1996

FPGA DATA BOOK AND DESIGN GUIDE

Actel

Take it to a higher level.

Designer Series

Are you an artist?

We think so.

Like an artist, you start with a blank canvas. And on that canvas you craft a design that's unique, inspired and, hopefully, the most efficient answer to a problem. That answer, in turn, translates into a product that gives your company a competitive edge in the market.

We call that creative.

At Actel, we understand this process and what you must go through. And we're committed to support your creativity and goals in every way that we can.

First, we support you with proven FPGAs that offer a full range of choices in speed, capacity, pin outs, and packaging. So that you can get the performance you need at a price you can afford.

That's performance/price value from Actel, without painful trade-offs.

Second, we have the advanced design and development tools you need to achieve fast, flexible and predictable design, with guaranteed gate utilization.

Our tools let you capture your vision in silicon, exactly the way you want it.

Third, we provide you with knowledgeable and responsive technical support. Real engineers to answer your questions and help you complete your masterpiece on time, on budget and to your specifications.

You have the ideas and the vision.

We provide the brushes, the paint and the canvas.

Actel

Actel
FPGA Data Book and Design Guide

1996

Contributors: Sam Beal, Larry Blessman, Jerome Fron, Steve Gurklys, Yousef Khalilollahi, Kamal Koraitem, Joel Landry, Warren Miller, Ken O'Neill, Mike Sarpa, Nelson Soria, Bruce Weyer, and Behrooz Zahiri.

Special thanks to many other contributors throughout Actel without whom this document could not have been completed.

Cover art was designed by Robert F. Tinney of Robert Tinney Graphics.

Actel Corporation reserves the right to make changes to any products or services herein at any time without notice. Actel does not assume any responsibility or liability arising out of the application or use of any product or service described as expressly agreed to in writing by Actel.

Order of Contents

Section 1: Component Data

Section 2: Software Development Systems

Section 3: Macro Libraries

Section 4: Designing with Actel Devices

Section 5: System Level Applications

Section 6: Customer-Authored Application Notes

Section 7: Test and Reliability Reports

Section 8: Technical Support Services

Section 9: Glossary

How to Use the Data Book

The *1996 Actel FPGA Data Book and Design Guide* contains a great deal of information to help designers make informed decisions about using Actel FPGAs.

General Overview

If you are not currently designing with Actel and are considering Actel as a possible solution for your needs, the following documents provide an overview of Actel devices and development software:

- Data book "Introduction" on page ix
- Introduction to each of the device data sheets
- "Introduction to Software" on page 2-1 in the Software Development Systems section
- "Introduction to Actel FPGA Architecture" application note on page 4-1

We also provide a method to compare and contrast the Actel design process with various design methodologies. The design methodologies covered include ASIC using VHDL and synthesis, PLD, and TTL, with the specifics covered in each of the following application notes:

- "Designing with FPGAs Compared with SSI/MSI Devices" on page 4-9
- "Designing with FPGAs Compared with PLD Devices" on page 4-13
- "FPGA Design for ASIC-Experienced Designers" on page 4-17
- "Comparing FPGA Solutions" on page 4-28

Selecting Devices for an Application

Appropriate devices for an application can be readily identified using a variety of information in the data book. Information on DC characteristics can be found in each device family's data sheet. Performance characteristics are also presented in each data sheet at the basic module level. Estimating performance at a higher or more abstract level is described in the "Estimating Performance and Capacity of Actel Devices" application note on page 4-25. This application note also provides a guide for estimating the size of device required for your application. In addition, the "Macro Libraries" section provides information on performance and capacity of various simple and complex functions.

A variety of application notes is also provided to assist in the evaluation process. These application notes include basic information such as board-level considerations, simple applications such as state machine design, and complex designs such as a 100 Mbit/s Ethernet controller. See the table of contents for a complete list of the application material available.

What's New in the 1996 Data Book?

The 1996 data book covers new products in both device hardware and development system software. The information includes the new PCI-Compliant ACT 3 devices described in the "Accelerator Series—ACT 3 PCI Family" data sheet and also the new 3200DX family with JTAG, fast decode, and RAM capabilities. For more information on the 3200DX family, see the following:

- "Integrator Series FPGAs – 1200XL and 3200DX Families" data sheet on page 1-7
- "3200DX Dual-Port Random Access Memory" application note on page 4-29
- "3200DX Wide Decode Modules" application note on page 4-31
- "IEEE Standard 1149.1 (JTAG) in the 3200DX Family" application note on page 4-37
- "3200DX Quadrant Clocks" application note on page 4-45

The data book also includes a comprehensive list of ACTgen-built macros, describing both performance and capacity ("Macro Libraries" section) and a description of the completely revamped Designer Series Development System, built around the new DirectTime timing-driven place-and-route engine. (See the "Software Development Systems" section.)

Introduction

The Designer's Challenge

Logic designers are constantly being faced with increasing design complexity, increasing demand for performance, increasing cost pressures, and the need for shorter time to market to ensure delivery of competitive products. These challenges have helped to make one solution—field programmable gate arrays (FPGAs)—the fastest growing segment in the ASIC industry. One difficulty, however, with FPGAs has been meeting all these challenges simultaneously. Often, performance or cost or complexity must be sacrificed to meet the remaining challenges. Actel has taken a unique approach in developing the hardware and software required to convert your creative ideas into working products that satisfy cost, performance, complexity, and scheduling needs concurrently. The key is Actel's device architecture, in conjunction with development software designed to meet the time-to-market challenge. FPGAs in general can help your products to compete, but Actel FPGAs can help your products to excel.

It Starts with Architecture

Achieving high complexity, low cost, and high performance simultaneously is no simple challenge for programmable logic. In its simplest form, programmable logic consists of small functional blocks connected by programmable elements. High complexity is achievable only by having plenty of programmable elements available to connect the functional blocks automatically. Low cost is achievable by ensuring that silicon is dedicated to efficient functional blocks, and not to the programmable network. Low cost is also facilitated by 100 percent usage of the available functional blocks. And finally, high performance is achievable by maintaining low propagation delays, which are a combination of logic block delay and the impedance of the programmable network. Actel's architecture has been designed to meet these challenges. Figures 1, 2, and 3 illustrate the Actel device architecture.

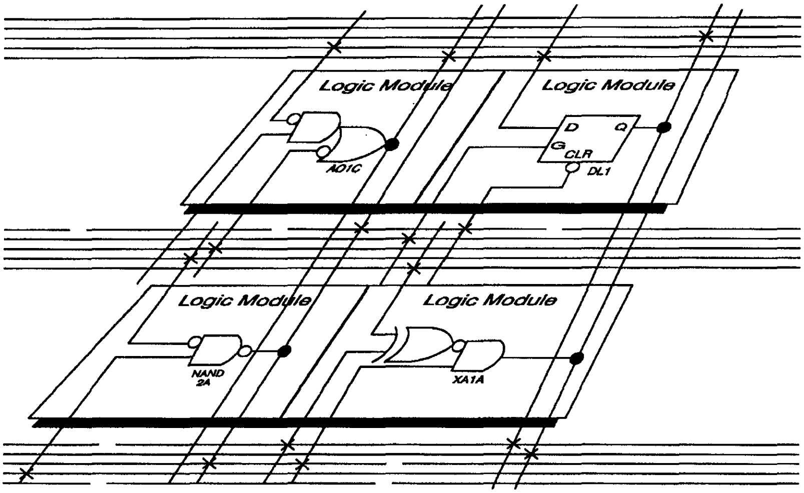


Figure 1 • Displayed is the basic Actel architecture, showing a combination of logic blocks, segmented routing tracks, and programmable interconnects (antifuses). Actel's multiplexer-based logic module is extremely efficient at implementing a variety of logic functions as shown. A combination of the antifuse (low resistance) and the segmented routing tracks (low capacitance) ensures a minimum of total impedance, resulting in high performance. Programmed antifuses are marked by the X on the diagram.

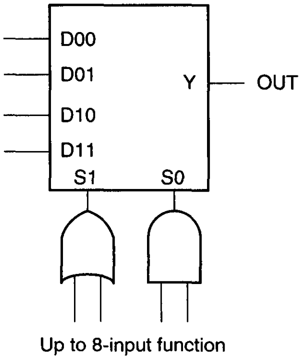


Figure 2 • The multiplexer-based Actel logic module is an efficient model for maximizing user function in a minimum amount of die space. In addition to area efficiency, the logic module is an excellent target for synthesis engines, allowing Actel devices to work well in synthesis environments.

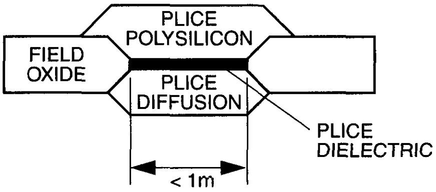


Figure 3 • Small size allows the antifuse to sit between the segmented metal tracks, requiring no additional space on the silicon. With little to no die size impact, antifuses can be applied liberally to enhance the overall interconnect scheme, providing 100 percent usage of the functional blocks. Impedance variability is also controlled by limiting the allowable number of antifuses in any given path (maximum of four), resulting in predictable delays.

Meeting the Time-to-Market Challenge

The Actel architecture also provides benefits that help improve time to market. Predictability of both cost and performance at the beginning of the design cycle is key to a successful product. The end of the design cycle is too late to find out you're over budget or under performance. The low impedance of the programmable network supports performance predictability, because the routing delay variation is small (and thus predictable) and because most of the delay is required by the logic rather than by the routing. Cost predictability is also enhanced by the large number of interconnects, since logic blocks, not interconnect, are the limiting factor in the device. The large number of antifuses also supports design flexibility, meaning that knowledge of architectural details is not required to achieve low cost or high performance.

Once the silicon challenge is met, the implementation challenge remains. To support the ever-decreasing time-to-market challenge, Actel provides an array of tools for functional definition, performance definition, fully automated layout, and timing analysis. Design definition supports third-party tools of choice (both synthesis and schematic) in addition to ACTmap VHDL Synthesis, Actel's own VHDL synthesis engine. Each of these tools has been optimized to minimize cost and maximize performance when implementing functions to maximize total silicon value. Performance definition and analysis are supported by Actel's DirectTime suite of timing-driven layout tools.

It Ends with Great Products

To build great products that excel in today's competitive marketplace, you have to start with great components—components that deliver high performance, low cost, complexity, and fast time to market. Actel has the architecture, the understanding, and the commitment to make these things possible. When you're ready to take your products to a higher level, we'll be there to support you. Give your products the best. Give them Actel.

Actel's Family of Devices

Actel has developed a variety of device families designed to meet particular cost, density, and performance trade-offs. The Integrator Series was designed to achieve low cost, yet maintain relatively high performance by leveraging the strengths of the Actel architecture. It consists of the ACT 2, 1200XL, and 3200DX families. The Accelerator Series (ACT 3 family) was designed to maximize FPGA performance. The ACT 1 family is Actel's original low-cost family of devices.

The Integrator Series

The combination of low cost and high performance makes the Integrator Series one of the best programmable values in the industry. (See Table 1.) In addition, the Integrator Series offers various options including high-performance dual-port SRAM (in excess of 100 MHz), wide decode capability, and JTAG, depending on the family and device size. The need for SRAM is driven by more complex applications at higher gate counts. Wide decode helps improve address decoding both on-chip and off-chip (7.5 ns and 15.0 ns, respectively, up to 35 bits). JTAG improves manufacturability for higher I/O count devices.

Table 1 • The Integrator Series Characteristics

Device	Usable Gates	User I/O	SRAM Bits	Wide Decode	JTAG
A1225XL	2,500	83	N/A	No	No
A1240XL	4,000	104	N/A	No	No
A3265DX	6,500	126	N/A	Yes	No
A1280XL	8,000	140	N/A	No	No
A32100DX	10,000	156	2048	Yes	Yes
A32140DX	14,000	176	N/A	Yes	Yes
A32200DX	20,000	206	2560	Yes	Yes
A32300DX	30,000	254	3072	Yes	Yes
A32400DX	40,000	292	4096	Yes	Yes

The Accelerator Series

These devices are especially well suited for very high-speed synchronous applications, given the excellent clk-q (pad-pad) delays and external setup times approaching 1 ns (with 0 ns hold). Internal performance also benefits from increased segmentation, larger internal drivers, lower-resistance antifuses, and special clock networks. I/O counts are also increased to improve interaction with wide data bus devices. (See Table 2.) The A1400 family is available in both 3.3 V and 5 V versions. Special PCI versions of the 5 V device are also available.

Table 2 • The Accelerator Series Characteristics

Device	Usable Gates	User I/O
A1415A	1,500	80
A1425A	2,500	100
A1440A	4,000	140
A1460A	6,000	168
A14100A	10,000	228

Table 3 • Performance comparisons between Integrator and Accelerator devices under worst-case commercial conditions. More complex functional comparisons can be found in the "Macro Libraries" section of the data book.

	Clk-Q (Pad-Pad) Performance		Maximum Internal Performance*	
Gate Count	Accelerator	Integrator	Accelerator	Integrator
1,500	7.5 ns	N/A	250 MHz	N/A
2,500	7.5 ns	10.0 ns	250 MHz	165 MHz
4,000	8.5 ns	10.5 ns	250 MHz	160 MHz
6,000	9.0 ns	11.0 ns	200 MHz	135 MHz
10,000	9.5 ns	11.0 ns	200 MHz	135 MHz

** Representative of shift registers, simple state machines, and prescaled counters; limited by maximum frequency of internal clock*

ACT 1 Family

The ACT 1 family consists of the A1010B and A1020B devices, providing an extremely cost-effective solution for low levels of integration. The A1010B provides up to 1200 usable gates, allowing integration for up to 30 TTL or 12 PLD devices into a single device. The A1020B provides up to 2000 usable gates and can integrate up to 50 TTL or 20 PLD devices. Typical system speeds in ACT 1 devices are below 30 MHz, with maximum internal performance peaking at 70 MHz. ACT 1 devices are available in both 3.3 V and 5 V versions.

Military and Space

Actel Corporation is committed to providing FPGA solutions for military and space customers. Actel offers devices in a wide variety of processing options, allowing control of device costs throughout the design process: tested to Commercial (0°C to 70°C) and Military (–55°C to 125°C) temperature ranges; MIL-STD-883 Class B compliant; and DESC SMD Class M compliant. Devices are available today in the popular ACT 1, ACT 2, and Accelerator (ACT 3) families, and the Integrator (1200XL and 3200DX) families will be introduced later in 1996.

Actel also offers radiation-hardened devices for space applications. Available in 2000- and 8000-gate capacities and compliant with QML Class V, these devices are guaranteed for radiation survivability and high reliability.

Designer Series Development Software

Designer Series, Actel's development system, takes your idea to working silicon as quickly and as easily as possible. It operates in both PC and workstation (Sun and HP700) environments. Actel's software is designed to work with various third-party CAE tools for function generation (behavioral and schematic) and simulation. Actel offers ACTmap VHDL Synthesis, a VHDL synthesis engine, for designers with no existing CAE tools. Designer Series streamlines the entire FPGA development flow with intuitive, easy-to-use tools, which address the following issues:

- Functional definition (including timing and I/O definitions)
- Layout (conversion of the definition to silicon)
- Timing verification
- Programming and lab verification

Functional Definition

Actel provides libraries to support both synthesis- and schematic-based functional descriptions. Random logic and state machines can be described in behavioral languages (VHDL, Verilog, or PAL). Structured logic such as counters, adders, and comparators can be automatically built using the ACTgen Macro Builder. Functions can be exclusively behavioral, schematic, or a combination of the two. Timing definition is supported by Actel's DirectTime suite of timing-driven layout tools. I/O definitions can be described in the original source or within Designer Series.

Layout

Actel has always featured fully automatic layout, maintaining high performance at high levels of utilization, to improve designer productivity. Beginning with Designer Series 3.0, Actel introduced DirectTime Layout (optional) to support fully automatic timing-driven place and route to give the user the ability to define succinctly all critical paths and thus squeeze every nanosecond out of the device. Standard layout remains for non-performance-critical applications. Layout iterations take only a few minutes, using the Incremental Layout option to improve productivity further.

Timing Verification

Actel supports backannotation into third-party simulators in addition to automatically generated timing reports and interactive static timing analysis tools.

Programming and Lab Verification

Device programming is accomplished using Actel's Activator 2, Activator 2S, or Data I/O programmers. Once device programming is complete, problems in the lab can be quickly debugged using the Actionprobe diagnostics tool (optional). This capability gives the designer the ability to probe any internal node during normal device operation.

Meeting the Synthesis Challenge

Top-down design methodologies are being driven by increasing design complexity and the need to improve designer productivity; however, many FPGAs have failed to meet the needs of synthesis because of poor architecture and unpredictable performance. To meet this challenge, Actel has developed a readily synthesizable architecture and fully automatic place and route that excel in a synthesis environment. In addition, Actel's excellent performance predictability makes synthesis expectations and hardware realities converge. With Actel's DirectTime Layout tools, performance descriptions used to drive the synthesis engine can also be passed directly to the layout engine, thus automating the entire process.

Summary

Actel recognizes the challenges facing today's design engineer, and we are delivering the hardware and software tools to meet those challenges. We started with an architecture that drives low cost, high performance, and predictability, from which we built a variety of families to address a wide range of applications. To meet the time-to-market challenge, high-quality synthesis results and automated tools put the designer in control of all phases of the development process including design, implementation, and debugging. So, when you are ready to take your products to a higher level, look to Actel for great FPGA products.

About the Company

Actel Corporation is dedicated to providing logic designers with the capability and confidence to move up to higher-complexity designs. Actel is the world's leading manufacturer of antifuse-based field programmable gate arrays (FPGAs) and associated software development tools. Actel FPGAs are used in voice and data communication, computing, medical, industrial control, military, and aerospace applications worldwide. Actel is traded on the NASDAQ market, using the symbol ACTL. Our contact information is as follows:

955 East Arques Avenue
Sunnyvale, CA 94086
Telephone: 408-739-1010
Internet: http://www.actel.com

Other Sources of Data

The data book is intended to provide information most designers would need to evaluate Actel as a possible solution. Additional information is available to supplement the data book for more specific needs and is described below. To ensure that you have the latest information available, check with your local sales representative or local Actel office.

Supplemental Data Sheets

As new device families and enhanced versions of existing families are introduced, information is provided to designers in the form of supplemental data sheets that will be incorporated into future data books. The latest device family data sheets can also be found on Actel's product Web site (http://www.actel.com/product.html).

Application Information

Additional application information can be found in various places including Actel's technical services Web site (http://www.actel.com /techserv.html) and on the Action Fax (800-262-1062). Information includes both customer applications and design examples. Action Fax also provides information such as bonding diagrams, pin cross-references, and socketing information (part numbers and mechanical drawings).

Synthesis Information

A variety of synthesis methodology guides have been developed to improve productivity and results when working with Actel in a synthesis environment. Additional information can be found on Actel's synthesis Web site (http://www.actel.com/synthesis.html). See the "High-Level Design Resources" document on page 2-27 in the data book for more details.

FPGA Integration Primer

The FPGA Integration Primer is a powerful evaluation tool designed to assess Actel as a potential FPGA solution targeted specifically for TTL and PLD designers. The primer is built around the ACTgen Macro Builder software tool, providing the ability to create many complex functions quickly and easily to assist in cost and performance analysis. The FPGA Integration Primer includes an evaluation copy of ACTgen so you can see the tool in action.

Development System Multimedia Demo

A CD-ROM multimedia interactive demo introduces you to FPGA benefits, Actel architecture and devices, design flow, and a realistic evaluation of the Designer Series tools.

Test and Reliability Information

You can obtain the latest and complete historical test and reliability information for Actel devices from the Actel product Web site (http://www.actel.com/product.html).

Component Data

Section 1: Component Data

Component Selector Guide

5 Volt FPGA Selector Guide

		No.		Temp.[3]	User	Logic	Wide	SRAM	Flip-Flops		Equiv. Pkgs.	
Device Type	Pkg[1]	Pins	Speed Option[2]	Range	I/O	Gates	Decodes	Bits	Fixed	Max	TTLs	PALs
Integrator Series												
A1225XL	PG	100	Std, -1, -2	C	83	2,500	0	0	231	382	63	25
	PL	84	Std, -1, -2	C, I	72	2,500	0	0	231	382	63	25
	PQ	100	Std, -1, -2	C, I	83	2,500	0	0	231	382	63	25
	VQ	100	Std, -1, -2	C, I	83	2,500	0	0	231	382	63	25
A1240XL	PG	132	Std, -1, -2	C	104	4,000	0	0	348	568	100	40
	PL	84	Std, -1, -2	C, I	72	4,000	0	0	348	568	100	40
	PQ	144	Std, -1, -2	C, I	104	4,000	0	0	348	568	100	40
	TQ	176	Std, -1, -2	C, I	104	4,000	0	0	348	568	100	40
A3265DX	PL	84	Std, -1, -2	C, I	72	6,500	20	0	510	810	160	65
	PQ	100	Std, -1, -2	C, I	83	6,500	20	0	510	810	160	65
	PQ	160	Std, -1, -2	C, I	125	6,500	20	0	510	810	160	65
	TQ	176	Std, -1, -2	C, I	126	6,500	20	0	510	810	160	65
A1280XL	CQ	172	Std, -1, -2*	C, M, B	140	8,000	0	0	624	998	200	80
	PG	176	Std, -1, -2*	C, M, B	140	8,000	0	0	624	998	200	80
	PL	84	Std, -1, -2	C, I	72	8,000	0	0	624	998	200	80
	PQ	160	Std, -1, -2	C, I	125	8,000	0	0	624	998	200	80
	TQ	176	Std, -1, -2	C, I	140	8,000	0	0	624	998	200	80
A32100DX	BG	240	Std, -1, -2	C, I	152	10,000	20	2,048	700	1,107	250	100
	PL	84	Std, -1, -2	C, I	72	10,000	20	2,048	700	1,107	250	100
	PQ	160	Std, -1, -2	C, I	125	10,000	20	2,048	700	1,107	250	100
	PQ	208	Std, -1, -2	C, I	152	10,000	20	2,048	700	1,107	250	100
	TQ	176	Std, -1, -2	C, I	150	10,000	20	2,048	700	1,107	250	100
A32140DX	BG	240	Std, -1, -2	C, I	176	14,000	24	0	954	1,498	350	140
	PQ	160	Std, -1, -2	C, I	125	14,000	24	0	954	1,498	350	140
	PQ	208	Std, -1, -2	C, I	176	14,000	24	0	954	1,498	350	140
	TQ	176	Std, -1, -2	C, I	150	14,000	24	0	954	1,498	350	140
A32200DX	BG	TBD	Std, -1, -2	C	202	20,000	24	2,560	1,230	1,923	500	200
	PQ	208	Std, -1, -2	C	176	20,000	24	2,560	1,230	1,923	500	200
	RQ	208	Std, -1, -2	C	176	20,000	24	2,560	1,230	1,923	500	200
	RQ	240	Std, -1, -2	C	202	20,000	24	2,560	1,230	1,923	500	200
A32300DX	BG	TBD	Std, -1, -2	C	250	30,000	28	3,072	1,888	2,929	750	300
	RQ	208	Std, -1, -2	C	176	30,000	28	3,072	1,888	2,929	750	300
	RQ	240	Std, -1, -2	C	tbd	30,000	28	3,072	1,888	2,929	750	300
A32400DX	BG	TBD	Std, -1, -2	C	288	40,000	28	4,096	2,526	3,903	1,000	400
	RQ	240	Std, -1, -2	C	tbd	40,000	28	4,096	2,526	3,903	1,000	400

** Offered for Commercial (C) devices only.*

5 Volt FPGA Selector Guide (continued)

		No.		Temp.[3]	User	Logic	Wide	SRAM	Flip-Flops		Equiv. Pkgs.	
Device Type	Pkg[1]	Pins	Speed Option[2]	Range	I/O	Gates	Decodes	Bits	Fixed	Max	TTLs	PALs
Accelerator Series												
A1415A	PG	100	Std, -1, -2, -3	C	80	1,500	0	0	264	312	38	15
	PL	84	Std, -1, -2, -3	C, I	70	1,500	0	0	264	312	38	15
	PQ	100	Std, -1, -2, -3	C, I	80	1,500	0	0	264	312	38	15
	VQ	100	Std, -1, -2, -3	C, I	80	1,500	0	0	264	312	38	15
A1425A	CQ	132	Std, -1	C, M, B	100	2,500	0	0	360	435	63	25
	PG	133	Std, -1, -2*, -3*	C, M, B	100	2,500	0	0	360	435	63	25
	PL	84	Std, -1, -2, -3	C, I	70	2,500	0	0	360	435	63	25
	PQ	100	Std, -1, -2, -3	C, I	80	2,500	0	0	360	435	63	25
	PQ	160	Std, -1, -2, -3	C, I	100	2,500	0	0	360	435	63	25
	VQ	100	Std, -1, -2, -3	C, I	83	2,500	0	0	360	435	63	25
A1440A	PG	175	Std, -1, -2, -3	C	140	4,000	0	0	568	706	100	40
	PL	84	Std, -1, -2, -3	C, I	70	4,000	0	0	568	706	100	40
	PQ	160	Std, -1, -2, -3	C, I	130	4,000	0	0	568	706	100	40
	TQ	176	Std, -1, -2, -3	C, I	140	4,000	0	0	568	706	100	40
	VQ	100	Std, -1, -2, -3	C, I	83	4,000	0	0	568	706	100	40
A1440B	PG	175	Std, -1, -2, -3	C	140	4,000	0	0	568	706	100	40
	PL	84	Std, -1, -2, -3	C, I	70	4,000	0	0	568	706	100	40
	PQ	160	Std, -1, -2, -3	C, I	130	4,000	0	0	568	706	100	40
	TQ	176	Std, -1, -2, -3	C, I	140	4,000	0	0	568	706	100	40
	VQ	100	Std, -1, -2, -3	C, I	83	4,000	0	0	568	706	100	40
A1440BP	PQ	160	Std, -1, -2	C, I	130	4,000	0	0	568	706	100	40
(PCI compliant)	TQ	176	Std, -1, -2	C, I	140	4,000	0	0	568	706	100	40
A1460A	BG	225	Std, -1, -2	C, I	168	6,000	0	0	768	976	150	60
	CQ	196	Std, -1	C, M, B	168	6,000	0	0	768	976	150	60
	PG	207	Std, -1, -2*	C, M, B	168	6,000	0	0	768	976	150	60
	PQ	160	Std, -1, -2	C, I	130	6,000	0	0	768	976	150	60
	PQ	208	Std, -1, -2	C, I	167	6,000	0	0	768	976	150	60
	TQ	176	Std, -1, -2	C, I	151	6,000	0	0	768	976	150	60
A1460B	BG	225	Std, -1, -2	C, I	168	6,000	0	0	768	976	150	60
	PG	207	Std, -1, -2	C	168	6,000	0	0	768	976	150	60
	PQ	160	Std, -1, -2	C, I	130	6,000	0	0	768	976	150	60
	PQ	208	Std, -1, -2	C, I	167	6,000	0	0	768	976	150	60
	TQ	176	Std, -1, -2	C, I	151	6,000	0	0	768	976	150	60
A1460BP	BG	225	Std, -1, -2	C, I	168	6,000	0	0	768	976	150	60
(PCI compliant)	PQ	160	Std, -1, -2	C, I	130	6,000	0	0	768	976	150	60
	PQ	208	Std, -1, -2	C, I	167	6,000	0	0	768	976	150	60
	TQ	176	Std, -1, -2	C, I	151	6,000	0	0	768	976	150	60
A14100A	BG	313	Std, -1, -2, -3	C, I	228	10,000	0	0	1,153	1,493	250	100
	CQ	256	Std, -1	C, M, B	228	10,000	0	0	1,153	1,493	250	100
	PG	257	Std, -1, -2*, -3*	C, M, B	228	10,000	0	0	1,153	1,493	250	100
	RQ	208	Std, -1, -2, -3	C, I	175	10,000	0	0	1,153	1,493	250	100
A14100B	BG	313	Std, -1, -2, -3	C, I	228	10,000	0	0	1,153	1,493	250	100
	PG	257	Std, -1, -2, -3	C	228	10,000	0	0	1,153	1,493	250	100
	RQ	208	Std, -1, -2, -3	C, I	175	10,000	0	0	1,153	1,493	250	100
A14100BP	BG	313	Std, -1, -2	C, I	228	10,000	0	0	1,153	1,493	250	100
(PCI compliant)	RQ	208	Std, -1, -2	C, I	175	10,000	0	0	1,153	1,493	250	100

** Offered for Commercial (C) devices only.*

5 Volt FPGA Selector Guide (continued)

		No.		Temp.[3]	User	Logic	Wide	SRAM	Flip-Flops		Equiv. Pkgs.	
Device Type	Pkg[1]	Pins	Speed Option[2]	Range	I/O	Gates	Decodes	Bits	Fixed	Max	TTLs	PALs
ACT 1 Family												
A1010B	PG	84	Std, -1	C, M, B	57	1,200	0	0	0	147	30	12
	PL	44	Std, -1, -2, -3	C, I	34	1,200	0	0	0	147	30	12
	PL	68	Std, -1, -2, -3	C, I	57	1,200	0	0	0	147	30	12
	PQ	100	Std, -1, -2, -3	C, I	57	1,200	0	0	0	147	30	12
	VQ	80	Std, -1, -2, -3	C, I	57	1,200	0	0	0	147	30	12
A1020B	CQ	84	Std, -1	C, M, B	69	2,000	0	0	0	273	50	20
	PG	84	Std, -1	C, M, B	69	2,000	0	0	0	273	50	20
	PL	44	Std, -1, -2, -3	C, I	34	2,000	0	0	0	273	50	20
	PL	68	Std, -1, -2, -3	C, I	57	2,000	0	0	0	273	50	20
	PL	84	Std, -1, -2, -3	C, I	69	2,000	0	0	0	273	50	20
	PQ	100	Std, -1, -2, -3	C, I	69	2,000	0	0	0	273	50	20
	VQ	80	Std, -1, -2, -3	C, I	69	2,000	0	0	0	273	50	20
ACT 2 Family												
A1225A	PG	100	Std, -1, -2	C	83	2,500	0	0	231	382	63	25
	PL	84	Std, -1, -2	C, I	72	2,500	0	0	231	382	63	25
	PQ	100	Std, -1, -2	C, I	83	2,500	0	0	231	382	63	25
	VQ	100	Std, -1, -2	C	83	2,500	0	0	231	382	63	25
A1240A	PG	132	Std, -1, -2*	C, M, B	104	4,000	0	0	348	568	100	40
	PL	84	Std, -1, -2	C, I	72	4,000	0	0	348	568	100	40
	PQ	144	Std, -1, -2	C, I	104	4,000	0	0	348	568	100	40
	TQ	176	Std, -1, -2	C	104	4,000	0	0	348	568	100	40
A1280A	CQ	172	Std, -1	C, M, B	140	8,000	0	0	624	998	200	80
	PG	176	Std, -1, -2*	C, M, B	140	8,000	0	0	624	998	200	80
	PL	84	Std, -1, -2	C, I	72	8,000	0	0	624	998	200	80
	PQ	160	Std, -1, -2	C, I	125	8,000	0	0	624	998	200	80
	TQ	176	Std, -1, -2	C	140	8,000	0	0	624	998	200	80
Rad-Hard FPGAs												
RH1020	CQ	84	Std	V	69	2,000	0	0	0	273	50	20
RH1280	CQ	172	Std	V	140	8,000	0	0	624	998	200	80

** Offered for Commercial (C) devices only.*

3.3 VOLT FPGA Selector Guide

		No.		Temp.[3]	User	Logic	Wide	SRAM	Flip-Flops		Equiv. Pkgs.	
Device Type	Pkg[1]	Pins	Speed Option[2]	Range	I/O	Gates	Decodes	Bits	Fixed	Max	TTLs	PALs
Integrator Series												
A1225XLV	PL	84	Std	C	72	2,500	0	0	231	382	63	25
	VQ	100	Std	C	83	2,500	0	0	231	382	63	25
A1240XLV	PL	84	Std	C	72	4,000	0	0	348	568	100	40
	TQ	176	Std	C	104	4,000	0	0	348	568	100	40
A3265DXV	PL	84	Std	C	72	6,500	20	0	510	810	160	65
	TQ	176	Std	C	126	6,500	20	0	510	810	160	65
A1280XLV	PL	84	Std	C	72	8,000	0	0	624	998	200	80
	TQ	176	Std	C	140	8,000	0	0	624	998	200	80
Accelerator Series												
A14V15A	PL	84	Std	C	70	1,500	0	0	264	312	38	15
	VQ	100	Std	C	80	1,500	0	0	264	312	38	15
A14V25A	PL	84	Std	C	70	2,500	0	0	360	435	63	25
	PQ	160	Std	C	100	2,500	0	0	360	435	63	25
	VQ	100	Std	C	83	2,500	0	0	360	435	63	25
A14V40A	PL	84	Std	C	70	4,000	0	0	568	706	100	40
	PQ	160	Std	C	130	4,000	0	0	568	706	100	40
	TQ	176	Std	C	140	4,000	0	0	568	706	100	40
	VQ	100	Std	C	83	4,000	0	0	568	706	100	40
A14V60A	PQ	160	Std	C	130	6,000	0	0	768	976	150	60
	PQ	208	Std	C	167	6,000	0	0	768	976	150	60
	TQ	176	Std	C	151	6,000	0	0	768	976	150	60
A14V100A	RQ	208	Std	C	175	10,000	0	0	1,153	1,493	250	100
ACT 1 Family												
A10V10B	PL	68	Std	C	57	1,200	0	0	0	147	30	12
	VQ	80	Std	C	57	1,200	0	0	0	147	30	12
A10V20B	PL	68	Std	C	57	2,000	0	0	0	273	50	20
	PL	84	Std	C	69	2,000	0	0	0	273	50	20
	VQ	80	Std	C	69	2,000	0	0	0	273	50	20

Masked Programmable Gate Arrays (MPGAs)

MPGAs are available for most Actel devices. For further details, see page 1-257.

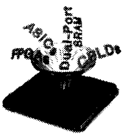
Notes for Product Selector Guides

1. Package Types:
   BG = Plastic Ball Grid Array
   CQ = Ceramic Quad Flat Pack
   PG = Ceramic Pin Grid Array
   PL = Plastic J-Leaded Chip Carrier
   PQ = Plastic Quad Flat Pack
   RQ = Power Plastic Quad Flat Pack
   TQ = Thin (1.4mm) Plastic Quad Flat Pack
   VQ = Very Thin (1.0mm) Plastic Quad Flat Pack
2. Speed Options:
   Std = Standard Speed
   –1 = Approximately 15% faster than Standard
   –2 = Approximately 25% faster than Standard
   –3 = Approximately 35% faster than Standard
3. Temperature Ranges:
   C = Commercial Temperature (0°C to +75°C)
   I = Industrial (–40°C to +85°C)
   M = Military (–55°C to +125°C)
   B = MIL-STD-883
   V = QML Class V

Package Options: User I/Os per Package

							Integrator Series								Accelerator Series					
		ACT 1		ACT 2			1200XL			3200DX						ACT 3				
Package	Pins	A1010B	A1020B	A1225A	A1240A	A1280A	A1225A	A1240A	A1280A	A3265DX	A32100DX	A32140DX	A32200DX	A32300DX	A32400DX	A1415A	A1425A	A1440A/B	A1460A/B	A14100A/B
PLCC	44	34	34																	
	68	57	57																	
	84		69	72	72	72	72	72	72	72	72			72	72	70	70	70		
PQFP	100	57	69	83			83	83								80	80			
	144				104			104												
	160					125			125	125	125	125					100	131	131	
	208								140		152	176	176						167	
RQFP	208												176	176						175
	240												202	TBD	TBD					
VQFP	80	57	69																	
	100			83			83									80	83	83		
TQFP	176				104	140		104	140	126	150	150						140	151	
BGA	225																		168	
	240										152	176								
	313																			228
	TBD												202	250	288					
CPGA	84	57	69																	
	100			83			83									80				
	132/133				104			104									100			
	175/176					140			140									140		
	207																		168	
	257																			228
CQFP	84		69																	
	132																100			
	172					140			140											
	196																		168	
	256																			228

Integrator Series FPGAs – 1200XL and 3200DX Familes



Features

High Capacity

- 2,500 to 40,000 logic gates
- Up to 4 Kbits configurable dual-port SRAM
- Fast wide-decode circuitry
- Up to 288 user-programmable I/O Pins

High Performance

- 225 MHz performance
- 5 ns Dual-Port SRAM Access
- 100 MHz FIFOs
- 7.5 ns 35-bit Address Decode

Ease-of-Integration

- Synthesis-friendly architecture supports ASIC design methodologies
- 95-100% device utilization using automatic Place and Route Tools
- Deterministic, user-controllable timing via DirectTime software tools
- Supported by Actel Designer Series development system with interfaces to popular design environments such as Cadence, Escalade, Exemplar, IST, Mentor Graphics, Synopsys and Viewlogic
- JTAG 1149.1 Boundary Scan Testing

General Description

Actel's Integrator Series FPGAs are the first programmable logic devices optimized for high-speed system logic integration. Based on Actel's proprietary PLICE antifuse technology and state-of-the-art 0.6-micron double metal CMOS process, the Integrator Series devices offer a fine-grained, register-rich architecture with the industry's fastest embedded dual-port SRAM and wide decode circuitry.

3200DX and 1200XL FPGAs were designed to integrate system logic which is typically implemented in multiple CPLDs, PALs and FPGAs. These devices provide the features and performance required for today's complex, high-speed digital logic systems. The 3200DX family offers the industry's fastest dual-port SRAM for implementing fast FIFOs, LIFOs and temporary data storage. The large number of storage elements can efficiently address applications requiring wide datapath manipulation and transformation functions such as telecommunications, networking and DSP.

Integrator Series Product Profile

Device	A1225XL	A1240XL	A3265DX	A1280XL	A32100DX	A32140DX	A32200DX	A32300DX	A32400DX
Capacity									
Logic Gates[1]	2,500	4,000	6,500	8,000	10,000	14,000	20,000	30,000	40,000
SRAM Bits	N/A	N/A	N/A	N/A	2,048	N/A	2,560	3,072	4,096
Logic Modules									
Sequential	231	348	510	624	738	954	1276	1944	2592
Combinatorial	220	336	475	608	698	912	1226	1885	2560
Decode	N/A	N/A	20	N/A	22	24	28	32	34
SRAM Modules (64x4 or 32x8)	NA	NA	NA	NA	8	NA	10	12	16
Dedicated Flip-Flops	231	348	510	624	738	954	1,276	1,944	2,592
Clocks	2	2	2	2	6	2	6	6	6
User I/O (maximum)	83	104	126	140	152	176	202	250	288
JTAG	No	No	No	No	Yes	Yes	Yes	Yes	Yes
Packages	PL84 PQ100 VQ100 PG100	PL84 PQ100 PQ144 TQ176 PG132	PL84 PQ160 TQ176	PL84 PQ160 PQ208 TQ176 PG176 CQ172	PL84 PQ160 TQ176 BG240	PQ160 PQ208 TQ176 BG240	PQ208 RQ208 RQ240 BG240 BG432	BG432	BG432

Note 1: Logic gate capacity does not include SRAM bits as logic.

Ordering Information

A1225 XL V – ___ PQ 100 C

Application (Temperature Range)

C	=	Commercial (0 to +70°C)
I	=	Industrial (–40 to +85°C)
M	=	Military (–55 to +125°C)
B	=	MIL-STD-883

Package Lead Count

Package Type

PG	=	Ceramic Pin Grid Array
PL	=	Plastic Leaded Chip Carrier
PQ	=	Plastic Quad Flatpack
RQ	=	Plastic Power Quad Flatpack
TQ	=	Thin (1.4 mm) Quad Flatpack
CQ	=	Ceramic Quad Flatpack
BG	=	Plastic Ball Grid Array
VQ	=	Very Thin (1.0 mm) Quad Flatpack

Speed Grade

Blank	=	Standard Speed
1	=	Approximately 15% faster than Standard
2	=	Approximately 25% faster than Standard

Operating Voltage

V	=	3.3 Volt
Blank	=	5.0 Volt

Die Revision

XL	=	1200XL Family
DX	=	3200DX Family

Part Number

A1225	=	2500 Gates
A1240	=	4000 Gates
A3265	=	6500 Gates
A1280	=	8000 Gates
A32100	=	10000 Gates
A32140	=	14000 Gates
A32200	=	20000 Gates
A32300	=	30000 Gates
A32400	=	40000 Gates

Product Plan

	Speed Grade			Application			
	Std	–1*	–2*	C	I	M	B
A1225XL Device							
84-pin Plastic Leaded Chip Carrier (PLCC)	✔	✔	✔	✔	✔	—	—
100-pin Plastic Quad Flatpack (PQFP)	✔	✔	✔	✔	✔	—	—
100-pin Very Thin Plastic Quad Flatpack (VQFP)	✔	✔	✔	✔	✔	—	
100-pin Ceramic Pin Grid Array (CPGA)	✔	✔	✔	✔	—	—	—
A1225XLV Device							
84-pin Plastic Leaded Chip Carrier (PLCC)	P	—	—	P	—	—	—
100-pin Very Thin Plastic Quad Flatpack (VQFP)	P	—	—	P	—	—	—
A1240XL Device							
84-pin Plastic Leaded Chip Carrier (PLCC)	✔	✔	✔	✔	✔	—	—
100-pin Plastic Quad Flatpack (PQFP)	P	P	P	P	P	—	—
132-pin Ceramic Pin Grid Array (CPGA)	✔	✔	✔	✔	—	—	—
144-pin Plastic Quad Flatpack (PQFP)	✔	✔	✔	✔	✔	—	—
176-pin Thin Plastic Quad Flatpack (TQFP)	✔	✔	✔	✔	✔	—	—
A1240XLV Device							
84-pin Plastic Leaded Chip Carrier (PLCC)	P	—	—	P	—	—	—
176-pin Thin Plastic Quad Flatpack (TQFP)	P	—	—	P	—	—	—
A3265DX Device							
84-pin Plastic Leaded Chip Carrier (PLCC)	✔	✔	✔	✔	✔	—	—
160-pin Plastic Quad Flatpack (PQFP)	✔	✔	✔	✔	✔	—	—
176-pin Thin Plastic Quad Flatpack (TQFP)	✔	✔	✔	✔	✔	—	—
A3265DXV Device							
84-pin Plastic Leaded Chip Carrier (PLCC)	P	—	—	P	—	—	—
176-pin Thin Plastic Quad Flatpack (TQFP)	P	—	—	P	—	—	—
A1280XL Device							
84-pin Plastic Leaded Chip Carrier (PLCC)	✔	✔	✔	✔	✔	—	—
160-pin Plastic Quad Flatpack (PQFP)	✔	✔	✔	✔	✔	—	—
172-pin Ceramic Quad Flatpack (CQFP)	✔	✔	✔	✔	—	P	P
176-pin Thin Plastic Quad Flatpack (TQFP)	✔	✔	✔	✔	✔	—	—
176-pin Ceramic Pin Grid Array (CPGA)	✔	✔	✔	✔	—	P	P
208-pin Plastic Quad Flatpack (PQFP)	P	P	P	P	P	—	—
A1280XLV Device							
84-pin Plastic Leaded Chip Carrier (PLCC)	P	—	—	P	—	—	—
176-pin Thin Plastic Quad Flatpack (TQFP)	P	—	—	P	—	—	—

Applications: *C = Commercial*
I = Industrial
M = Military
B = MIL-STD-883

Availability: ✔ *= Available*
P = Planned
— = Not Planned

** Speed Grade:* *–1 = Approx. 15% faster than Standard*
–2 = Approx. 25% faster than Standard

Product Plan (continued)

	Speed Grade			Application			
	Std	–1*	–2*	C	I	M	B
A32100DX Device							
84-pin Plastic Leaded Chip Carrier (PLCC)	P	P	P	P	P	—	—
160-pin Plastic Quad Flatpack (PQFP)	P	P	P	P	P	—	—
176-pin Thin Plastic Quad Flatpack (TQFP)	P	P	P	P	P	—	—
240-pin Plastic Super Ball Grid Array (SBGA)	P	P	P	P	P	—	—
A32140DX Device							
160-pin Plastic Quad Flatpack (PQFP)	✓	✓	✓	✓	✓	—	—
176-pin Thin Plastic Quad Flatpack (TQFP)	✓	✓	✓	✓	✓	—	—
208-pin Plastic Quad Flatpack (PQFP)	✓	✓	✓	✓	✓	—	—
240-pin Plastic Super Ball Grid Array (SBGA)	✓	✓	✓	✓	✓	—	—
A32140DXV Device							
176-pin Thin Plastic Quad Flatpack (TQFP)	P	—	—	P	—	—	—
208-pin Plastic Quad Flatpack (PQFP)	P	—	—	P	—	—	—
A32200DX Device							
208-pin Plastic Quad Flatpack (PQFP)	✓	✓	P	✓	P	—	—
208-pin Plastic Power Quad Flatpack (RQFP)	✓	✓	P	✓	P	—	—
240-pin Plastic Power Quad Flatpack (RQFP)	✓	✓	P	✓	P	—	—
240-pin Plastic Super Ball Grid Array (SBGA)	P	P	P	P	P	—	—
432-pin Plastic Super Ball Grid Array (SBGA)	P	P	P	P	P	—	—
A32300DX Device							
432-pin Plastic Super Ball Grid Array (SBGA)	P	P	P	P	P	—	—
A32400DX Device							
432-pin Plastic Super Ball Grid Array (SBGA)	P	P	P	P	P	—	—

Applications: C = *Commercial*, I = *Industrial*, M = *Military*, B = *MIL-STD-883*

Availability: ✓ = *Available*, P = *Planned*, — = *Not Planned*

** Speed Grade:* –1 = *Approx. 15% faster than Standard*, –2 = *Approx. 25% faster than Standard*

Integrator Series devices are supported by Actel's Designer Series Development software which provides a seamless integration into any ASIC design flow. The Designer Series development tools offer automatic placement and routing (even with pre-assigned pins), static timing analysis, user programming, and debug and diagnostic probe capabilities. In addition, the DirectTime tool provides deterministic as well as controllable timing. DirectTime allows the designer to specify the performance requirements of individual paths and system clock(s). Using these specifications, the software will automatically optimize the placement and routing of the logic to meet these constraints. Included with the Designer Series tools is Actel's ACTGen™ Macro Builder. ACTGen allows the designer to quickly build fast, efficient logic functions such as counters, adders, FIFOs, and RAM.

The Designer Series tools provide designers the capability to move up to High-Level Description Languages, such as VHDL and Verilog, or use schematic design entry with interfaces to most EDA tools. Designer Series is supported on the following development platforms: 486 and Pentium PC, Sun® and HP® workstations. The software provides CAE interfaces to Cadence, Mentor Graphics®, Escalade, OrCAD™ and Viewlogic® design environments. Additional development tools are supported through Actel's Industry Alliance Program, including DATA I/O (ABEL FPGA) and MINC.

Actel's FPGAs are an ideal solution for shortening the system design and development cycle and offers a cost-effective alternative for low volume production runs. The 3200DX and 1200XL devices are an excellent choice for integrating logic that is currently implemented in multiple PALs, CPLDs and FPGAs. Some example applications include high-speed controllers and address decoding, peripheral bus interfaces, DSP, and co-processor functions.

Plastic Device Resources

	User I/Os									
Device	PLCC 84-pin	VQFP 100-pin	PQFP 100-pin	PQFP 144-pin	PQFP 160-pin	PQFP 208-pin	RQFP 240-pin	TQFP 176-pin	SBGA 240-pin	SBGA 432-pin
A1225XL	72	83	83	—	—	—	—	—	—	—
A1240XL	72	—	83	104	—	—	—	103	—	—
A3265DX	72	—	—	—	125	—	—	126	—	—
A1280XL	72	—	—	—	125	140	—	140	—	—
A32100DX	72	—	—	—	125	152	—	150	152	—
A32140DX	—	—	—	—	125	176	—	150	176	—
A32200DX	—	—	—	—	—	176*	202	—	—	202
A32300DX	—	—	—	—	—	—	—	—	—	250
A32400DX	—	—	—	—	—	—	—	—	—	288

Package Definitions *(Consult your local Actel Sales Representative for product availability.)*

PLCC = Plastic Leaded Chip Carrier, PQFP = Plastic Quad Flat Pack, TQFP = Thin Quad Flat Pack, BGA = Ball Grid Array, VQFP = Very Thin Quad Flat Pack, RQFP = Plastic Power Quad Flat Pack

• Also available in RQFP 208-pin.

Hermetic Device Resources

	User I/Os	
Device	CPGA 176-pin	CQFP 172-pin
A1280XL	140	140

Package Definitions *(Consult your local Actel Sales Representative for product availability.)*

CPGA = Ceramic Pin Grid Array, CQFP = Ceramic Quad Flat Pack

Pin Description

CLKA, CLKB Clock A and Clock B (input)

TTL Clock inputs for clock distribution networks. The Clock input is buffered prior to clocking the logic modules. This pin can also be used as an I/O.

DCLK Diagnostic Clock (Input)

TTL Clock input for diagnostic probe and device programming. DCLK is active when the MODE pin is HIGH. This pin functions as an I/O when the MODE pin is LOW.

GND Ground (Input)

Input LOW supply voltage.

I/O Input/Output (Input, Output)

I/O pin functions as an input, output, three-state or bi-directional buffer. Input and output levels are compatible with standard TTL and CMOS specifications. Unused I/O pins are automatically driven LOW by the Designer Series software.

MODE Mode (Input)

The MODE pin controls the use of multi-function pins (DCLK, PRA, PRB, SDI, TDO). When the MODE pin is HIGH, the special functions are active. To provide Actionprobe capability, the MODE pin should be terminated to GND through a 10K resistor so that the MODE pin can be pulled high when required.

NC No Connection

This pin is not connected to circuitry within the device.

PRA/I/O Probe A (Output)

The Probe A pin is used to output data from any user-defined design node within the device. This independent diagnostic pin is used in conjunction with the Probe B pin to allow real-time diagnostic output of any signal path within the device. The Probe A pin can be used as a user-defined I/O when debugging has been completed. The pin's probe capabilities can be permanently disabled to protect programmed design confidentiality. PRA is active when the MODE pin is HIGH. This pin functions as an I/O when the MODE pin is LOW.

PRB/I/O Probe B (Output)

The Probe B pin is used to output data from any user-defined design node within the device. This independent diagnostic pin is used in conjunction with the Probe A pin to allow real-time diagnostic output of any signal path within the device. The Probe B pin can be used as a user-defined I/O when debugging has been completed. The pin's probe capabilities can be permanently disabled to protect programmed design confidentiality. PRB is active when the MODE pin is HIGH. This pin functions as an I/O when the MODE pin is LOW.

QCLKA/B,C,D Quadrant Clock (Input/Output)

These four pins are the quadrant clock inputs. When not used as a register control signal, these pins can function as general purpose I/O.

SDI Serial Data Input (Input)

Serial data input for diagnostic probe and device programming. SDI is active when the MODE pin is HIGH. This pin functions as an I/O when the MODE pin is LOW.

TCK Test Clock

Clock signal to shift the JTAG data into the device. This pin functions as an I/O when the JTAG fuse is not programmed.

TDI Test Data In

Serial data input for JTAG instructions and data. Data is shifted in on the rising edge of TCLK. This pin functions as an I/O when the JTAG fuse is not programmed.

TDO Test Data Out

Serial data output for JTAG instructions and test data. This pin functions as an I/O when the JTAG fuse is not programmed.

TMS Test Mode Select

Serial data input for JTAG test mode. Data is shifted in on the rising edge of TCLK. This pin functions as an I/O when the JTAG fuse is not programmed.

V_{CC} Supply Voltage (Input)

Input HIGH supply voltage.

Note: *TCK, TDI, TDO, TMS are only available on devices containing JTAG circuitry.*

Integrator Series Architectural Overview

The 1200XL and 3200DX architecture is composed of fine-grained building blocks which produce fast, efficient logic designs. All devices within the Integrator Series are composed of Logic Modules, Routing Resources, Clock Networks, and I/O modules which are the building blocks to design fast logic designs. In addition, a subset of devices contain embedded dual-port SRAM and wide decode modules. The dual-port SRAM modules are optimized for high-speed data path functions such as FIFOs, LIFOs, and scratchpad memory. "Integrator Series Product Profile" on page 1-7 lists the specific logic resources contained within each device.

Logic Modules

3200DX and 1200XL devices contain three types of logic modules: combinatorial (C-modules), sequential (S-modules), and decode (D-modules). 1200XL devices contain only the C-module and S-module, while the 3200DX devices contain D-modules and dual-port SRAM modules; in addition to the S-module and C-module.

The C-module is shown in Figure 1 and implements the following function:

Y=!S1*!S0*D00+!S1*S0*D01+S1*!S0*D01+S1*S0*D11

where:

S0=A0*B0

S1=A1+B1

The S-module shown in Figure 2 is designed to implement high-speed sequential functions within a single logic module. The S-module implements the same combinatorial logic function as the C-module while adding a sequential element. The sequential element can be configured as either a D flip-flop or a transparent latch. To increase flexibility, the S-module register can be by-passed so that it implements purely combinatorial logic.

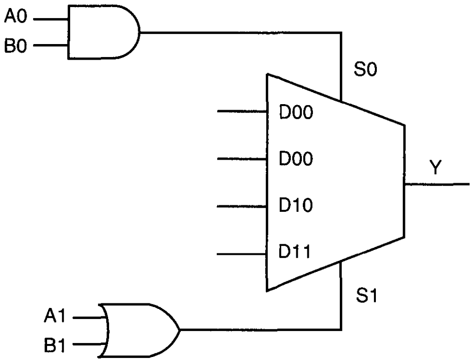


Figure 1 *• C-module Implementation*

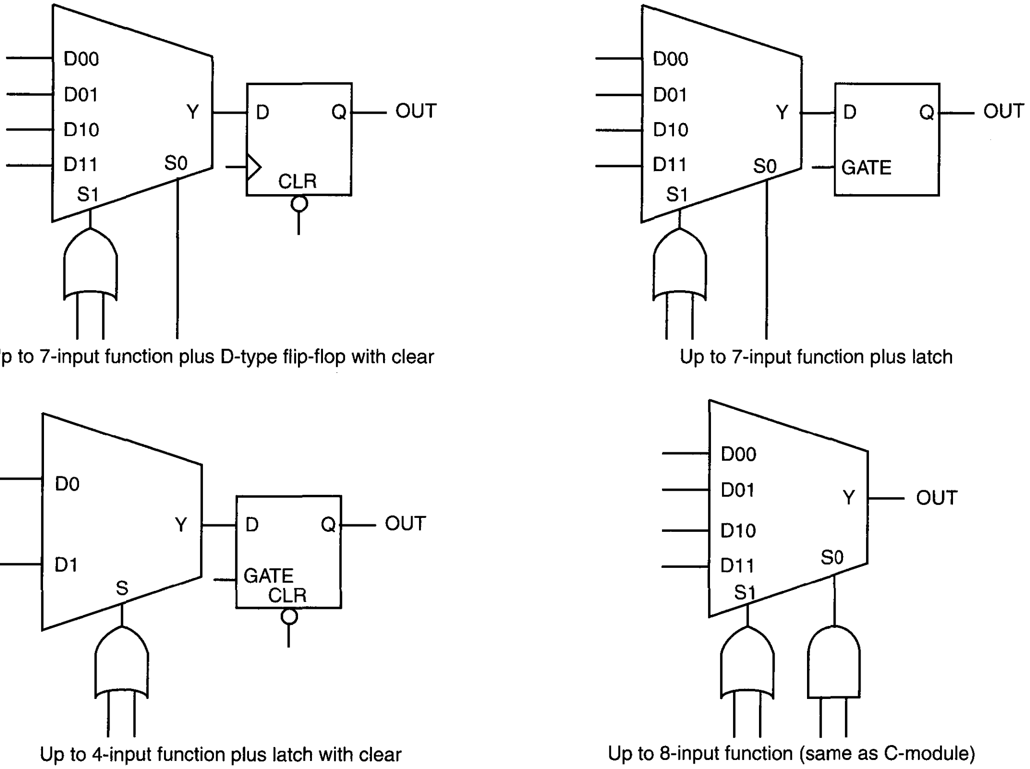

D00
D01
D10
D11
S1
S0
Y
D
Q
OUT
CLR
Up to 7-input function plus D-type flip-flop with clear
D00
D01
D10
D11
S1
S0
Y
D
Q
OUT
GATE
Up to 7-input function plus latch
D0
D1
S
Y
D
Q
OUT
GATE
CLR
Up to 4-input function plus latch with clear
D00
D01
D10
D11
S1
S0
Y
OUT
Up to 8-input function (same as C-module)

Figure 2 *• S-module Implementation*

3200DX devices contain a third type of logic module, D-modules, which are arranged around the periphery of device. D-modules contain wide decode circuitry which provides a fast, wide-input AND function similar to that found in product term architectures (Figure 3). The D-module allows 3200DX devices to perform wide decode functions at speeds comparable CPLDs and PAL devices. The output of the D-module has a programmable inverter for active HIGH or LOW assertion. The D-module output is hardwired to an output pin or can be fed back into the array to be incorporated into other logic.

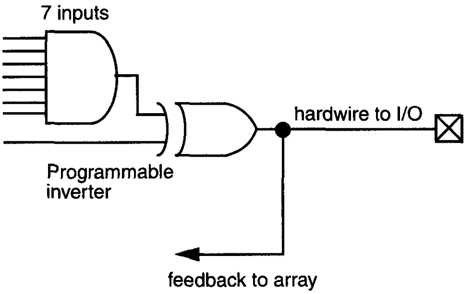


Figure 3 • *D-Module Implementation*

Dual-Port SRAM Modules

Several 3200DX devices contain dual-port SRAM modules that have been optimized for synchronous or asynchronous applications. The SRAM modules are arranged in 256 bit blocks which can be configured as 32 x 8 or 64 x 4 (refer to "Integrator Series Product Profile" on page 7 for the number of SRAM blocks within a particular device). SRAM modules can be cascaded together to form memory spaces of user-definable width and depth. A block diagram of the 3200DX dual-port SRAM block is shown in Figure 4.

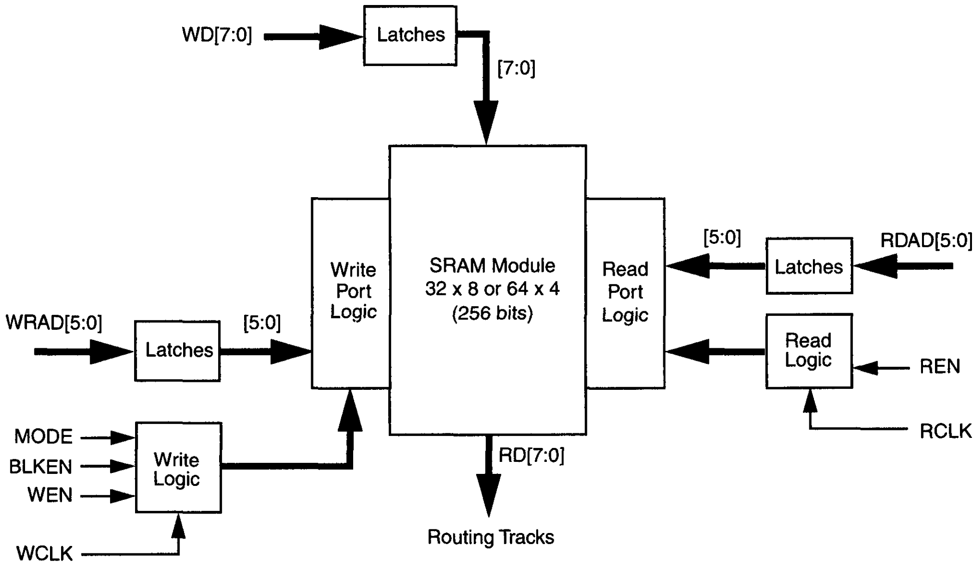


Figure 4 • *3200DX Dual-Port SRAM Block*

The 3200DX SRAM modules are true dual-port structures containing independent READ and WRITE ports. Each SRAM module contains six bits of read and write addressing (RDAD[5:0] and WRAD[5:0] respectively) for 64x4 bit blocks. When configured in byte mode, the highest order address bits (RDAD5 and WRAD5) are not used. The read and write ports of the SRAM block contain independent clocks (RCLK and WCLK) with programmable polarities offering active HIGH or LOW implementation. The SRAM block contains eight data inputs (WD[7:0]) and eight outputs (RD[7:0]) which are connected to segmented vertical routing tracks.

The 3200DX dual-port SRAM blocks are ideal for high-speed buffered applications requiring fast FIFO and LIFO queues. Actel's ACTGen Macro Builder provides the capability to quickly design memory functions, such as FIFOs, LIFOs, and

RAM arrays. Additionally, unused SRAM blocks need not be wasted since they can be used to implement registers for other logic within the design.

I/O Modules

The I/O modules provide the interface between the device pins and the logic array. Figure 5 is a block diagram of the I/O module. A variety of user functions, determined by a library macro selection, can be implemented in the module (refer to the Macro Library Guide for more information). I/O modules contain a tri-state buffer, input and output latches which can be configured for input, output, or bi-directional pins (Figure 5).

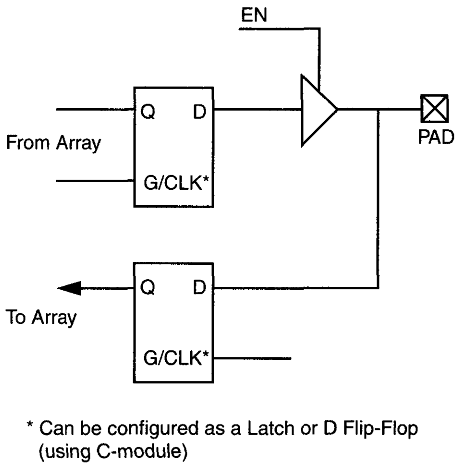

EN
Q
D
From Array
G/CLK*
PAD
Q
D
To Array
G/CLK*
* Can be configured as a Latch or D Flip-Flop (using C-module)

Figure 5 • *I/O Module*

The Integrator Series devices contain flexible I/O structures in that each output pin has a dedicated output enable control. The I/O module can be used to latch input and/or output data, providing a fast setup time. In addition, the Actel Designer software tools can build a D flip-flop, using a C-module, to register input and/or output signals.

Actel's Designer Series development tools provide a design library of I/O macros. The I/O macro library provides macrofunctions which can implement all I/O configurations supported by the Integrator Series FPGAs.

Routing Structure

The Integrator Series architecture uses Vertical and Horizontal routing tracks to interconnect the various logic and I/O modules. These routing tracks are metal interconnects that may either be of continuous length or broken into pieces called segments. Varying segment lengths allows the interconnect of over 90% of design tracks to occur with only two antifuse connections. Segments can be joined together at the ends, using antifuses, to increase their lengths up to the full length of the track. All interconnects can be accomplished with a maximum of four antifuses.

Horizontal Routing

Horizontal channels are located between the rows of modules and are composed of several routing tracks. The horizontal routing tracks within the channel are divided into one or more segments. The minimum horizontal segment length is the width of a module-pair, and the maximum horizontal segment length is the full length of the channel. Any segment that spans more than one-third the row length is considered a long horizontal segment. A typical channel is shown in Figure 6. Non-dedicated horizontal routing tracks are used to route signal nets. Dedicated routing tracks are used for the global clock networks and for power and ground tie-off tracks.

Vertical Routing

Another set of routing tracks run vertically through the module. Vertical tracks are of three types: input, output, and long. Vertical tracks are also divided into one or more segments. Each segment in an input track is dedicated to the input of a particular module. Each segment in an output track is dedicated to the output of a particular module. Long segments are uncommitted and can be assigned during routing. Each output segment spans four channels (two above and two below), except near the top and bottom of the array where edge effects occur. LVTs contain either one or two segments. An example of vertical routing tracks and segments is shown in Figure 6.

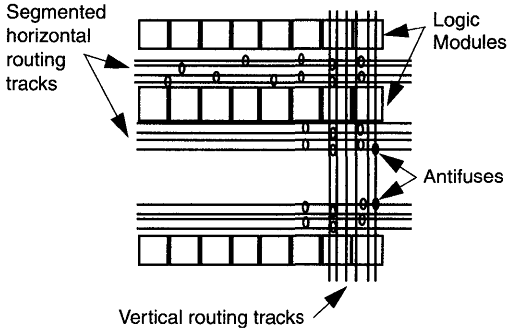


Figure 6 • *Routing Structure*

Antifuse Structures

An antifuse is a "normally open" structure as opposed to the normally closed fuse structure used in PROMs or PALs. The use of antifuses to implement a Programmable Logic Device results in highly testable structures as well as efficient programming algorithms. The structure is highly testable because there are no pre-existing connections; therefore, temporary connections can be made using pass transistors. These temporary connections can isolate individual antifuses to be programmed as well as isolate individual circuit structures to be tested. This can be done both before and after programming. For example, all metal tracks can be tested for continuity and shorts between adjacent tracks, and the functionality of all logic modules can be verified.

Clock Networks

Two low-skew, high fanout clock distribution networks are provided in each 3200DX device. These networks are referred to as CLK0 and CLK1. Each network has a clock module (CLKMOD) that selects the source of the clock signal and may be driven as follows:

1. Externally from the CLKA pad
2. Externally from the CLKB pad
3. Internally from the CLKINA input
4. Internally from the CLKINB input

The clock modules are located in the top row of I/O modules. Clock drivers and a dedicated horizontal clock track are located in each horizontal routing channel.

The user controls the clock module by selecting one of two clock macros from the macro library. The macro CLKBUF is used to connect one of the two external clock pins to a clock network, and the macro CLKINT is used to connect an internally generated clock signal to a clock network. Since both clock networks are identical, the user does not care whether CLK0 or CLK1 is being used. The clock input pads may also be used as normal I/Os, bypassing the clock networks (see Figure 7).

The 3200DX devices which contain SRAM modules (all except A3265DX and A32140DX) have four additional register control resources, called Quadrant Clock Networks (Figure 8). Each quadrant clock provides a local, high-fanout resource to the contiguous logic modules within its quadrant of the device. Quadrant clock signals can originate from specific I/O pins or from the internal array and can be used as a secondary register clock, register clear, or output enable.

Test Circuitry

Both 3200DX and 1200XL devices provide the means to test and debug a design once it is programmed into a device. 3200DX and 1200XL devices contain Actel's Actionprobe® test facility. Once a device has been programmed, the Actionprobe test facility allows the designer to probe any internal node during device operation to aid in debugging a design. In addition, 3200DX devices contain JTAG 1149.1 Boundary Scan Test.

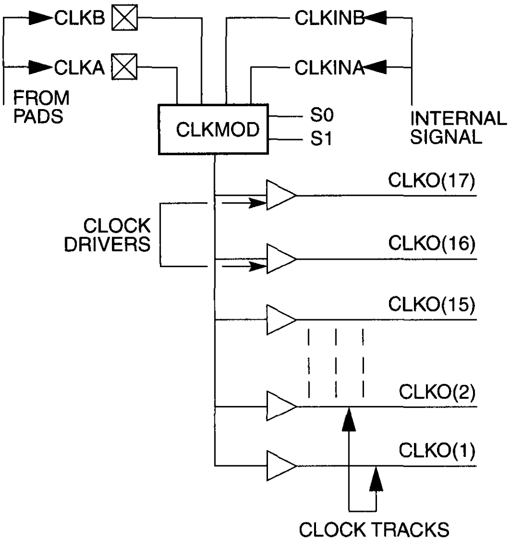


Figure 7 • *Clock Networks*

JTAG Boundary Scan Testing (BST)

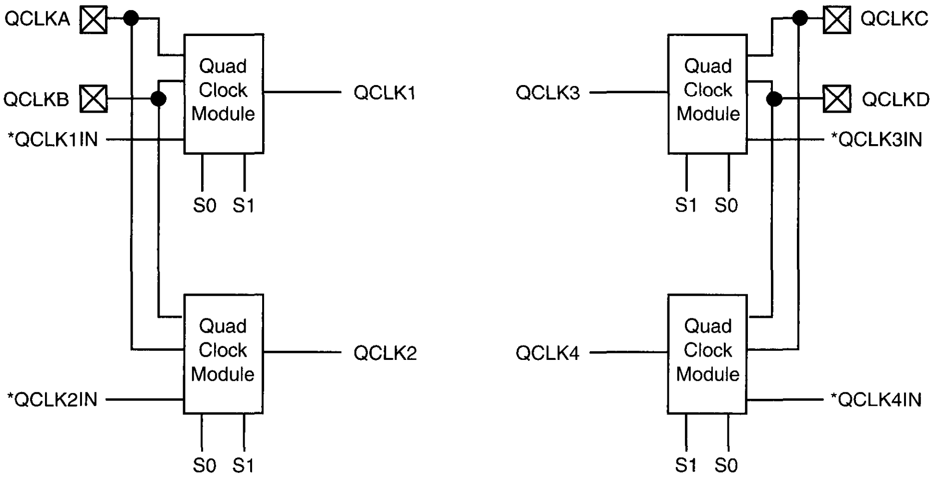
Device pin spacing is decreasing with the advent of fine-pitch packages such as TQFP and BGA packages and manufacturers are routinely implementing surface-mount technology with multi-layer PC boards. Boundary scan is becoming an attractive tool to help systems manufacturers test their PC boards. The Joint Test Action Group (JTAG) developed the IEEE Boundary Scan standard 1149.1 to facilitate board-level testing during manufacturing.

IEEE Standard 1149.1 defines a 4-pin Test Access Port (TAP) interface for testing integrated circuits in a system. The 3200DX family provides four JTAG BST pins: Test Data In (TDI), Test Data Out (TDO), Test Clock (TCLK) and Test Mode Select (TMS). Devices are configured in a JTAG "chain" where BST data can be transmitted serially between devices via TDO to TDI interconnections. The TMS and TCLK signals are shared between all devices in the JTAG chain so that all components operate in the same state.

The 3200DX family implements a subset of the IEEE 1149.1 Boundary Scan Test (BST) instruction in addition to a private instruction to allow the use of Actel's Actionprobe facility with JTAG BST. Refer to the IEEE 1149.1 specification for detailed information regarding JTAG testing.

JTAG Architecture

The 3200DX JTAG BST circuitry consist of a Test Access Port (TAP) controller, JTAG instruction register, JPROBE register, bypass register and boundary scan register. Figure 9 is a block diagram of the 3200DX JTAG circuitry.



*QCLK1IN, QCLK2IN, QCLK3IN, and QCKL4IN are internally generated signals.

Figure 8 • *Quadrant Clock Network*

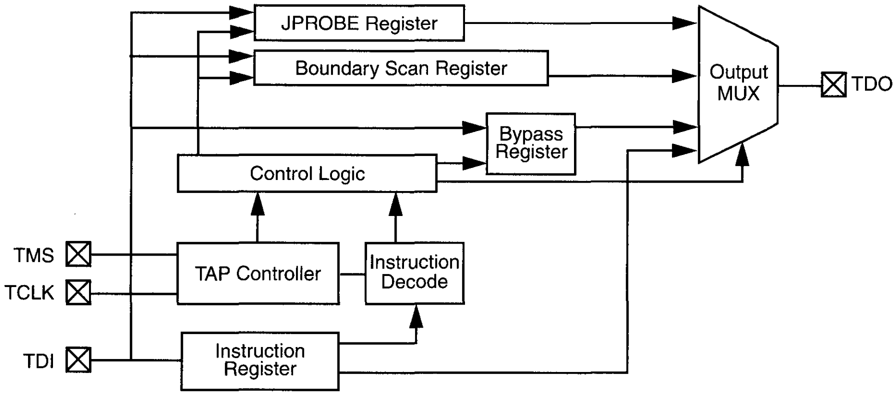


Figure 9 • *JTAG BST Circuitry*

When a device is operating in JTAG BST mode, four I/O pins are used for the TDI, TDO, TMS, and TCLK signals. An active reset (nTRST) pin is not supported, however the 3200DX contains power-on reset circuitry which resets the JTAG BST circuitry upon power-up. During normal device operation, the JTAG pins should be held LOW to disable the JTAG circuitry. The following table summarizes the functions of the JTAG BST signals.

JTAG Signal	Name	Function
TDI	Test Data In	Serial data input for JTAG instructions and data. Data is shifted in on the rising edge of TCLK.
TDO	Test Data Out	Serial data output for JTAG instructions and test data.
TMS	Test Mode Select	Serial data input for JTAG test mode. Data is shifted in on the rising edge of TCLK.
TCLK	Test Clock	Clock signal to shift the JTAG data into the device.

JTAG BST Instructions

JTAG BST testing within the 3200DX devices is controlled by a Test Access Port (TAP) state machine. The TAP controller drives the three-bit instruction register, a bypass register, and the boundary scan data registers within the device. The TAP controller uses the TMS signal to control the JTAG testing of the device. The JTAG test mode is determined by the bit stream entered on the TMS pin. The table in the next column describes the JTAG instructions supported by the 3200DX.

Actionprobe

If a device has been successfully programmed and the security fuse has not been programmed, any internal logic or I/O module output can be observed using the Actionprobe circuitry and the PRA and/or PRB pins. The Actionprobe diagnostic system provides the software and hardware required to perform real-time debugging. Refer to "Using the Actionprobe for System-Level Debug" application note on page 4-123 for further information.

Test Mode	Code	Description
EXTEST	000	Allows the external circuitry and board-level interconnections to be tested by forcing a test pattern at the output pins and capturing test results at the input pins.
SAMPLE/ PRELOAD	001	Allows a snapshot of the signals at the device pins to be captured and examined during device operation.
INTEST	010	Refer to IEEE 1149.1 *Specification*
JPROBE	011	A private instruction allowing the user to connect Actel's Micro Probe registers to the JTAG chain.
USER INSTRUCTION	100	Allows the user to build application-specific instructions such as RAM READ and RAM WRITE.
HIGH Z	101	Refer to IEEE 1149.1 *Specification*
CLAMP	110	Refer to IEEE 1149.1 Specification
BYPASS	111	Enables the by bypass register between the TDI and TDO pins. The test data passes through the selected device to adjacent devices in the JTAG chain.

5.0V Operating Conditions

Absolute Maximum Ratings[1]

Free air temperature range

Symbol	Parameter	Limits	Units
V_{CC}	DC Supply Voltage	–0.5 to +7.0	V
V_I	Input Voltage	–0.5 to V_{CC} +0.5	V
V_O	Output Voltage	–0.5 to V_{CC} +0.5	V
I_{IO}	I/O Source/Sink Current[2]	±20	mA
T_{STG}	Storage Temperature	–65 to +150	°C

Notes:

1. *Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum rated conditions for extended periods may affect device reliability. Device should not be operated outside the Recommended Operating Conditions.*
2. *Device inputs are normally high impedence and draw extremely low current. However, when input voltage is greater than V_{CC} + 0.5 V or less than GND – 0.5 V, the internal protection diode will be forward biased and can draw excessive current.*

Recommended Operating Conditions

Parameter	Commercial	Industrial	Military	Units
Temperature Range[1]	0 to +70	–40 to +85	–55 to +125	°C
Power Supply Tolerance	±5	±10	±10	%V_{CC}

Note:

1. *Ambient temperature (T_A) is used for commercial and industrial; case temperature (T_C) is used for military.*

Electrical Specifications

Symbol	Parameter	Commercial		Industrial		Military		Units
		Min.	Max.	Min.	Max.	Min.	Max.	
V_{OH}[1]	(I_{OH} = –10 mA)[2]	2.4						V
	(I_{OH} = –6 mA)	3.84						V
	(I_{OH} = –4 mA)			3.7		3.7		V
V_{OL}[1]	(I_{OL} = 10 mA)[2]		0.5					V
	(I_{OL} = 6 mA)		0.33		0.40		0.40	V
V_{IL}		–0.3	0.8	–0.3	0.8	–0.3	0.8	V
V_{IH}		2.0	V_{CC} + 0.3	2.0	V_{CC} + 0.3	2.0	V_{CC} + 0.3	V
Input Transition Time t_R, t_F[2]			500		500		500	ns
C_{IO} I/O Capacitance[2, 3]			10		10		10	pF
Standby Current, I_{CC}[4] (typical = 1 mA)			1.5		10		20	mA
$I_{CC(D)}$ Dynamic V_{CC} Supply Current		See "Power Dissipation" on page 1-21						

Notes:

1. *Only one output tested at a time. V_{CC} = min.*
2. *Not tested, for information only.*
3. *Includes worst-case 176 CPGA package capacitance. V_{OUT} = 0 V, f = 1 MHz.*
4. *All outputs unloaded. All inputs = V_{CC} or GND, typical I_{CC} = 1 mA. I_{CC} limit includes I_{PP} and I_{SV} during normal operation.*

3.3V Operating Conditions

Absolute Maximum Ratings[1]

Free air temperature range

Symbol	Parameter	Limits	Units
V_{CC}	DC Supply Voltage	–0.5 to +7.0	V
V_I	Input Voltage	–0.5 to V_{CC} +0.5	V
V_O	Output Voltage	–0.5 to V_{CC} +0.5	V
I_{IO}	I/O Source Sink Current[2]	±20	mA
T_{STG}	Storage Temperature	–65 to +150	°C

Notes:

1. *Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. Exposure to absolute maximum rated conditions for extended periods may affect device reliability. Device should not be operated outside the Recommended Operating Conditions.*
2. *Device inputs are normally high impedance and draw extremely low current. However, when input voltage is greater than V_{CC} + 0.5 V or less than GND – 0.5 V, the internal protection diodes will forward bias and can draw excessive current.*

Recommended Operating Conditions

Parameter	Commercial	Units
Temperature Range[1]	0 to +70	°C
Power Supply Tolerance	±5	%V

Note:

1. *Ambient temperature (T_A) is used for commercial.*

Electrical Specifications

Parameter		Commercial		Units
		Min.	Max.	
V_{OH}[1]	(I_{OH} = –4 mA)	2.15		V
	(I_{OH} = –3.2 mA)	2.4		V
V_{OL}[1]	(I_{OL} = 6 mA)		0.4	V
V_{IL}		–0.3	0.8	V
V_{IH}		2.0	V_{CC} + 0.3	V
Input Transition Time t_R, t_F[2]			500	ns
C_{IO} I/O Capacitance[2, 3]			10	pF
Standby Current, I_{CC}[4] (typical = 0.3 mA)			0.75	mA
$I_{CC(D)}$ Dynamic V_{CC} Supply Current		See “Power Dissipation” on page 1-21		

Notes:

1. *Only one output tested at a time. V_{CC} = min.*
2. *Not tested, for information only.*
3. *Includes worst-case 84-pin PLCC package capacitance. V_{OUT} = 0 V, f = 1 MHz.*
4. *Typical standby current = 0.3 mA. All outputs unloaded. All inputs = V_{CC} or GND.*

Package Thermal Characteristics

The device junction to case thermal characteristic is θjc, and the junction to ambient air characteristic is θja. The thermal characteristics for θja are shown with two different air flow rates.

Maximum junction temperature is 150°C.

A sample calculation of the absolute maximum power dissipation allowed for a PQFP 160-pin package at commercial temperature is as follows:

$$\frac{\text{Max. junction temp. (°C)} - \text{Max. commercial temp.}}{\theta\text{ja (°C/W)}} = \frac{150°\text{C} - 70°\text{C}}{30°\text{C/W}} = 2.6\text{W}$$

Package Type	Pin Count	θja		Maximum Power Dissipation	
		Still Air	300 ft/min	Still Air	300 ft/min
Plastic Quad Flatpack	160	36 °C/W	30 °C/W	2.2 W	2.6 W
Plastic Quad Flatpack	208	25 °C/W	16.2 °C/W	3.2 W	4.9 W
Plastic Leaded Chip Carrier	84	37 °C/W	28 °C/W	2.2 W	2.9 W
Thin Quad Flatpack	176	32 °C/W	25 °C/W	2.5 W	3.2 W
Power Quad Flatpack	208	16.8 °C/W	11.4 °C/W	4.8 W	7.0 W
Power Quad Flatpack	240	16.1 °C/W	10.6 °C/W	5.0 W	7.5 W
Ball Grid Array	240	14.0 °C/W	10.0 °C/W	5.7 W	8.0 W
Ball Grid Array	432	10.0 °C/W	8.0 ° C/W	8.0 W	10.0 W

Power Dissipation

General Power Equation

$$P = [I_{CC}\text{standby} + I_{CC}\text{active}] * V_{CC} + I_{OL} * V_{OL} * N + I_{OH} * (V_{CC} - V_{OH}) * M$$

Where:

I_{CC}standby is the current flowing when no inputs or outputs are changing.

I_{CC}active is the current flowing due to CMOS switching.

I_{OL}, I_{OH} are TTL sink/source currents.

V_{OL}, V_{OH} are TTL level output voltages.

N equals the number of outputs driving TTL loads to V_{OL}.

M equals the number of outputs driving TTL loads to V_{OH}.

An accurate determination of N and M is problematic because their values depend on the family type, design details, and on the system I/O. The power can be divided into two components: static and active.

Static Power Component

Actel FPGAs have small static power components that result in lower power dissipation than PALs or PLDs. By integrating multiple PALs/PLDs into one FPGA, an even greater reduction in board-level power dissipation can be achieved.

The power due to standby current is typically a small component of the overall power. Standby power is calculated below for commercial, worst case conditions.

I_{CC}	V_{CC}	Power
2 mA	5.25 V	10.5 mW

The static power dissipation by TTL loads depends on the number of outputs driving high or low and the DC load current. Again, this number is typically small. For instance, a 32-bit bus sinking 4 mA at 0.33 V will generate 42 mW with all outputs driving low and 140 mW with all outputs driving high. The actual dissipation will average somewhere between as I/Os switch states with time.

Active Power Component

Power dissipation in CMOS devices is usually dominated by the active (dynamic) power dissipation. This component is frequency dependent, a function of the logic and the external I/O. Active power dissipation results from charging internal chip capacitances of the interconnect, unprogrammed antifuses, module inputs, and module outputs, plus external capacitance due to PC board traces and load device inputs. An additional component of the active power dissipation is the totem-pole current in the CMOS transistor pairs. The net effect can be associated with an equivalent capacitance that can be combined with frequency and voltage to represent active power dissipation.

Equivalent Capacitance

The power dissipated by a CMOS circuit can be expressed by Equation 1.

$$\text{Power } (\mu W) = C_{EQ} * V_{CC}^2 * F \qquad (1)$$

Where:

C_{EQ} is the equivalent capacitance expressed in picofarads (pF).

V_{CC} is power supply in volts (V).

F is the switching frequency in megahertz (MHz).

Equivalent capacitance is calculated by measuring $I_{CCactive}$ at a specified frequency and voltage for each circuit component of interest. Measurements have been made over a range of frequencies at a fixed value of V_{CC}. Equivalent capacitance is frequency independent so that the results may be used over a wide range of operating conditions. Equivalent capacitance values are shown below.

C_{EQ} Values for Actel FPGAs

Modules (C_{EQM})	5.2
Input Buffers (C_{EQI})	11.6
Output Buffers (C_{EQO})	23.8
Routed Array Clock Buffer Loads (C_{EQCR})	3.5

To calculate the active power dissipated from the complete design, the switching frequency of each part of the logic must be known. Equation 2 shows a piece-wise linear summation over all components.

$$\text{Power} = V_{CC}^2 * [(m \times C_{EQM} * f_m)_{Modules} +$$
$$(n * C_{EQI} * f_n)_{Inputs} + (p * (C_{EQO} + C_L) * f_p)_{outputs} +$$
$$0.5 * (q_1 * C_{EQCR} * f_{q1})_{routed\_Clk1} + (r_1 * f_{q1})_{routed\_Clk1} +$$
$$0.5 * (q_2 * C_{EQCR} * f_{q2})_{routed\_Clk2} + (r_2 * f_{q2})_{routed\_Clk2} \qquad (2)$$

Where:

m	=	Number of logic modules switching at frequency f_m
n	=	Number of input buffers switching at frequency f_n
p	=	Number of output buffers switching at frequency f_p
q_1	=	Number of clock loads on the first routed array clock
q_2	=	Number of clock loads on the second routed array clock
r_1	=	Fixed capacitance due to first routed array clock
r_2	=	Fixed capacitance due to second routed array clock
C_{EQM}	=	Equivalent capacitance of logic modules in pF
C_{EQI}	=	Equivalent capacitance of input buffers in pF
C_{EQO}	=	Equivalent capacitance of output buffers in pF
C_{EQCR}	=	Equivalent capacitance of routed array clock in pF
C_L	=	Output load capacitance in pF
f_m	=	Average logic module switching rate in MHz
f_n	=	Average input buffer switching rate in MHz
f_p	=	Average output buffer switching rate in MHz
f_{q1}	=	Average first routed array clock rate in MHz
f_{q2}	=	Average second routed array clock rate in MHz

Fixed Capacitance Values for Actel FPGAs (pF)

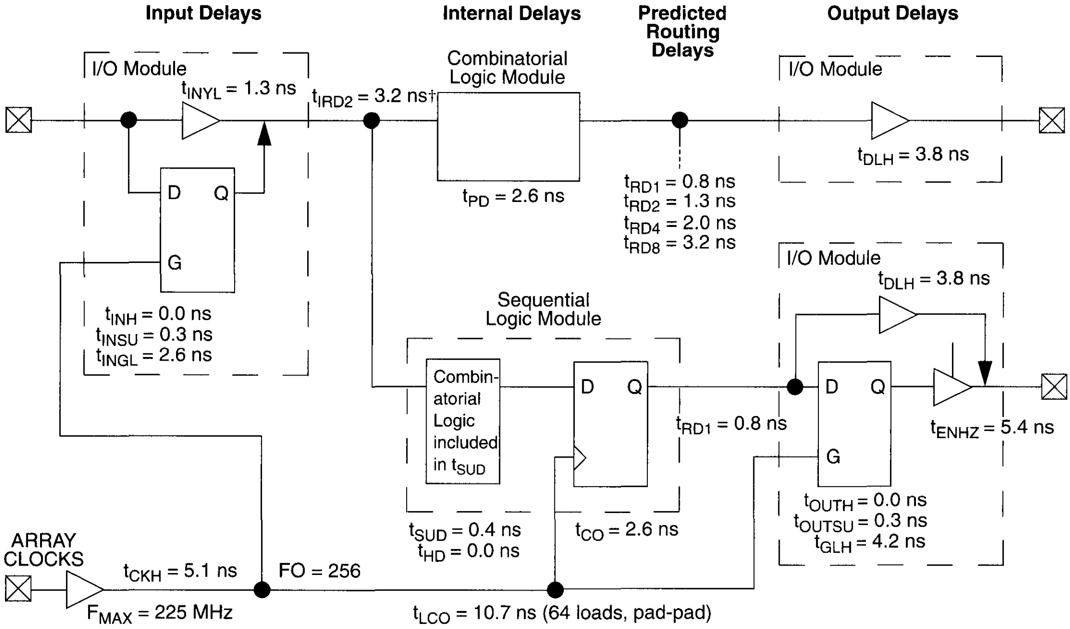
	r_1	r_2
Device Type	routed_Clk1	routed_Clk2
A1225XL	106	106
A1240XL	134	134
A3265DX	158	158
A1280XL	168	168
A32100DX	178	178
A32140DX	190	190
A32200DX	230	230
A32300DX	285	285

Determining Average Switching Frequency

To determine the switching frequency for a design, you must have a detailed understanding of the data input values to the circuit. The following guidelines are meant to represent worst-case scenarios so that they can be generally used to predict the upper limits of power dissipation. These guidelines are as follows:

Logic Modules (m)	= 80% of combinatorial modules
Inputs switching (n)	= # of inputs/4
Outputs switching (p)	= # outputs/4
First routed array clock loads (q_1)	= 40% of sequential modules
Second routed array clock loads (q_2)	= 40% of sequential modules
Load capacitance (C_L)	= 35 pF
Average logic module switching rate (f_m)	= F/10
Average input switching rate (f_n)	= F/5
Average output switching rate (f_p)	= F/10
Average first routed array clock rate (f_{q1})	= F
Average second routed array clock rate (f_{q2})	= F/2

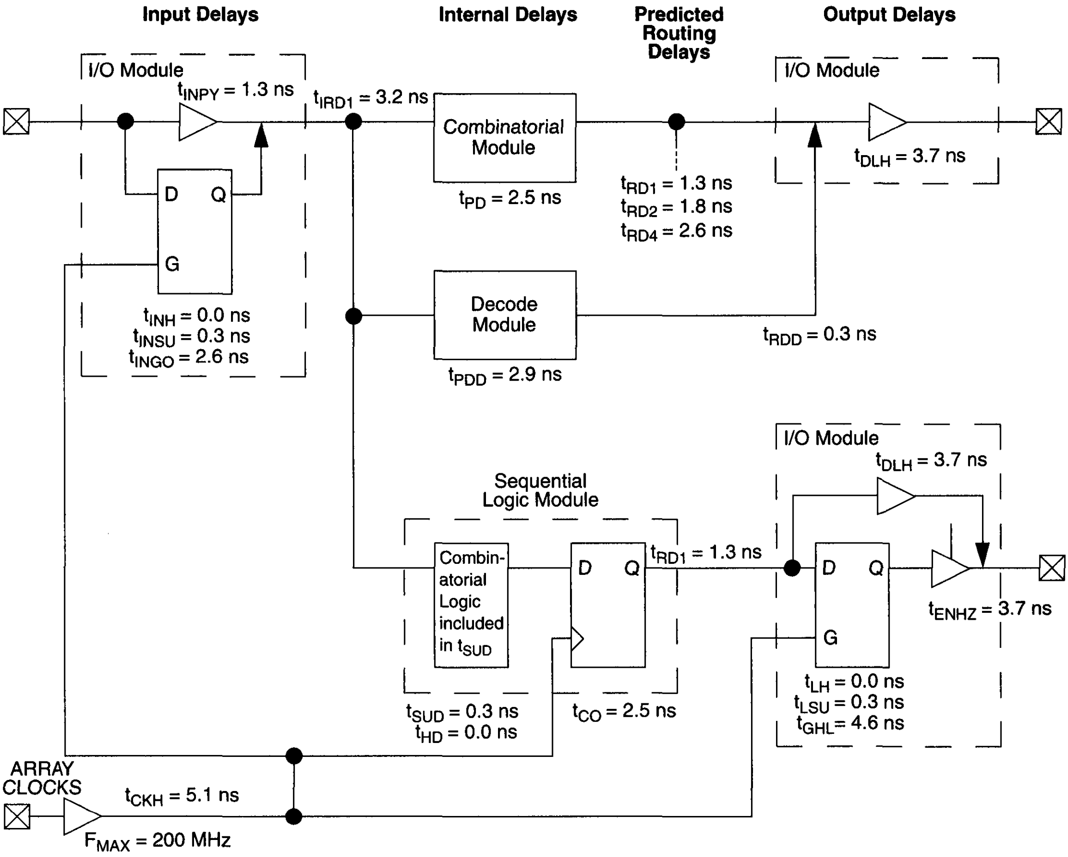
1200XL Timing Model*



*Values shown for A1225XL-2 at worst-case commercial conditions.

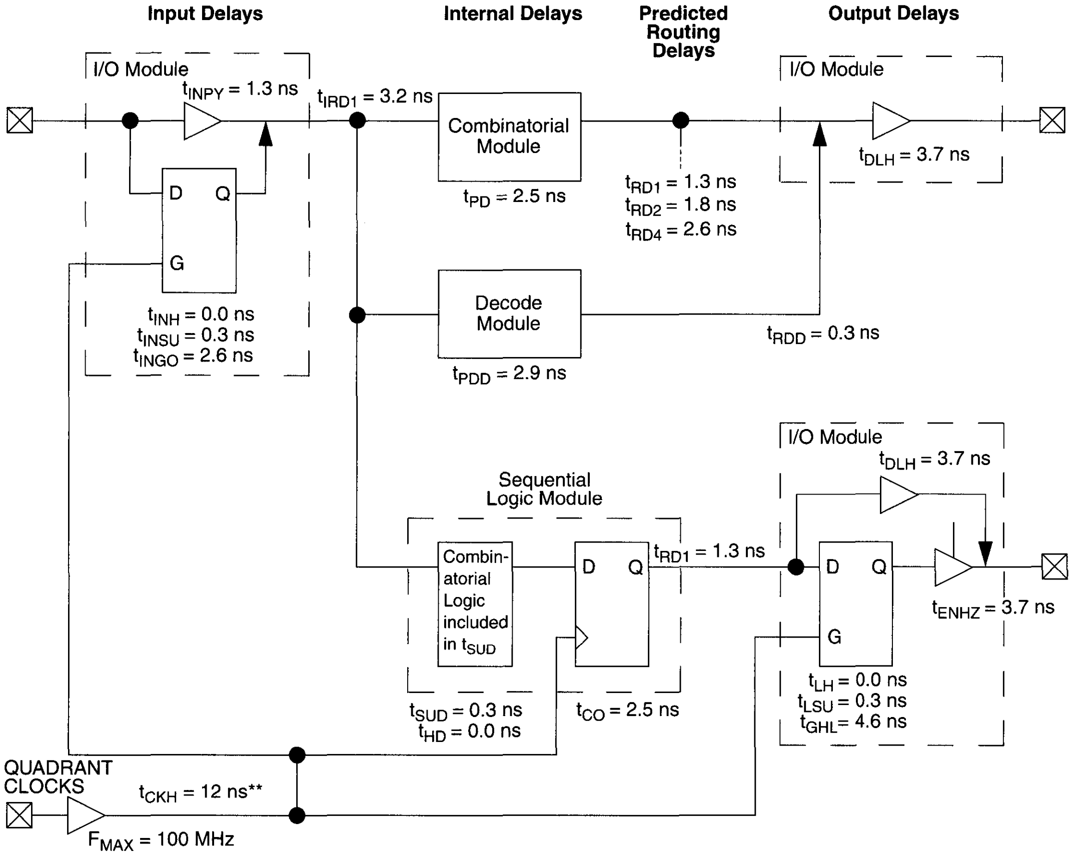
† Input Module Predicted Routing Delay

3200DX Timing Model (Logic Functions using Array Clocks)*



Values shown for A3265DX-2 at worst-case commercial conditions.

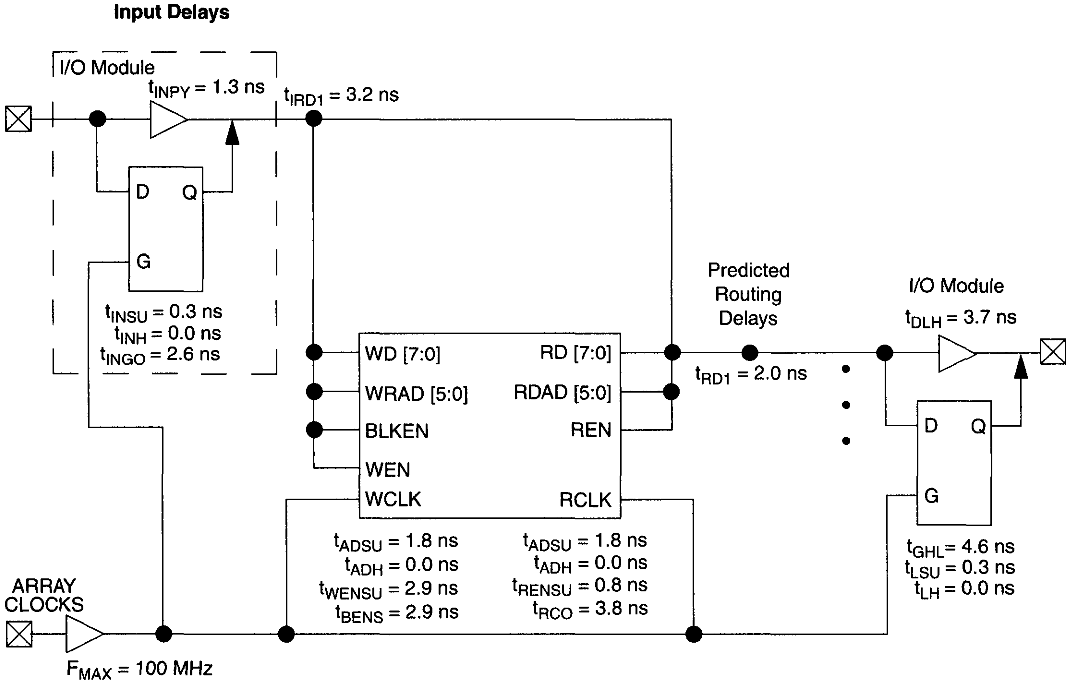
3200DX Timing Model (Logic Functions using Quadrant Clocks)*



** Preliminary values shown for A32200DX-2 at worst-case commercial conditions.*

*** Load dependent.*

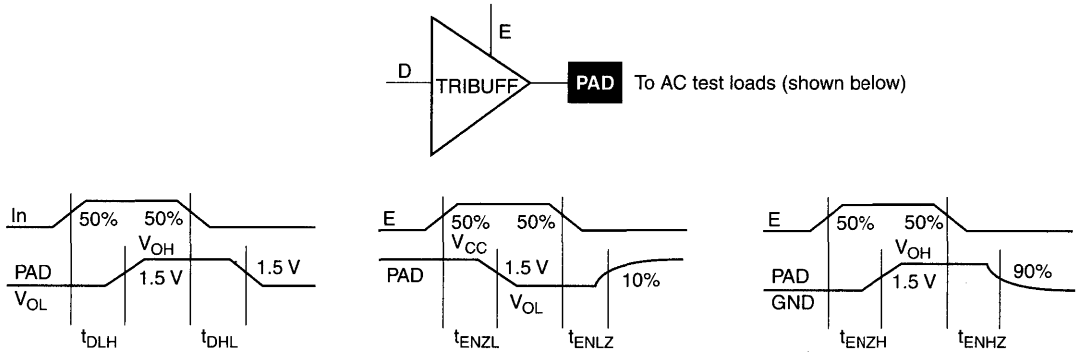
3200DX Timing Model (SRAM Functions)*



**Values shown for A32200DX-2 at worst-case commercial conditions.*

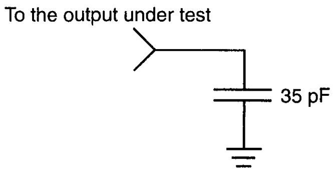
Parameter Measurement

Output Buffer Delays

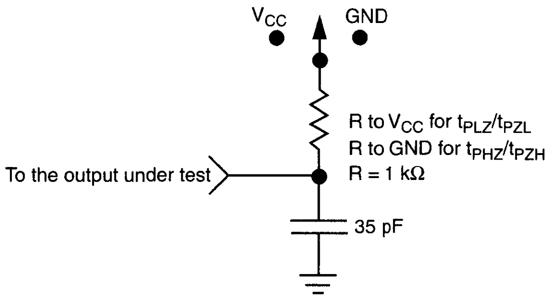


AC Test Loads

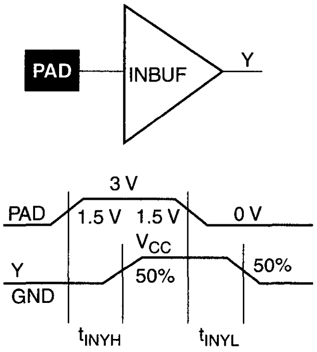
Load 1
(Used to measure propagation delay)



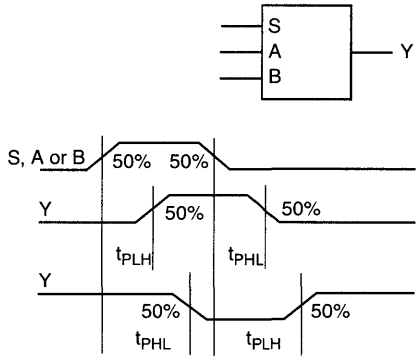
Load 2
(Used to measure rising/falling edges)


V_{CC}
GND
R to V_{CC} for t_{PLZ}/t_{PZL}
R to GND for t_{PHZ}/t_{PZH}
R = 1 kΩ
To the output under test
35 pF

Input Buffer Delays

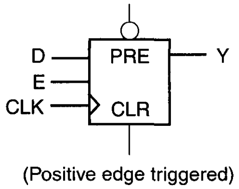


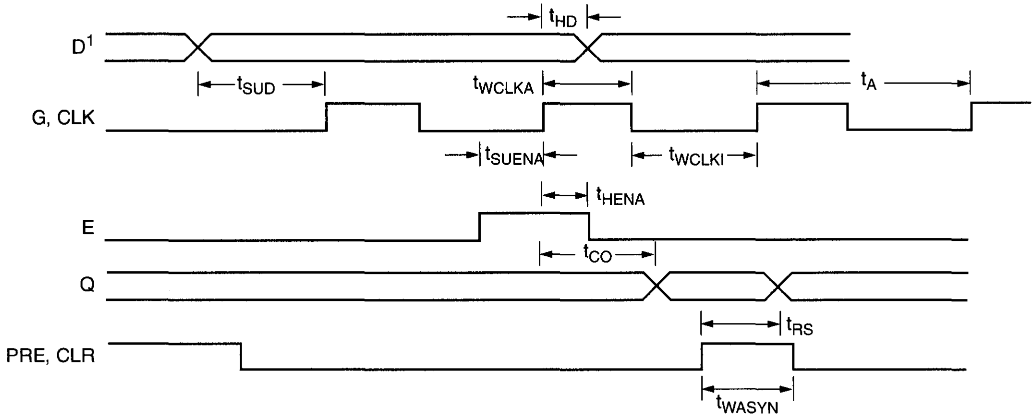
Module Delays



Sequential Module Timing Characteristics

Flip-Flops and Latches

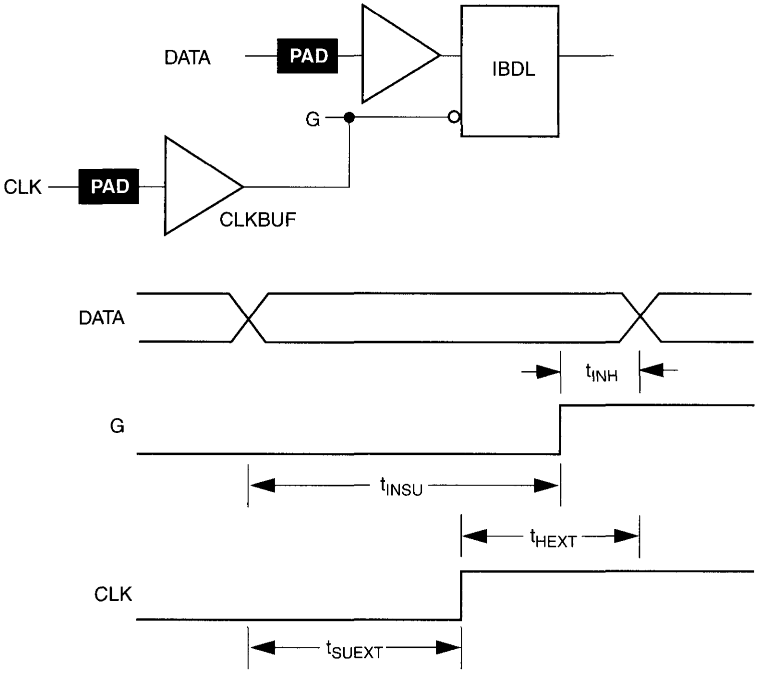




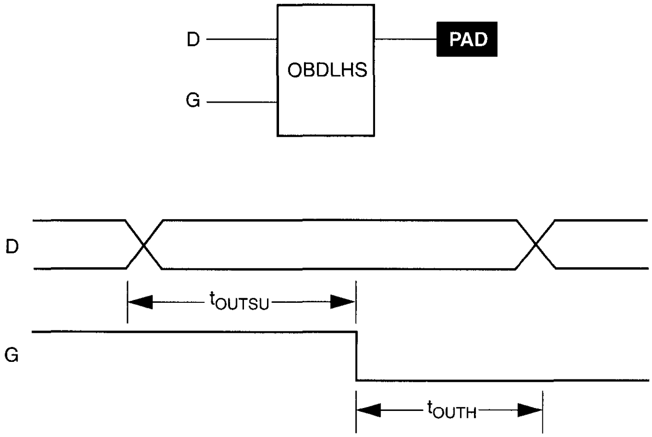
Note: *D represents all data functions involving A, B, and S for multiplexed flip-flops.*

Sequential Timing Characteristics (continued)

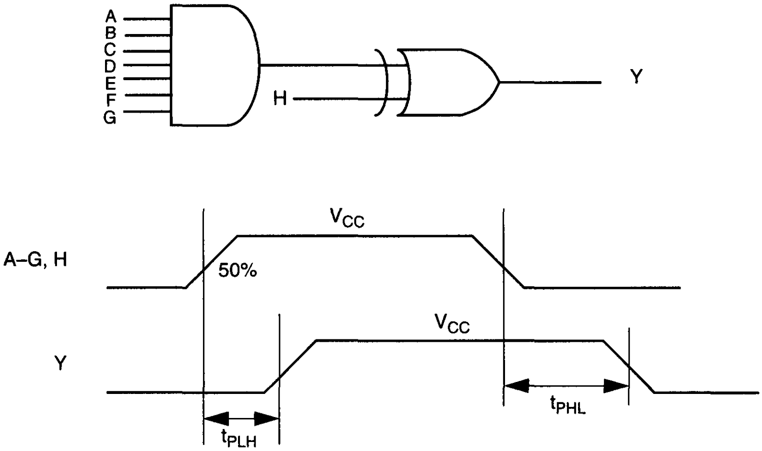
Input Buffer Latches



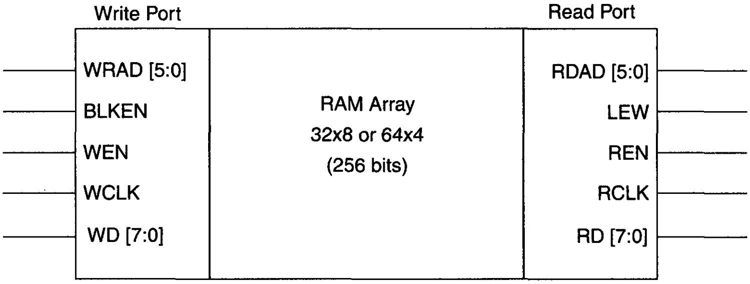
Output Buffer Latches



Decode Module Timing

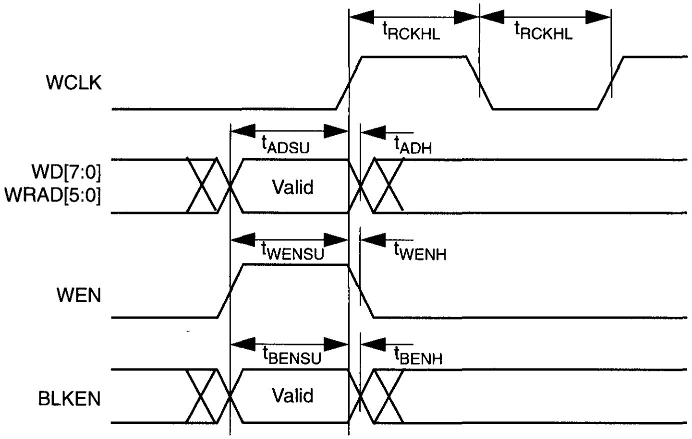


SRAM Timing Characteristics



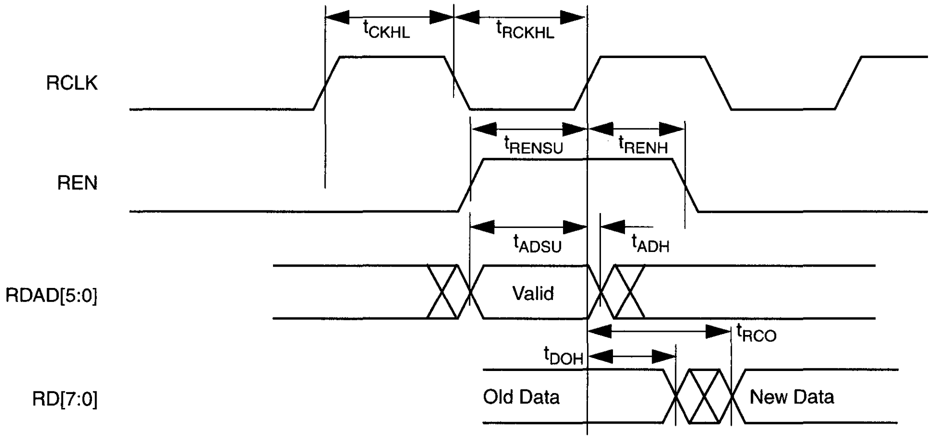
Dual-Port SRAM Timing Waveforms

3200DX SRAM Write Operation



Note: *Identical timing for falling-edge clock.*

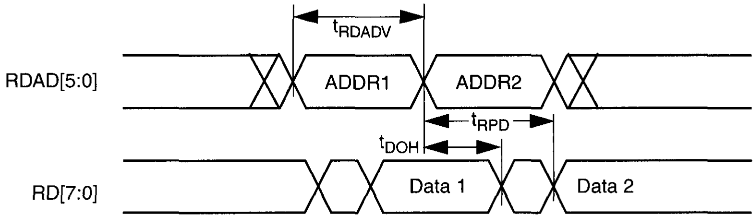
3200DX SRAM Synchronous Read Operation



Note: *Identical timing for falling-edge clock.*

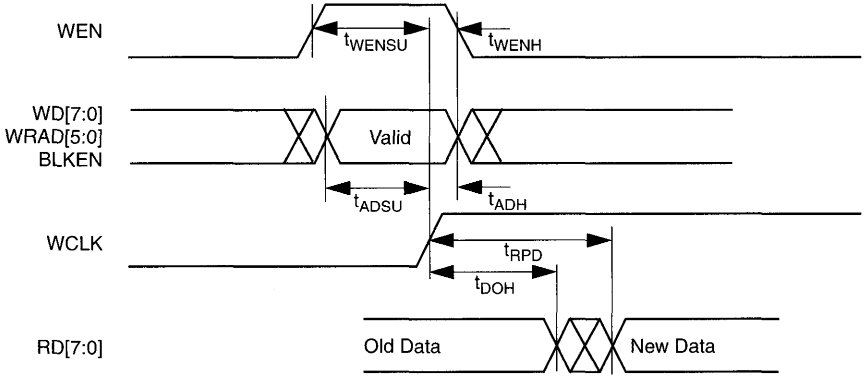
3200DX SRAM Asynchronous Read Operation—Type 1

(Read Address Controlled)



3200DX SRAM Asynchronous Read Operation—Type 2

(Write Address Controlled)



Predictable Performance: Tight Delay Distributions

Propagation delay between logic modules depends on the resistive and capacitive loading of the routing tracks, the interconnect elements, and the module inputs being driven. Propagation delay increases as the length of routing tracks, the number of interconnect elements, or the number of inputs increases.

From a design perspective, the propagation delay can be statistically correlated or modeled by the fanout (number of loads) driven by a module. Higher fanout usually requires some paths to have longer routing tracks.

The Integrator Series delivers a very tight fanout delay distribution. This tight distribution is achieved in two ways: by decreasing the delay of the interconnect elements and by decreasing the number of interconnect elements per path.

Actel's patented PLICE antifuse offers a very low resistive/capacitive interconnect. The antifuses, fabricated in 0.6 micron lithography, offer nominal levels of 100 ohms resistance and 7.0 femtofarad (fF) capacitance per antifuse.

The Integrator Series fanout distribution is also tight due to the low number of antifuses required for each interconnect path. The proprietary architecture limits the number of antifuses per path to a maximum of four, with 90% of interconnects using two antifuses.

Timing Characteristics

Timing characteristics for devices fall into three categories: family dependent, device dependent, and design dependent. The input and output buffer characteristics are common to all Integrator Series members. Internal routing delays are device dependent. Design dependency means actual delays are not determined until after placement and routing of the user's design is complete. Delay values may then be determined by using the Designer Series utility or performing simulation with post-layout delays.

Critical Nets and Typical Nets

Propagation delays are expressed only for typical nets, which are used for initial design performance evaluation. Since the architecture provides deterministic timing and abundant routing resources, Actel's Designer Series development tools offers DirectTime; a timing-driven place and route tool. Using DirectTime, the designer may specify timing-critical nets and system clock frequency. Using these timing specifications, the place and route software optimized the layout of the design to meet the user's specifications.

Long Tracks

Some nets in the design use long tracks. Long tracks are special routing resources that span multiple rows, columns, or modules. Long tracks employ three and sometimes four antifuse connections. This increases capacitance and resistance, resulting in longer net delays for macros connected to long tracks. Typically, up to 6% of nets in a fully utilized device require long tracks. Long tracks contribute approximately 3 ns to 6 ns delay. This additional delay is represented statistically in higher fanout (FO=8) routing delays in the data sheet specifications section.

Timing Derating

A best case timing derating factor of 0.45 is used to reflect best case processing. Note that this factor is relative to the "standard speed" timing parameters, and must be multiplied by the appropriate voltage and temperature derating factors for a given application.

Timing Derating Factor (Temperature and Voltage)

	Industrial		Military	
	Min.	Max.	Min.	Max.
(Commercial Specification) x	0.69	1.11	0.67	1.23

Timing Derating Factor for Designs at Typical Temperature (T_J = 25°C) and Voltage (5.0 V)

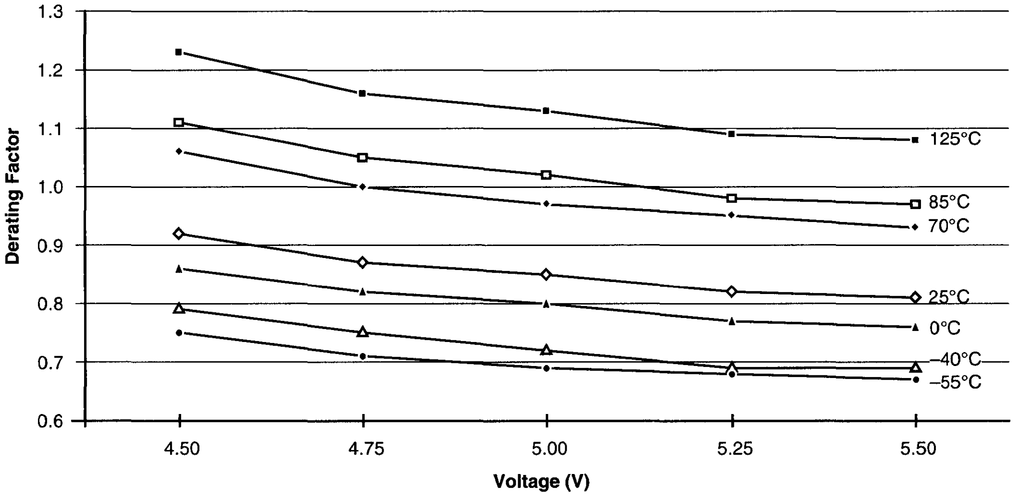
(Maximum Specification, Worst-Case Condition) x	0.85

Note: *This derating factor applies to all routing and propagation delays.*

Temperature and Voltage Derating Factors (normalized to Worst-Case Commercial, T_J = 4.75 V, 70°C)

	–55	–40	0	25	70	85	125
4.50	0.75	0.79	0.86	0.92	1.06	1.11	1.23
4.75	0.71	0.75	0.82	0.87	1.00	1.05	1.16
5.00	0.69	0.72	0.80	0.85	0.97	1.02	1.13
5.25	0.68	0.69	0.77	0.82	0.95	0.98	1.09
5.50	0.67	0.69	0.76	0.81	0.93	0.97	1.08

Junction Temperature and Voltage Derating Curves
(normalized to Worst-Case Commercial, T_J = 4.75 V, 70°C)



Note: *This derating factor applies to all routing and propagation delays.*

A1225XL Timing Characteristics

(Worst-Case Commercial Conditions, V_{CC} = 4.75 V, T_J = 70°C)

Logic Module Propagation Delays[1]		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed[5]		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{PD1}	Single Module		2.6		3.0		3.5		4.2	ns
t_{CO}	Sequential Clk to Q		2.6		3.0		3.5		4.2	ns
t_{GO}	Latch G to Q		2.6		3.0		3.5		4.2	ns
t_{RS}	Flip-Flop (Latch) Reset to Q		2.6		3.0		3.5		4.2	ns
Predicted Routing Delays[2]										
t_{RD1}	FO=1 Routing Delay		0.8		0.9		1.1		1.3	ns
t_{RD2}	FO=2 Routing Delay		1.3		1.4		1.7		2.0	ns
t_{RD3}	FO=3 Routing Delay		1.7		1.8		2.2		2.6	ns
t_{RD4}	FO=4 Routing Delay		2.0		2.3		2.7		3.2	ns
t_{RD8}	FO=8 Routing Delay		3.2		3.5		4.2		5.0	ns
Sequential Timing Characteristics[3,4]										
t_{SUD}	Flip-Flop (Latch) Data Input Setup	0.4		0.4		0.5		0.6		ns
t_{HD}	Flip-Flop (Latch) Data Input Hold	0.0		0.0		0.0		0.0		ns
t_{SUENA}	Flip-Flop (Latch) Enable Setup	0.8		0.9		1.0		1.2		ns
t_{HENA}	Flip-Flop (Latch) Enable Hold	0.0		0.0		0.0		0.0		ns
t_{WCLKA}	Flip-Flop (Latch) Clock Active Pulse Width	3.2		3.6		4.3		5.2		ns
t_{WASYN}	Flip-Flop (Latch) Asynchronous Pulse Width	3.2		3.6		4.3		5.2		ns
t_A	Flip-Flop Clock Input Period	6.5		7.4		8.7		10.4		ns
t_{INH}	Input Buffer Latch Hold	0.0		0.0		0.0		0.0		ns
t_{INSU}	Input Buffer Latch Setup	0.3		0.4		0.4		0.5		ns
t_{OUTH}	Output Buffer Latch Hold	0.0		0.0		0.0		0.0		ns
t_{OUTSU}	Output Buffer Latch Setup	0.3		0.4		0.4		0.5		ns
f_{MAX}	Flip-Flop (Latch) Clock Frequency		225.0		200.0		170.0		115.0	MHz

Notes:

1. *For dual-module macros, use $t_{PD1} + t_{RD1} + t_{PDn}$, $t_{CO} + t_{RD1} + t_{PDn}$ or $t_{PD1} + t_{RD1} + t_{SUD}$, whichever is appropriate.*
2. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*
3. *Data applies to macros based on the S-module. Timing parameters for sequential macros constructed from C-modules can be obtained from the DirectTime Analyzer utility.*
4. *Setup and hold timing parameters for the Input Buffer Latch are defined with respect to the PAD and the D input. External setup/hold timing parameters must account for delay from an external PAD signal to the G inputs. Delay from an external PAD signal to the G input subtracts (adds) to the internal setup (hold) time.*
5. *V_{CC} = 3.0 V for 3.3V specifications.*

A1225XL Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

Input Module Propagation Delays			'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed		
Parameter	**Description**		**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{INYH}	Pad to Y High			1.1		1.2		1.4		1.7	ns
t_{INYL}	Pad to Y Low			1.3		1.4		1.7		2.0	ns
t_{INGH}	G to Y High			2.0		2.3		2.7		3.2	ns
t_{INGL}	G to Y Low			2.6		3.0		3.5		4.2	ns
Input Module Predicted Routing Delays[1]											
t_{IRD1}	FO=1 Routing Delay			2.9		3.3		3.9		4.7	ns
t_{IRD2}	FO=2 Routing Delay			3.2		3.6		4.3		5.2	ns
t_{IRD3}	FO=3 Routing Delay			3.8		4.2		5.0		6.0	ns
t_{IRD4}	FO=4 Routing Delay			4.1		4.6		5.4		6.5	ns
t_{IRD8}	FO=8 Routing Delay			5.2		5.9		6.9		8.3	ns
Global Clock Network											
t_{CKH}	Input Low to High	FO = 32		5.1		5.8		6.8		8.2	ns
		FO = 256		5.7		6.5		7.6		9.1	
t_{CKL}	Input High to Low	FO = 32		5.0		5.7		6.7		8.0	ns
		FO = 256		5.7		6.5		7.6		9.1	
t_{PWH}	Minimum Pulse Width High	FO = 32	2.6		3.0		3.5		4.2		ns
		FO = 256	2.7		3.1		3.6		4.3		
t_{PWL}	Minimum Pulse Width Low	FO = 32	2.6		3.0		3.5		4.2		ns
		FO = 256	2.7		3.1		3.6		4.3		
t_{CKSW}	Maximum Skew	FO = 32		0.8		0.9		1.0		1.2	ns
		FO = 256		0.8		0.9		1.0		1.2	
t_{SUEXT}	Input Latch External Setup	FO = 32	0.0		0.0		0.0		0.0		ns
		FO = 256	0.0		0.0		0.0		0.0		
t_{HEXT}	Input Latch External Hold	FO = 32	2.6		2.9		3.4		4.1		ns
		FO = 256	3.2		3.7		4.3		5.2		
t_P	Minimum Period	FO = 32	5.4		6.1		7.2		8.6		ns
		FO = 256	5.6		6.3		7.4		8.9		
f_{MAX}	Maximum Frequency	FO = 32		225.0		200.0		170.0		115.0	MHz
		FO = 256		200.0		180.0		155.0		105.0	

Note:

1. *These parameters should be used for estimating device performance. Optimization techniques may further reduce delays by 0 to 3 ns. Routing delays are for typical designs across worst-case operating conditions. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

A1225XL Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

Output Module Timing		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
TTL Output Module Timing[1]										
t_{DLH}	Data to Pad High		3.8		4.3		5.0		6.0	ns
t_{DHL}	Data to Pad Low		4.1		4.6		5.4		6.5	ns
t_{ENZH}	Enable Pad Z to High		3.8		4.3		5.0		6.0	ns
t_{ENZL}	Enable Pad Z to Low		4.1		4.7		5.5		6.5	ns
t_{ENHZ}	Enable Pad High to Z		5.4		6.1		7.2		8.6	ns
t_{ENLZ}	Enable Pad Low to Z		5.4		6.1		7.2		8.6	ns
t_{GLH}	G to Pad High		4.2		4.8		5.6		6.7	ns
t_{GHL}	G to Pad Low		4.7		5.4		6.3		7.6	ns
t_{LCO}	I/O Latch Clock-Out (pad-to-pad), 64 clock loading		9.0		10.0		12.0		14.4	ns
t_{ACO}	Array Clock-Out (pad-to-pad), 64 clock loading		12.8		14.4		17.0		20.4	ns
d_{TLH}	Capacitive Loading, Low to High		0.04		0.04		0.05		0.06	ns/pF
d_{THL}	Capacitive Loading, High to Low		0.05		0.06		0.07		0.08	ns/pF
CMOS Output Module Timing[1]										
t_{DLH}	Data to Pad High		4.8		5.4		6.4		7.7	ns
t_{DHL}	Data to Pad Low		3.4		3.8		4.5		5.4	ns
t_{ENZH}	Enable Pad Z to High		3.8		4.3		5.0		6.0	ns
t_{ENZL}	Enable Pad Z to Low		4.1		4.7		5.5		6.6	ns
t_{ENHZ}	Enable Pad High to Z		5.4		6.1		7.2		8.6	ns
t_{ENLZ}	Enable Pad Low to Z		5.4		6.1		7.2		8.6	ns
t_{GLH}	G to Pad High		4.2		4.8		5.6		6.7	ns
t_{GHL}	G to Pad Low		4.7		5.4		6.3		7.6	ns
t_{LCO}	I/O Latch Clock-Out (pad-to-pad), 64 clock loading		10.7		11.8		14.2		17.0	ns
t_{ACO}	Array Clock-Out (pad-to-pad), 64 clock loading		15.0		17.0		20.0		24.0	ns
d_{TLH}	Capacitive Loading, Low to High		0.05		0.06		0.07		0.08	ns/pF
d_{THL}	Capacitive Loading, High to Low		0.05		0.05		0.06		0.07	ns/pF

Notes:

1. *Delays based on 35 pF loading.*
2. *SSO information can be found in the "Simultaneously Switching Output Limits for Actel FPGAs" application note on page 4-125.*

A1240XL Timing Characteristics

(Worst-Case Commercial Conditions, V_{CC} = 4.75 V, T_J = 70°C)

Logic Module Propagation Delays[1]		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed[5]		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{PD1}	Single Module		2.6		3.0		3.5		4.2	ns
t_{CO}	Sequential Clk to Q		2.6		3.0		3.5		4.2	ns
t_{GO}	Latch G to Q		2.6		3.0		3.5		4.2	ns
t_{RS}	Flip-Flop (Latch) Reset to Q		2.6		3.0		3.5		4.2	ns
Predicted Routing Delays[2]										
t_{RD1}	FO=1 Routing Delay		1.1		1.2		1.4		1.7	ns
t_{RD2}	FO=2 Routing Delay		1.3		1.4		1.7		2.0	ns
t_{RD3}	FO=3 Routing Delay		1.7		1.9		2.2		2.6	ns
t_{RD4}	FO=4 Routing Delay		2.3		2.6		3.0		3.6	ns
t_{RD8}	FO=8 Routing Delay		3.4		3.8		4.5		5.4	ns
Sequential Timing Characteristics[3, 4]										
t_{SUD}	Flip-Flop (Latch) Data Input Setup	0.4		0.4		0.5		0.6		ns
t_{HD}	Flip-Flop (Latch) Data Input Hold	0.0		0.0		0.0		0.0		ns
t_{SUENA}	Flip-Flop (Latch) Enable Setup	0.8		0.9		1.0		1.2		ns
t_{HENA}	Flip-Flop (Latch) Enable Hold	0.0		0.0		0.0		0.0		ns
t_{WCLKA}	Flip-Flop (Latch) Clock Active Pulse Width	3.4		3.8		4.5		5.4		ns
t_{WASYN}	Flip-Flop (Latch) Asynchronous Pulse Width	3.4		3.8		4.5		5.4		ns
t_A	Flip-Flop Clock Input Period	6.8		7.7		9.1		10.9		ns
t_{INH}	Input Buffer Latch Hold	0.0		0.0		0.0		0.0		ns
t_{INSU}	Input Buffer Latch Setup	0.3		0.4		0.4		0.5		ns
t_{OUTH}	Output Buffer Latch Hold	0.0		0.0		0.0		0.0		ns
t_{OUTSU}	Output Buffer Latch Setup	0.3		0.4		0.4		0.5		ns
f_{MAX}	Flip-Flop (Latch) Clock Frequency		215.0		190.0		160.0		105.0	MHz

Notes:

1. *For dual-module macros, use $t_{PD1} + t_{RD1} + t_{PDn}$, $t_{CO} + t_{RD1} + t_{PDn}$ or $t_{PD1} + t_{RD1} + t_{SUD}$, whichever is appropriate.*
2. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*
3. *Data applies to macros based on the S-module. Timing parameters for sequential macros constructed from C-modules can be obtained from the DirectTime Analyzer utility.*
4. *Setup and hold timing parameters for the Input Buffer Latch are defined with respect to the PAD and the D input. External setup/hold timing parameters must account for delay from an external PAD signal to the G inputs. Delay from an external PAD signal to the G input subtracts (adds) to the internal setup (hold) time.*
5. *V_{CC} = 3.0 V for 3.3V specifications.*

A1240XL Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

Input Module Propagation Delays			'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed		
Parameter	Description		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Units
t_{INYH}	Pad to Y High			1.1		1.2		1.4		1.7	ns
t_{INYL}	Pad to Y Low			1.3		1.4		1.7		2.0	ns
t_{INGH}	G to Y High			2.0		2.3		2.7		3.2	ns
t_{INGL}	G to Y Low			2.6		3.0		3.5		4.2	ns
Input Module Predicted Routing Delays[1]											
t_{IRD1}	FO=1 Routing Delay			2.9		3.3		3.9		4.7	ns
t_{IRD2}	FO=2 Routing Delay			3.4		3.8		4.5		5.4	ns
t_{IRD3}	FO=3 Routing Delay			3.8		4.3		5.1		6.1	ns
t_{IRD4}	FO=4 Routing Delay			4.1		4.7		5.5		6.6	ns
t_{IRD8}	FO=8 Routing Delay			5.6		6.3		7.4		8.9	ns
Global Clock Network											
t_{CKH}	Input Low to High	FO = 32		5.1		5.8		6.8		8.2	ns
		FO = 256		5.7		6.5		7.6		9.1	
t_{CKL}	Input High to Low	FO = 32		5.0		5.7		6.7		8.0	ns
		FO = 256		5.7		6.5		7.6		9.1	
t_{PWH}	Minimum Pulse Width High	FO = 32	2.7		3.1		3.6		4.3		ns
		FO = 256	2.9		3.3		3.9		4.7		
t_{PWL}	Minimum Pulse Width Low	FO = 32	2.7		3.1		3.6		4.3		ns
		FO = 256	2.9		3.3		3.9		4.7		
t_{CKSW}	Maximum Skew	FO = 32		0.8		0.9		1.0		1.2	ns
		FO = 256		0.8		0.9		1.0		1.2	
t_{SUEXT}	Input Latch External Setup	FO = 32	0.0		0.0		0.0		0.0		ns
		FO = 256	0.0		0.0		0.0		0.0		
t_{HEXT}	Input Latch External Hold	FO = 32	2.6		2.9		3.4		4.1		ns
		FO = 256	3.2		3.7		4.3		5.2		
t_P	Minimum Period	FO = 32	5.6		6.3		7.4		8.9		ns
		FO = 256	6.0		6.8		8.0		9.6		
f_{MAX}	Maximum Frequency	FO = 32		215.0		190.0		160.0		105.0	MHz
		FO = 256		195.0		170.0		144.0		95.0	

Note:

[1] *These parameters should be used for estimating device performance. Optimization techniques may further reduce delays by 0 to 3 ns. Routing delays are for typical designs across worst-case operating conditions. Post-route timing analysis or simulation is required to determine actual worst-case*

A1240XL Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

Output Module Timing		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
TTL Output Module Timing[1]										
t_{DLH}	Data to Pad High		3.8		4.3		5.0		6.0	ns
t_{DHL}	Data to Pad Low		4.1		4.6		5.4		6.5	ns
t_{ENZH}	Enable Pad Z to High		3.8		4.3		5.0		6.0	ns
t_{ENZL}	Enable Pad Z to Low		4.1		4.7		5.5		6.6	ns
t_{ENHZ}	Enable Pad High to Z		5.4		6.1		7.2		8.6	ns
t_{ENLZ}	Enable Pad Low to Z		5.4		6.1		7.2		8.6	ns
t_{GLH}	G to Pad High		4.2		4.8		5.6		6.7	ns
t_{GHL}	G to Pad Low		4.7		5.4		6.3		7.6	ns
t_{LCO}	I/O Latch Clock-Out (pad-to-pad), 64 clock loading		9.2		10.5		12.3		14.8	ns
t_{ACO}	Array Clock-Out (pad-to-pad), 64 clock loading		12.9		14.6		17.2		20.6	ns
d_{TLH}	Capacity Loading, Low to High		0.04		0.04		0.05		0.06	ns/pF
d_{THL}	Capacity Loading, High to Low		0.05		0.06		0.07		0.08	ns/pF
CMOS Output Module Timing[1]										
t_{DLH}	Data to Pad High		4.8		5.4		6.4		7.7	ns
t_{DHL}	Data to Pad Low		3.4		3.8		4.5		5.4	ns
t_{ENZH}	Enable Pad Z to High		3.8		4.3		5.0		6.0	ns
t_{ENZL}	Enable Pad Z to Low		4.1		4.7		5.5		6.6	ns
t_{ENHZ}	Enable Pad High to Z		5.4		6.1		7.2		8.6	ns
t_{ENLZ}	Enable Pad Low to Z		5.4		6.1		7.2		8.6	ns
t_{GLH}	G to Pad High		4.2		4.8		5.6		6.7	ns
t_{GHL}	G to Pad Low		4.7		5.4		6.3		7.6	ns
t_{LCO}	I/O Latch Clock-Out (pad-to-pad), 64 clock loading		10.9		12.4		14.5		17.4	ns
t_{ACO}	Array Clock-Out (pad-to-pad), 64 clock loading		15.2		17.2		20.3		24.4	ns
d_{TLH}	Capacity Loading, Low to High		0.05		0.06		0.07		0.08	ns/pF
d_{THL}	Capacity Loading, High to Low		0.05		0.05		0.06		0.07	ns/pF

Notes:

1. *Delays based on 35 pF loading.*
2. *SSO information can be found in the "Simultaneously Switching Output Limits for Actel FPGAs" application note on page 4-125.*

A3265DX Timing Characteristics

(Worst-Case Commercial Conditions)

		Advanced Information		Preliminary Information						
Logic Module Propagation Delays[1]		‘–2’ Speed		‘–1’ Speed		‘Std’ Speed		3.3V Speed[5]		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
Combinatorial Functions										
t_{PD}	Internal Array Module Delay		2.6		3.0		3.5		4.2	ns
t_{PDD}	Internal Decode Module Delay		2.9		3.3		3.9		4.7	ns
Predicted Routing Delays[2]										
t_{RD1}	FO=1 Routing Delay		1.3		1.4		1.7		2.0	ns
t_{RD2}	FO=2 Routing Delay		1.8		2.0		2.4		2.9	ns
t_{RD3}	FO=3 Routing Delay		2.2		2.5		2.9		3.5	ns
t_{RD4}	FO=4 Routing Delay		2.6		3.0		3.5		4.2	ns
t_{RD5}	FO=8 Routing Delay		5.0		5.7		6.7		8.0	ns
t_{RDD}	Decode-to-Output Routing Delay		0.3		0.4		0.5		0.6	ns
Sequential Timing Chatacteristics[3, 4]										
t_{CO}	Flip-Flop Clock-to-Output		2.5		3.0		3.5		4.2	ns
t_{GO}	Latch Gate-to-Output		2.5		3.0		3.5		4.2	ns
t_{SU}	Flip-Flop (Latch) Setup Time	0.3		0.4		0.5		0.6		ns
t_{H}	Flip-Flop (Latch) Hold Time	0.0		0.0		0.0		0.0		ns
t_{RO}	Flip-Flop (Latch) Reset to Output		2.5		3.0		3.5		4.2	ns
t_{SUENA}	Flip-Flop (Latch) Enable Setup	0.8		0.9		1.0		1.2		ns
t_{HENA}	Flip-Flop (Latch) Enable Hold	0.0		0.0		0.0		0.0		ns
t_{WCLKA}	Flip-Flop (Latch) Clock Active Pulse Width	3.7		4.2		4.9		5.9		ns
t_{WASYN}	Flip-Flop (Latch) Asynchronous Pulse Width	3.7		4.2		4.9		5.9		ns

Notes:

1. *For dual-module macros, use* $t_{PD1} + t_{RD1} + t_{PDn}$, $t_{CO} + t_{RD1} + t_{PDn}$ *or* $t_{PD1} + t_{RD1} + t_{SUD}$, *whichever is appropriate.*
2. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*
3. *Data applies to macros based on the S-module. Timing parameters for sequential macros constructed from C-modules can be obtained from the DirectTime Analyzer utility.*
4. *Setup and hold timing parameters for the Input Buffer Latch are defined with respect to the PAD and the D input. External setup/hold timing parameters must account for delay from an external PAD signal to the G inputs. Delay from an external PAD signal to the G input subtracts (adds) to the internal setup (hold) time.*
5. V_{CC} *= 3.0 V for 3.3V specifications.*

A3265DX Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

			Advanced Information		Preliminary Information						
Input Module Propagation Delays			**'–2' Speed**		**'–1' Speed**		**'Std' Speed**		**3.3V Speed**		
Parameter	**Description**		**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{INPY}	Input Data Pad to Y			1.3		1.4		1.7		2.0	ns
t_{INGO}	Input Latch Gate-to-Output			2.6		3.0		3.5		4.2	ns
t_{INH}	Input Latch Hold		0.0		0.0		0.0		0.0		ns
t_{INSU}	Input Latch Setup		0.3		0.3		0.4		0.5		ns
t_{ILA}	Latch Active Pulse Width		3.7		4.2		4.9		5.9		ns
Input Module Predicted Routing Delays[1]											
t_{IRD1}	FO=1 Routing Delay			3.2		3.7		4.3		5.2	ns
t_{IRD2}	FO=2 Routing Delay			3.7		4.2		4.9		5.9	ns
t_{IRD3}	FO=3 Routing Delay			4.0		4.5		5.3		6.4	ns
t_{IRD4}	FO=4 Routing Delay			4.6		5.2		6.1		7.3	ns
t_{IRD5}	FO=8 Routing Delay			6.6		7.5		8.8		10.6	ns
Global Clock Network											
t_{CKH}	Input Low to High	FO=32		5.1		5.8		6.8		8.2	ns
		FO=256		5.7		6.5		7.6		9.1	ns
t_{CKL}	Input High to Low	FO=32		5.0		5.7		6.7		8.0	ns
		FO=256		5.7		6.5		7.6		9.1	ns
t_{PW}	Minimum Pulse Width	FO=32	2.5		3.0		3.5		4.2		ns
		FO=256	3.0		3.7		4.6		5.5		ns
t_{CKSW}	Maximum Skew	FO=32		0.8		0.9		1.0		1.2	ns
		FO=256		0.8		0.9		1.0		1.2	ns
t_{SUEXT}	Input Latch External Setup	FO=32	0.0		0.0		0.0		0.0		ns
		FO=256	0.0		0.0		0.0		0.0		ns
t_{HEXT}	Input Latch External Hold	FO=32	2.6		2.6		3.4		4.1		ns
		FO=256	3.2		3.2		4.3		5.2		ns
t_P	Minimum Period (1/fmax)	FO=32	4.5		6.0		7.0		8.4		ns
		FO=256	6.0		7.4		8.7		10.4		ns
f_{MAX}	Maximum Datapath Frequency	FO=32		200.0		167.0		143.0		120.0	MHz
		FO=256		180.0		150.0		130.0		110.0	MHz

Note:

1. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance.*

A3265DX Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

		Advanced Information		Preliminary Information						
Output Module Timing		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
TTL Output Module Timing[1]										
t_{DLH}	Data to Pad High		3.7		4.3		5.0		6.0	ns
t_{DHL}	Data to Pad Low		3.9		4.6		5.4		6.5	ns
t_{ENZH}	Enable Pad Z to High		3.7		4.3		5.0		6.0	ns
t_{ENZL}	Enable Pad Z to Low		4.0		4.7		5.5		6.6	ns
t_{ENHZ}	Enable Pad High to Z		5.2		6.1		7.2		8.6	ns
t_{ENLZ}	Enable Pad Low to Z		5.2		6.1		7.2		8.6	ns
t_{GLH}	G to Pad High		4.1		4.8		5.6		6.7	ns
t_{GHL}	G to Pad Low		4.6		5.4		6.3		7.6	ns
t_{LSU}	I/O Latch Output Setup	0.3		0.3		0.4		0.5		ns
t_{LH}	I/O Latch Output Hold	0.0		0.0		0.0		0.0		ns
t_{LCO}	I/O Latch Clock-Out (Pad-to-Pad) 32 I/O		9.4		11.0		13.1		15.7	ns
t_{ACO}	Array Latch Clock-Out (Pad-to-Pad) 32 I/O		13.3		15.7		18.5		22.2	ns
d_{TLH}	Capacitive Loading, Low to High		0.03		0.04		0.05		0.06	ns/pF
d_{THL}	Capacitive Loading, High to Low		0.03		0.04		0.07		0.08	ns/pF
CMOS Output Module Timing[1]										
t_{DLH}	Data to Pad High		4.6		5.4		6.4		7.7	ns
t_{DHL}	Data to Pad Low		3.2		3.8		4.5		5.4	ns
t_{ENZH}	Enable Pad Z to High		3.7		4.3		5.0		6.0	ns
t_{ENZL}	Enable Pad Z to Low		4.0		4.7		5.5		6.6	ns
t_{ENHZ}	Enable Pad High to Z		5.2		6.1		7.2		8.6	ns
t_{ENLZ}	Enable Pad Low to Z		5.2		6.1		7.2		8.6	ns
t_{GLH}	G to Pad High		4.1		4.8		5.6		6.7	ns
t_{GHL}	G to Pad Low		4.6		5.4		6.3		7.6	ns
t_{LSU}	I/O Latch Setup	0.4		0.4		0.4		0.5		ns
t_{LH}	I/O Latch Hold	0.0		0.0		0.0		0.0		ns
t_{LCO}	I/O Latch Clock-Out (Pad-to-Pad) 32 I/O		11.1		13.0		15.5		18.6	ns/pF
t_{ACO}	Array Latch Clock-Out (Pad-to-Pad) 32 I/O		15.7		18.5		21.8		26.2	ns/pF
d_{TLH}	Capacitive Loading, Low to High		0.05		0.06		0.07		0.08	ns/pF
d_{THL}	Capacitive Loading, High to Low		0.04		0.05		0.06		0.07	ns/pF

Notes:

1. *Delays based on 35pF loading.*
2. *SSO information can be found in the "Simultaneously Switching Output Limits for Actel FPGAs" application note on page 4-125.*

A1280XL Timing Characteristics

(Worst-Case Commercial Conditions, V_{CC} = 4.75 V, T_J = 70°C)

Logic Module Propagation Delays[1]		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed[5]		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{PD1}	Single Module		2.6		3.0		3.5		4.2	ns
t_{CO}	Sequential Clk to Q		2.6		3.0		3.5		4.2	ns
t_{GO}	Latch G to Q		2.6		3.0		3.5		4.2	ns
t_{RS}	Flip-Flop (Latch) Reset to Q		2.6		3.0		3.5		4.2	ns
Predicted Routing Delays[2]										
t_{RD1}	FO=1 Routing Delay		1.3		1.4		1.7		2.0	ns
t_{RD2}	FO=2 Routing Delay		1.8		2.0		2.4		2.9	ns
t_{RD3}	FO=3 Routing Delay		2.2		2.5		2.9		3.5	ns
t_{RD4}	FO=4 Routing Delay		2.6		3.0		3.5		4.2	ns
t_{RD8}	FO=8 Routing Delay		5.0		5.7		6.7		8.0	ns
Sequential Timing Characteristics[3,4]										
t_{SUD}	Flip-Flop (Latch) Data Input Setup	0.4		0.4		0.5		0.6		ns
t_{HD}	Flip-Flop (Latch) Data Input Hold	0.0		0.0		0.0		0.0		ns
t_{SUENA}	Flip-Flop (Latch) Enable Setup	0.8		0.9		1.0		1.2		ns
t_{HENA}	Flip-Flop (Latch) Enable Hold	0.0		0.0		0.0		0.0		ns
t_{WCLKA}	Flip-Flop (Latch) Clock Active Pulse Width	3.7		4.3		4.9		5.9		ns
t_{WASYN}	Flip-Flop (Latch) Asynchronous Pulse Width	3.7		4.3		4.9		5.9		ns
t_A	Flip-Flop Clock Input Period	8.0		8.7		10		12		ns
t_{INH}	Input Buffer Latch Hold	0.0		0.0		0.0		0.0		ns
t_{INSU}	Input Buffer Latch Setup	0.3		0.4		0.4		0.5		ns
t_{OUTH}	Output Buffer Latch Hold	0.0		0.0		0.0		0.0		ns
t_{OUTSU}	Output Buffer Latch Setup	0.3		0.4		0.4		0.5		ns
f_{MAX}	Flip-Flop (Latch) Clock Frequency		200.0		167.0		130.0		110.0	MHz

Notes:

1. *For dual-module macros, use $t_{PD1} + t_{RD1} + t_{PDn}$, $t_{CO} + t_{RD1} + t_{PDn}$, or $t_{PD1} + t_{RD1} + t_{SUD}$, whichever is appropriate.*
2. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*
3. *Data applies to macros based on the S-module. Timing parameters for sequential macros constructed from C-modules can be obtained from the DirectTime Analyzer utility.*
4. *Setup and hold timing parameters for the Input Buffer Latch are defined with respect to the PAD and the D input. External setup/hold timing parameters must account for delay from an external PAD signal to the G inputs. Delay from an external PAD signal to the G input subtracts (adds) to the internal setup (hold) time.*
5. *V_{CC} = 3.0 V for 3.3V specifications.*

A1280XL Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

Input Module Propagation Delays			'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed		
Parameter	Description		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Units
t_{INYH}	Pad to Y High			1.1		1.2		1.4		1.7	ns
t_{INYL}	Pad to Y Low			1.3		1.4		1.7		2.0	ns
t_{INGH}	G to Y High			2.0		2.3		2.7		3.2	ns
t_{INGL}	G to Y Low			2.6		3.0		3.5		4.2	ns
Input Module Predicted Routing Delays[1]											
t_{IRD1}	FO=1 Routing Delay			3.2		3.7		4.3		5.2	ns
t_{IRD2}	FO=2 Routing Delay			3.7		4.2		4.9		5.9	ns
t_{IRD3}	FO=3 Routing Delay			4.0		4.5		5.3		6.4	ns
t_{IRD4}	FO=4 Routing Delay			4.6		5.2		6.1		7.3	ns
t_{IRD8}	FO=8 Routing Delay			6.6		7.5		8.8		10.6	ns
Global Clock Network											
t_{CKH}	Input Low to High	FO = 32 FO = 384		5.1 5.7		5.8 6.5		6.8 7.6		8.2 9.1	ns
t_{CKL}	Input High to Low	FO = 32 FO = 384		5.0 5.7		5.7 6.5		6.7 7.6		8.0 9.1	ns
t_{PWH}	Minimum Pulse Width High	FO = 32 FO = 384	3.2 3.5		3.5 3.9		4.3 4.6		5.2 5.5		ns
t_{PWL}	Minimum Pulse Width Low	FO = 32 FO = 384	3.2 3.5		3.5 3.9		4.3 4.6		5.2 5.5		ns
t_{CKSW}	Maximum Skew	FO = 32 FO = 384		0.8 0.8		0.9 0.9		1.0 1.0		1.2 1.2	ns
t_{SUEXT}	Input Latch External Setup	FO = 32 FO = 384	0.0 0.0		0.0 0.0		0.0 0.0		0.0 0.0		ns
t_{HEXT}	Input Latch External Hold	FO = 32 FO = 384	2.6 3.2		2.9 3.7		3.4 4.3		4.1 5.2		ns
t_P	Minimum Period	FO = 32 FO = 384	6.5 7.2		7.4 8.0		8.7 9.6		10.4 11.5		ns
f_{MAX}	Maximum Frequency	FO = 32 FO = 384		200.0 180.0		167.0 150.0		143.0 130.0		120.0 110.0	MHz

Note:

1. *These parameters should be used for estimating device performance. Optimization techniques may further reduce delays by 0 to 4 ns. Routing delays are for typical designs across worst-case operating conditions. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

A1280XL Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

Output Module Timing		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
TTL Output Module Timing[1]										
t_{DLH}	Data to Pad High		3.8		4.3		5.0		6.0	ns
t_{DHL}	Data to Pad Low		4.1		4.6		5.4		6.5	ns
t_{ENZH}	Enable Pad Z to High		3.8		4.3		5.0		6.0	ns
t_{ENZL}	Enable Pad Z to Low		4.1		4.7		5.5		6.6	ns
t_{ENHZ}	Enable Pad High to Z		5.4		6.1		7.2		8.6	ns
t_{ENLZ}	Enable Pad Low to Z		5.4		6.1		7.2		8.6	ns
t_{GLH}	G to Pad High		4.2		4.8		5.6		6.7	ns
t_{GHL}	G to Pad Low		4.7		5.4		6.3		7.6	ns
t_{LCO}	I/O Latch Clock-Out (pad-to-pad), 64 clock loading		9.8		11.0		13.1		15.7	ns
t_{ACO}	Array Clock-Out (pad-to-pad), 64 clock loading		13.9		15.7		18.5		22.2	ns
d_{TLH}	Capacitive Loading, Low to High		0.04		0.04		0.05		0.06	ns/pF
d_{THL}	Capacitive Loading, High to Low		0.05		0.06		0.07		0.08	ns/pF
CMOS Output Module Timing[1]										
t_{DLH}	Data to Pad High		4.8		5.4		6.4		7.7	ns
t_{DHL}	Data to Pad Low		3.4		3.8		4.5		5.4	ns
t_{ENZH}	Enable Pad Z to High		3.8		4.3		5.0		6.0	ns
t_{ENZL}	Enable Pad Z to Low		4.1		4.7		5.5		6.6	ns
t_{ENHZ}	Enable Pad High to Z		5.4		6.1		7.2		8.6	ns
t_{ENLZ}	Enable Pad Low to Z		5.4		6.1		7.2		8.6	ns
t_{GLH}	G to Pad High		4.2		4.8		5.6		6.7	ns
t_{GHL}	G to Pad Low		4.7		5.4		6.3		7.6	ns
t_{LCO}	I/O Latch Clock-Out (pad-to-pad), 64 clock loading		11.6		13.0		15.5		18.6	ns
t_{ACO}	Array Clock-Out (pad-to-pad), 64 clock loading		16.4		18.5		21.8		26.2	ns
d_{TLH}	Capacitive Loading, Low to High		0.05		0.06		0.07		0.08	ns/pF
d_{THL}	Capacitive Loading, High to Low		0.05		0.05		0.06		0.07	ns/pF

Notes:

1. *Delays based on 35 pF loading.*
2. *SSO information can be found in the "Simultaneously Switching Output Limits for Actel FPGAs" application note on page 4-125.*

A32140DX Timing Characteristics

(Worst-Case Commercial Conditions)

		Advanced Information		Preliminary Information				
Logic Module Propagation Delays[1]		'–2 Speed		'–1' Speed		'Std' Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
Combinatorial Functions								
t_{PD}	Internal Array Module Delay		2.6		3.0		3.5	ns
t_{PDD}	Internal Decode Module Delay		2.9		3.3		3.9	ns
Predicted Routing Delays[2]								
t_{RD1}	FO=1 Routing Delay		1.3		1.4		1.7	ns
t_{RD2}	FO=2 Routing Delay		2.0		2.2		2.6	ns
t_{RD3}	FO=3 Routing Delay		2.9		3.3		3.8	ns
t_{RD4}	FO=4 Routing Delay		3.5		3.9		4.6	ns
t_{RD5}	FO=8 Routing Delay		6.3		7.1		8.4	ns
t_{RDD}	Decode-to-Output Routing Delay		0.3		0.4		0.5	ns
Sequential Timing Characteristics[3, 4]								
t_{CO}	Flip-Flop Clock-to-Output		2.6		3.0		3.5	ns
t_{GO}	Latch Gate-to-Output		2.6		3.0		3.5	ns
t_{SU}	Flip-Flop (Latch) Setup Time	0.4		0.4		0.5		ns
t_H	Flip-Flop (Latch) Hold Time	0.0		0.0		0.0		ns
t_{RO}	Flip-Flop (Latch) Reset to Output		2.6		3.0		3.5	ns
t_{SUENA}	Flip-Flop (Latch) Enable Setup	0.8		0.9		1.0		ns
t_{HENA}	Flip-Flop (Latch) Enable Hold	0.0		0.0		0.0		ns
t_{WCLKA}	Flip-Flop (Latch) Clock Active Pulse Width	3.7		4.2		4.9		ns
t_{WASYN}	Flip-Flop (Latch) Asynchronous Pulse Width	3.7		4.2		4.9		ns

Notes:

1. *For dual-module macros, use $t_{PD1} + t_{RD1} + t_{PDn}$, $t_{CO} + t_{RD1} + t_{PDn}$, or $t_{PD1} + t_{RD1} + t_{SUD}$, whichever is appropriate.*
2. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*
3. *Data applies to macros based on the S-module. Timing parameters for sequential macros constructed from C-modules can be obtained from the DirectTime Analyzer utility.*
4. *Setup and hold timing parameters for the Input Buffer Latch are defined with respect to the PAD and the D input. External setup/hold timing parameters must account for delay from an external PAD signal to the G inputs. Delay from an external PAD signal to the G input subtracts (adds) to the internal setup (hold) time.*

A32140DX Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

			Advanced Information		Preliminary Information				
Input Module Propagation Delays			**'–2' Speed**		**'–1' Speed**		**'Std' Speed**		
Parameter	**Description**		**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{INPY}	Input Data Pad to Y			1.3		1.4		1.7	ns
t_{INGO}	Input Latch Gate-to-Output			2.6		3.0		3.5	ns
t_{INH}	Input Latch Hold		0.0		0.0		0.0		ns
t_{INSU}	Input Latch Setup		0.3		0.3		0.4		ns
t_{ILA}	Latch Active Pulse Width		3.7		4.2		4.9		ns
Input Module Predicted Routing Delays[1]									
t_{IRD1}	FO=1 Routing Delay			3.2		3.7		4.3	ns
t_{IRD2}	FO=2 Routing Delay			3.7		4.2		4.9	ns
t_{IRD3}	FO=3 Routing Delay			4.0		4.5		5.3	ns
t_{IRD4}	FO=4 Routing Delay			4.6		5.2		6.1	ns
t_{IRD5}	FO=8 Routing Delay			6.6		7.5		8.8	ns
Global Clock Network									
t_{CKH}	Input Low to High	FO=32		5.1		5.8		6.8	ns
		FO=486		5.7		6.5		7.6	ns
t_{CKL}	Input High to Low	FO=32		5.0		5.7		6.7	ns
		FO=486		5.7		6.5		7.6	ns
t_{PW}	Minimum Pulse Width	FO=32	3.2		3.7		4.3		ns
		FO=486	3.5		3.9		4.6		ns
t_{CKSW}	Maximum Skew	FO=32		0.8		0.9		1.0	ns
		FO=486		0.8		0.9		1.0	ns
t_{SUEXT}	Input Latch External Setup	FO=32	0.0		0.0		0.0		ns
		FO=486	0.0		0.0		0.0		ns
t_{HEXT}	Input Latch External Hold	FO=32	2.6		2.9		3.4		ns
		FO=486	3.2		3.7		4.3		ns
t_P	Minimum Period (1/fmax)	FO=32	6.5		7.4		8.7		ns
		FO=486	7.2		8.2		9.6		ns
f_{MAX}	Maximum Datapath Frequency	FO=32		153.0		133.0		115.0	MHz
		FO=486		140.0		123.0		105.0	MHz

Note:

[1] *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance.*

A32140DX Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

		Advanced Information		Preliminary Information				
Output Module Timing		**'–2 Speed**		**'–1' Speed**		**'Std' Speed**		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
TTL Output Module Timing[1]								
t_{DLH}	Data to Pad High		3.7		4.3		5.0	ns
t_{DHL}	Data to Pad Low		3.9		4.6		5.4	ns
t_{ENZH}	Enable Pad Z to High		3.7		4.3		5.0	ns
t_{ENZL}	Enable Pad Z to Low		4.0		4.7		5.5	ns
t_{ENHZ}	Enable Pad High to Z		5.2		6.1		7.2	ns
t_{ENLZ}	Enable Pad Low to Z		5.2		6.1		7.2	ns
t_{GLH}	G to Pad High		4.1		4.8		5.6	ns
t_{GHL}	G to Pad Low		4.6		5.4		6.3	ns
t_{LSU}	I/O Latch Output Setup	0.3		0.3		0.4		ns
t_{LH}	I/O Latch Output Hold	0.0		0.0		0.0		ns
t_{LCO}	I/O Latch Clock-Out (Pad-to-Pad) 32 I/O		9.4		11.0		13.1	ns
t_{ACO}	Array Latch Clock-Out (Pad-to-Pad) 32 I/O		13.3		15.7		18.5	ns
d_{TLH}	Capacitive Loading, Low to High		0.03		0.04		0.05	ns/pF
d_{THL}	Capacitive Loading, High to Low		0.03		0.04		0.07	ns/pF
CMOS Output Module Timing[1]								
t_{DLH}	Data to Pad High		4.6		5.4		6.4	ns
t_{DHL}	Data to Pad Low		3.2		3.8		4.5	ns
t_{ENZH}	Enable Pad Z to High		3.7		4.3		5.0	ns
t_{ENZL}	Enable Pad Z to Low		4.0		4.7		5.5	ns
t_{ENHZ}	Enable Pad High to Z		5.2		6.1		7.2	ns
t_{ENLZ}	Enable Pad Low to Z		5.2		6.1		7.2	ns
t_{GLH}	G to Pad High		4.1		4.8		5.6	ns
t_{GHL}	G to Pad Low		4.6		5.4		6.3	ns
t_{LSU}	I/O Latch Setup	0.4		0.4		0.4		ns
t_{LH}	I/O Latch Hold	0.0		0.0		0.0		ns
t_{LCO}	I/O Latch Clock-Out (Pad-to-Pad) 32 I/O		11.1		13.0		15.6	ns/pF
t_{ACO}	Array Latch Clock-Out (Pad-to-Pad) 32 I/O		15.7		18.5		21.8	ns/pF
d_{TLH}	Capacitive Loading, Low to High		0.05		0.06		0.07	ns/pF
d_{THL}	Capacitive Loading, High to Low		0.04		0.05		0.06	ns/pF

Notes:

1. *Delays based on 35 pF loading.*
2. *SSO information can be found in the "Simultaneously Switching Output Limits for Actel FPGAs" application note on page 4-125.*

A32200DX Timing Characteristics

(Worst-Case Commercial Conditions)

		Advanced Information						
Logic Module Propagation Delays		**'–2 Speed**		**'–1' Speed**		**'Std' Speed**		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
Combinatorial Functions								
t_{PD}	Internal Array Module Delay		2.6		3.0		3.5	ns
t_{PDD}	Internal Decode Module Delay		2.9		3.3		3.9	ns
Predicted Module Routing Delays								
t_{RD1}	FO=1 Routing Delay		2.0		2.2		2.6	ns
t_{RD2}	FO=2 Routing Delay		2.5		2.9		3.4	ns
t_{RD3}	FO=3 Routing Delay		2.9		3.3		3.8	ns
t_{RD4}	FO=4 Routing Delay		3.4		3.9		4.6	ns
t_{RD5}	FO=8 Routing Delay		6.8		7.7		9.1	ns
t_{RDD}	Decode-to-Output Routing Delay		0.3		0.4		0.5	ns
Sequential Timing Characteristics								
t_{CO}	Flip-Flop Clock-to-Output		2.6		3.0		3.5	ns
t_{GO}	Latch Gate-to-Output		2.6		3.0		3.5	ns
t_{SU}	Flip-Flop (Latch) Setup Time	0.4		0.4		0.5		ns
t_{H}	Flip-Flop (Latch) Hold Time	0.0		0.0		0.0		ns
t_{RO}	Flip-Flop (Latch) Reset to Output		2.6		3.0		3.5	ns
t_{SUENA}	Flip-Flop (Latch) Enable Setup	0.8		0.9		1.0		ns
t_{HENA}	Flip-Flop (Latch) Enable Hold	0.0		0.0		0.0		ns
t_{WCLKA}	Flip-Flop (Latch) Clock Active Pulse Width	3.7		4.2		4.9		ns
t_{WASYN}	Flip-Flop (Latch) Asynchronous Pulse Width	3.7		4.2		4.9		ns

A32200DX Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

		Advanced Information						
Logic Module Timing		**'–2 Speed**		**'–1' Speed**		**'Std' Speed**		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
Synchronous SRAM Operations								
t_{RC}	Read Cycle Time	7.5		8.5		10.0		ns
t_{WC}	Write Cycle Time	7.5		8.5		10.0		ns
t_{RCKHL}	Clock High/Low Time	3.8		4.3		5.0		ns
t_{RCO}	Data Valid After Clock High/Low		3.8		4.3		5.0	ns
t_{ADSU}	Address/Data Setup Time	1.8		2.0		2.4		ns
t_{ADH}	Address/Data Hold Time	0.0		0.0		0.0		ns
t_{RENSU}	Read Enable Setup	0.8		0.9		1.0		ns
t_{RENH}	Read Enable Hold	0.4		0.4		0.5		ns
t_{WENSU}	Write Enable Setup	2.9		3.2		3.8		ns
t_{WENH}	Write Enable Hold	0.0		0.0		0.0		ns
t_{BENS}	Block Enable Setup	2.9		3.3		3.9		ns
t_{BENH}	Block Enable Hold	0.0		0.0		0.0		ns
Asynchronous SRAM Operations								
t_{RPD}	Asynchronous Access Time		9.0		10.2		12.0	ns
t_{RDADV}	Read Address Valid	9.8		11.0		13.0		ns
t_{ADSU}	Address/Data Setup Time	1.8		2.0		2.4		ns
t_{ADH}	Address/Data Hold Time	0.0		0.0		0.0		ns
t_{RENSUA}	Read Enable Setup to Address Valid	0.8		0.9		1.0		ns
t_{RENHA}	Read Enable Hold	0.4		0.4		0.5		ns
t_{WENSU}	Write Enable Setup	2.9		3.2		3.8		ns
t_{WENH}	Write Enable Hold	0.0		0.0		0.0		ns
t_{DOH}	Data Out Hold Time		1.4		1.5		1.8	ns

A32200DX Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

			Advanced Information						
Input Module Propagation Delays			**'–2' Speed**		**'–1' Speed**		**'Std' Speed**		
Parameter	**Description**		**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{INPY}	Input Data Pad to Y			1.3		1.4		1.7	ns
t_{INGO}	Input Latch Gate-to-Output[1]			2.6		3.0		3.5	ns
t_{INH}	Input Latch Hold[1]		0.0		0.0		0.0		ns
t_{INSU}	Input Latch Setup[1]		0.3		0.3		0.4		ns
t_{ILA}	Latch Active Pulse Width[1]		3.7		4.2		4.9		ns
Input Module Predicted Routing Delays									
t_{IRD1}	FO=1 Routing Delay			3.2		3.7		4.3	ns
t_{IRD2}	FO=2 Routing Delay			3.7		4.2		4.9	ns
t_{IRD3}	FO=3 Routing Delay			4.0		4.5		5.3	ns
t_{IRD4}	FO=4 Routing Delay			4.6		5.2		6.1	ns
t_{IRD8}	FO=8 Routing Delay			6.6		7.5		8.8	ns
Global Clock Network									
t_{CKH}	Input Low to High	FO=32		5.1		5.8		6.8	ns
		FO=635		5.7		6.5		7.6	ns
t_{CKL}	Input High to Low	FO=32		5.0		5.8		6.7	ns
		FO=635		5.7		6.5		7.6	ns
t_{PWH}	Minimum Pulse Width High	FO=32	3.2		3.7		4.3		ns
		FO=635	3.5		4.1		4.6		ns
t_{PWL}	Minimum Pulse Width Low	FO=32	3.2		3.7		4.3		ns
		FO=635	3.5		4.1		4.6		ns
t_{CKSW}	Maximum Skew	FO=32		0.8		0.9		1.0	ns
		FO=635		0.8		0.9		1.0	ns
t_{SUEXT}	Input Latch External Setup	FO=32	0.0		0.0		0.0		ns
		FO=635	0.0		0.0		0.0		ns
t_{HEXT}	Input Latch External Hold	FO=32	2.6		2.9		3.4		ns
		FO=635	3.2		3.7		4.3		ns
t_P	Minimum Period (1/fmax)	FO=32	6.5		7.4		8.7		ns
		FO=635	7.2		8.2		9.6		ns
f_{HMAX}	Maximum Datapath Frequency	FO=32		153.0		133.0		115.0	MHz
		FO=635		140.0		123.0		105.0	MHz

Note:

[1] *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance.*

A32200DX Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

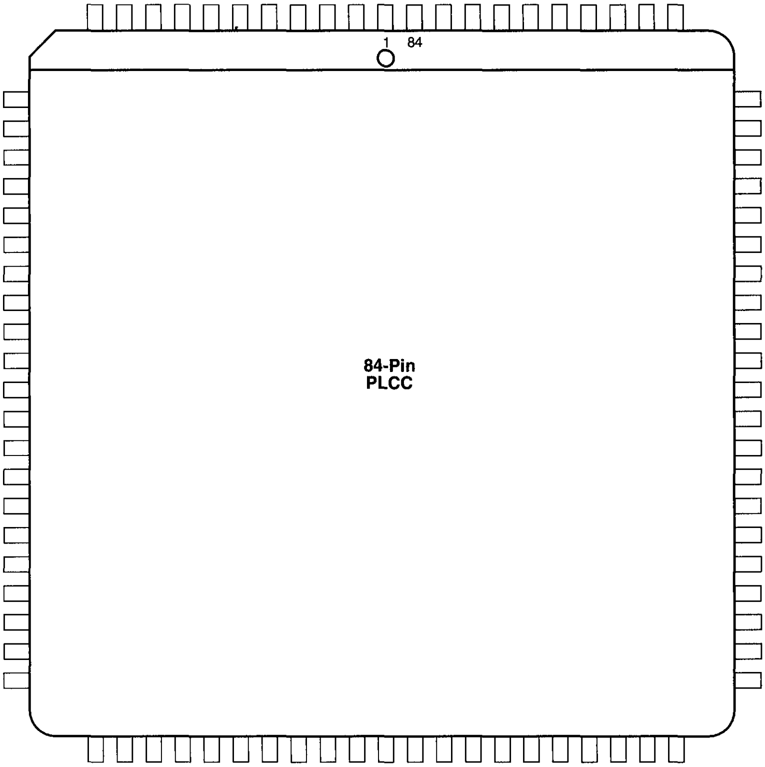
		Advanced Information						
Output Module Timing		**'–2' Speed**		**'–1' Speed**		**'Std' Speed**		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
TTL Output Module Timing[1]								
t_{DLH}	Data to Pad High		3.7		4.3		5.4	ns
t_{DHL}	Data to Pad Low		3.9		4.6		5.4	ns
t_{ENZH}	Enable Pad Z to High		3.7		4.3		5.0	ns
t_{ENZL}	Enable Pad Z to Low		4.0		4.7		5.5	ns
t_{ENHZ}	Enable Pad High to Z		5.2		6.1		7.2	ns
t_{ENLZ}	Enable Pad Low to Z		5.2		6.1		7.2	ns
t_{GLH}	G to Pad High		4.1		4.8		5.6	ns
t_{GHL}	G to Pad Low		4.6		5.4		6.3	ns
t_{LSU}	I/O Latch Output Setup	0.3		0.3		0.4		ns
t_{LH}	I/O Latch Output Hold	0.0		0.0		0.0		ns
t_{LCO}	I/O Latch Clock-Out (Pad-to-Pad) 32 I/O		9.4		11.0		13.1	ns
t_{ACO}	Array Latch Clock-Out (Pad-to-Pad) 32 I/O		13.3		15.7		18.5	ns
d_{TLH}	Capacitive Loading, Low to High		0.03		0.04		0.05	ns/pF
d_{THL}	Capacitive Loading, High to Low		0.03		0.04		0.07	ns/pF
t_{WDO}	Hard-Wired Wide Decode Output				0.05			ns
CMOS Output Module Timing[1]								
t_{DLH}	Data to Pad High		4.6		5.4		6.4	ns
t_{DHL}	Data to Pad Low		3.2		3.8		4.5	ns
t_{ENZH}	Enable Pad Z to High		3.7		4.3		5.0	ns
t_{ENZL}	Enable Pad Z to Low		4.0		4.7		5.5	ns
t_{ENHZ}	Enable Pad High to Z		5.2		6.1		7.2	ns
t_{ENLZ}	Enable Pad Low to Z		5.2		6.1		7.2	ns
t_{GLH}	G to Pad High		4.1		4.8		5.6	ns
t_{GHL}	G to Pad Low		4.6		5.4		6.3	ns
t_{LSU}	I/O Latch Setup	0.4		0.4		0.4		ns
t_{LH}	I/O Latch Hold	0.0		0.0		0.0		ns
t_{LCO}	I/O Latch Clock-Out (Pad-to-Pad) 32 I/O		11.1		13.0		15.5	ns/pF
t_{ACO}	Array Latch Clock-Out (Pad-to-Pad) 32 I/O		15.7		18.5		21.8	ns/pF
d_{TLH}	Capacitive Loading, Low to High		0.05		0.06		0.07	ns/pF
d_{THL}	Capacitive Loading, High to Low		0.04		0.05		0.06	ns/pF
t_{WDO}	Hard-Wired Wide Decode Output		0.04		0.05		0.06	ns/pF

Notes:

1. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance.*
2. *SSO information can be found in the "Simultaneously Switching Output Limits for Actel FPGAs" application note on page 4-125.*

Package Pin Assignments

84-Pin PLCC Package (Top View)



84-Pin PLCC Package

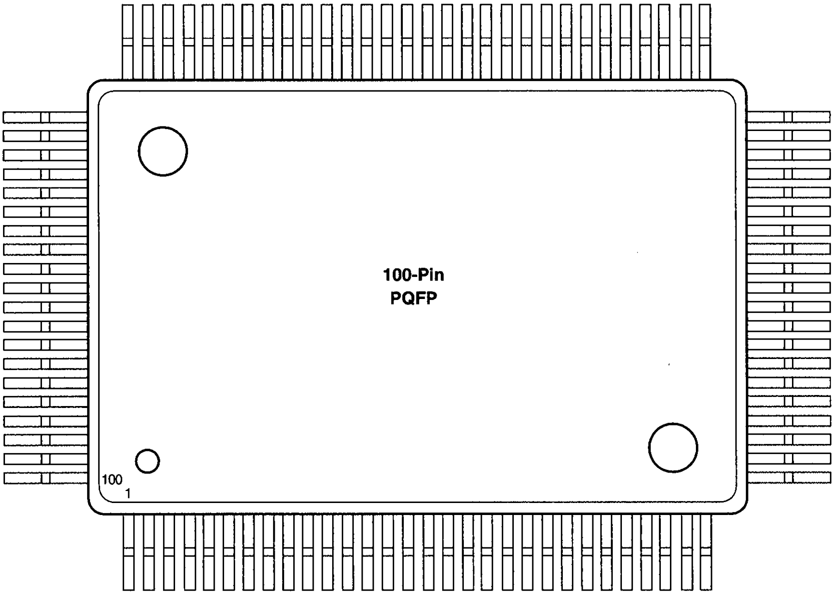
Pin Number	A1225XL Function	A1240XL Function	A3265DX Function	A1280XL Function
2	CLKB,I/O	CLKB,I/O	CLKB,I/O	CLKB,I/O
4	PRB,I/O	PRB,I/O	PRB,I/O	PRB,I/O
5	I/O	I/O	I/O (WD)	I/O
6	GND	GND	GND	GND
8	I/O	I/O	I/O (WD)	I/O
9	I/O	I/O	I/O (WD)	I/O
10	DCLK,I/O	DCLK,I/O	DCLK,I/O	DCLK,I/O
12	MODE	MODE	MODE	MODE
22	VCC	VCC	VCC	VCC
23	VCC	VCC	VCC	VCC
28	GND	GND	GND	GND
38	I/O	I/O	I/O (WD)	I/O
39	I/O	I/O	I/O (WD)	I/O
43	VCC	VCC	VCC	VCC
44	I/O	I/O	I/O (WD)	I/O
46	I/O	I/O	I/O (WD)	I/O
47	I/O	I/O	I/O (WD)	I/O
49	GND	GND	GND	GND
63	GND	GND	GND	GND
64	VCC	VCC	VCC	VCC
65	VCC	VCC	VCC	VCC
70	GND	GND	GND	GND
76	SDI,I/O	SDI,I/O	SDI,I/O	SDI,I/O
78	I/O	I/O	I/O (WD)	I/O
79	I/O	I/O	I/O (WD)	I/O
80	I/O	I/O	I/O (WD)	I/O
81	PRA,I/O	PRA,I/O	PRA,I/O	PRA,I/O
83	CLKA,I/O	CLKA,I/O	CLKA,I/O	CLKA,I/O
84	VCC	VCC	VCC	VCC

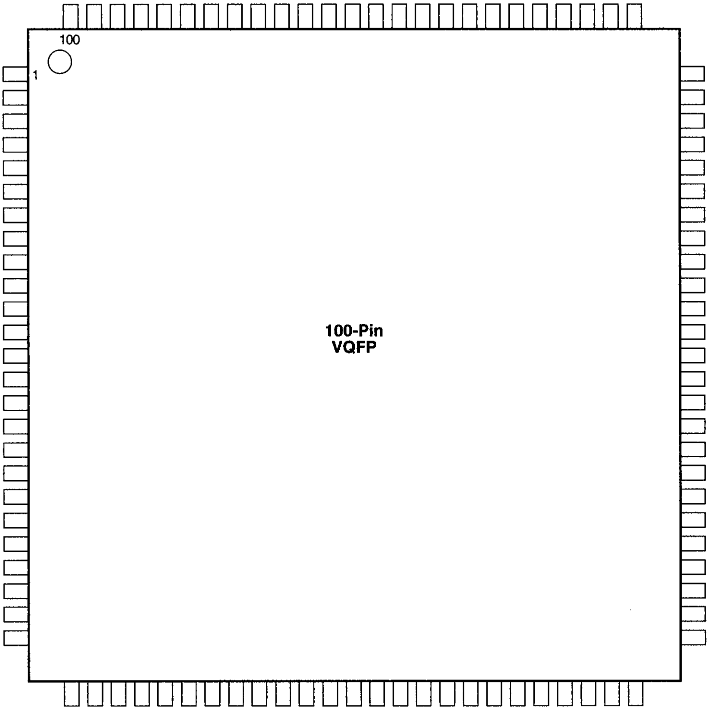
Notes:

1. *I/O (WD) : Denotes I/O pin with an associated Wide Decode Module*
2. *Wide Decode I/O (WD) can also be general purpose user I/O*
3. *NC : Denotes No Connection*
4. *All unlisted pin numbers are user I/O's*
5. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

100-pin PQFP Package, 100-pin VQFP Package (Top View)





100-pin PQFP Package, 100-pin VQFP Package

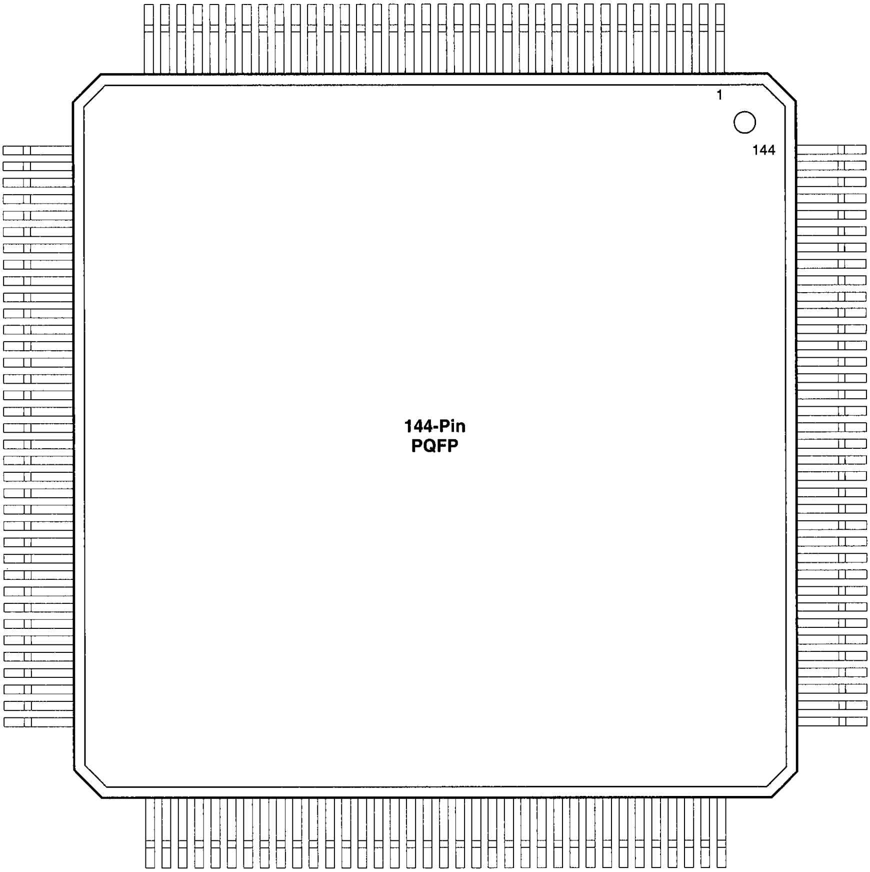
Pin Number	A1225XL-PQ100 Function	A1225XL-VQ100 Function	A1240XL-PQ100 Function
2	DCLK, I/O	MODE	DCLK, I/O
4	MODE	I/O	MODE
7	I/O	GND	I/O
9	GND	I/O	GND
14	I/O	VCC	I/O
15	I/O	VCC	I/O
16	VCC	I/O	VCC
17	VCC	I/O	VCC
20	I/O	GND	I/O
22	GND	I/O	GND
32	I/O	GND	I/O
34	GND	I/O	GND
38	I/O	VCC	I/O
40	VCC	I/O	VCC
44	I/O	GND	I/O
46	GND	I/O	GND
55	I/O	GND	I/O
57	GND	I/O	GND
62	I/O	GND	I/O
63	I/O	VCC	I/O
64	GND	VCC	GND
65	VCC	VCC	VCC
66	VCC	I/O	VCC
67	VCC	I/O	VCC
70	I/O	GND	I/O
72	GND	I/O	GND
77	I/O	SDI, I/O	I/O
79	SDI, I/O	I/O	SDI, I/O
82	I/O	GND	I/O
84	GND	I/O	GND
85	I/O	PRA, I/O	I/O
87	PRA, I/O	CLKA, I/O	PRA, I/O
88	I/O	VCC	I/O
89	CLKA, I/O	I/O	CLKA, I/O
90	VCC	CLKB, I/O	VCC
92	CLKB, I/O	PRB, I/O	CLKB, I/O
94	PRB, I/O	GND	PRB, I/O
96	GND	I/O	GND
100	I/O	DCLK, I/O	I/O

Notes:

1. *NC : Denotes No Connection*
2. *All unlisted pin numbers are user I/O's*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

144-Pin PQFP Package (Top View)



144-Pin PQFP Package

Pin Number	A1240XL Function
2	MODE
9	GND
10	GND
11	GND
18	VCC
19	VCC
20	VCC
21	VCC
28	GND
29	GND
30	GND
44	GND
45	GND
46	GND
54	VCC
55	VCC
56	VCC
64	GND
65	GND
79	GND
80	GND
81	GND
88	GND

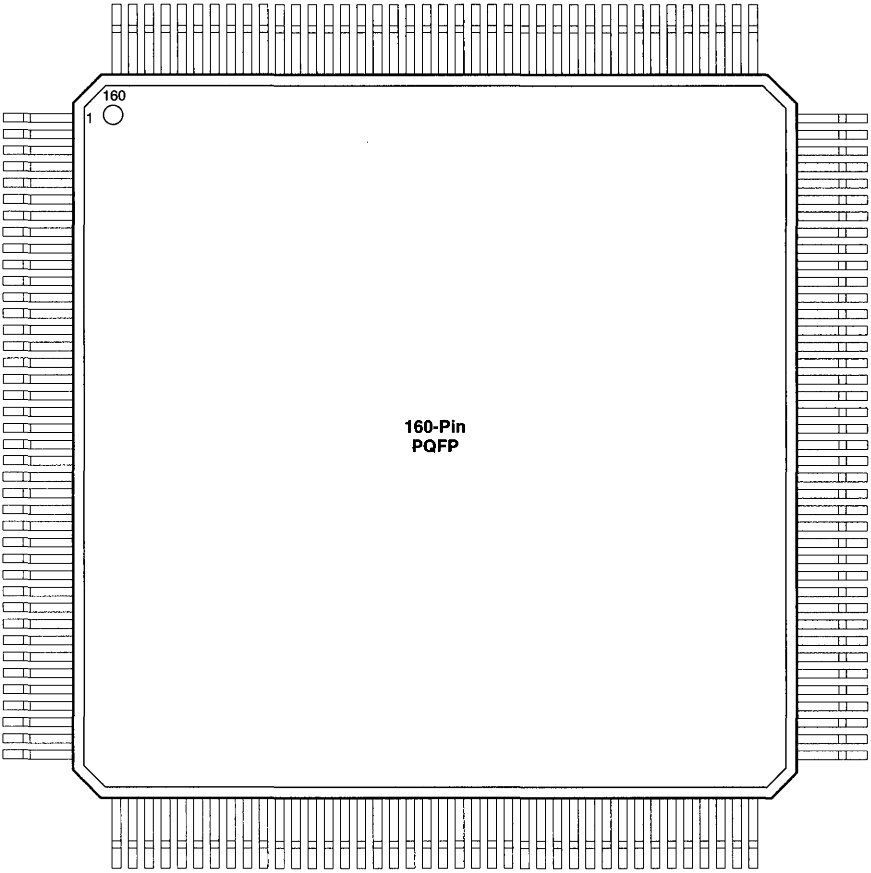
Pin Number	A1240XL Function
89	VCC
90	VCC
91	VCC
92	VCC
93	VCC
100	GND
101	GND
102	GND
110	SDI, I/O
116	GND
117	GND
118	GND
123	PRA, I/O
125	CLKA, I/O
126	VCC
127	VCC
128	VCC
130	CLKB, I/O
132	PRB, I/O
136	GND
137	GND
138	GND
144	DCLK, I/O

Notes:

1. *NC : Denotes No Connection*
2. *All unlisted pin numbers are user I/O's*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

160-pin PQFP Package (Top View)



Notes:

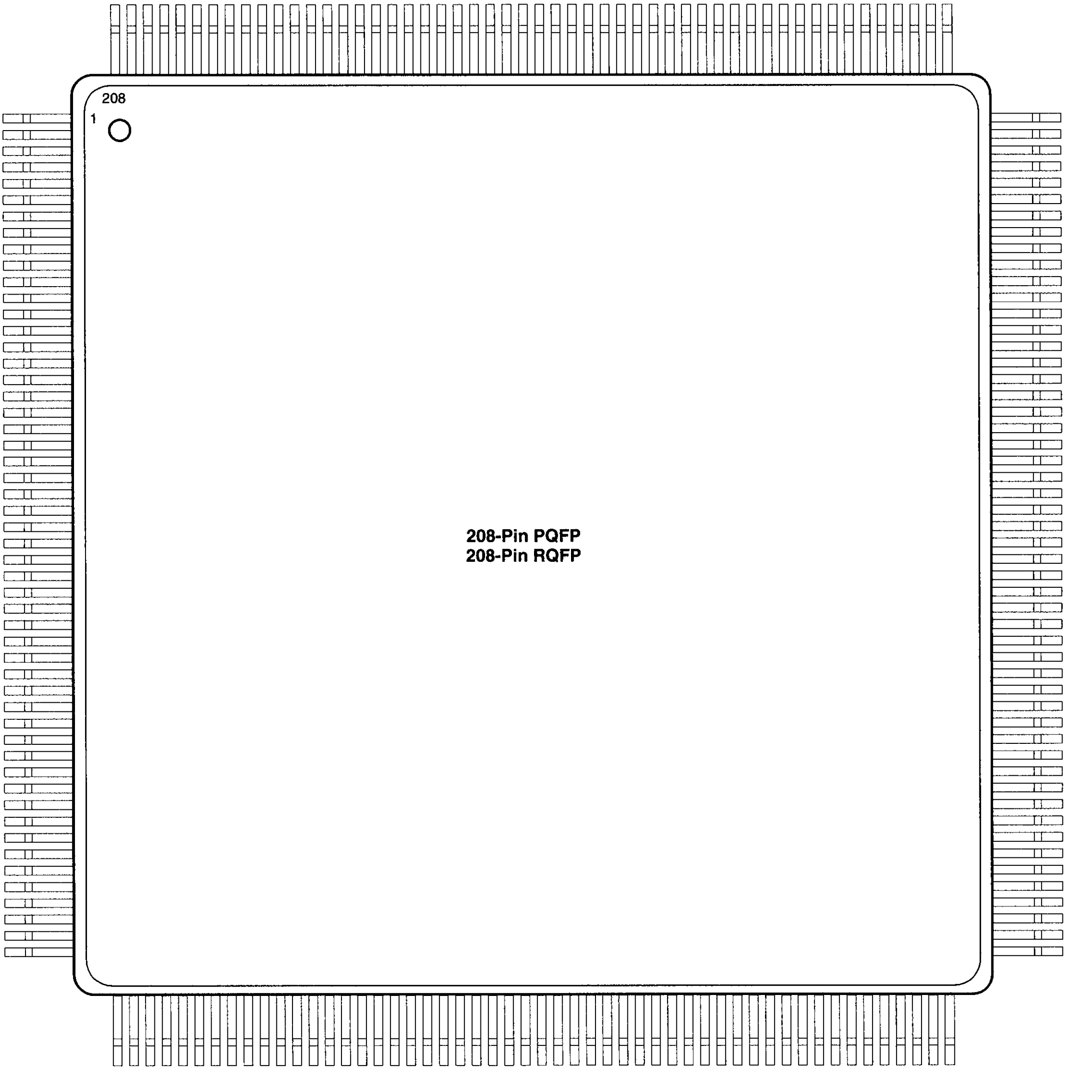
1. *I/O (WD) : Denotes I/O pin with an associated Wide Decode Module*
2. *Wide Decode I/O (WD) can also be general purpose user I/O*
3. *NC : Denotes No Connection*
4. *All unlisted pin numbers are user I/O's*
5. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

160-Pin PQFP Package

Pin Number	A3265DX Function	A1280XL Function	A32140DX Function
2	DCLK,I/O	DCLK,I/O	DCLK,I/O
4	I/O	I/O	I/O (WD)
5	I/O (WD)	I/O	I/O (WD)
6	VCC	VCC	VCC
7	I/O (WD)	I/O	I/O
11	GND	GND	GND
13	I/O (WD)	I/O	I/O (WD)
14	I/O (WD)	I/O	I/O (WD)
16	PRB,I/O	PRB,I/O	PRB,I/O
18	CLKB,I/O	CLKB,I/O	CLKB,I/O
20	VCC	VCC	VCC
21	CLKA,I/O	CLKA,I/O	CLKA,I/O
23	PRA,I/O	PRA,I/O	PRA,I/O
24	I/O	I/O	I/O (WD)
25	I/O (WD)	I/O	I/O (WD)
26	I/O (WD)	I/O	I/O
29	I/O (WD)	I/O	I/O (WD)
30	GND	GND	GND
31	I/O (WD)	I/O	I/O (WD)
34	I/O (WD)	I/O	I/O
35	VCC	VCC	VCC
36	I/O (WD)	I/O	I/O (WD)
37	I/O	I/O	I/O (WD)
38	SDI,I/O	SDI,I/O	SDI,I/O
40	GND	GND	GND
44	GND	GND	GND
49	GND	GND	GND
54	VCC	VCC	VCC
57	VCC	VCC	VCC
58	VCC	VCC	VCC
59	GND	GND	GND
60	VCC	VCC	VCC
61	GND	GND	GND
62	I/O	I/O	TCK, I/O
64	GND	GND	GND
69	GND	GND	GND
80	GND	GND	GND
82	I/O	I/O	TDO, I/O
83	I/O	I/O	I/O (WD)
84	I/O	I/O	I/O (WD)
86	VCC	VCC	VCC
87	I/O (WD)	I/O	I/O
88	I/O (WD)	I/O	I/O (WD)
89	GND	GND	GND
92	I/O (WD)	I/O	I/O
93	I/O (WD)	I/O	I/O
96	I/O (WD)	I/O	I/O (WD)
97	I/O (WD)	I/O	I/O
98	VCC	VCC	VCC
99	GND	GND	GND
106	I/O (WD)	I/O	I/O (WD)
107	I/O (WD)	I/O	I/O (WD)
109	GND	GND	GND
111	I/O (WD)	I/O	I/O (WD)
112	I/O (WD)	I/O	I/O (WD)
114	VCC	VCC	VCC
115	I/O	I/O	I/O (WD)
116	I/O	I/O	I/O (WD)
118	I/O	I/O	TDI, I/O
119	I/O	I/O	TMS, I/O
120	GND	GND	GND
125	GND	GND	GND
130	GND	GND	GND
135	VCC	VCC	VCC
138	VCC	VCC	VCC
139	VCC	VCC	VCC
140	GND	GND	GND
145	GND	GND	GND
150	VCC	VCC	VCC
155	GND	GND	GND
159	MODE	MODE	MODE
160	GND	GND	GND

Package Pin Assignments (continued)

208-Pin PQFP Package, 208-pin RQFP Package (Top View)



Notes:

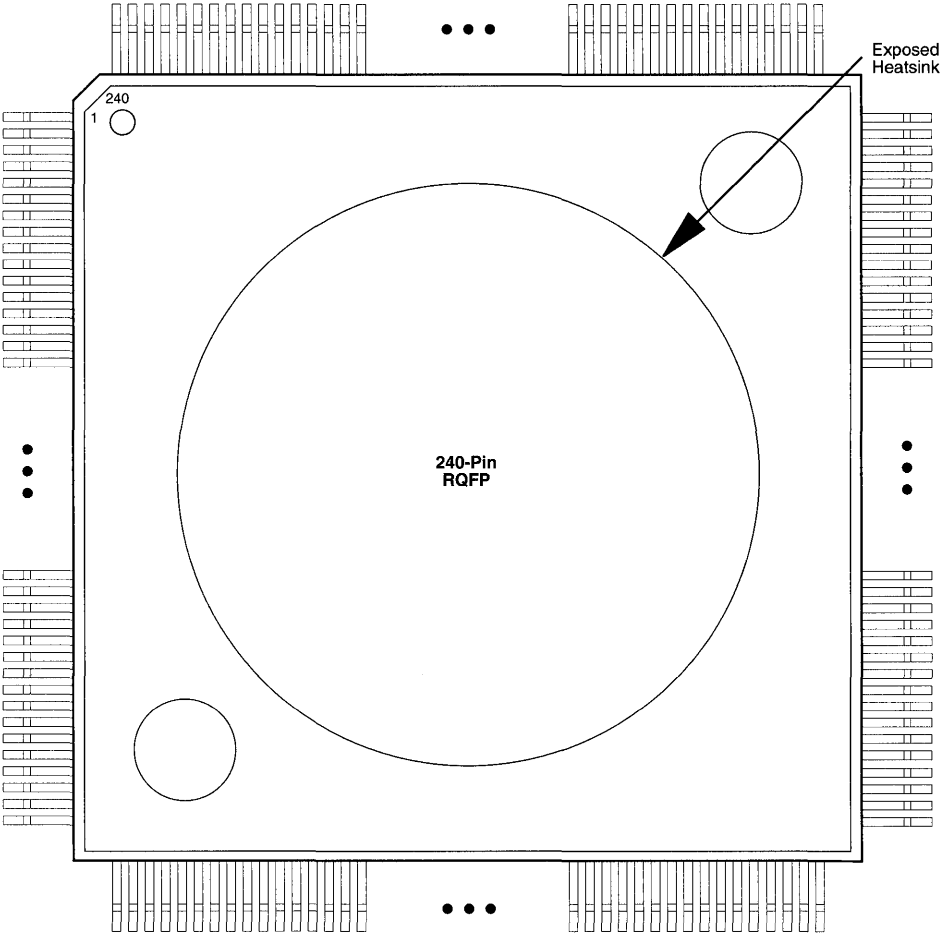
1. *I/O (WD) : Denotes I/O pin with an associated Wide Decode Module*
2. *Wide Decode I/O (WD) can also be general purpose user I/O*
3. *NC : Denotes No Connection*
4. *All unlisted pin numbers are user I/O's*
5. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*
6. *RQFP has an exposed circular metal heatsink on the top surface.*

208-Pin PQFP Package, 208-pin RQFP Package

Pin Number	A1280XL Function	A32140DX Function	A32200DX-PQ208 Function	A32200DX-RQ208 Function
1	GND	GND	GND	I/O
2	NC	VCC	VCC	DCLK, I/O
3	MODE	MODE	MODE	I/O
5	I/O	I/O	I/O	I/O (WD)
6	I/O	I/O	I/O	I/O (WD)
7	I/O	I/O	I/O	VCC
9	NC	I/O	I/O	I/O
10	NC	I/O	I/O	I/O
11	NC	I/O	I/O	I/O
13	I/O	I/O	I/O	QCLKC, I/O
15	I/O	I/O	I/O	I/O (WD)
16	NC	I/O	I/O	I/O (WD)
17	VCC	VCC	VCC	I/O
19	I/O	I/O	I/O	I/O (WD)
20	I/O	I/O	I/O	I/O (WD)
22	GND	GND	GND	PRB, I/O
24	I/O	I/O	I/O	CLKB, I/O
26	I/O	I/O	I/O	GND
27	GND	GND	GND	VCC
28	VCC	VCC	VCC	I/O
29	VCC	VCC	VCC	CLKA, I/O
30	I/O	I/O	I/O	PRA, I/O
32	VCC	VCC	VCC	I/O (WD)
33	I/O	I/O	I/O	I/O (WD)
38	I/O	I/O	I/O	QCLKD, I/O
40	I/O	I/O	I/O	I/O (WD)
41	NC	I/O	I/O	I/O (WD)
42	NC	I/O	I/O	I/O
43	NC	I/O	I/O	I/O
45	I/O	I/O	I/O	VCC
47	I/O	I/O	I/O	I/O (WD)
48	I/O	I/O	I/O	I/O (WD)
50	NC	I/O	I/O	SDI, I/O
51	NC	I/O	I/O	I/O
52	GND	GND	GND	GND
53	GND	GND	GND	I/O
54	I/O	TMS, I/O	TMS, I/O	I/O
55	I/O	TDI, I/O	TDI, I/O	I/O
57	I/O	I/O (WD)	I/O (WD)	I/O
58	I/O	I/O (WD)	I/O (WD)	I/O
59	I/O	I/O	I/O	GND
60	VCC	VCC	VCC	I/O
61	NC	I/O	I/O	I/O
62	NC	I/O	I/O	I/O
65	I/O	I/O	QCLKA, I/O	I/O
66	I/O	I/O (WD)	I/O (WD)	I/O
67	NC	I/O (WD)	I/O (WD)	I/O
68	NC	I/O	I/O	I/O
70	I/O	I/O (WD)	I/O (WD)	I/O
71	I/O	I/O (WD)	I/O (WD)	I/O
74	I/O	I/O	I/O	VCC
77	I/O	I/O	I/O	VCC
78	GND	GND	GND	VCC
79	VCC	VCC	VCC	VCC
80	NC	VCC	VCC	GND
81	I/O	I/O	I/O	TCK, I/O
83	I/O	I/O	I/O	GND
85	I/O	I/O (WD)	I/O (WD)	I/O
86	I/O	I/O (WD)	I/O (WD)	I/O
89	NC	I/O	I/O	I/O
90	NC	I/O	I/O	I/O
91	I/O	I/O	QCLKB, I/O	I/O
93	I/O	I/O (WD)	I/O (WD)	I/O
94	I/O	I/O (WD)	I/O (WD)	I/O
95	NC	I/O	I/O	I/O
96	NC	I/O	I/O	I/O
97	NC	I/O	I/O	I/O
98	VCC	VCC	VCC	I/O
100	I/O	I/O (WD)	I/O (WD)	I/O
101	I/O	I/O (WD)	I/O (WD)	I/O
103	I/O	TDO, I/O	TDO, I/O	VCC
104	I/O	I/O	I/O	GND
105	GND	GND	GND	I/O
106	NC	VCC	VCC	TDO, I/O
107	I/O	I/O	I/O	I/O (WD)
108	I/O	I/O	I/O	I/O (WD)
110	I/O	I/O	I/O	VCC
112	NC	I/O	I/O	I/O
113	NC	I/O	I/O	I/O
114	NC	I/O	I/O	I/O (WD)
115	NC	I/O	I/O	I/O (WD)
117	I/O	I/O	I/O	QCLKB, I/O
121	I/O	I/O	I/O	I/O (WD)
122	I/O	I/O	I/O	I/O (WD)
126	GND	GND	GND	I/O
128	I/O	TCK, I/O	TCK, I/O	I/O
129	GND	GND	GND	VCC
130	VCC	VCC	VCC	GND
131	GND	GND	GND	I/O
132	VCC	VCC	VCC	I/O
133	VCC	VCC	VCC	I/O
136	VCC	VCC	VCC	I/O
137	I/O	I/O	I/O	I/O (WD)
138	I/O	I/O	I/O	I/O (WD)
141	NC	I/O	I/O	I/O (WD)
142	I/O	I/O	I/O	I/O (WD)
144	I/O	I/O	I/O	QCLKA, I/O
146	NC	I/O	I/O	I/O
147	NC	I/O	I/O	I/O
148	NC	I/O	I/O	I/O
149	NC	I/O	I/O	VCC
150	GND	GND	GND	I/O
151	I/O	I/O	I/O	I/O (WD)
152	I/O	I/O	I/O	I/O (WD)
154	I/O	I/O	I/O	TDI, I/O
155	I/O	I/O	I/O	TMS, I/O
156	I/O	I/O	I/O	GND
157	GND	GND	GND	VCC
159	SDI,I/O	SDI,I/O	SDI,I/O	I/O
161	I/O	I/O (WD)	I/O (WD)	I/O
162	I/O	I/O (WD)	I/O (WD)	I/O
164	VCC	VCC	VCC	I/O
165	NC	I/O	I/O	I/O
166	NC	I/O	I/O	I/O
168	I/O	I/O (WD)	I/O (WD)	I/O
169	I/O	I/O (WD)	I/O (WD)	I/O
171	NC	I/O	QCLKD, I/O	I/O
176	I/O	I/O (WD)	I/O (WD)	I/O
177	I/O	I/O (WD)	I/O (WD)	I/O
178	PRA,I/O	PRA,I/O	PRA,I/O	VCC
180	CLKA,I/O	CLKA,I/O	CLKA,I/O	I/O
181	NC	I/O	I/O	VCC
182	NC	VCC	VCC	VCC
183	VCC	VCC	VCC	I/O
184	GND	GND	GND	I/O
186	CLKB,I/O	CLKB,I/O	CLKB,I/O	I/O
187	I/O	I/O	I/O	GND
188	PRB,I/O	PRB,I/O	PRB,I/O	I/O
190	I/O	I/O (WD)	I/O (WD)	I/O
191	I/O	I/O (WD)	I/O (WD)	I/O
193	NC	I/O	I/O	I/O
194	NC	I/O (WD)	I/O (WD)	I/O
195	NC	I/O (WD)	I/O (WD)	I/O
196	I/O	I/O	QCLKC, I/O	I/O
197	NC	I/O	I/O	I/O
201	NC	I/O	I/O	I/O
202	VCC	VCC	VCC	I/O
203	I/O	I/O (WD)	I/O (WD)	I/O
204	I/O	I/O (WD)	I/O (WD)	I/O
206	I/O	I/O	I/O	MODE
207	DCLK,I/O	DCLK,I/O	DCLK,I/O	VCC
208	I/O	I/O	I/O	GND

Package Pin Assignments (continued)

240-Pin RQFP Package (Top View)



Notes:

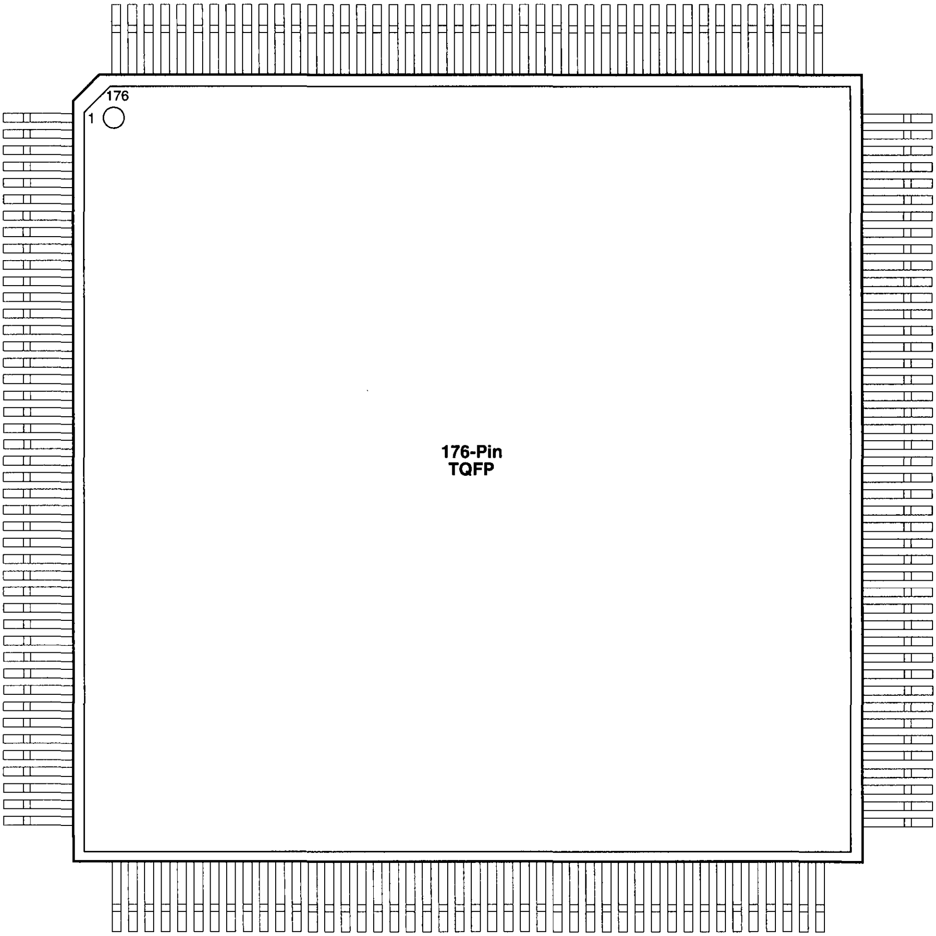
1. *I/O (WD) : Denotes I/O pin with an associated Wide Decode Module*
2. *Wide Decode I/O (WD) can also be general purpose user I/O*
3. *NC : Denotes No Connection*
4. *All unlisted pin numbers are user I/O's*
5. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*
6. *RQFP has an exposed circular metal heatsink on the top surface.*

240-Pin RQFP Package

Pin Number	A32200DX Function
2	DCLK, I/O
6	I/O (WD)
7	I/O (WD)
8	VCC
15	QCLKC, I/O
17	I/O (WD)
18	I/O (WD)
21	I/O (WD)
22	I/O (WD)
24	PRB, I/O
26	CLKB, I/O
28	GND
29	VCC
30	VCC
32	CLKA, I/O
34	PRA, I/O
37	I/O (WD)
38	I/O (WD)
45	QCLKD, I/O
47	I/O (WD)
48	I/O (WD)
52	VCC
54	I/O (WD)
55	I/O (WD)
57	SDI, I/O
59	VCC
60	GND
61	GND
71	VCC
85	VCC
88	VCC
89	VCC
90	VCC
91	GND
92	TCK, I/O
94	GND
108	VCC
118	VCC
119	GND
120	GND
121	GND
123	TDO, I/O
125	I/O (WD)
126	I/O (WD)
128	VCC
132	I/O (WD)
133	I/O (WD)
135	QCLKB, I/O
142	I/O (WD)
143	I/O (WD)
150	VCC
151	VCC
152	GND
159	I/O (WD)
160	I/O (WD)
163	I/O (WD)
164	I/O (WD)
166	QCLKA, I/O
172	VCC
174	I/O (WD)
175	I/O (WD)
178	TDI, I/O
179	TMS, I/O
180	GND
181	VCC
182	GND
192	VCC
206	VCC
209	VCC
210	VCC
219	VCC
227	VCC
237	GND
238	MODE
239	VCC
240	GND

Package Pin Assignments (continued)

176-Pin TQFP Package (Top View)



Notes:

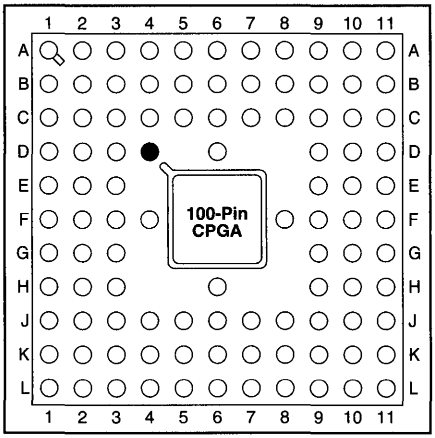
1. *I/O (WD) : Denotes I/O pin with an associated Wide Decode Module*
2. *Wide Decode I/O (WD) can also be general purpose user I/O*
3. *NC : Denotes No Connection*
4. *All unlisted pin numbers are user I/O's*
5. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

176-pin TQFP Package

Pin Number	A1240XL Function	A3265DX Function	A1280XL Function	A32140DX Function
1	GND	GND	GND	GND
2	MODE	MODE	MODE	MODE
8	NC	NC	NC	I/O
10	NC	NC	I/O	I/O
11	NC	NC	I/O	I/O
13	NC	VCC	VCC	VCC
18	GND	GND	GND	GND
19	NC	I/O	I/O	I/O
20	NC	I/O	I/O	I/O
22	NC	I/O	I/O	I/O
23	GND	GND	GND	GND
24	NC	VCC	VCC	VCC
25	VCC	VCC	VCC	VCC
26	NC	I/O	I/O	I/O
27	NC	I/O	I/O	I/O
28	VCC	VCC	VCC	VCC
29	NC	NC	I/O	I/O
33	NC	NC	NC	I/O
37	NC	NC	I/O	I/O
38	NC	NC	NC	I/O
45	GND	GND	GND	GND
46	I/O	I/O	I/O	TMS, I/O
47	I/O	I/O	I/O	TDI, I/O
48	I/O	NC	I/O	I/O
49	I/O	I/O	I/O	I/O (WD)
50	I/O	I/O	I/O	I/O (WD)
52	NC	VCC	VCC	VCC
54	NC	I/O (WD)	I/O	I/O
55	NC	I/O (WD)	I/O	I/O (WD)
56	I/O	I/O	I/O	I/O (WD)
57	NC	NC	NC	I/O
59	I/O	I/O (WD)	I/O	I/O (WD)
60	I/O	I/O (WD)	I/O	I/O (WD)
61	NC	I/O	I/O	I/O
64	NC	I/O	I/O	I/O
66	NC	I/O	I/O	I/O
67	GND	GND	GND	GND
68	VCC	VCC	VCC	VCC
69	I/O	I/O (WD)	I/O	I/O (WD)
70	I/O	I/O (WD)	I/O	I/O (WD)
73	I/O	I/O (WD)	I/O	I/O
74	NC	NC	I/O	I/O
75	I/O	I/O (WD)	I/O	I/O
77	NC	NC	NC	I/O (WD)
78	NC	NC	I/O	I/O (WD)
80	NC	I/O (WD)	I/O	I/O
81	I/O	I/O (WD)	I/O	I/O
82	NC	VCC	VCC	VCC
84	I/O	I/O	I/O	I/O (WD)
85	I/O	I/O	I/O	I/O (WD)
86	NC	NC	I/O	I/O
87	I/O	I/O	I/O	TDO, I/O
89	GND	GND	GND	GND
96	NC	NC	I/O	I/O
97	NC	I/O	I/O	I/O
101	NC	NC	NC	I/O
103	NC	I/O	I/O	I/O
106	GND	GND	GND	GND
107	NC	I/O	I/O	I/O
108	NC	I/O	I/O	TCK, I/O
109	GND	GND	GND	GND
110	VCC	VCC	VCC	VCC
111	GND	GND	GND	GND
112	VCC	VCC	VCC	VCC
113	VCC	VCC	VCC	VCC
114	NC	I/O	I/O	I/O
115	NC	I/O	I/O	I/O
116	NC	VCC	VCC	VCC
117	I/O	NC	I/O	I/O
121	NC	NC	NC	I/O
124	NC	NC	I/O	I/O
125	NC	NC	I/O	I/O
126	NC	NC	NC	I/O
133	GND	GND	GND	GND
135	SDI,I/O	SDI,I/O	SDI,I/O	SDI,I/O
136	NC	NC	I/O	I/O
137	I/O	I/O	I/O	I/O (WD)
138	I/O	I/O	I/O	I/O (WD)
139	I/O	I/O (WD)	I/O	I/O
140	NC	VCC	VCC	VCC
141	I/O	I/O (WD)	I/O	I/O
143	NC	I/O	I/O	I/O
144	NC	I/O (WD)	I/O	I/O (WD)
145	NC	NC	NC	I/O (WD)
146	I/O	I/O (WD)	I/O	I/O
147	NC	I/O	I/O	I/O
149	I/O	I/O (WD)	I/O	I/O
150	I/O	I/O (WD)	I/O	I/O (WD)
151	NC	I/O	I/O	I/O (WD)
152	PRA,I/O	PRA,I/O	PRA,I/O	PRA,I/O
154	CLKA,I/O	CLKA,I/O	CLKA,I/O	CLKA,I/O
155	VCC	VCC	VCC	VCC
156	GND	GND	GND	GND
158	CLKB,I/O	CLKB,I/O	CLKB,I/O	CLKB,I/O
160	PRB,I/O	PRB,I/O	PRB,I/O	PRB,I/O
161	NC	I/O	I/O	I/O (WD)
162	I/O	I/O (WD)	I/O	I/O (WD)
163	I/O	I/O (WD)	I/O	I/O
165	NC	NC	NC	I/O (WD)
166	NC	I/O	I/O	I/O (WD)
168	NC	I/O	I/O	I/O
169	I/O	I/O (WD)	I/O	I/O
170	NC	VCC	VCC	VCC
171	I/O	I/O (WD)	I/O	I/O (WD)
172	I/O	I/O	I/O	I/O (WD)
173	NC	NC	I/O	I/O
175	DCLK,I/O	DCLK,I/O	DCLK,I/O	DCLK,I/O

Package Pin Assignments (continued)

100-Pin CPGA (Top View)



● Orientation Pin

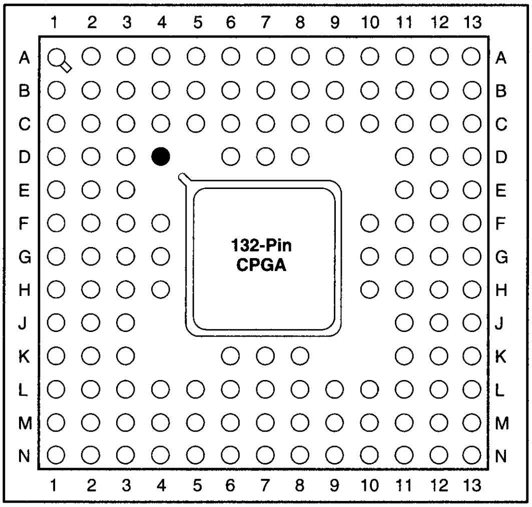
Signal	Pad Number	Location
PRA or I/O	85	A7
PRB or I/O	92	A4
MODE	2	C2
SDI or I/O	77	C8
DCLK or I/O	100	C3
CLKA or I/O	87	C6
CLKB or I/O	90	D6
GND	7, 20, 32, 44, 55, 70, 82, 94	E3, G3, J5, J7, G9, F11, D10, C7, C5
V_{CC}	15, 38, 64, 88	F3, G1, K6, F9, F10, E11, B6

Notes:

1. *Unused I/O pins are designated as outputs by ALS and are driven low.*
2. *All unassigned pins are available for use as I/Os.*
3. *MODE = GND, except during device programming or debugging.*

Package Pin Assignments (continued)

132-Pin CPGA (Top View)



● Orientation Pin

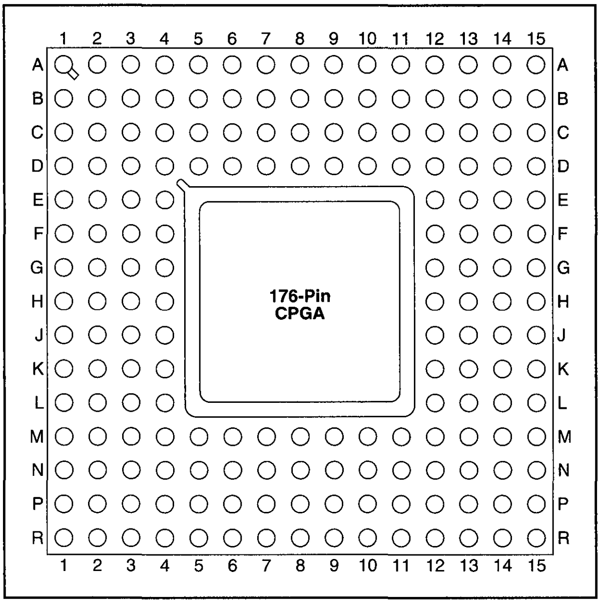
Signal	Pad Number	Location
PRA or I/O	113	B8
PRB or I/O	121	C6
MODE	2	A1
SDI or I/O	101	B12
DCLK or I/O	132	C3
CLKA or I/O	115	B7
CLKB or I/O	119	B6
GND	9, 10, 26, 27, 41, 58, 59, 73, 74, 92, 93, 107, 108, 125, 126	E3, F4, J2, J3, L5, L9, M9, K12, J11, H13, E12, E11, C9, B9, B5, C5
V_{CC}	18, 19, 49, 50, 83, 84, 116, 117	G3, G2, G4, L7, K7, G10, G11, G12, G13, D7, C7

Notes:

1. *Unused I/O pins are designated as outputs by ALS and are driven low.*
2. *All unassigned pins are available for use as I/Os.*
3. *MODE = GND, except during device programming or debugging.*

Package Pin Assignments (continued)

176-Pin CPGA (Top View)



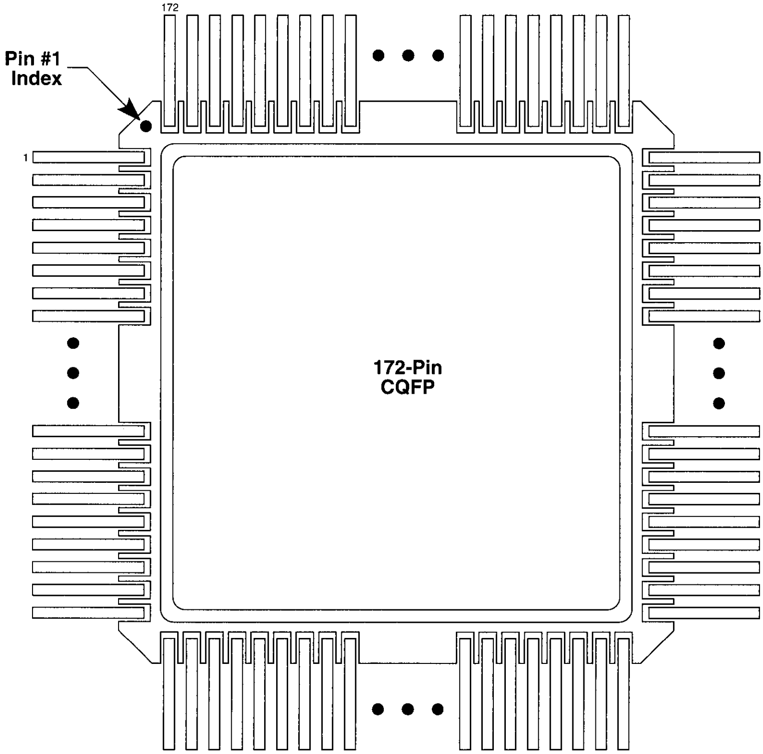
Signal	Pad Number	Location
PRA or I/O	152	C9
PRB or I/O	160	D7
MODE	2	C3
SDI or I/O	135	B14
DCLK or I/O	175	B3
CLKA or I/O	154	A9
CLKB or I/O	158	B8
GND	1, 8, 18, 23, 33, 38, 45, 57, 67, 77, 89 101, 106, 111, 121, 126, 133, 145, 156, 165	D4, E4, G4, H4, K4, L4, M4, M6, M8, M10, M12 K12, J12, J13, H12, F12, E12, D12, D10, C8, D6
V_{CC}	13, 24, 28, 52, 68, 82, 112, 116, 140, 155, 170	F4, H2, H3, J4, M5, N8, M11, J14, H13, H14, G12, D11, D8, D5

Notes:

1. *Unused I/O pins are designated as outputs by ALS and are driven low.*
2. *All unassigned pins are available for use as I/Os.*
3. *MODE = GND, except during device programming or debugging.*

Package Pin Assignments (continued)

172-Pin CQFP



Signal	Pad Number
CLKA or I/O	150
CLKB or I/O	154
DCLK or I/O	171
GND	7, 17, 22, 32, 37, 55, 65, 75, 98, 103, 106, 118, 123, 141, 152, 161
MODE	1
PRA or I/O	148
PRB or I/O	156
SDI or I/O	131
V_{CC}	12, 23, 24, 27, 50, 66, 80, 107, 109, 110, 113, 136, 151, 166

Notes:

1. *Unused I/O pins are designated as outputs by ALS and are driven low.*
2. *All unassigned pins are available for use as I/Os.*
3. *MODE = GND, except during device programming or debugging.*

Accelerator Series FPGAs – ACT™ 3 Family

Features

- Up to 10,000 Gate Array Equivalent Gates (up to 25,000 equivalent PLD Gates)
- Highly Predictable Performance with 100% Automatic Placement and Routing
- 7.5 ns Clock-to-Output Times
- Up to 250 MHz On-Chip Performance
- Up to 228 User-Programmable I/O Pins
- Four Fast, Low-Skew Clock Networks
- More than 500 Macro Functions
- Replaces up to twenty 32 macro-cell CPLDs
- Replaces up to one hundred 20-pin PAL® Packages
- Up to 1153 Dedicated Flip-Flops
- VQFP, TQFP, BGA, and PQFP Packages
- Nonvolatile, User Programmable
- Low-power 0.8 micron CMOS Technology
- Fully Tested Prior to Shipment
- 5.0V and 3.3V Versions
- Optimized for Logic Synthesis Methodologies

Device	A1415	A1425	A1440	A1460	A14100
Capacity					
Gate Array Equivalent Gates	1,500	2,500	4,000	6,000	10,000
PLD Equivalent Gates	3,750	6,250	10,000	15,000	25,000
TTL Equivalent Packages (40 gates)	40	60	100	150	250
20-Pin PAL Equivalent Packages (100 gates)	15	25	40	60	100
Logic Modules	200	310	564	848	1,377
S-Module	104	160	288	432	697
C-Module	96	150	276	416	680
Dedicated Flip-Flops[1]	264	360	568	768	1,153
User I/Os (maximum)	80	100	140	168	228
Packages[2] (by pin count)					
CPGA	100	133	175	207	257
PLCC	84	84	84	—	—
PQFP	100	100, 160	160	160, 208	—
RQFP	—	—	—	—	208
VQFP	100	100	100	—	—
TQFP	—	—	176	176	—
BGA	—	—	—	225	313
CQFP	—	132	—	196	256
Performance[3] (maximum, worst-case commercial)					
Chip-to-Chip[4]	108 MHz	108 MHz	100 MHz	97 MHz	93 MHz
Accumulators (16-bit)	63 MHz	63 MHz	63 MHz	63 MHz	63 MHz
Loadable Counter (16-bit)	110 MHz	110 MHz	110 MHz	110 MHz	105 MHz
Prescaled Loadable Counters (16-bit)	250 MHz	250 MHz	250 MHz	200 MHz	200 MHz
Datapath, Shift Registers	250 MHz	250 MHz	250 MHz	200 MHz	200 MHz
Clock-to-Output (pad-to-pad)	7.5 ns	7.5 ns	8.5 ns	9.0 ns	9.5 ns

Notes:

1. *One flip-flop per S-Module, two flip-flops per I/O-Module.*
2. *See product plan on page 1-76 for package availability.*
3. *Based on A1415A-3, A1425A-3, A1440B-3, A1460B-3, and A14100B-3.*
4. *Clock-to-Output + Setup*

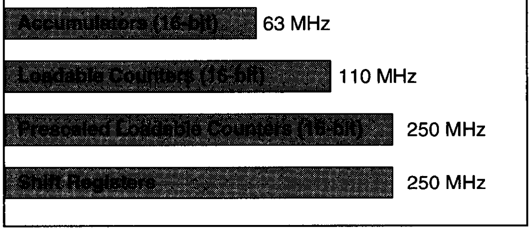
Description

Actel's ACT 3 Accelerator Series of FPGAs offers the industry's fastest high-capacity programmable logic device. ACT 3 FPGAs offer a high perfomance, PCI compliant programmable solution capable of 250 MHz on-chip performance and 7.5 nanosecond clock-to-output, with capacities spanning from 1,500 to 10,000 gate array equivalent gates. For further information regarding PCI compliance of ACT 3 devices, see "Accelerator Series FPGAs—ACT 3 PCI Compliant Family."

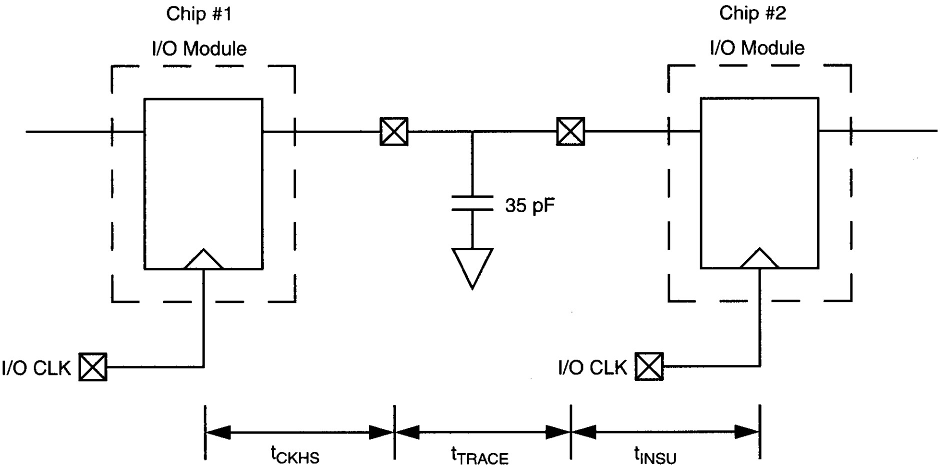
The ACT 3 family builds on the proven two-module architecture consisting of combinatorial and sequential logic modules used in Actel's 3200DX and 1200XL families. In addition, the ACT 3 I/O modules contain registers which deliver 7.5 nanosecond clock-to-out times. The devices contain four clock distribution networks, including dedicated array and I/O clocks, supporting very fast synchronous and asynchronous designs. In addition, routed clocks can be used to drive high fanout signals such as flip-flop resets and output enables.

The ACT 3 family is supported by Actel's Designer Series Development System which offers automatic placement and routing (with automatic or fixed pin assignments), static timing anlaysis, user programming, and debug and diagnostic probe capabilities. The Designer Series is supported on the following platforms: 486/Pentium class PC's, Sun®, and HP®, workstations. The software provides CAE interfaces to Cadence, Mentor Graphics®, OrCAD™ and Viewlogic®, design environments. Additional platforms are supported through Actel's Industry Alliance Program, including DATA I/O (ABEL FPGA) and MINC.

Predictable Performance* (Worst-Case Commercial)

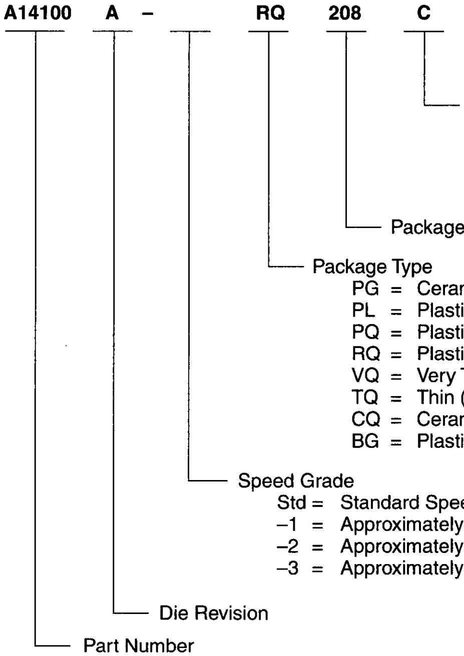


System Performance Model



Chip-to-Chip Performance (Worst-Case Commercial)					
	t_{CKHS}	t_{TRACE}	t_{INSU}	Total	MHz
A1425A-3	7.5	1.0	1.8	10.3 ns	97
A1460A-3	9.0	1.0	1.3	11.3 ns	88

Ordering Information



Application (Temperature Range)

C	=	Commercial (0 to +70°C)
I	=	Industrial (–40 to +85°C)
M	=	Military (–55 to +125°C)
B	=	MIL-STD-883

Package Lead Count

Package Type

PG	=	Ceramic Pin Grid Array
PL	=	Plastic Leaded Chip Carrier
PQ	=	Plastic Quad Flatpack
RQ	=	Plastic Power Quad Flatpack
VQ	=	Very Thin (1.0 mm) Quad Flatpack
TQ	=	Thin (1.4 mm) Quad Flatpack
CQ	=	Ceramic Quad Flatpack
BG	=	Plastic Ball Grid Array

Speed Grade

Std	=	Standard Speed
–1	=	Approximately 15% faster than Standard
–2	=	Approximately 25% faster than Standard
–3	=	Approximately 35% faster than Standard

Die Revision

Part Number

A1415A	=	1500 Gates
A14V15A	=	1500 Gates (3.3V)
A1425A	=	2500 Gates
A14V25A	=	2500 Gates (3.3V)
A1440A	=	4000 Gates
A14V40A	=	4000 Gates (3.3V)
A1440B	=	4000 Gates
A1460A	=	6000 Gates
A14V60A	=	6000 Gates (3.3V)
A1460B	=	6000 Gates
A14100A	=	10000 Gates
A14V100A	=	10000 Gates (3.3V)
A14100B	=	10000 Gates

Product Plan

	Speed Grade*				Application			
	Std	–1	–2	–3	C	I	M	B
A1415A Device								
84-pin Plastic Leaded Chip Carrier (PLCC)	✔	✔	✔	P	✔	✔	—	—
100-pin Plastic Quad Flatpack (PQFP)	✔	✔	✔	P	✔	✔	—	—
100-pin Very Thin Quad Flatpack (VQFP)	✔	✔	✔	P	✔	✔	—	—
100-pin Ceramic Pin Grid Array (CPGA)	✔	✔	✔	P	✔	—	—	—
A14V15A Device								
84-pin Plastic Leaded Chip Carrier (PLCC)	✔	—	—	—	✔	—	—	—
100-pin Very Thin Quad Flatpack (VQFP)	✔	—	—	—	✔	—	—	—
A1425A Device								
84-pin Plastic Leaded Chip Carrier (PLCC)	✔	✔	✔	P	✔	✔	—	—
100-pin Plastic Quad Flatpack (PQFP)	✔	✔	✔	P	✔	✔	—	—
100-pin Very Thin Quad Flatpack (VQFP)	✔	✔	✔	P	✔	✔	—	—
132-pin Ceramic Quad Flatpack (CQFP)	✔	✔	—	—	✔	—	✔	✔
133-pin Ceramic Pin Grid Array (CPGA)	✔	✔	✔†	P†	✔	—	✔	✔
160-pin Plastic Quad Flatpack (PQFP)	✔	✔	✔	P	✔	✔	—	—
A14V25A Device								
84-pin Plastic Leaded Chip Carrier (PLCC)	✔	—	—	—	✔	—	—	—
100-pin Very Thin Quad Flatpack (VQFP)	✔	—	—	—	✔	—	—	—
160-pin Plastic Quad Flatpack (PQFP)	✔	—	—	—	✔	—	—	—
A1440A Device								
84-pin Plastic Leaded Chip Carrier (PLCC)	✔	✔	✔	P	✔	✔	—	—
100-pin Very Thin Quad Flatpack (VQFP)	✔	✔	✔	P	✔	—	—	—
160-pin Plastic Quad Flatpack (PQFP)	✔	✔	✔	P	✔	✔	—	—
175-pin Ceramic Pin Grid Array (CPGA)	✔	✔	✔	P	✔	—	—	—
176-pin Thin Quad Flatpack (TQFP)	✔	✔	✔	P	✔	—	—	—
A14V40A Device								
84-pin Plastic Leaded Chip Carrier (PLCC)	✔	—	—	—	✔	—	—	—
100-pin Very Thin Quad Flatpack (VQFP)	✔	—	—	—	✔	—	—	—
160-pin Plastic Quad Flatpack (PQFP)	✔	—	—	—	✔	—	—	—
176-pin Thin Quad Flatpack (TQFP)	✔	—	—	—	✔	—	—	—
A1440B Device								
84-pin Plastic Leaded Chip Carrier (PLCC)	P	P	P	P	P	P	—	—
100-pin Very Thin Quad Flatpack (VQFP)	P	P	P	P	P	P	—	—
160-pin Plastic Quad Flatpack (PQFP)	P	P	P	P	P	P	—	—
176-pin Thin Quad Flatpack (TQFP)	P	P	P	P	P	P	—	—
A1460A Device								
160-pin Plastic Quad Flatpack (PQFP)	✔	✔	✔	P	✔	✔	—	—
176-pin Thin Quad Flatpack (TQFP)	✔	✔	✔	P	✔	✔	—	—
196-pin Ceramic Quad Flatpack (CQFP)	✔	✔	—	—	✔	—	✔	✔
207-pin Ceramic Pin Grid Array (CPGA)	✔	✔	✔†	P†	✔	—	✔	✔
208-pin Plastic Quad Flatpack (PQFP)	✔	✔	✔	P	✔	✔	—	—
225-pin Platic Ball Grid Array (BGA)	✔	✔	✔	P	✔	—	—	—

Applications: C = *Commercial*
I = *Industrial*
M = *Military*
B = *MIL-STD-883*
† *Commercial Only*

Availability: ✔ = *Available*
P = *Planned*
— = *Not Planned*

* *Speed Grade:* –1 = *Approx. 15% faster than Standard*
–2 = *Approx. 25% faster than Standard*
–3 = *Approx. 35 % faster than Standard.*

Product Plan (continued)

	Speed Grade*				Application			
	Std	–1	–2	–3	C	I	M	B
A14V60A Device								
160-pin Plastic Quad Flatpack (PQFP)	✔	—	—	—	✔	—	—	—
176-pin Thin Quad Flatpack (TQFP)	✔	—	—	—	✔	—	—	—
208-pin Plastic Quad Flatpack (PQFP)	✔	—	—	—	✔	—	—	—
A1460B Device								
160-pin Plastic Quad Flatpack (PQFP)	P	P	P	P	P	P	—	—
176-pin Thin Quad Flatpack (TQFP)	P	P	P	P	P	P	—	—
208-pin Plastic Quad Flatpack (PQFP)	P	P	P	P	P	P	—	—
225-pin Plastic Ball Grid Array (BGA)	P	P	P	P	P	—	—	—
A14100A Device								
208-pin Power Quad Flatpack (RQFP)	✔	✔	✔	P	✔	✔	—	—
257-pin Ceramic Pin Grid Array (CPGA)	✔	✔	✔†	P†	✔	—	✔	✔
313-pin Plastic Ball Grid Array (BGA)	✔	✔	✔	P	✔	—	—	—
256-pin Ceramic Quad Flatpack (CQFP)	✔	✔	—	—	✔	—	✔	✔
A14V100A Device								
208-pin Power Quad Flatpack (RQFP)	✔	—	—	—	✔	—	—	—
313-pin Plastic Ball Grid Array (BGA)	✔	—	—	—	✔	—	—	—
A14100B Device								
208-pin Power Quad Flatpack (RQFP)	P	P	P	P	P	P	—	—
313-pin Plastic Ball Grid Array (BGA)	P	P	P	P	P	—	—	—

Applications: *C = Commercial* *Availability:* ✔ *= Available* ** Speed Grade:* *–1 = Approx. 15% faster than Standard*
I = Industrial *P = Planned* *–2 = Approx. 25% faster than Standard*
M = Military *— = Not Planned* *–3 = Approx. 35 % faster than Standard.*
B = MIL-STD-883
† Commercial Only

Plastic Device Resources

			User I/Os							
			PLCC	PQFP, RQFP			VQFP	TQFP	BGA	
Device Series	Logic Modules	Gates	84-pin	100-pin	160-pin	208-pin	100-pin	176-pin	225-pin	313-pin
A1415	200	1500	70	80	—	—	80	—	—	—
A1425	310	2500	70	80	100	—	83	—	—	—
A1440	564	4000	70	—	131	—	83	140	—	—
A1460	848	6000	—	—	131	167	—	151	168	—
A14100	1377	10000	—	—	—	175	—	—	—	228

Hermetic Device Resources

			User I/Os							
			CPGA					CQFP		
Device Series	Logic Modules	Gates	100-pin	133-pin	175-pin	207-pin	257-pin	132-pin	196-pin	256-pin
A1415	200	1500	80	—	—	—	—	—	—	—
A1425	310	2500	—	100	—	—	—	100	—	—
A1440	564	4000	—	—	140	—	—	—	—	—
A1460	848	6000	—	—	—	168	—	—	168	—
A14100	1377	10000	—	—	—	—	228	—	—	228

Pin Description

CLKA Clock A (Input)

TTL Clock input for clock distribution networks. The Clock input is buffered prior to clocking the logic modules. This pin can also be used as an I/O.

CLKB Clock B (Input)

TTL Clock input for clock distribution networks. The Clock input is buffered prior to clocking the logic modules. This pin can also be used as an I/O.

DCLK Diagnostic Clock (Input)

TTL Clock input for diagnostic probe and device programming. DCLK is active when the MODE pin is HIGH. This pin functions as an I/O when the MODE pin is LOW.

GND Ground

LOW supply voltage.

HCLK Dedicated (Hard-wired) Array Clock (Input)

TTL Clock input for sequential modules. This input is directly wired to each S-Module and offers clock speeds independent of the number of S-Modules being driven. This pin can also be used as an I/O.

I/O Input/Output (Input, Output)

The I/O pin functions as an input, output, three-state, or bidirectional buffer. Input and output levels are compatible with standard TTL and CMOS specifications. Unused I/O pins are automatically driven LOW by the Designer Series software.

IOCLK Dedicated (Hard-wired) I/O Clock (Input)

TTL Clock input for I/O modules. This input is directly wired to each I/O module and offers clock speeds independent of the number of I/O modules being driven. This pin can also be used as an I/O.

IOPCL Dedicated (Hard-wired) I/O Preset/Clear (Input)

TTL input for I/O preset or clear. This global input is directly wired to the preset and clear inputs of all I/O registers. This pin functions as an I/O when no I/O preset or clear macros are used.

MODE Mode (Input)

The MODE pin controls the use of diagnostic pins (DCLK, PRA, PRB, SDI). When the MODE pin is HIGH, the special functions are active. When the MODE pin is LOW, the pins function as I/Os. To provide Actionprobe capability, the MODE pin should be terminated to GND through a 10K resistor so that the MODE pin can be pulled high when required.

NC No Connection

This pin is not connected to circuitry within the device.

PRA Probe A (Output)

The Probe A pin is used to output data from any user-defined design node within the device. This independent diagnostic pin can be used in conjunction with the Probe B pin to allow real-time diagnostic output of any signal path within the device. The Probe A pin can be used as a user-defined I/O when debugging has been completed. The pin's probe capabilities can be permanently disabled to protect programmed design confidentiality. PRA is accessible when the MODE pin is HIGH. This pin functions as an I/O when the MODE pin is LOW.

PRB Probe B (Output)

The Probe B pin is used to output data from any user-defined design node within the device. This independent diagnostic pin can be used in conjunction with the Probe A pin to allow real-time diagnostic output of any signal path within the device. The Probe B pin can be used as a user-defined I/O when debugging has been completed. The pin's probe capabilities can be permanently disabled to protect programmed design confidentiality. PRB is accessible when the MODE pin is HIGH. This pin functions as an I/O when the MODE pin is LOW.

SDI Serial Data Input (Input)

Serial data input for diagnostic probe and device programming. SDI is active when the MODE pin is HIGH. This pin functions as an I/O when the MODE pin is LOW.

V_{CC} 5 V Supply Voltage

HIGH supply voltage.

Architecture

This section of the data sheet is meant to familiarize the user with the architecture of the ACT 3 family of FPGA devices. A generic description of the family will be presented first, followed by a detailed description of the logic blocks, the routing structure, the antifuses, and the special function circuits. The on-chip circuitry required to program the devices is not covered.

Topology

The ACT 3 family architecture is composed of six key elements: Logic modules, I/O modules, I/O Pad Drivers, Routing Tracks, Clock Networks, and Programming and Test Circuits. The basic structure is similar for all devices in the family, differing only in the number of rows, columns, and I/Os. The array itself consists of alternating rows of modules and channels. The logic modules and channels are in the center of the array; the I/O modules are located along the array periphery. A simplified floor plan is depicted in Figure 1.

Logic Modules

ACT 3 logic modules are enhanced versions of the 1200XL family logic modules. As in the 1200XL family, there are two types of modules: C-modules and S-modules. The C-module is functionally equivalent to the 1200XL C-module and implements high fanin combinatorial macros, such as 5-input AND, 5-input OR, and so on. It is available for use as the CM8 hard macro. The S-module is designed to implement high-speed sequential functions within a single module. S-modules consist of a full C-module driving a flip-flop, which allows an additional level of logic to be implemented without additional propagation delay. It is available for use as the DFM8A/B and DLM8A/B hard macros. C-modules and S-modules are arranged in pairs called module-pairs. Module-pairs are arranged in alternating patterns and make up the bulk of the array. This arrangement allows the placement software to support two-module macros of four types (CC, CS, SC, and SS). The C-module implements the following function:

$Y = !S1 * !S0 * D00 + !S1 * S0 * D01 + S1 * !S0 * D10 + S1 * S0 * D11$

where: $S0 = A0 * B0$ and $S1 = A1 + B1$

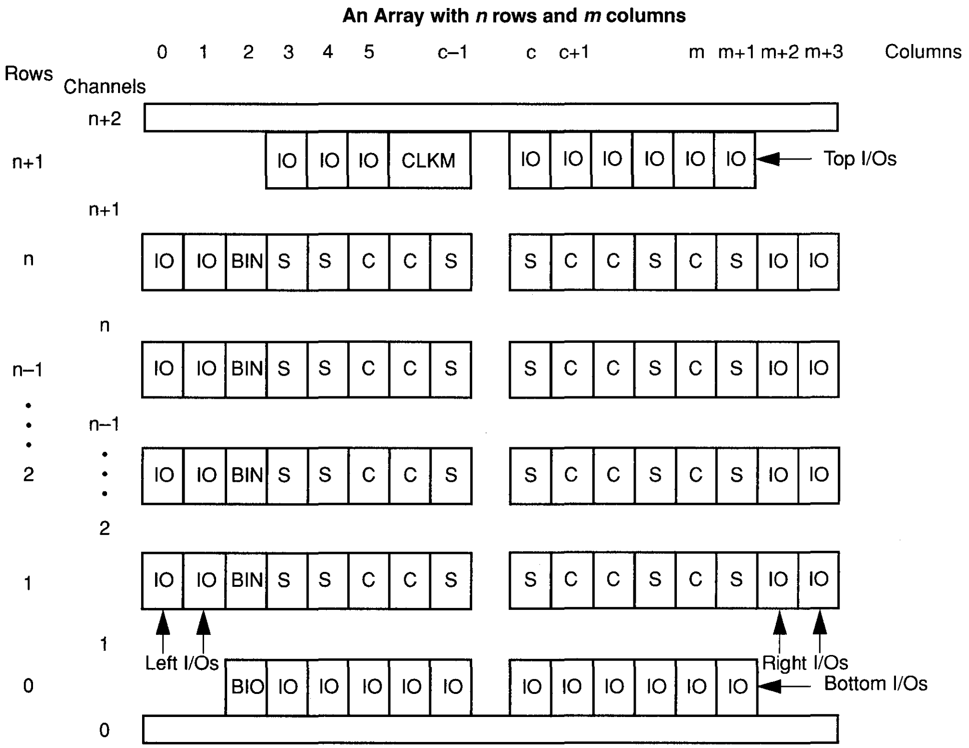

An Array with *n* rows and *m* columns
0 1 2 3 4 5 c–1 c c+1 m m+1 m+2 m+3 Columns
Rows
Channels
n+2
n+1
IO IO IO CLKM IO IO IO IO IO IO
Top I/Os
n+1
n
IO IO BIN S S C C S S C C S C S IO IO
n
n–1
IO IO BIN S S C C S S C C S C S IO IO
n–1
2
IO IO BIN S S C C S S C C S C S IO IO
2
1
IO IO BIN S S C C S S C C S C S IO IO
1
Left I/Os
Right I/Os
0
BIO IO IO IO IO IO IO IO IO IO IO IO
Bottom I/Os
0

Figure 1 • Generalized Floor Plan of ACT 3 Device

The S-module contains a full implementation of the C-module plus a clearable sequential element that can either implement a latch or flip-flop function. The S-module can therefore implement any function implemented by the C-module. This allows complex combinatorial-sequential functions to be implemented with no delay penalty. The Designer Series Development System will automatically combine any C-module macro driving an S-module macro into the S-module, thereby freeing up a logic module and eliminating a module delay.

The clear input CLR is accessible from the routing channel. In addition, the clock input may be connected to one of three clock networks: CLK0, CLK1, or HCLK. The C-module and S-module functional descriptions are shown in Figures 2 and 3. The clock selection multiplexor selects the clock input to the S-module.

I/Os

I/O Modules

I/O modules provide an interface between the array and the I/O Pad Drivers. I/O modules are located in the array and access the routing channels in a similar fashion to logic modules. There are two types of I/O modules: side and top/bottom. The I/O module schematic is shown in Figure 4. UO1 and UO2 are inputs from the routing channel, one for the routing channel above and one for the routing channel below the module. The top/bottom I/O modules interact with only one channel and therefore have only one UO input. The signals DataIn and DataOut connect to the I/O pad driver. Each I/O module contains two D-type flip-flops. Each flip-flop is connected to the dedicated I/O clock (IOCLK). Each flip-flop can be bypassed by nonsequential I/Os. In addition, each flip-flop contains a data enable input that can be accessed from the routing channels (ODE and IDE). The asynchronous preset/clear input is driven by the dedicated preset/clear network (IOPCL). Either preset or clear can be selected individually on an I/O module by I/O module basis.

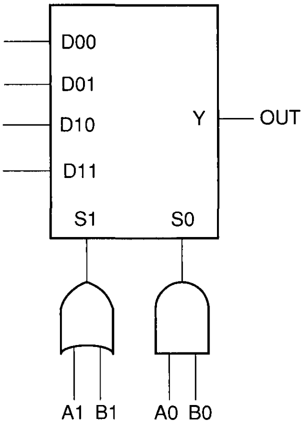


Figure 2 • C-Module Diagram

The I/O module output Y is used to bring Pad signals into the array *or* to feed the output register back into the array. This allows the output register to be used in high-speed state machine applications. Side I/O modules have a dedicated output segment for Y extending into the routing channels above and below (similar to logic modules). Top/Bottom I/O modules have no dedicated output segment. Signals coming into the chip from the top or bottom are routed using F-fuses and LVTs (F-fuses and LVTs are explained in detail in the routing section).

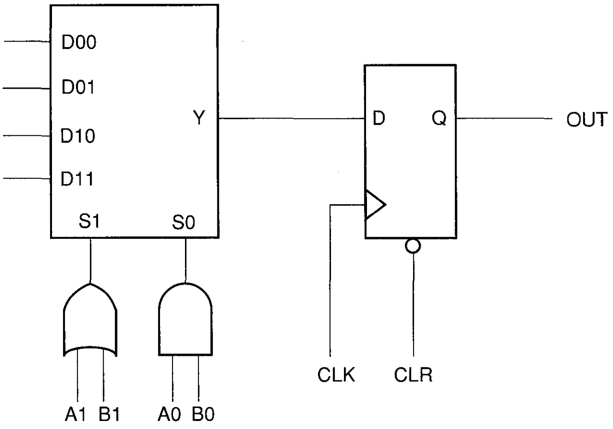


Figure 3 • S-Module Diagram

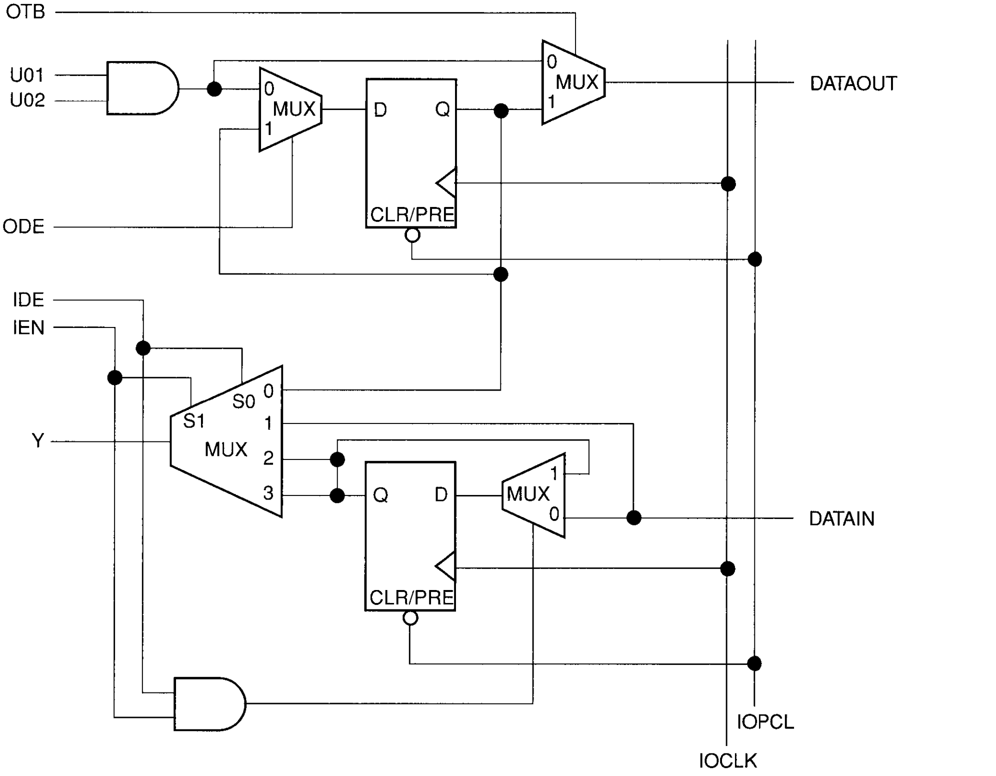


Figure 4 • Functional Diagram for I/O Module

I/O Pad Drivers

All pad drivers are capable of being tristate. Each buffer connects to an associated I/O module with four signals: OE (Output Enable), IE (Input Enable), DataOut, and DataIn. Certain special signals used only during programming and test also connect to the pad drivers: OUTEN (global output enable), INEN (global input enable), and SLEW (individual slew selection). See Figure 5.

Special I/Os

The special I/Os are of two types: temporary and permanent. Temporary special I/Os are used during programming and testing. They function as normal I/Os when the MODE pin is inactive. Permanent special I/Os are user programmed as either normal I/Os or special I/Os. Their function does not change once the device has been programmed. The permanent special I/Os consist of the array clock input buffers (CLKA and CLKB), the hard-wired array clock input buffer (HCLK), the hard-wired I/O clock input buffer (IOCLK), and the hard-wired I/O register preset/clear input buffer (IOPCL). Their function is determined by the I/O macros selected.

Clock Networks

The ACT 3 architecture contains four clock networks: two high-performance dedicated clock networks and two general purpose routed networks. The high-performance networks function up to 200 MHz, while the general purpose routed networks function up to 150 MHz.

Dedicated Clocks

Dedicated clock networks support high performance by providing sub-nanosecond skew and guaranteed performance. Dedicated clock networks contain no programming elements in the path from the I/O Pad Driver to the input of S-modules or I/O modules. There are two dedicated clock networks: one for the array registers (HCLK), and one for the I/O registers (IOCLK). The clock networks are accessed by special I/Os.

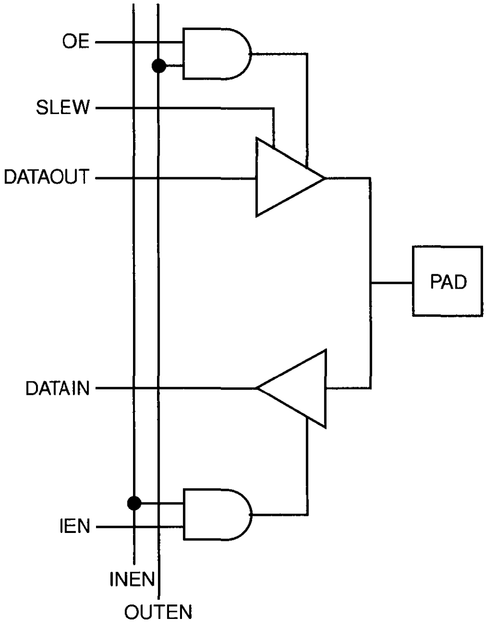


Figure 5 • Function Diagram for I/O Pad Driver

Routed Clocks

The routed clock networks are referred to as CLK0 and CLK1. Each network is connected to a clock module (CLKMOD) that selects the source of the clock signal and may be driven as follows (see Figure 6):

- externally from the CLKA pad
- externally from the CLKB pad
- internally from the CLKINA input
- internally from the CLKINB input

The clock modules are located in the top row of I/O modules. Clock drivers and a dedicated horizontal clock track are located in each horizontal routing channel. The function of the clock module is determined by the selection of clock macros from the macro library. The macro CLKBUF is used to connect one of the two external clock pins to a clock network, and the macro CLKINT is used to connect an internally generated clock signal to a clock network. Since both clock networks are identical, the user does not care whether CLK0 or CLK1 is being used. Routed clocks can also be used to drive high fanout nets like resets, output enables, or data enables. This saves logic modules and results in performance increases in some cases.

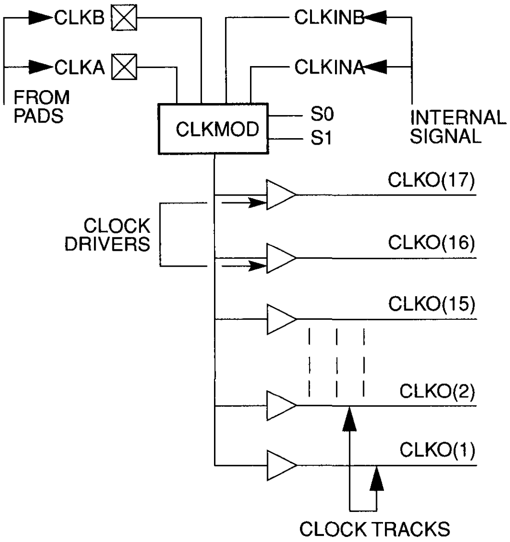


Figure 6 • Clock Networks

Routing Structure

The ACT 3 architecture uses vertical and horizontal routing tracks to connect the various logic and I/O modules. These routing tracks are metal interconnects that may either be of continuous length or broken into segments. Segments can be joined together at the ends using antifuses to increase their lengths up to the full length of the track.

Horizontal Routing

Horizontal channels are located between the rows of modules and are composed of several routing tracks. The horizontal routing tracks within the channel are divided into one or more segments. The minimum horizontal segment length is the width of a module-pair, and the maximum horizontal segment length is the full length of the channel. Any segment that spans more than one-third the row length is considered a long horizontal segment. A typical channel is shown in Figure 7. Undedicated horizontal routing tracks are used to route signal nets. Dedicated routing tracks are used for the global clock networks and for power and ground tie-off tracks.

Vertical Routing

Other tracks run vertically through the modules. Vertical tracks are of three types: input, output, and long. Vertical tracks are also divided into one or more segments. Each segment in an input track is dedicated to the input of a particular module. Each segment in an output track is dedicated to the output of a particular module. Long segments are uncommitted and can be assigned during routing. Each output segment spans four channels (two above and two below), except near the top and bottom of the array where edge effects occur. LVTs contain either one or two segments. An example of vertical routing tracks and segments is shown in Figure 8.

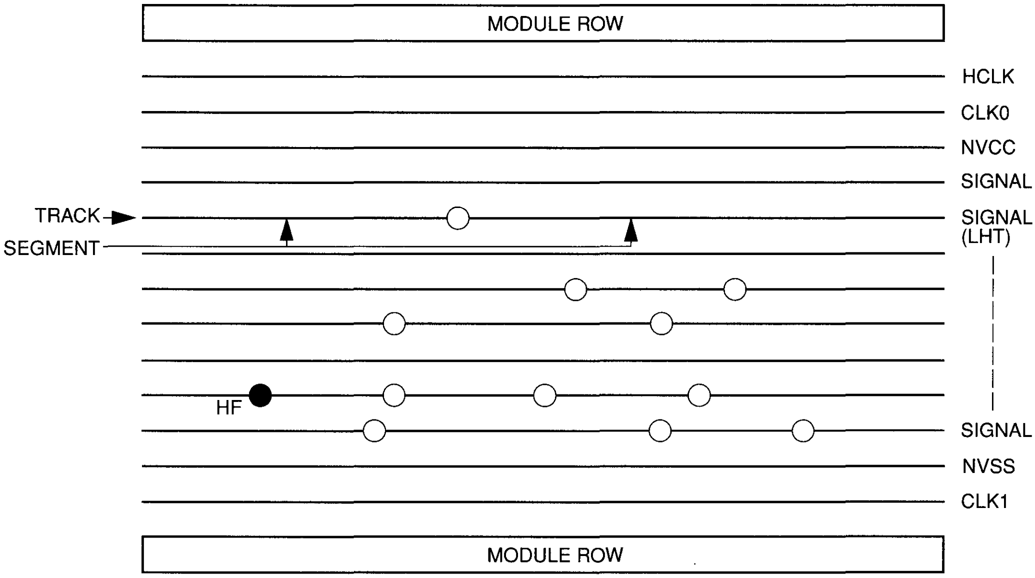


Figure 7 • *Horizontal Routing Tracks and Segments*

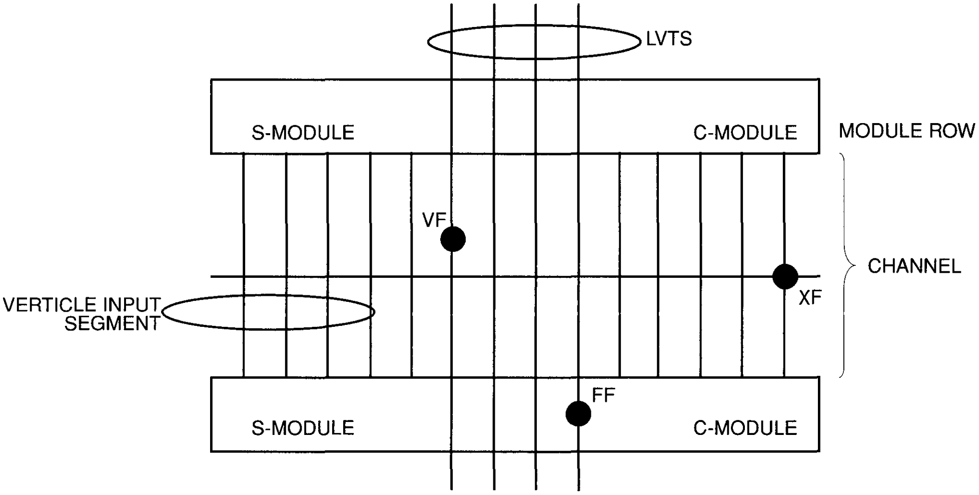


Figure 8 • *Vertical Routing Tracks and Segments*

Antifuse Connections

An antifuse is a "normally open" structure as opposed to the normally closed fuse structure used in PROMs or PALs. The use of antifuses to implement a programmable logic device results in highly testable structures as well as an efficient programming architecture. The structure is highly testable because there are no preexisting connections; temporary connections can be made using pass transistors. These temporary connections can isolate individual antifuses to be programmed as well as isolate individual circuit structures to be tested. This can be done both before and after programming. For example, all metal tracks can be tested for continuity and shorts between adjacent tracks, and the functionality of all logic modules can be verified.

Four types of antifuse connections are used in the routing structure of the ACT 3 array. (The physical structure of the antifuse is identical in each case; only the usage differs.) Table 1 shows four types of antifuses.

Table 1 • *Antifuse Types*

XF	Horizontal-to-Vertical Connection
HF	Horizontal-to-Horizontal Connection
VF	Vertical-to-Vertical Connection
FF	"Fast" Vertical Connection

Examples of all four types of connections are shown in Figures 7 and 8.

Module Interface

Connections to Logic and I/O modules are made through vertical segments that connect to the module inputs and outputs. These vertical segments lie on vertical tracks that span the entire height of the array.

Module Input Connections

The tracks dedicated to module inputs are segmented by pass transistors in each module row. During normal user operation, the pass transistors are inactive, which isolates the inputs of a module from the inputs of the module directly above or below it. During certain test modes, the pass transistors are active to verify the continuity of the metal tracks. Vertical input segments span only the channel above or the channel below. The logic modules are arranged such that half of the inputs are connected to the channel above and half of the inputs to segments in the channel below as shown in Figure 9.

Module Output Connections

Module outputs have dedicated output segments. Output segments extend vertically two channels above and two channels below, except at the top or bottom of the array. Output segments twist, as shown in Figure 10, so that only four vertical tracks are required.

LVT Connections

Outputs may also connect to nondedicated segments called Long Vertical Tracks (LVTs). Each module pair in the array shares four LVTs that span the length of the column. Any module in the column pair can connect to one of the LVTs in the column using an FF connection. The FF connection uses antifuses connected directly to the driver stage of the module output, bypassing the isolation transistor. FF antifuses are programmed at a higher current level than HF, VF, or XF antifuses to produce a lower resistance value.

Antifuse Connections

In general every intersection of a vertical segment and a horizontal segment contains an unprogrammed antifuse (XF-type). One exception is in the case of the clock networks.

Clock Connections

To minimize loading on the clock networks, a subset of inputs has antifuses on the clock tracks. Only a few of the C-module and S-module inputs can be connected to the clock networks. To further reduce loading on the clock network, only a subset of the horizontal routing tracks can connect to the clock inputs of the S-module.

Programming and Test Circuits

The array of logic and I/O modules is surrounded by test and programming circuits controlled by the temporary special I/O pins MODE, SDI, and DCLK. The function of these pins is similar to all ACT family devices. The ACT 3 family also includes support for two Actionprobe® circuits allowing complete observability of any logic or I/O module in the array using the temporary special I/O pins, PRA and PRB.

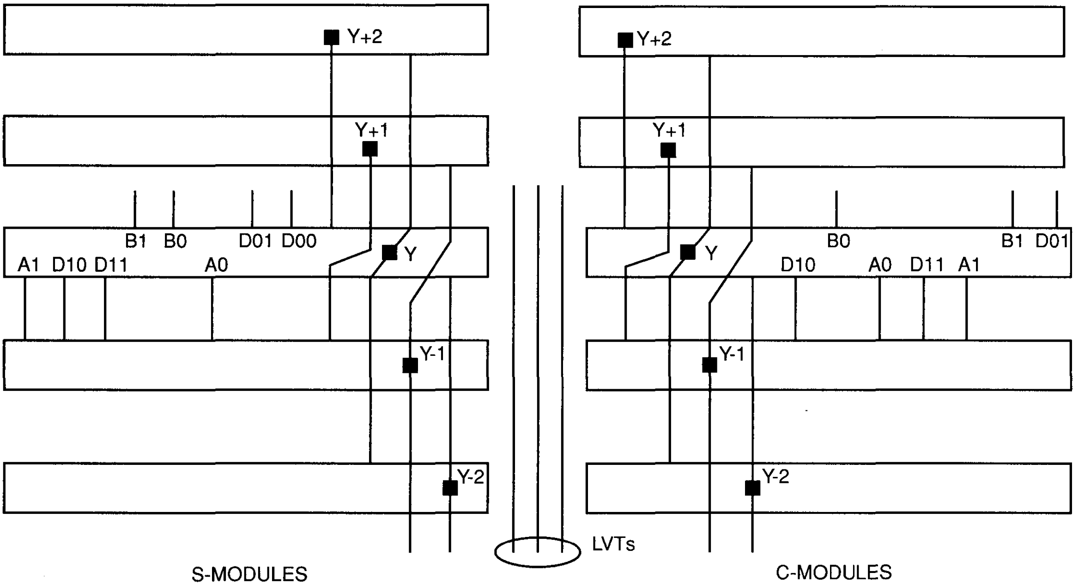


Figure 9* • *Logic Module Routing Interface

5V Operating Conditions

Absolute Maximum Ratings[1]

Free air temperature range

Symbol	Parameter	Limits	Units
V_{CC}	DC Supply Voltage[2]	–0.5 to +7.0	V
V_I	Input Voltage	–0.5 to V_{CC} +0.5	V
V_O	Output Voltage	–0.5 to V_{CC} +0.5	V
I_{IO}	I/O Source Sink Current[3]	±20	mA
T_{STG}	Storage Temperature	–65 to +150	°C

Notes:

1. *Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum rated conditions for extended periods may affect device reliability. Device should not be operated outside the Recommended Operating Conditions.*
2. *V_{PP}, V_{SV} = V_{CC}, except during device programming.*
3. *Device inputs are normally high impedance and draw extremely low current. However, when input voltage is greater than V_{CC} + 0.5 V or less than GND – 0.5 V, the internal protection diodes will forward bias and can draw excessive current.*

Recommended Operating Conditions

Parameter	Commercial	Industrial	Military	Units
Temperature Range[1]	0 to +70	–40 to +85	–55 to +125	°C
5V Power Supply Tolerance	±5	±10	±10	%V_{CC}

Note:

1. *Ambient temperature (T_A) is used for commercial and industrial; case temperature (T_C) is used for military.*

Electrical Specifications

			Commercial		Industrial		Military		
Symbol	**Parameter**	**Test Condition**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
V_{OH}[1,2]	HIGH Level Output	I_{OH} = –4 mA (CMOS)			3.7		3.7		V
		I_{OH} = –6 mA (CMOS)	3.84						V
		I_{OH} = –10 mA (TTL)[3]	2.40						V
V_{OL}[1,2]	LOW Level Output	I_{OL} = +6 mA (CMOS)		0.33		0.4		0.4	V
		I_{OL} = +12 mA (TTL)[3]		0.50					V
V_{IH}	HIGH Level Input	TTL Inputs	2.0	V_{CC} + 0.3	2.0	V_{CC} + 0.3	2.0	V_{CC} + 0.3	V
V_{IL}	LOW Level Input	TTL Inputs	–0.3	0.8	–0.3	0.8	–0.3	0.8	V
I_{IN}	Input Leakage	V_I = V_{CC} or GND	–10	+10	–10	+10	–10	+10	µA
I_{OZ}	3-state Output Leakage	V_O = V_{CC} or GND	–10	+10	–10	+10	–10	+10	µA
C_{IO}	I/O Capacitance[3,4]			10		10		10	pF
$I_{CC(S)}$	Standby V_{CC} Supply Current	(typical = 0.7 mA)		2		10		20	mA
$I_{CC(D)}$	Dynamic V_{CC} Supply Current	See "Power Dissipation" Section							

Notes:

1. *Actel devices can drive and receive either CMOS or TTL signal levels. No assignment of I/Os as TTL or CMOS is required.*
2. *Tested one output at a time, V_{CC} = min.*
3. *Not tested, for information only.*
4. *V_{OUT} = 0V, f = 1 MHz.*
5. *Typical standby current = 0.7 mA. All outputs unloaded. All inputs = V_{CC} or GND.*

3.3V Operating Conditions

Absolute Maximum Ratings[1]

Free air temperature range

Symbol	Parameter	Limits	Units
V_{CC}	DC Supply Voltage	–0.5 to +7.0	V
V_I	Input Voltage	–0.5 to V_{CC} +0.5	V
V_O	Output Voltage	–0.5 to V_{CC} +0.5	V
I_{IO}	I/O Source Sink Current[2]	±20	mA
T_{STG}	Storage Temperature	–65 to +150	°C

Notes:

1. *Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum rated conditions for extended periods may affect device reliability. Device should not be operated outside the Recommended Operating Conditions.*
2. *Device inputs are normally high impedance and draw extremely low current. However, when input voltage is greater than V_{CC} + 0.5 V or less than GND – 0.5 V, the internal protection diodes will forward bias and can draw excessive current.*

Recommended Operating Conditions

Parameter	Commercial	Units
Temperature Range[1]	0 to +70	°C
Power Supply Tolerance	3.0 to 3.6	V

Note:

1. *Ambient temperature (T_A) is used for commercial.*

Electrical Specifications

Parameter		Commercial Min.	Commercial Max.	Units
V_{OH}[1]	(I_{OH} = –4 mA)	2.15		V
	(I_{OH} = –3.2 mA)	2.4		V
V_{OL}[1]	(I_{OL} = 6 mA)		0.4	V
V_{IL}		–0.3	0.8	V
V_{IH}		2.0	V_{CC} + 0.3	V
Input Transition Time t_R, t_F[2]			500	ns
C_{IO} I/O Capacitance[2, 3]			10	pF
Standby Current, I_{CC}[4] (typical = 0.3 mA)			0.75	mA
Leakage Current[5]		–10	10	µA

Notes:

1. *Only one output tested at a time. V_{CC} = min.*
2. *Not tested, for information only.*
3. *Includes worst-case 84-pin PLCC package capacitance. V_{OUT} = 0 V, f = 1 MHz.*
4. *Typical standby current = 0.3 mA. All outputs unloaded. All inputs = V_{CC} or GND.*
5. *V_O, V_{IN} = V_{CC} or GND.*

Package Thermal Characteristics

The device junction to case thermal characteristic is θjc, and the junction to ambient air characteristic is θja. The thermal characteristics for θja are shown with two different air flow rates.

Maximum junction temperature is 150°C.

A sample calculation of the absolute maximum power dissipation allowed for a CPGA 175-pin package at commercial temperature and still air is as follows:

$$\text{Absolute Maximum Power Allowed} = \frac{\text{Max. junction temp. (°C)} - \text{Max. ambient temp. (°C)}}{\theta\text{ja (°C/W)}} = \frac{150°\text{C} - 70°\text{C}}{25°\text{C/W}} = 3.2\text{ W}$$

Package Type[1]	Pin Count	θjc	θja Still Air	θja 300 ft/min	Units
Ceramic Pin Grid Array	100	20	35	17	°C/W
	133	20	30	15	°C/W
	175	20	25	14	°C/W
	207	20	22	13	°C/W
	257	20	15	8	°C/W
Ceramic Quad Flatpack	132	13	55	30	°C/W
	196	13	36	24	°C/W
	256	13	30	18	°C/W
Plastic Quad Flatpack	100	13	51	40	°C/W
	160	10	33	26	°C/W
	208	10	33	26	°C/W
Very Thin Quad Flatpack	100	12	43	35	°C/W
Thin Quad Flatpack	176	11	32	25	°C/W
Power Quad Flatpack	208	0.4	17	13	°C/W
Plastic Leaded Chip Carrier	84	12	37	28	°C/W
Plastic Ball Grid Array	225	10	25	19	°C/W
	313	10	23	17	°C/W

Note:

1. *Maximum Power Dissipation in Still Air for 160-pin PQFP package is 2.4 Watts, 208-pin PQFP package is 2.4 Watts, 100-pin PQFP package is 1.6 Watts, 100-pin VQFP package is 1.9 Watts, 176-pin TQFP package is 2.5 Watts, 84-pin PLCC package is 2.2 Watts, 208-pin RQFP package is 4.7 Watts, 225-pin BGA package is 3.2 Watts, 313-pin BGA package is 3.5 Watts.*

Power Dissipation

$$P = [I_{CC\ \text{standby}} + I_{\text{active}}] * V_{CC} + I_{OL} * V_{OL} * N + I_{OH} * (V_{CC} - V_{OH}) * M \qquad (1)$$

Where:

$I_{CC\ \text{standby}}$ is the current flowing when no inputs or outputs are changing.

I_{active} is the current flowing due to CMOS switching.

I_{OL}, I_{OH} are TTL sink/source currents.

V_{OL}, V_{OH} are TTL level output voltages.

N equals the number of outputs driving TTL loads to V_{OL}.

M equals the number of outputs driving TTL loads to V_{OH}.

An accurate determination of N and M is problematical because their values depend on the design and on the system I/O. The power can be divided into two components: static and active.

Static Power Component

Actel FPGAs have small static power components that result in lower power dissipation than PALs or PLDs. By integrating multiple PALs/PLDs into one FPGA, an even greater reduction in board-level power dissipation can be achieved.

The power due to standby current is typically a small component of the overall power. Standby power is calculated below for commercial, worst case conditions.

I_{CC}	V_{CC}	Power
2mA	5.25 V	10.5 mW

The static power dissipated by TTL loads depends on the number of outputs driving high or low and the DC load current. Again, this value is typically small. For instance, a 32-bit bus sinking 4 mA at 0.33 V will generate 42 mW with all outputs driving low, and 140 mW with all outputs driving high. The actual dissipation will average somewhere between as I/Os switch states with time.

Active Power Component

Power dissipation in CMOS devices is usually dominated by the active (dynamic) power dissipation. This component is frequency dependent, a function of the logic and the external I/O. Active power dissipation results from charging internal chip capacitances of the interconnect, unprogrammed antifuses, module inputs, and module outputs, plus external capacitance due to PC board traces and load device inputs. An additional component of the active power dissipation is the totem-pole current in CMOS transistor pairs. The net effect can be associated with an equivalent capacitance that can be combined with frequency and voltage to represent active power dissipation.

Equivalent Capacitance

The power dissipated by a CMOS circuit can be expressed by the Equation 2.

$$\text{Power (uW)} = C_{EQ} * V_{CC}^2 * F \qquad (2)$$

Where:

C_{EQ} is the equivalent capacitance expressed in pF.

V_{CC} is the power supply in volts.

F is the switching frequency in MHz.

Equivalent capacitance is calculated by measuring I_{CC}active at a specified frequency and voltage for each circuit component of interest. Measurements have been made over a range of frequencies at a fixed value of V_{CC}. Equivalent capacitance is frequency independent so that the results may be used over a wide range of operating conditions. Equivalent capacitance values are shown below.

C_{EQ} Values for Actel FPGAs

Modules (C_{EQM})	6.7
Input Buffers ($_{CEQI}$)	7.2
Output Buffers (C_{EQO})	10.4
Routed Array Clock Buffer Loads (C_{EQCR})	1.6
Dedicated Clock Buffer Loads (C_{EQCD})	0.7
I/O Clock Buffer Loads (C_{EQCI})	0.9

To calculate the active power dissipated from the complete design, the switching frequency of each part of the logic must be known. Equation 3 shows a piece-wise linear summation over all components.

$$\begin{aligned}
&\text{Power} = V_{CC}2 * [(m * C_{EQM} * f_m)_{modules} + (n * C_{EQI} * f_n)_{inputs} \\
&+ (p * (C_{EQO} + C_L) * f_p)_{outputs} \\
&+ 0.5 * (q_1 * C_{EQCR} * f_{q1})_{routed\_Clk1} + (r_1 * f_{q1})_{routed\_Clk1} \\
&+ 0.5 * (q_2 * C_{EQCR} * f_{q2})_{routed\_Clk2} \\
&+ (r_2 * f_{q2})_{routed\_Clk2} + 0.5 * (s_1 * {}_{CEQCD} * f_{s1})_{dedicated\_Clk} \\
&+ (s_2 * CEQCI * f_{s2})_{IO\_Clk}] \qquad (3)
\end{aligned}$$

Where:

m = Number of logic modules switching at f_m

n = Number of input buffers switching at f_n

p = Number of output buffers switching at f_p

q_1 = Number of clock loads on the first routed array clock

q_2 = Number of clock loads on the second routed array clock

r_1 = Fixed capacitance due to first routed array clock

r_2 = Fixed capacitance due to second routed array clock

s_1 = Fixed number of clock loads on the dedicated array clock

s_2 = Fixed number of clock loads on the dedicated I/O clock

C_{EQM} = Equivalent capacitance of logic modules in pF

C_{EQI} = Equivalent capacitance of input buffers in pF

C_{EQO} = Equivalent capacitance of output buffers in pF

C_{EQCR} = Equivalent capacitance of routed array clock in pF

C_{EQCD} = Equivalent capacitance of dedicated array clock in pF

C_{EQCI} = Equivalent capacitance of dedicated I/O clock in pF

C_L = Output lead capacitance in pF

f_m = Average logic module switching rate in MHz

f_n = Average input buffer switching rate in MHz

fp = Average output buffer switching rate in MHz

f_{q1} = Average first routed array clock rate in MHz

f_{q2} = Average second routed array clock rate in MHz

f_{s1} = Average dedicated array clock rate in MHz

f_{s2} = Average dedicated I/O clock rate in MHz

Fixed Capacitance Values for Actel FPGAs (pF)

Device Type	r_1 routed_Clk1	r_2 routed_Clk2
A1415A	60	60
A14V15A	57	57
A1425A	75	75
A14V25A	72	72
A1440A	105	105
A14V40A	100	100
A1440B	105	105
A1460A	165	165
A14V60A	157	157
A1460B	165	165
A14100A	195	195
A14V100A	185	185
A14100B	195	195

Fixed Clock Loads (s_1/s_2)

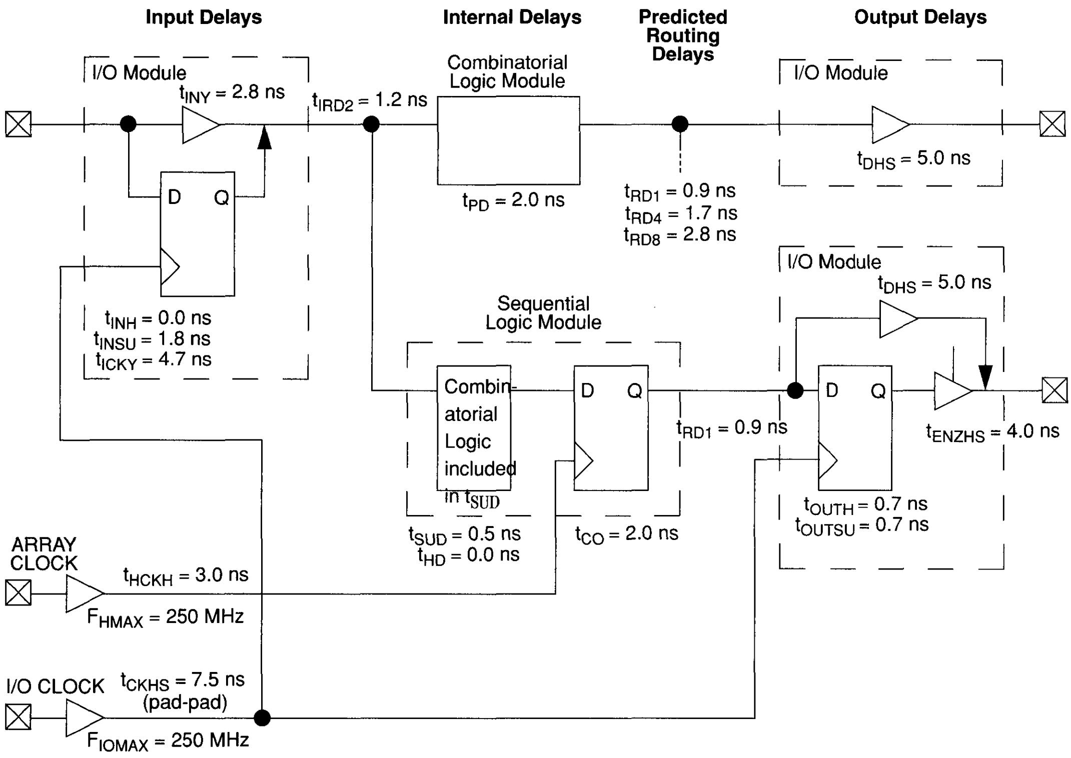
Device Type	s_1 Clock Loads on dedicated array clock	s_2 Clock Loads on dedicated I/O clock
A1415A	104	80
A14V15A	104	80
A1425A	160	100
A14V25A	160	100
A1440A	288	140
A14V40A	288	140
A1440B	288	140
A1460A	432	168
A14V60A	432	168
A1460B	432	168
A14100A	697	228
A14V100A	697	228
A14100B	697	228

Determining Average Switching Frequency

To determine the switching frequency for a design, you must have a detailed understanding of the data input values to the circuit. The following guidelines are meant to represent worst-case scenarios so that they can be generally used to predict the upper limits of power dissipation. These guidelines are as follows:

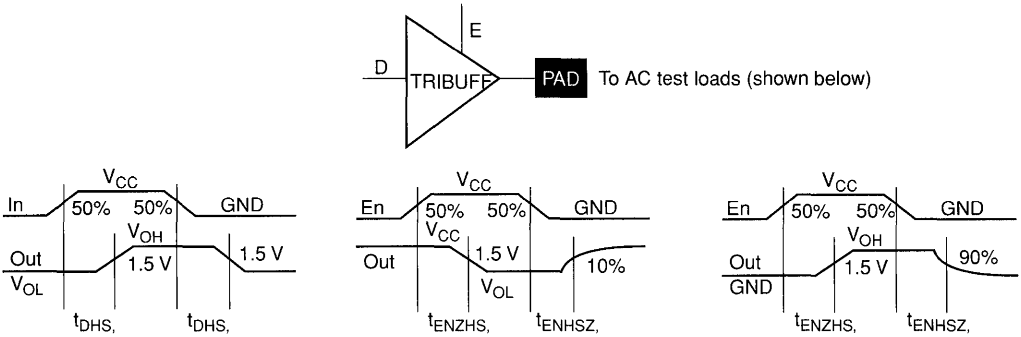
Logic Modules (m)	=	80% of modules
Inputs switching (n)	=	# inputs/4
Outputs switching (p)	=	# output/4
First routed array clock loads (q_1)	=	40% of sequential modules
Second routed array clock loads (q_2)	=	40% of sequential modules
Load capacitance (C_L)	=	35 pF
Average logic module switching rate (f_m)	=	F/10
Average input switching rate (f_n)	=	F/5
Average output switching rate (f_p)	=	F/10
Average first routed array clock rate (f_{q1})	=	F/2
Average second routed array clock rate (f_{q2})	=	F/2
Average dedicated array clock rate (f_{s1})	=	F
Average dedicated I/O clock rate (f_{s2})	=	F

ACT 3 Timing Model*

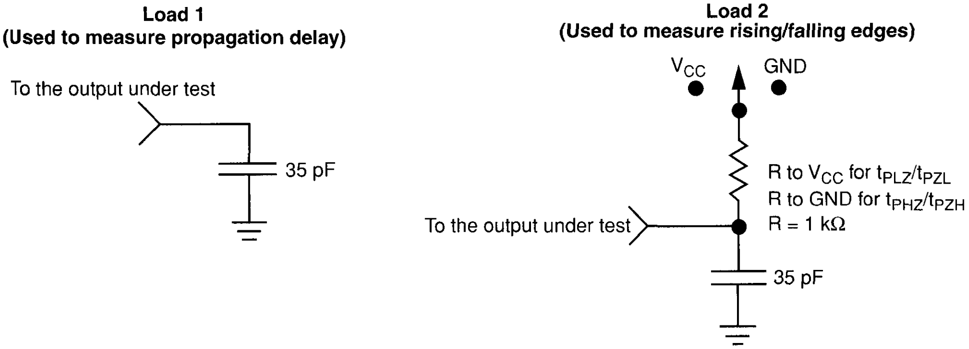


**Values shown for A1425A-3.*

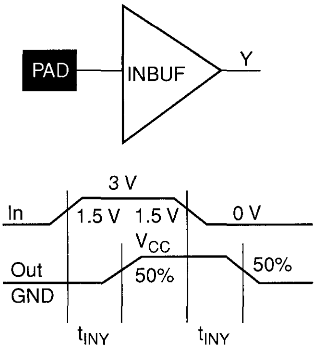
Output Buffer Delays



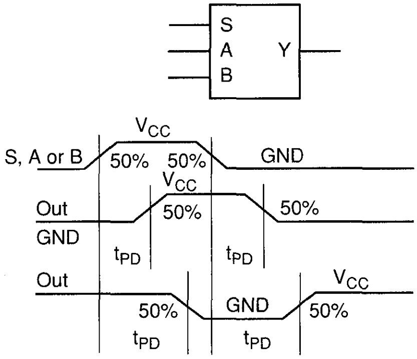
AC Test Loads



Input Buffer Delays

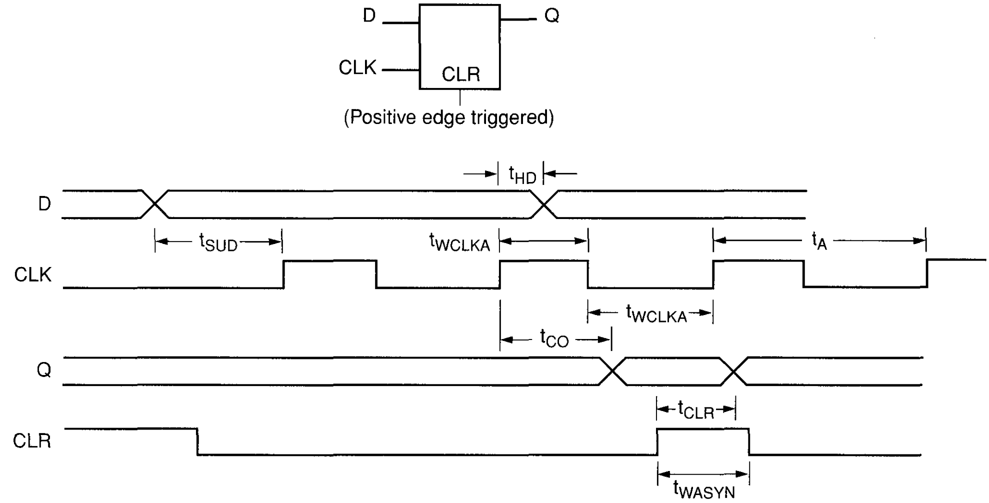


Module Delays

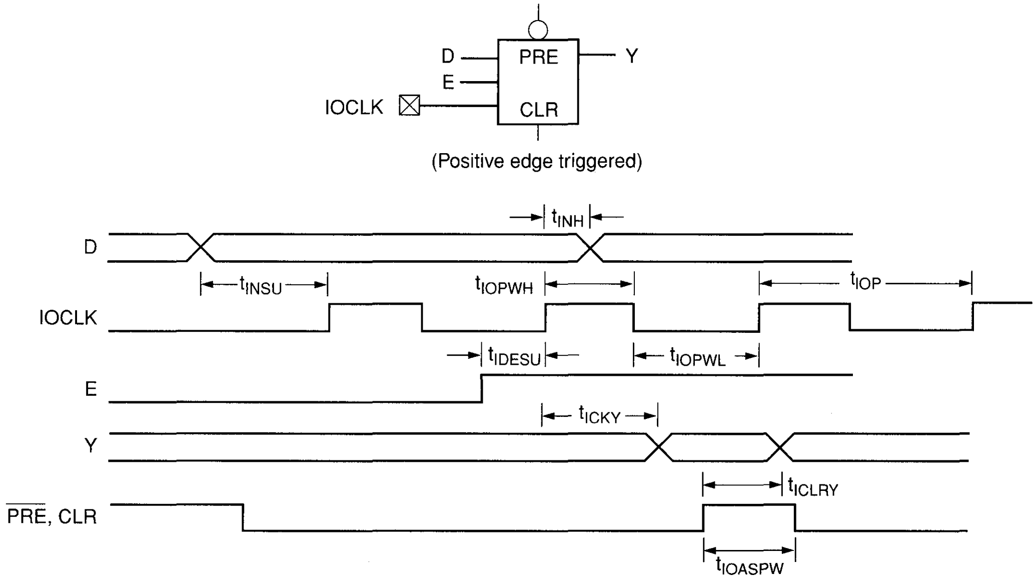


Sequential Module Timing Characteristics

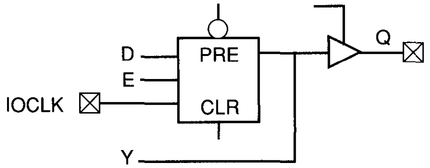
Flip-Flops



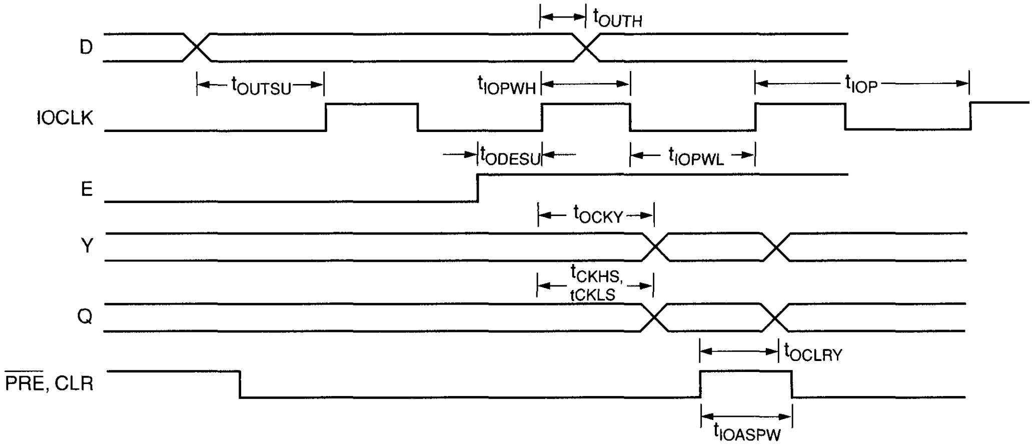
I/O Module: Sequential Input Timing Characteristics



I/O Module: Sequential Output Timing Characteristics



(Positive edge triggered)



Predictable Performance: Tightest Delay Distributions

Propagation delay between logic modules depends on the resistive and capacitive loading of the routing tracks, the interconnect elements, and the module inputs being driven. Propagation delay increases as the length of routing tracks, the number of interconnect elements, or the number of inputs increases.

From a design perspective, the propagation delay can be statistically correlated or modeled by the fanout (number of loads) driven by a module. Higher fanout usually requires some paths to have longer lengths of routing track.

The ACT 3 family delivers the tightest fanout delay distribution of any FPGA. This tight distribution is achieved in two ways: by decreasing the delay of the interconnect elements and by decreasing the number of interconnect elements per path.

Actel's patented PLICE antifuse offers a very low resistive/capacitive interconnect. The ACT 3 family's antifuses, fabricated in 0.8 micron m lithography, offer nominal levels of 200Ω resistance and 6 femtofarad (fF) capacitance per antifuse.

The ACT 3 fanout distribution is also tighter than alternative devices due to the low number of antifuses required per interconnect path. The ACT 3 family's proprietary architecture limits the number of antifuses per path to only four, with 90% of interconnects using only two antifuses.

The ACT 3 family's tight fanout delay distribution offers an FPGA design environment in which fanout can be traded for the increased performance of reduced logic level designs. This also simplifies performance estimates when designing with ACT 3 devices.

Table 2 • *Logic Module and Routing Delay by Fanout (ns) (Worst-Case Commercial Conditions)*

Speed	FO=1	FO=2	FO=3	FO=4	FO=8
ACT 3 –3	2.9	3.2	3.4	3.7	4.8

Timing Characteristics

Timing characteristics for ACT 3 devices fall into three categories: family dependent, device dependent, and design dependent. The input and output buffer characteristics are common to all ACT 3 family members. Internal routing delays. are device dependent. Design dependency means actual delays are not determined until after placement and routing of the user's design is complete. Delay values may then be determined by using the ALS Timer utility or performing simulation with post-layout delays.

Critical Nets and Typical Nets

Propagation delays are expressed only for typical nets, which are used for initial design performance evaluation. Critical net delays can then be applied to the most time-critical paths. Critical nets are determined by net property assignment prior to placement and routing. Up to 6% of the nets in a design may be designated as critical, while 90% of the nets in a design are typical.

Long Tracks

Some nets in the design use long tracks. Long tracks are special routing resources that span multiple rows, columns, or modules. Long tracks employ three and sometimes four antifuse connections. This increases capacitance and resistance, resulting in longer net delays for macros connected to long tracks. Typically up to 6% of nets in a fully utilized device require long tracks. Long tracks contribute approximatley 4 ns to 14 ns delay. This additional delay is represented statistically in higher fanout (FO=8) routing delays in the data sheet specifications section.

Timing Derating

ACT 3 devices are manufactured in a CMOS process. Therefore, device performance varies according to temperature, voltage, and process variations. Minimum timing parameters reflect maximum operating voltage, minimum operating temperature, and best-case processing. Maximum timing parameters reflect minimum operating voltage, maximum operating temperature, and worst-case processing.

Timing Derating Factor (Temperature and Voltage)

	Industrial		Military	
	Min.	Max.	Min.	Max.
(Commercial Minimum/Maximum Specification) x	0.66	1.07	0.63	1.17

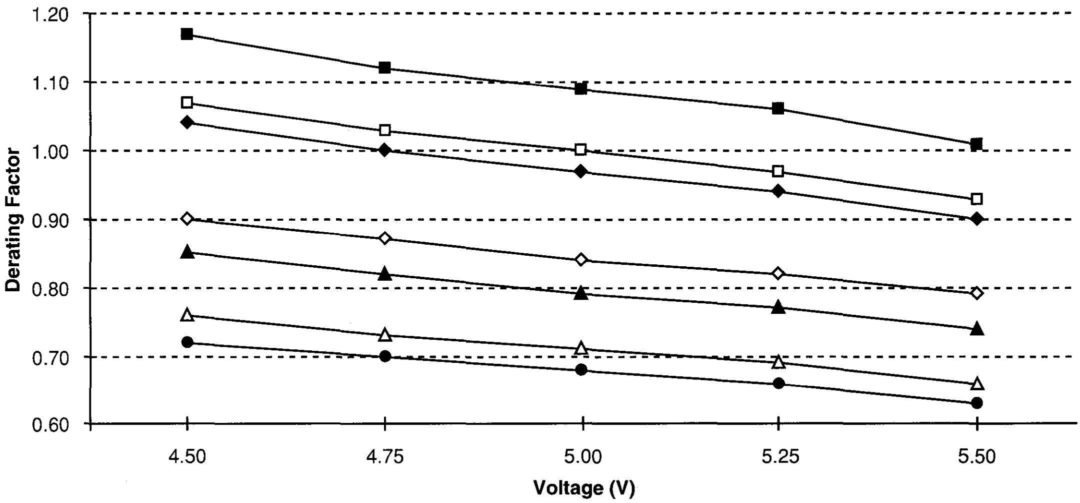
Timing Derating Factor for Designs at Typical Temperature (T_J = 25°C) and Voltage (5.0 V)

(Commercial Maximum Specification) x	0.85

Temperature and Voltage Derating Factors (normalized to Worst-Case Commercial, T_J = 4.75 V, 70°C)

	–55	–40	0	25	70	85	125
4.50	0.72	0.76	0.85	0.90	1.04	1.07	1.17
4.75	0.70	0.73	0.82	0.87	1.00	1.03	1.12
5.00	0.68	0.71	0.79	0.84	0.97	1.00	1.09
5.25	0.66	0.69	0.77	0.82	0.94	0.97	1.06
5.50	0.63	0.66	0.74	0.79	0.90	0.93	1.01

Junction Temperature and Voltage Derating Curves (normalized to Worst-Case Commercial, T_J = 4.75 V, 70°C)



Note: This derating factor applies to all routing and propagation dealys.

A1415A, A14V15A Timing Characteristics

(Worst-Case Commercial Conditions, V_{CC} = 4.75 V, T_J = 70°C)[1]

Logic Module Propagation Delays[2]		'–3' Speed		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed[1]		
Parameter	Description	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Units
t_{PD}	Internal Array Module		2.0		2.3		2.6		3.0		3.9	ns
t_{CO}	Sequential Clock to Q		2.0		2.3		2.6		3.0		3.9	ns
t_{CLR}	Asynchronous Clear to Q		2.0		2.3		2.6		3.0		3.9	ns
Predicted Routing Delays[3]												
t_{RD1}	FO=1 Routing Delay		0.9		1.0		1.1		1.3		1.7	ns
t_{RD2}	FO=2 Routing Delay		1.2		1.4		1.6		1.8		2.4	ns
t_{RD3}	FO=3 Routing Delay		1.4		1.6		1.8		2.1		2.8	ns
t_{RD4}	FO=4 Routing Delay		1.7		1.9		2.2		2.5		3.3	ns
t_{RD8}	FO=8 Routing Delay		2.8		3.2		3.6		4.2		5.5	ns
Logic Module Sequential Timing												
t_{SUD}	Flip-Flop Data Input Setup	0.5		0.6		0.7		0.8		0.8		ns
t_{HD}	Flip-Flop Data Input Hold	0.0		0.0		0.0		0.0		0.0		ns
t_{SUD}	Latch Data Input Setup	0.5		0.6		0.7		0.8		0.8		ns
t_{HD}	Latch Data Input Hold	0.0		0.0		0.0		0.0		0.0		ns
t_{WASYN}	Asynchronous Pulse Width	1.9		2.4		3.2		3.8		4.8		ns
t_{WCLKA}	Flip-Flop Clock Pulse Width	1.9		2.4		3.2		3.8		4.8		ns
t_A	Flip-Flop Clock Input Period	4.0		5.0		6.8		8.0		10.0		ns
f_{MAX}	Flip-Flop Clock Frequency		250		200		150		125		100	MHz

Notes:

1. *V_{CC} = 3.0 V for 3.3V specifications.*
2. *For dual-module macros, use $t_{PD} + t_{RD1} + t_{PDn}$, $t_{CO} + t_{RD1} + t_{PDn}$ or $t_{PD1} + t_{RD1} + t_{SUD}$, whichever is appropriate.*
3. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

A1415A, A14V15A Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

I/O Module Input Propagation Delays		'–3' Speed		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{INY}	Input Data Pad to Y		2.8		3.2		3.6		4.2		5.5	ns
t_{ICKY}	Input Reg IOCLK Pad to Y		4.7		5.3		6.0		7.0		9.2	ns
t_{OCKY}	Output Reg IOCLK Pad to Y		4.7		5.3		6.0		7.0		9.2	ns
t_{ICLRY}	Input Asynchronous Clear to Y		4.7		5.3		6.0		7.0		9.2	ns
t_{OCLRY}	Output Asynchronous Clear to Y		4.7		5.3		6.0		7.0		9.2	ns
Predicted Input Routing Delays[1]												
t_{IRD1}	FO=1 Routing Delay		0.9		1.0		1.1		1.3		1.7	ns
t_{IRD2}	FO=2 Routing Delay		1.2		1.4		1.6		1.8		2.4	ns
t_{IRD3}	FO=3 Routing Delay		1.4		1.6		1.8		2.1		2.8	ns
t_{IRD4}	FO=4 Routing Delay		1.7		1.9		2.2		2.5		3.3	ns
t_{IRD8}	FO=8 Routing Delay		2.8		3.2		3.6		4.2		5.5	ns
I/O Module Sequential Timing												
t_{INH}	Input F-F Data Hold (w.r.t. IOCLK Pad)	0.0		0.0		0.0		0.0		0.0		ns
t_{INSU}	Input F-F Data Setup (w.r.t. IOCLK Pad)	2.0		2.3		2.5		3.0		3.0		ns
t_{IDEH}	Input Data Enable Hold (w.r.t. IOCLK Pad)	0.0		0.0		0.0		0.0		0.0		ns
t_{IDESU}	Input Data Enable Setup (w.r.t. IOCLK Pad)	5.8		6.5		7.5		8.6		8.6		ns
t_{OUTH}	Output F-F Data Hold (w.r.t. IOCLK Pad)	0.7		0.8		0.9		1.0		1.0		ns
t_{OUTSU}	Output F-F Data Setup (w.r.t. IOCLK Pad)	0.7		0.8		0.9		1.0		1.0		ns
t_{ODEH}	Output Data Enable Hold (w.r.t. IOCLK Pad)	0.3		0.4		0.4		0.5		0.5		ns
t_{ODESU}	Output Data Enable Setup (w.r.t. IOCLK Pad)	1.3		1.5		1.7		2.0		2.0		ns

Note:

1. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

A1415A, A14V15A Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

I/O Module – TTL Output Timing[1]		'–3' Speed		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed		
Parameter	Description	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Units
t_{DHS}	Data to Pad, High Slew		5.0		5.6		6.4		7.5		9.8	ns
t_{DLS}	Data to Pad, Low Slew		8.0		9.0		10.2		12.0		15.6	ns
t_{ENZHS}	Enable to Pad, Z to H/L, Hi Slew		4.0		4.5		5.1		6.0		7.8	ns
t_{ENZLS}	Enable to Pad, Z to H/L, Lo Slew		7.4		8.3		9.4		11.0		14.3	ns
t_{ENHSZ}	Enable to Pad, H/L to Z, Hi Slew		6.5		7.5		8.5		10.0		13.0	ns
t_{ENLSZ}	Enable to Pad, H/L to Z, Lo Slew		6.5		7.5		8.5		10.0		13.0	ns
t_{CKHS}	IOCLK Pad to Pad H/L, Hi Slew		7.5		7.5		9.0		10.0		13.0	ns
t_{CKLS}	IOCLK Pad to Pad H/L, Lo Slew		11.3		11.3		13.5		15.0		19.5	ns
d_{TLHHS}	Delta Low to High, Hi Slew		0.02		0.02		0.03		0.03		0.04	ns/pF
d_{TLHLS}	Delta Low to High, Lo Slew		0.05		0.05		0.06		0.07		0.09	ns/pF
d_{THLHS}	Delta High to Low, Hi Slew		0.04		0.04		0.04		0.05		0.07	ns/pF
d_{THLLS}	Delta High to Low, Lo Slew		0.05		0.05		0.06		0.07		0.09	ns/pF
I/O Module – CMOS Output Timing[1]												
t_{DHS}	Data to Pad, High Slew		6.2		7.0		7.9		9.3		12.1	ns
t_{DLS}	Data to Pad, Low Slew		11.7		13.1		14.9		17.5		22.8	ns
t_{ENZHS}	Enable to Pad, Z to H/L, Hi Slew		5.2		5.9		6.6		7.8		10.1	ns
t_{ENZLS}	Enable to Pad, Z to H/L, Lo Slew		8.9		10.0		11.3		13.3		17.3	ns
t_{ENHSZ}	Enable to Pad, H/L to Z, Hi Slew		6.7		7.5		8.5		10.0		13.0	ns
t_{ENLSZ}	Enable to Pad, H/L to Z, Lo Slew		6.7		7.5		9.0		10.0		13.0	ns
t_{CKHS}	IOCLK Pad to Pad H/L, Hi Slew		8.9		8.9		10.7		11.8		15.3	ns
t_{CKLS}	IOCLK Pad to Pad H/L, Lo Slew		13.0		13.0		15.6		17.3		22.5	ns
d_{TLHHS}	Delta Low to High, Hi Slew		0.04		0.04		0.05		0.06		0.08	ns/pF
d_{TLHLS}	Delta Low to High, Lo Slew		0.07		0.08		0.09		0.11		0.14	ns/pF
d_{THLHS}	Delta High to Low, Hi Slew		0.03		0.03		0.03		0.04		0.05	ns/pF
d_{THLLS}	Delta High to Low, Lo Slew		0.04		0.04		0.04		0.05		0.07	ns/pF

Note:

[1] *Delays based on 35pF loading.*

A1415A, A14V15A Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

Dedicated (Hard-Wired) I/O Clock Network		'–3' Speed		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed		
Parameter	Description	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Units
t_{IOCKH}	Input Low to High (Pad to I/O Module Input)		2.0		2.3		2.6		3.0		3.5	ns
t_{IOPWH}	Minimum Pulse Width High	1.9		2.4		3.3		3.8		4.8		ns
t_{IOPWL}	Minimum Pulse Width Low	1.9		2.4		3.3		3.8		4.8		ns
t_{IOSAPW}	Minimum Asynchronous Pulse Width	1.9		2.4		3.3		3.8		4.8		ns
t_{IOCKSW}	Maximum Skew		0.4		0.4		0.4		0.4		0.4	ns
t_{IOP}	Minimum Period	4.0		5.0		6.8		8.0		10.0		ns
f_{IOMAX}	Maximum Frequency		250		200		150		125		100	MHz
Dedicated (Hard-Wired) Array Clock Network												
t_{HCKH}	Input Low to High (Pad to S-Module Input)		3.0		3.4		3.9		4.5		5.5	ns
t_{HCKL}	Input High to Low (Pad to S-Module Input)		3.0		3.4		3.9		4.5		5.5	ns
t_{HPWH}	Minimum Pulse Width High	1.9		2.4		3.3		3.8		4.8		ns
t_{HPWL}	Minimum Pulse Width Low	1.9		2.4		3.3		3.8		4.8		ns
t_{HCKSW}	Maximum Skew		0.3		0.3		0.3		0.3		0.3	ns
t_{HP}	Minimum Period	4.0		5.0		6.8		8.0		10.0		ns
f_{HMAX}	Maximum Frequency		250		200		150		125		100	MHz
Routed Array Clock Networks												
t_{RCKH}	Input Low to High (FO=64)		3.7		4.1		4.7		5.5		9.0	ns
t_{RCKL}	Input High to Low (FO=64)		4.0		4.5		5.1		6.0		9.0	ns
t_{RPWH}	Min. Pulse Width High (FO=64)	3.3		3.8		4.2		4.9		6.5		ns
t_{RPWL}	Min. Pulse Width Low (FO=64)	3.3		3.8		4.2		4.9		6.5		ns
t_{RCKSW}	Maximum Skew (FO=128)		0.7		0.8		0.9		1.0		1.0	ns
t_{RP}	Minimum Period (FO=64)	6.8		8.0		8.7		10.0		13.4		ns
f_{RMAX}	Maximum Frequency (FO=64)		150		125		115		100		75	MHz
Clock-to-Clock Skews												
$t_{IOHCKSW}$	I/O Clock to H-Clock Skew	0.0	1.7	0.0	1.8	0.0	2.0	0.0	2.2	0.0	3.0	ns
$t_{IORCKSW}$	I/O Clock to R-Clock Skew (FO = 64)	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	3.0	ns
t_{HRCKSW}	H-Clock to R-Clock Skew (FO = 64) (FO = 50% max.)	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0 0.0	1.0 3.0	ns

Note:

1. *Delays based on 35pF loading.*

A1425A, A14V25A Timing Characteristics

(Worst-Case Commercial Conditions, V_{CC} = 4.75 V, T_J = 70°C)[1]

Logic Module Propagation Delays[2]		'–3' Speed		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed[1]		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{PD}	Internal Array Module		2.0		2.3		2.6		3.0		3.9	ns
t_{CO}	Sequential Clock to Q		2.0		2.3		2.6		3.0		3.9	ns
t_{CLR}	Asynchronous Clear to Q		2.0		2.3		2.6		3.0		3.9	ns
Predicted Routing Delays[3]												
t_{RD1}	FO=1 Routing Delay		0.9		1.0		1.1		1.3		1.7	ns
t_{RD2}	FO=2 Routing Delay		1.2		1.4		1.6		1.8		2.4	ns
t_{RD3}	FO=3 Routing Delay		1.4		1.6		1.8		2.1		2.8	ns
t_{RD4}	FO=4 Routing Delay		1.7		1.9		2.2		2.5		3.3	ns
t_{RD8}	FO=8 Routing Delay		2.8		3.2		3.6		4.2		5.5	ns
Logic Module Sequential Timing												
t_{SUD}	Flip-Flop Data Input Setup	0.5		0.6		0.7		0.8		0.8		ns
t_{HD}	Flip-Flop Data Input Hold	0.0		0.0		0.0		0.0		0.0		ns
t_{SUD}	Latch Data Input Setup	0.5		0.6		0.7		0.8		0.8		ns
t_{HD}	Latch Data Input Hold	0.0		0.0		0.0		0.0		0.0		ns
t_{WASYN}	Asynchronous Pulse Width	1.9		2.4		3.2		3.8		4.8		ns
t_{WCLKA}	Flip-Flop Clock Pulse Width	1.9		2.4		3.2		3.8		4.8		ns
t_A	Flip-Flop Clock Input Period	4.0		5.0		6.8		8.0		10.0		ns
f_{MAX}	Flip-Flop Clock Frequency		250		200		150		125		100	MHz

Notes:

1. *V_{CC} = 3.0 V for 3.3V specifications.*
2. *For dual-module macros, use $t_{PD} + t_{RD1} + t_{PDn}$, $t_{CO} + t_{RD1} + t_{PDn}$ or $t_{PD1} + t_{RD1} + t_{SUD}$, whichever is appropriate.*
3. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

A1425A, A14V25A Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

I/O Module Input Propagation Delays		'–3' Speed		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{INY}	Input Data Pad to Y		2.8		3.2		3.6		4.2		5.5	ns
t_{ICKY}	Input Reg IOCLK Pad to Y		4.7		5.3		6.0		7.0		9.2	ns
t_{OCKY}	Output Reg IOCLK Pad to Y		4.7		5.3		6.0		7.0		9.2	ns
t_{ICLRY}	Input Asynchronous Clear to Y		4.7		5.3		6.0		7.0		9.2	ns
t_{OCLRY}	Output Asynchronous Clear to Y		4.7		5.3		6.0		7.0		9.2	ns
Predicted Input Routing Delays[1]												
t_{IRD1}	FO=1 Routing Delay		0.9		1.0		1.1		1.3		1.7	ns
t_{IRD2}	FO=2 Routing Delay		1.2		1.4		1.6		1.8		2.4	ns
t_{IRD3}	FO=3 Routing Delay		1.4		1.6		1.8		2.1		2.8	ns
t_{IRD4}	FO=4 Routing Delay		1.7		1.9		2.2		2.5		3.3	ns
t_{IRD8}	FO=8 Routing Delay		2.8		3.2		3.6		4.2		5.5	ns
I/O Module Sequential Timing												
t_{INH}	Input F-F Data Hold (w.r.t. IOCLK Pad)	0.0		0.0		0.0		0.0		0.0		ns
t_{INSU}	Input F-F Data Setup (w.r.t. IOCLK Pad)	1.8		2.0		2.3		2.7		3.0		ns
t_{IDEH}	Input Data Enable Hold (w.r.t. IOCLK Pad)	0.0		0.0		0.0		0.0		0.0		ns
t_{IDESU}	Input Data Enable Setup (w.r.t. IOCLK Pad)	5.8		6.5		7.5		8.6		8.6		ns
t_{OUTH}	Output F-F Data Hold (w.r.t. IOCLK Pad)	0.7		0.8		0.9		1.0		1.0		ns
t_{OUTSU}	Output F-F Data Setup (w.r.t. IOCLK Pad)	0.7		0.8		0.9		1.0		1.0		ns
t_{ODEH}	Output Data Enable Hold (w.r.t. IOCLK Pad)	0.3		0.4		0.4		0.5		0.5		ns
t_{ODESU}	Output Data Enable Setup (w.r.t. IOCLK Pad)	1.3		1.5		1.7		2.0		2.0		ns

Note:

1. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

A1425A, A14V25A Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

I/O Module – TTL Output Timing[1]		'–3' Speed		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{DHS}	Data to Pad, High Slew		5.0		5.6		6.4		7.5		9.8	ns
t_{DLS}	Data to Pad, Low Slew		8.0		9.0		10.2		12.0		15.6	ns
t_{ENZHS}	Enable to Pad, Z to H/L, Hi Slew		4.0		4.5		5.1		6.0		7.8	ns
t_{ENZLS}	Enable to Pad, Z to H/L, Lo Slew		7.4		8.3		9.4		11.0		14.3	ns
t_{ENHSZ}	Enable to Pad, H/L to Z, Hi Slew		6.5		7.5		8.5		10.0		13.0	ns
t_{ENLSZ}	Enable to Pad, H/L to Z, Lo Slew		6.5		7.5		8.5		10.0		13.0	ns
t_{CKHS}	IOCLK Pad to Pad H/L, Hi Slew		7.5		7.5		9.0		10.0		13.0	ns
t_{CKLS}	IOCLK Pad to Pad H/L, Lo Slew		11.3		11.3		13.5		15.0		19.5	ns
d_{TLHHS}	Delta Low to High, Hi Slew		0.02		0.02		0.03		0.03		0.04	ns/pF
d_{TLHLS}	Delta Low to High, Lo Slew		0.05		0.05		0.06		0.07		0.09	ns/pF
d_{THLHS}	Delta High to Low, Hi Slew		0.04		0.04		0.04		0.05		0.07	ns/pF
d_{THLLS}	Delta High to Low, Lo Slew		0.05		0.05		0.06		0.07		0.09	ns/pF
I/O Module – CMOS Output Timing[1]												
t_{DHS}	Data to Pad, High Slew		6.2		7.0		7.9		9.3		12.1	ns
t_{DLS}	Data to Pad, Low Slew		11.7		13.1		14.9		17.5		22.8	ns
t_{ENZHS}	Enable to Pad, Z to H/L, Hi Slew		5.2		5.9		6.6		7.8		10.1	ns
t_{ENZLS}	Enable to Pad, Z to H/L, Lo Slew		8.9		10.0		11.3		13.3		17.3	ns
t_{ENHSZ}	Enable to Pad, H/L to Z, Hi Slew		6.7		7.5		8.5		10.0		13.0	ns
t_{ENLSZ}	Enable to Pad, H/L to Z, Lo Slew		6.7		7.5		9.0		10.0		13.0	ns
t_{CKHS}	IOCLK Pad to Pad H/L, Hi Slew		8.9		8.9		10.7		11.8		15.3	ns
t_{CKLS}	IOCLK Pad to Pad H/L, Lo Slew		13.0		13.0		15.6		17.3		22.5	ns
d_{TLHHS}	Delta Low to High, Hi Slew		0.04		0.04		0.05		0.06		0.08	ns/pF
d_{TLHLS}	Delta Low to High, Lo Slew		0.07		0.08		0.09		0.11		0.14	ns/pF
d_{THLHS}	Delta High to Low, Hi Slew		0.03		0.03		0.03		0.04		0.05	ns/pF
d_{THLLS}	Delta High to Low, Lo Slew		0.04		0.04		0.04		0.05		0.07	ns/pF

Note:

1. Delays based on 35pF loading.

A1425A, A14V25A Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

Dedicated (Hard-Wired) I/O Clock Network		'–3' Speed		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{IOCKH}	Input Low to High (Pad to I/O Module Input)		2.0		2.3		2.6		3.0		3.5	ns
t_{IOPWH}	Minimum Pulse Width High	1.9		2.4		3.3		3.8		4.8		ns
t_{IOPWL}	Minimum Pulse Width Low	1.9		2.4		3.3		3.8		4.8		ns
t_{IOSAPW}	Minimum Asynchronous Pulse Width	1.9		2.4		3.3		3.8		4.8		ns
t_{IOCKSW}	Maximum Skew		0.4		0.4		0.4		0.4		0.4	ns
t_{IOP}	Minimum Period	4.0		5.0		6.8		8.0		10.0		ns
f_{IOMAX}	Maximum Frequency		250		200		150		125		100	MHz
Dedicated (Hard-Wired) Array Clock Network												
t_{HCKH}	Input Low to High (Pad to S-Module Input)		3.0		3.4		3.9		4.5		5.5	ns
t_{HCKL}	Input High to Low (Pad to S-Module Input)		3.0		3.4		3.9		4.5		5.5	ns
t_{HPWH}	Minimum Pulse Width High	1.9		2.4		3.3		3.8		4.8		ns
t_{HPWL}	Minimum Pulse Width Low	1.9		2.4		3.3		3.8		4.8		ns
t_{HCKSW}	Maximum Skew		0.3		0.3		0.3		0.3		0.3	ns
t_{HP}	Minimum Period	4.0		5.0		6.8		8.0		10.0		ns
f_{HMAX}	Maximum Frequency		250		200		150		125		100	MHz
Routed Array Clock Networks												
t_{RCKH}	Input Low to High (FO=64)		3.7		4.1		4.7		5.5		9.0	ns
t_{RCKL}	Input High to Low (FO=64)		4.0		4.5		5.1		6.0		9.0	ns
t_{RPWH}	Min. Pulse Width High (FO=64)	3.3		3.8		4.2		4.9		6.5		ns
t_{RPWL}	Min. Pulse Width Low (FO=64)	3.3		3.8		4.2		4.9		6.5		ns
t_{RCKSW}	Maximum Skew (FO=128)		0.7		0.8		0.9		1.0		1.0	ns
t_{RP}	Minimum Period (FO=64)	6.8		8.0		8.7		10.0		13.4		ns
f_{RMAX}	Maximum Frequency (FO=64)		150		125		115		100		75	MHz
Clock-to-Clock Skews												
$t_{IOHCKSW}$	I/O Clock to H-Clock Skew	0.0	1.7	0.0	1.8	0.0	2.0	0.0	2.2	0.0	3.0	ns
$t_{IORCKSW}$	I/O Clock to R-Clock Skew											
	(FO = 64)	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	3.0	ns
	(FO = 80)	0.0	3.0	0.0	3.0	0.0	3.0	0.0	3.0	0.0	3.0	ns
t_{HRCKSW}	H-Clock to R-Clock Skew											
	(FO = 64)	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	ns
	(FO = 80)	0.0	3.0	0.0	3.0	0.0	3.0	0.0	3.0	0.0	3.0	ns

Note:

1. *Delays based on 35pF loading.*

A1440A, A14V40A Timing Characteristics

(Worst-Case Commercial Conditions, V_{CC} = 4.75 V, T_J = 70°C)[1]

Logic Module Propagation Delays[2]		'–3' Speed		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed[1]		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{PD}	Internal Array Module		2.0		2.3		2.6		3.0		3.9	ns
t_{CO}	Sequential Clock to Q		2.0		2.3		2.6		3.0		3.9	ns
t_{CLR}	Asynchronous Clear to Q		2.0		2.3		2.6		3.0		3.9	ns
Predicted Routing Delays[3]												
t_{RD1}	FO=1 Routing Delay		0.9		1.0		1.1		1.3		1.7	ns
t_{RD2}	FO=2 Routing Delay		1.2		1.4		1.6		1.8		2.4	ns
t_{RD3}	FO=3 Routing Delay		1.4		1.6		1.8		2.1		2.8	ns
t_{RD4}	FO=4 Routing Delay		1.7		1.9		2.2		2.5		3.3	ns
t_{RD8}	FO=8 Routing Delay		2.8		3.2		3.6		4.2		5.5	ns
Logic Module Sequential Timing												
t_{SUD}	Flip-Flop Data Input Setup	0.5		0.6		0.7		0.8		0.8		ns
t_{HD}	Flip-Flop Data Input Hold	0.0		0.0		0.0		0.0		0.0		ns
t_{SUD}	Latch Data Input Setup	0.5		0.6		0.7		0.8		0.8		ns
t_{HD}	Latch Data Input Hold	0.0		0.0		0.0		0.0		0.0		ns
t_{WASYN}	Asynchronous Pulse Width	1.9		2.4		3.2		3.8		4.8		ns
t_{WCLKA}	Flip-Flop Clock Pulse Width	1.9		2.4		3.2		3.8		4.8		ns
t_A	Flip-Flop Clock Input Period	4.0		5.0		6.8		8.0		10.0		ns
f_{MAX}	Flip-Flop Clock Frequency		250		200		150		125		100	MHz

Notes:

1. *V_{CC} = 3.0 V for 3.3V specifications.*
2. *For dual-module macros, use $t_{PD} + t_{RD1} + t_{PDn}$, $t_{CO} + t_{RD1} + t_{PDn}$ or $t_{PD1} + t_{RD1} + t_{SUD}$, whichever is appropriate.*
3. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

A1440A, A14V40A Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

I/O Module Input Propagation Delays		'–3' Speed		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{INY}	Input Data Pad to Y		2.8		3.2		3.6		4.2		5.5	ns
t_{ICKY}	Input Reg IOCLK Pad to Y		4.7		5.3		6.0		7.0		9.2	ns
t_{OCKY}	Output Reg IOCLK Pad to Y		4.7		5.3		6.0		7.0		9.2	ns
t_{ICLRY}	Input Asynchronous Clear to Y		4.7		5.3		6.0		7.0		9.2	ns
t_{OCLRY}	Output Asynchronous Clear to Y		4.7		5.3		6.0		7.0		9.2	ns
Predicted Input Routing Delays[1]												
t_{IRD1}	FO=1 Routing Delay		0.9		1.0		1.1		1.3		1.7	ns
t_{IRD2}	FO=2 Routing Delay		1.2		1.4		1.6		1.8		2.4	ns
t_{IRD3}	FO=3 Routing Delay		1.4		1.6		1.8		2.1		2.8	ns
t_{IRD4}	FO=4 Routing Delay		1.7		1.9		2.2		2.5		3.3	ns
t_{IRD8}	FO=8 Routing Delay		2.8		3.2		3.6		4.2		5.5	ns
I/O Module Sequential Timing												
t_{INH}	Input F-F Data Hold (w.r.t. IOCLK Pad)	0.0		0.0		0.0		0.0		0.0		ns
t_{INSU}	Input F-F Data Setup (w.r.t. IOCLK Pad)	1.5		1.7		2.0		2.3		2.3		ns
t_{IDEH}	Input Data Enable Hold (w.r.t. IOCLK Pad)	0.0		0.0		0.0		0.0		0.0		ns
t_{IDESU}	Input Data Enable Setup (w.r.t. IOCLK Pad)	5.8		6.5		7.5		8.6		8.6		ns
t_{OUTH}	Output F-F Data Hold (w.r.t. IOCLK Pad)	0.7		0.8		0.9		1.0		1.0		ns
t_{OUTSU}	Output F-F Data Setup (w.r.t. IOCLK Pad)	0.7		0.8		0.9		1.0		1.0		ns
t_{ODEH}	Output Data Enable Hold (w.r.t. IOCLK Pad)	0.3		0.4		0.4		0.5		0.5		ns
t_{ODESU}	Output Data Enable Setup (w.r.t. IOCLK Pad)	1.3		1.5		1.7		2.0		2.0		ns

Note:

1. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

A1440A, A14V40A Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

I/O Module – TTL Output Timing[1]		'–3' Speed		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{DHS}	Data to Pad, High Slew		5.0		5.6		6.4		7.5		9.8	ns
t_{DLS}	Data to Pad, Low Slew		8.0		9.0		10.2		12.0		15.6	ns
t_{ENZHS}	Enable to Pad, Z to H/L, Hi Slew		4.0		4.5		5.1		6.0		7.8	ns
t_{ENZLS}	Enable to Pad, Z to H/L, Lo Slew		7.4		8.3		9.4		11.0		14.3	ns
t_{ENHSZ}	Enable to Pad, H/L to Z, Hi Slew		7.4		8.3		9.4		11.0		14.3	ns
t_{ENLSZ}	Enable to Pad, H/L to Z, Lo Slew		7.4		8.3		9.4		11.0		14.3	ns
t_{CKHS}	IOCLK Pad to Pad H/L, Hi Slew		8.5		8.5		9.5		11.0		14.3	ns
t_{CKLS}	IOCLK Pad to Pad H/L, Lo Slew		11.3		11.3		13.5		15.0		19.5	ns
d_{TLHHS}	Delta Low to High, Hi Slew		0.02		0.02		0.03		0.03		0.04	ns/pF
d_{TLHLS}	Delta Low to High, Lo Slew		0.05		0.05		0.06		0.07		0.09	ns/pF
d_{THLHS}	Delta High to Low, Hi Slew		0.04		0.04		0.04		0.05		0.07	ns/pF
d_{THLLS}	Delta High to Low, Lo Slew		0.05		0.05		0.06		0.07		0.09	ns/pF
I/O Module – CMOS Output Timing[1]												
t_{DHS}	Data to Pad, High Slew		6.2		7.0		7.9		9.3		12.1	ns
t_{DLS}	Data to Pad, Low Slew		11.7		13.1		14.9		17.5		22.8	ns
t_{ENZHS}	Enable to Pad, Z to H/L, Hi Slew		5.2		5.9		6.6		7.8		10.1	ns
t_{ENZLS}	Enable to Pad, Z to H/L, Lo Slew		8.9		10.0		11.3		13.3		17.3	ns
t_{ENHSZ}	Enable to Pad, H/L to Z, Hi Slew		7.4		8.3		9.4		11.0		14.3	ns
t_{ENLSZ}	Enable to Pad, H/L to Z, Lo Slew		7.4		8.3		9.4		11.0		14.3	ns
t_{CKHS}	IOCLK Pad to Pad H/L, Hi Slew		9.0		9.0		10.1		11.8		14.3	ns
t_{CKLS}	IOCLK Pad to Pad H/L, Lo Slew		13.0		13.0		15.6		17.3		22.5	ns
d_{TLHHS}	Delta Low to High, Hi Slew		0.04		0.04		0.05		0.06		0.08	ns/pF
d_{TLHLS}	Delta Low to High, Lo Slew		0.07		0.08		0.09		0.11		0.14	ns/pF
d_{THLHS}	Delta High to Low, Hi Slew		0.03		0.03		0.03		0.04		0.05	ns/pF
d_{THLLS}	Delta High to Low, Lo Slew		0.04		0.04		0.04		0.05		0.07	ns/pF

Note:

1. *Delays based on 35pF loading.*

A1440A, A14V40A Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

Dedicated (Hard-Wired) I/O Clock Network		'–3' Speed		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{IOCKH}	Input Low to High (Pad to I/O Module Input)		2.0		2.3		2.6		3.0		3.5	ns
t_{IOPWH}	Minimum Pulse Width High	1.9		2.4		3.3		3.8		4.8		ns
t_{IOPWL}	Minimum Pulse Width Low	1.9		2.4		3.3		3.8		4.8		ns
t_{IOSAPW}	Minimum Asynchronous Pulse Width	1.9		2.4		3.3		3.8		4.8		ns
t_{IOCKSW}	Maximum Skew		0.4		0.4		0.4		0.4		0.4	ns
t_{IOP}	Minimum Period	4.0		5.0		6.8		8.0		10.0		ns
f_{IOMAX}	Maximum Frequency		250		200		150		125		100	MHz
Dedicated (Hard-Wired) Array Clock Network												
t_{HCKH}	Input Low to High (Pad to S-Module Input)		3.0		3.4		3.9		4.5		5.5	ns
t_{HCKL}	Input High to Low (Pad to S-Module Input)		3.0		3.4		3.9		4.5		5.5	ns
t_{HPWH}	Minimum Pulse Width High	1.9		2.4		3.3		3.8		4.8		ns
t_{HPWL}	Minimum Pulse Width Low	1.9		2.4		3.3		3.8		4.8		ns
t_{HCKSW}	Maximum Skew		0.3		0.3		0.3		0.3		0.3	ns
t_{HP}	Minimum Period	4.0		5.0		6.8		8.0		10.0		ns
f_{HMAX}	Maximum Frequency		250		200		150		125		100	MHz
Routed Array Clock Networks												
t_{RCKH}	Input Low to High (FO=64)		3.7		4.1		4.7		5.5		9.0	ns
t_{RCKL}	Input High to Low (FO=64)		4.0		4.5		5.1		6.0		9.0	ns
t_{RPWH}	Min. Pulse Width High (FO=64)	3.3		3.8		4.2		4.9		6.5		ns
t_{RPWL}	Min. Pulse Width Low (FO=64)	3.3		3.8		4.2		4.9		6.5		ns
t_{RCKSW}	Maximum Skew (FO=128)		0.7		0.8		0.9		1.0		1.0	ns
t_{RP}	Minimum Period (FO=64)	6.8		8.0		8.7		10.0		13.4		ns
f_{RMAX}	Maximum Frequency (FO=64)		150		125		115		100		75	MHz
Clock-to-Clock Skews												
$t_{IOHCKSW}$	I/O Clock to H-Clock Skew	0.0	1.7	0.0	1.8	0.0	2.0	0.0	2.2	0.0	3.0	ns
$t_{IORCKSW}$	I/O Clock to R-Clock Skew											
	(FO = 64)	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	3.0	ns
	(FO = 144)	0.0	3.0	0.0	3.0	0.0	3.0	0.0	3.0	0.0	3.0	ns
t_{HRCKSW}	H-Clock to R-Clock Skew											
	(FO = 64)	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	ns
	(FO = 144)	0.0	3.0	0.0	3.0	0.0	3.0	0.0	3.0	0.0	3.0	ns

Note:

1. *Delays based on 35pF loading.*

A1460A, A1460B, A14V60A Timing Characteristics

(Worst-Case Commercial Conditions, V_{CC} = 4.75 V, T_J = 70°C)[1]

Logic Module Propagation Delays[2]		'–3' Speed		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed[1]		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{PD}	Internal Array Module		2.0		2.3		2.6		3.0		3.9	ns
t_{CO}	Sequential Clock to Q		2.0		2.3		2.6		3.0		3.9	ns
t_{CLR}	Asynchronous Clear to Q		2.0		2.3		2.6		3.0		3.9	ns
Predicted Routing Delays[3]												
t_{RD1}	FO=1 Routing Delay		0.9		1.0		1.1		1.3		1.7	ns
t_{RD2}	FO=2 Routing Delay		1.2		1.4		1.6		1.8		2.4	ns
t_{RD3}	FO=3 Routing Delay		1.4		1.6		1.8		2.1		2.8	ns
t_{RD4}	FO=4 Routing Delay		1.7		1.9		2.2		2.5		3.3	ns
t_{RD8}	FO=8 Routing Delay		2.8		3.2		3.6		4.2		5.5	ns
Logic Module Sequential Timing												
t_{SUD}	Flip-Flop Data Input Setup	0.5		0.6		0.7		0.8		0.8		ns
t_{HD}	Flip-Flop Data Input Hold	0.0		0.0		0.0		0.0		0.0		ns
t_{SUD}	Latch Data Input Setup	0.5		0.6		0.7		0.8		0.8		ns
t_{HD}	Latch Data Input Hold	0.0		0.0		0.0		0.0		0.0		ns
t_{WASYN}	Asynchronous Pulse Width	2.4		3.2		3.8		4.8		6.5		ns
t_{WCLKA}	Flip-Flop Clock Pulse Width	2.4		3.2		3.8		4.8		6.5		ns
t_A	Flip-Flop Clock Input Period	5.0		6.8		8.0		10.0		13.4		ns
f_{MAX}	Flip-Flop Clock Frequency		200		150		125		100		75	MHz

Notes:

1. V_{CC} = 3.0 V for 3.3V specifications.
2. For dual-module macros, use $t_{PD} + t_{RD1} + t_{PDn}$, $t_{CO} + t_{RD1} + t_{PDn}$ or $t_{PD1} + t_{RD1} + t_{SUD}$, whichever is appropriate.
3. Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.

A1460A, A1460B, A14V60A Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

I/O Module Input Propagation Delays		'–3' Speed		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{INY}	Input Data Pad to Y		2.8		3.2		3.6		4.2		5.5	ns
t_{ICKY}	Input Reg IOCLK Pad to Y		4.7		5.3		6.0		7.0		9.2	ns
t_{OCKY}	Output Reg IOCLK Pad to Y		4.7		5.3		6.0		7.0		9.2	ns
t_{ICLRY}	Input Asynchronous Clear to Y		4.7		5.3		6.0		7.0		9.2	ns
t_{OCLRY}	Output Asynchronous Clear to Y		4.7		5.3		6.0		7.0		9.2	ns
Predicted Input Routing Delays[1]												
t_{IRD1}	FO=1 Routing Delay		0.9		1.0		1.1		1.3		1.7	ns
t_{IRD2}	FO=2 Routing Delay		1.2		1.4		1.6		1.8		2.4	ns
t_{IRD3}	FO=3 Routing Delay		1.4		1.6		1.8		2.1		2.8	ns
t_{IRD4}	FO=4 Routing Delay		1.7		1.9		2.2		2.5		3.3	ns
t_{IRD8}	FO=8 Routing Delay		2.8		3.2		3.6		4.2		5.5	ns
I/O Module Sequential Timing												
t_{INH}	Input F-F Data Hold (w.r.t. IOCLK Pad)	0.0		0.0		0.0		0.0		0.0		ns
t_{INSU}	Input F-F Data Setup (w.r.t. IOCLK Pad)	1.3		1.5		1.8		2.0		2.0		ns
t_{IDEH}	Input Data Enable Hold (w.r.t. IOCLK Pad)	0.0		0.0		0.0		0.0		0.0		ns
t_{IDESU}	Input Data Enable Setup (w.r.t. IOCLK Pad)	5.8		6.5		7.5		8.6		8.6		ns
t_{OUTH}	Output F-F Data Hold (w.r.t. IOCLK Pad)	0.7		0.8		0.9		1.0		1.0		ns
t_{OUTSU}	Output F-F Data Setup (w.r.t. IOCLK Pad)	0.7		0.8		0.9		1.0		1.0		ns
t_{ODEH}	Output Data Enable Hold (w.r.t. IOCLK Pad)	0.3		0.4		0.4		0.5		0.5		ns
t_{ODESU}	Output Data Enable Setup (w.r.t. IOCLK Pad)	1.3		1.5		1.7		2.0		2.0		ns

Note:

1. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

A1460A, A1460B, A14V60A Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

I/O Module – TTL Output Timing[1]		'–3' Speed		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{DHS}	Data to Pad, High Slew		5.0		5.6		6.4		7.5		9.8	ns
t_{DLS}	Data to Pad, Low Slew		8.0		9.0		10.2		12.0		15.6	ns
t_{ENZHS}	Enable to Pad, Z to H/L, Hi Slew		4.0		4.5		5.1		6.0		7.8	ns
t_{ENZLS}	Enable to Pad, Z to H/L, Lo Slew		7.4		8.3		9.4		11.0		14.3	ns
t_{ENHSZ}	Enable to Pad, H/L to Z, Hi Slew		7.8		8.7		9.9		11.6		15.1	ns
t_{ENLSZ}	Enable to Pad, H/L to Z, Lo Slew		7.4		8.3		9.4		11.0		14.3	ns
t_{CKHS}	IOCLK Pad to Pad H/L, Hi Slew		9.0		9.0		10.0		11.5		15.0	ns
t_{CKLS}	IOCLK Pad to Pad H/L, Lo Slew		12.8		12.8		15.3		17.0		22.1	ns
d_{TLHHS}	Delta Low to High, Hi Slew		0.02		0.02		0.03		0.03		0.04	ns/pF
d_{TLHLS}	Delta Low to High, Lo Slew		0.05		0.05		0.06		0.07		0.09	ns/pF
d_{THLHS}	Delta High to Low, Hi Slew		0.04		0.04		0.04		0.05		0.07	ns/pF
d_{THLLS}	Delta High to Low, Lo Slew		0.05		0.05		0.06		0.07		0.09	ns/pF
I/O Module – CMOS Output Timing[1]												
t_{DHS}	Data to Pad, High Slew		6.2		7.0		7.9		9.3		12.1	ns
t_{DLS}	Data to Pad, Low Slew		11.7		13.1		14.9		17.5		22.8	ns
t_{ENZHS}	Enable to Pad, Z to H/L, Hi Slew		5.2		5.9		6.6		7.8		10.1	ns
t_{ENZLS}	Enable to Pad, Z to H/L, Lo Slew		8.9		10.0		11.3		13.3		17.3	ns
t_{ENHSZ}	Enable to Pad, H/L to Z, Hi Slew		7.4		8.3		9.4		11.0		14.3	ns
t_{ENLSZ}	Enable to Pad, H/L to Z, Lo Slew		7.4		8.3		9.4		11.0		14.3	ns
t_{CKHS}	IOCLK Pad to Pad H/L, Hi Slew		10.4		10.4		12.1		13.8		17.9	ns
t_{CKLS}	IOCLK Pad to Pad H/L, Lo Slew		14.5		14.5		17.4		19.3		25.1	ns
d_{TLHHS}	Delta Low to High, Hi Slew		0.04		0.04		0.05		0.06		0.08	ns/pF
d_{TLHLS}	Delta Low to High, Lo Slew		0.07		0.08		0.09		0.11		0.14	ns/pF
d_{THLHS}	Delta High to Low, Hi Slew		0.03		0.03		0.03		0.04		0.05	ns/pF
d_{THLLS}	Delta High to Low, Lo Slew		0.04		0.04		0.04		0.05		0.07	ns/pF

Note:

[1] *Delays based on 35pF loading.*

A1460A, A1460B, A14V60A Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

Dedicated (Hard-Wired) I/O Clock Network		'–3' Speed		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{IOCKH}	Input Low to High (Pad to I/O Module Input)		2.3		2.6		3.0		3.5		4.5	ns
t_{IOPWH}	Minimum Pulse Width High	2.4		3.2		3.8		4.8		6.5		ns
t_{IOPWL}	Minimum Pulse Width Low	2.4		3.2		3.8		4.8		6.5		ns
t_{IOSAPW}	Minimum Asynchronous Pulse Width	2.4		3.2		3.8		4.8		6.5		ns
t_{IOCKSW}	Maximum Skew		0.6		0.6		0.6		0.6		0.6	ns
t_{IOP}	Minimum Period	5.0		6.8		8.0		10.0		13.4		ns
f_{IOMAX}	Maximum Frequency		200		150		125		100		75	MHz
Dedicated (Hard-Wired) Array Clock Network												
t_{HCKH}	Input Low to High (Pad to S-Module Input)		3.7		4.1		4.7		5.5		7.0	ns
t_{HCKL}	Input High to Low (Pad to S-Module Input)		3.7		4.1		4.7		5.5		7.0	ns
t_{HPWH}	Minimum Pulse Width High	2.4		3.2		3.8		4.8		6.5		ns
t_{HPWL}	Minimum Pulse Width Low	2.4		3.2		3.8		4.8		6.5		ns
t_{HCKSW}	Maximum Skew		0.6		0.6		0.6		0.6		0.6	ns
t_{HP}	Minimum Period	5.0		6.8		8.0		10.0		13.4		ns
f_{HMAX}	Maximum Frequency		200		150		125		100		75	MHz
Routed Array Clock Networks												
t_{RCKH}	Input Low to High (FO=256)		6.0		6.8		7.7		9.0		11.8	ns
t_{RCKL}	Input High to Low (FO=256)		6.0		6.8		7.7		9.0		11.8	ns
t_{RPWH}	Min. Pulse Width High (FO=256)	4.1		4.5		5.4		6.1		8.2		ns
t_{RPWL}	Min. Pulse Width Low (FO=256)	4.1		4.5		5.4		6.1		8.2		ns
t_{RCKSW}	Maximum Skew (FO=128)		1.2		1.4		1.6		1.8		1.8	ns
t_{RP}	Minimum Period (FO=256)	8.3		9.3		11.1		12.5		16.7		ns
f_{RMAX}	Maximum Frequency (FO=256)		120		105		90		80		60	MHz
Clock-to-Clock Skews												
$t_{IOHCKSW}$	I/O Clock to H-Clock Skew	0.0	2.6	0.0	2.7	0.0	2.9	0.0	3.0	0.0	3.0	ns
$t_{IORCKSW}$	I/O Clock to R-Clock Skew											
	(FO = 64)	0.0	1.7	0.0	1.7	0.0	1.7	0.0	1.7	0.0	5.0	ns
	(FO = 216)	0.0	5.0	0.0	5.0	0.0	5.0	0.0	5.0	0.0	5.0	ns
t_{HRCKSW}	H-Clock to R-Clock Skew											
	(FO = 64)	0.0	1.3	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	ns
	(FO = 216)	0.0	3.0	0.0	3.0	0.0	3.0	0.0	3.0	0.0	3.0	ns

Note:

1. Delays based on 35pF loading.

A14100A, A14100B, A14V100A Timing Characteristics

(Worst-Case Commercial Conditions, V_{CC} = 4.75 V, T_J = 70°C)[1]

Logic Module Propagation Delays[2]		'–3' Speed		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed[1]		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{PD}	Internal Array Module		2.0		2.3		2.6		3.0		3.9	ns
t_{CO}	Sequential Clock to Q		2.0		2.3		2.6		3.0		3.9	ns
t_{CLR}	Asynchronous Clear to Q		2.0		2.3		2.6		3.0		3.9	ns
Predicted Routing Delays[3]												
t_{RD1}	FO=1 Routing Delay		0.9		1.0		1.1		1.3		1.7	ns
t_{RD2}	FO=2 Routing Delay		1.2		1.4		1.6		1.8		2.4	ns
t_{RD3}	FO=3 Routing Delay		1.4		1.6		1.8		2.1		2.8	ns
t_{RD4}	FO=4 Routing Delay		1.7		1.9		2.2		2.5		3.3	ns
t_{RD8}	FO=8 Routing Delay		2.8		3.2		3.6		4.2		5.5	ns
Logic Module Sequential Timing												
t_{SUD}	Flip-Flop Data Input Setup	0.5		0.6		0.8		0.8		0.8		ns
t_{HD}	Flip-Flop Data Input Hold	0.0		0.0		0.5		0.5		0.5		ns
t_{SUD}	Latch Data Input Setup	0.5		0.6		0.8		0.8		0.8		ns
t_{HD}	Latch Data Input Hold	0.0		0.0		0.5		0.5		0.5		ns
t_{WASYN}	Asynchronous Pulse Width	2.4		3.2		3.8		4.8		6.5		ns
t_{WCLKA}	Flip-Flop Clock Pulse Width	2.4		3.2		3.8		4.8		6.5		ns
t_A	Flip-Flop Clock Input Period	5.0		6.8		8.0		10.0		13.4		ns
f_{MAX}	Flip-Flop Clock Frequency		200		150		125		100		75	MHz

Notes:

1. V_{CC} = 3.0 V *for 3.3V specifications.*
2. *For dual-module macros, use* $t_{PD} + t_{RD1} + t_{PDn}$, $t_{CO} + t_{RD1} + t_{PDn}$ *or* $t_{PD1} + t_{RD1} + t_{SUD}$, *whichever is appropriate.*
3. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

A14100A, A14100B, A14V100A Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

I/O Module Input Propagation Delays		'–3' Speed		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{INY}	Input Data Pad to Y		2.8		3.2		3.6		4.2		5.5	ns
t_{ICKY}	Input Reg IOCLK Pad to Y		4.7		5.3		6.0		7.0		9.2	ns
t_{OCKY}	Output Reg IOCLK Pad to Y		4.7		5.3		6.0		7.0		9.2	ns
t_{ICLRY}	Input Asynchronous Clear to Y		4.7		5.3		6.0		7.0		9.2	ns
t_{OCLRY}	Output Asynchronous Clear to Y		4.7		5.3		6.0		7.0		9.2	ns
Predicted Input Routing Delays[1]												
t_{IRD1}	FO=1 Routing Delay		0.9		1.0		1.1		1.3		1.7	ns
t_{IRD2}	FO=2 Routing Delay		1.2		1.4		1.6		1.8		2.4	ns
t_{IRD3}	FO=3 Routing Delay		1.4		1.6		1.8		2.1		2.8	ns
t_{IRD4}	FO=4 Routing Delay		1.7		1.9		2.2		2.5		3.3	ns
t_{IRD8}	FO=8 Routing Delay		2.8		3.2		3.6		4.2		5.5	ns
I/O Module Sequential Timing												
t_{INH}	Input F-F Data Hold (w.r.t. IOCLK Pad)	0.0		0.0		0.0		0.0		0.0		ns
t_{INSU}	Input F-F Data Setup (w.r.t. IOCLK Pad)	1.2		1.4		1.5		1.8		1.8		ns
t_{IDEH}	Input Data Enable Hold (w.r.t. IOCLK Pad)	0.0		0.0		0.0		0.0		0.0		ns
t_{IDESU}	Input Data Enable Setup (w.r.t. IOCLK Pad)	5.8		6.5		7.5		8.6		8.6		ns
t_{OUTH}	Output F-F Data Hold (w.r.t. IOCLK Pad)	0.7		0.8		1.0		1.0		1.0		ns
t_{OUTSU}	Output F-F Data Setup (w.r.t. IOCLK Pad)	0.7		0.8		1.0		1.0		1.0		ns
t_{ODEH}	Output Data Enable Hold (w.r.t. IOCLK Pad)	0.3		0.4		0.5		0.5		0.5		ns
t_{ODESU}	Output Data Enable Setup (w.r.t. IOCLK Pad)	1.3		1.5		2.0		2.0		2.0		ns

Note:

1. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

A14100A, A14100B, A14V100A Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

I/O Module – TTL Output Timing[1]		'–3' Speed		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{DHS}	Data to Pad, High Slew		5.0		5.6		6.4		7.5		9.8	ns
t_{DLS}	Data to Pad, Low Slew		8.0		9.0		10.2		12.0		15.6	ns
t_{ENZHS}	Enable to Pad, Z to H/L, Hi Slew		4.0		4.5		5.1		6.0		7.8	ns
t_{ENZLS}	Enable to Pad, Z to H/L, Lo Slew		7.4		8.3		9.4		11.0		14.3	ns
t_{ENHSZ}	Enable to Pad, H/L to Z, Hi Slew		8.0		9.0		10.2		12.0		15.6	ns
t_{ENLSZ}	Enable to Pad, H/L to Z, Lo Slew		7.4		8.3		9.4		11.0		14.3	ns
t_{CKHS}	IOCLK Pad to Pad H/L, Hi Slew		9.5		9.5		10.5		12.0		15.6	ns
t_{CKLS}	IOCLK Pad to Pad H/L, Lo Slew		12.8		12.8		15.3		17.0		22.1	ns
d_{TLHHS}	Delta Low to High, Hi Slew		0.02		0.02		0.03		0.03		0.04	ns/pF
d_{TLHLS}	Delta Low to High, Lo Slew		0.05		0.05		0.06		0.07		0.09	ns/pF
d_{THLHS}	Delta High to Low, Hi Slew		0.04		0.04		0.04		0.05		0.07	ns/pF
d_{THLLS}	Delta High to Low, Lo Slew		0.05		0.05		0.06		0.07		0.09	ns/pF
I/O Module – CMOS Output Timing[1]												
t_{DHS}	Data to Pad, High Slew		6.2		7.0		7.9		9.3		12.1	ns
t_{DLS}	Data to Pad, Low Slew		11.7		13.1		14.9		17.5		22.8	ns
t_{ENZHS}	Enable to Pad, Z to H/L, Hi Slew		5.2		5.9		6.6		7.8		10.1	ns
t_{ENZLS}	Enable to Pad, Z to H/L, Lo Slew		8.9		10.0		11.3		13.3		17.3	ns
t_{ENHSZ}	Enable to Pad, H/L to Z, Hi Slew		8.0		9.0		10.0		12.0		15.6	ns
t_{ENLSZ}	Enable to Pad, H/L to Z, Lo Slew		7.4		8.3		9.4		11.0		14.3	ns
t_{CKHS}	IOCLK Pad to Pad H/L, Hi Slew		10.4		10.4		12.4		13.8		17.9	ns
t_{CKLS}	IOCLK Pad to Pad H/L, Lo Slew		14.5		14.5		17.4		19.3		25.1	ns
d_{TLHHS}	Delta Low to High, Hi Slew		0.04		0.04		0.05		0.06		0.08	ns/pF
d_{TLHLS}	Delta Low to High, Lo Slew		0.07		0.08		0.09		0.11		0.14	ns/pF
d_{THLHS}	Delta High to Low, Hi Slew		0.03		0.03		0.03		0.04		0.05	ns/pF
d_{THLLS}	Delta High to Low, Lo Slew		0.04		0.04		0.04		0.05		0.07	ns/pF

Note:

[1] *Delays based on 35pF loading.*

A14100A, A14100B, A14V100A Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

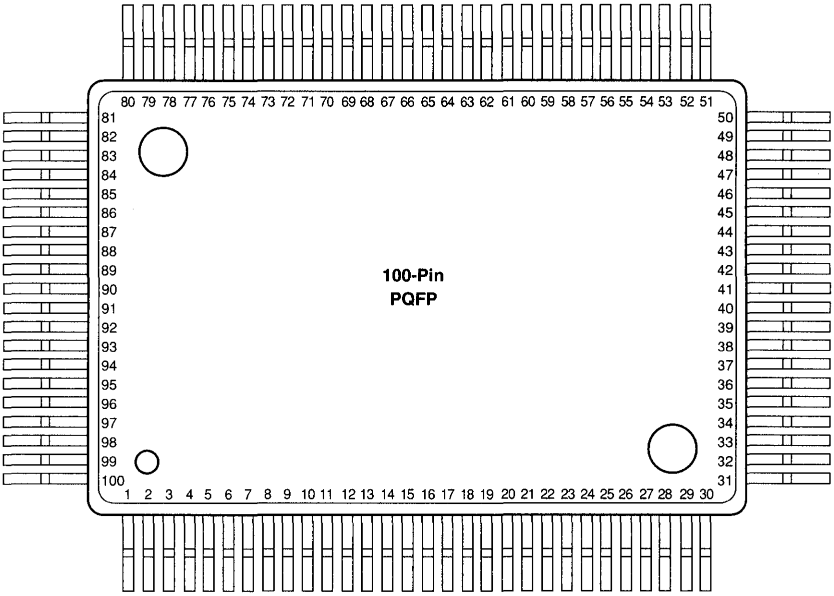
Dedicated (Hard-Wired) I/O Clock Network		'–3' Speed		'–2' Speed		'–1' Speed		'Std' Speed		3.3V Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{IOCKH}	Input Low to High (Pad to I/O Module Input)		2.3		2.6		3.0		3.5		4.5	ns
t_{IOPWH}	Minimum Pulse Width High	2.4		3.3		3.8		4.8		6.5		ns
t_{IOPWL}	Minimum Pulse Width Low	2.4		3.3		3.8		4.8		6.5		ns
t_{IOSAPW}	Minimum Asynchronous Pulse Width	2.4		3.3		3.8		4.8		6.5		ns
t_{IOCKSW}	Maximum Skew		0.6		0.6		0.7		0.8		0.6	ns
t_{IOP}	Minimum Period	5.0		6.8		8.0		10.0		13.4		ns
f_{IOMAX}	Maximum Frequency		200		150		125		100		75	MHz
Dedicated (Hard-Wired) Array Clock Network												
t_{HCKH}	Input Low to High (Pad to S-Module Input)		3.7		4.1		4.7		5.5		7.0	ns
t_{HCKL}	Input High to Low (Pad to S-Module Input)		3.7		4.1		4.7		5.5		7.0	ns
t_{HPWH}	Minimum Pulse Width High	2.4		3.3		3.8		4.8		6.5		ns
t_{HPWL}	Minimum Pulse Width Low	2.4		3.3		3.8		4.8		6.5		ns
t_{HCKSW}	Maximum Skew		0.6		0.6		0.7		0.8		0.6	ns
t_{HP}	Minimum Period	5.0		6.8		8.0		10.0		13.4		ns
f_{HMAX}	Maximum Frequency		200		150		125		100		75	MHz
Routed Array Clock Networks												
t_{RCKH}	Input Low to High (FO=256)		6.0		6.8		7.7		9.0		11.8	ns
t_{RCKL}	Input High to Low (FO=256)		6.0		6.8		7.7		9.0		11.8	ns
t_{RPWH}	Min. Pulse Width High (FO=256)	4.1		4.5		5.4		6.1		8.2		ns
t_{RPWL}	Min. Pulse Width Low (FO=256)	4.1		4.5		5.4		6.1		8.2		ns
t_{RCKSW}	Maximum Skew (FO=128)		1..2		1.4		1.6		1.8		1.8	ns
t_{RP}	Minimum Period (FO=256)	8.3		9.3		11.1		12.5		16.7		ns
f_{RMAX}	Maximum Frequency (FO=256)		120		105		90		80		60	MHz
Clock-to-Clock Skews												
$t_{IOHCKSW}$	I/O Clock to H-Clock Skew	0.0	2.6	0.0	2.7	0.0	2.9	0.0	3.0	0.0	3.0	ns
$t_{IORCKSW}$	I/O Clock to R-Clock Skew											
	(FO = 64)	0.0	1.7	0.0	17	0.0	1.7	0.0	1.7	0.0	5.0	
	(FO = 350)	0.0	5.0	0.0	5.0	0.0	5.0	0.0	5.0	0.0	5.0	ns
t_{HRCKSW}	H-Clock to R-Clock Skew											
	(FO = 64)	0.0	1.3	0.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	
	(FO = 350)	0.0	3.0	0.0	3.0	0.0	3.0	0.0	3.0	0.0	3.0	ns

Note:

1. *Delays based on 35pF loading.*

Package Pin Assignments

100-Pin PQFP (Top View)



Pin Number	A1415 Function	A1425 Function
2	IOCLK, I/O	IOCLK, I/O
14	CLKA, I/O	CLKA, I/O
15	CLKB, I/O	CLKB, I/O
16	VCC	VCC
17	GND	GND
18	VCC	VCC
19	GND	GND
20	PRA, I/O	PRA, I/O
27	DCLK, I/O	DCLK, I/O
28	GND	GND
29	SDI, I/O	SDI, I/O
34	MODE	MODE
35	VCC	VCC
36	GND	GND
47	GND	GND

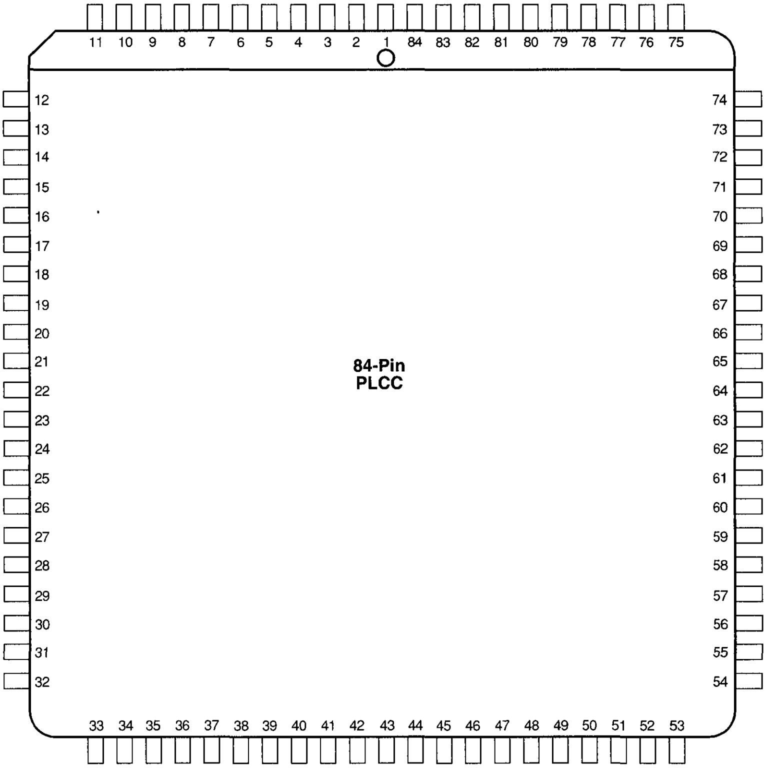
Pin Number	A1415 Function	A1425 Function
48	VCC	VCC
61	PRB, I/O	PRB, I/O
62	GND	GND
63	VCC	VCC
64	GND	GND
65	VCC	VCC
67	HCLK, I/O	HCLK, I/O
78	IOPCL, I/O	IOPCL, I/O
79	GND	GND
85	VCC	VCC
86	VCC	VCC
87	GND	GND
96	VCC	VCC
97	GND	GND

Notes:

1. *All unlisted pin numbers are user I/Os.*
2. *NC : Denotes No Connection*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

84-Pin PLCC (Top View)



84-Pin PLCC

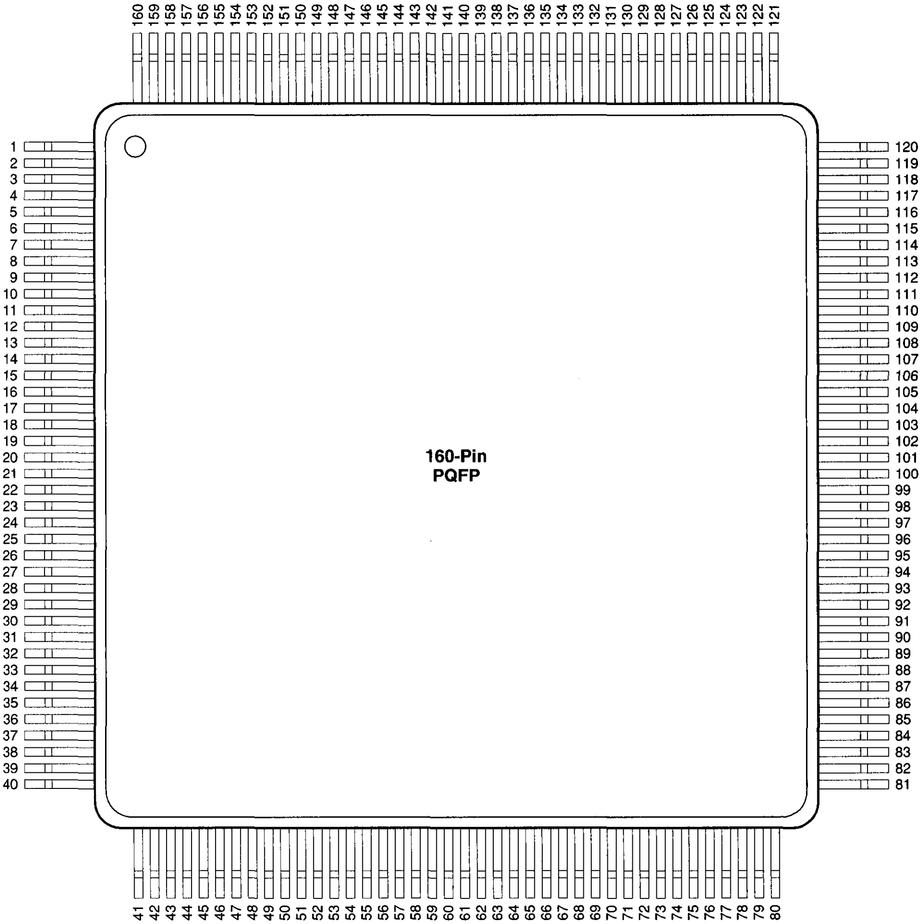
Pin Number	A1415 A14V15 Function	A1425 A14V25 Function	A1440 A14V40 Function
1	VCC	VCC	VCC
2	GND	GND	GND
3	VCC	VCC	VCC
4	PRA, I/O	PRA, I/O	PRA, I/O
11	DCLK, I/O	DCLK, I/O	DCLK, I/O
12	SDI, I/O	SDI, I/O	SDI, I/O
16	MODE	MODE	MODE
27	GND	GND	GND
28	VCC	VCC	VCC
40	PRB, I/O	PRB, I/O	PRB, I/O
41	VCC	VCC	VCC
42	GND	GND	GND
43	VCC	VCC	VCC
45	HCLK, I/O	HCLK, I/O	HCLK, I/O
53	IOPCL, I/O	IOPCL, I/O	IOPCL, I/O
59	VCC	VCC	VCC
60	VCC	VCC	VCC
61	GND	GND	GND
68	VCC	VCC	VCC
69	GND	GND	GND
74	IOCLK, I/O	IOCLK, I/O	IOCLK, I/O
83	CLKA, I/O	CLKA, I/O	CLKA, I/O
84	CLKB, I/O	CLKB, I/O	CLKB, I/O

Notes:

1. *All unlisted pin numbers are user I/Os.*
2. *NC : Denotes No Connection*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

160-Pin PQFP (Top View)



160-Pin PQFP

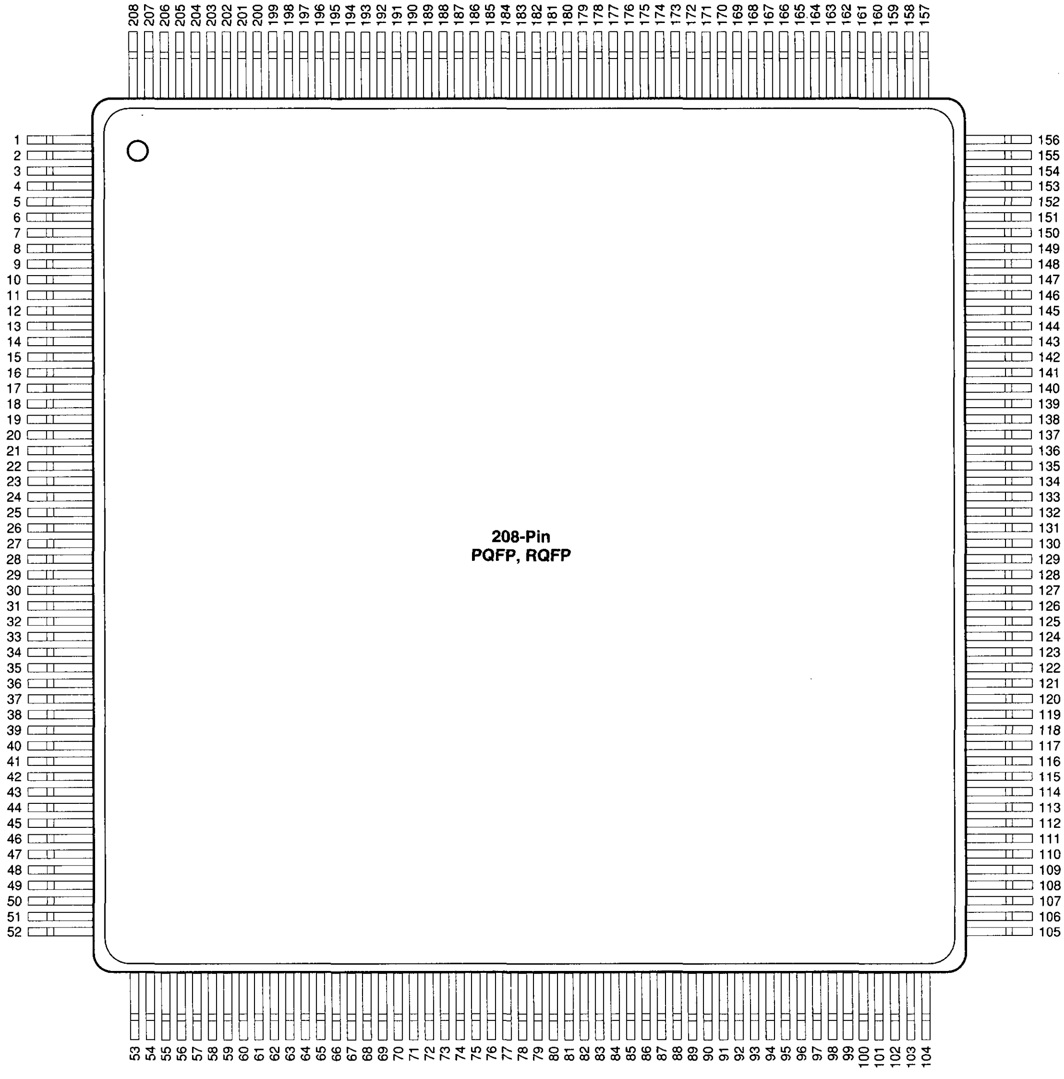
Pin Number	A1425 A14V25 Function	A1440 A14V40 Function	A1460 A14V60 Function
1	GND	GND	GND
2	SDI, I/O	SDI, I/O	SDI, I/O
5	NC	I/O	I/O
9	MODE	MODE	MODE
10	VCC	VCC	VCC
14	NC	I/O	I/O
15	GND	GND	GND
18	VCC	VCC	VCC
19	GND	GND	GND
20	NC	I/O	I/O
24	NC	I/O	I/O
27	NC	I/O	I/O
28	VCC	VCC	VCC
29	VCC	VCC	VCC
40	GND	GND	GND
41	NC	I/O	I/O
43	NC	I/O	I/O
45	NC	I/O	I/O
46	VCC	VCC	VCC
47	NC	I/O	I/O
49	NC	I/O	I/O
51	NC	I/O	I/O
53	NC	I/O	I/O
58	PRB, I/O	PRB, I/O	PRB, I/O
59	GND	GND	GND
60	VCC	VCC	VCC
62	HCLK, I/O	HCLK, I/O	HCLK, I/O
63	GND	GND	GND
74	NC	I/O	I/O
75	VCC	VCC	VCC
76	NC	I/O	I/O
77	NC	I/O	I/O
78	NC	I/O	I/O
80	IOPCL, I/O	IOPCL, I/O	IOPCL, I/O
81	GND	GND	GND
90	VCC	VCC	VCC
91	VCC	VCC	VCC
92	NC	I/O	I/O
93	NC	I/O	I/O
98	GND	GND	GND
99	VCC	VCC	VCC
100	NC	I/O	I/O
103	GND	GND	GND
107	NC	I/O	I/O
109	NC	I/O	I/O
110	VCC	VCC	VCC
111	GND	GND	GND
112	VCC	VCC	VCC
113	NC	I/O	I/O
119	NC	I/O	I/O
120	IOCLK, I/O	IOCLK, I/O	IOCLK, I/O
121	GND	GND	GND
124	NC	I/O	I/O
127	NC	I/O	I/O
136	CLKA, I/O	CLKA, I/O	CLKA, I/O
137	CLKB, I/O	CLKB, I/O	CLKB, I/O
138	VCC	VCC	VCC
139	GND	GND	GND
140	VCC	VCC	VCC
141	GND	GND	GND
142	PRA, I/O	PRA, I/O	PRA, I/O
143	NC	I/O	I/O
145	NC	I/O	I/O
147	NC	I/O	I/O
149	NC	I/O	I/O
151	NC	I/O	I/O
153	NC	I/O	I/O
154	VCC	VCC	VCC
160	DCLK, I/O	DCLK, I/O	DCLK, I/O

Notes:

1. *All unlisted pin numbers are user I/Os.*
2. *NC : Denotes No Connection*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

208-Pin PQFP, RQFP (Top View)



208-Pin PQFP, RQFP

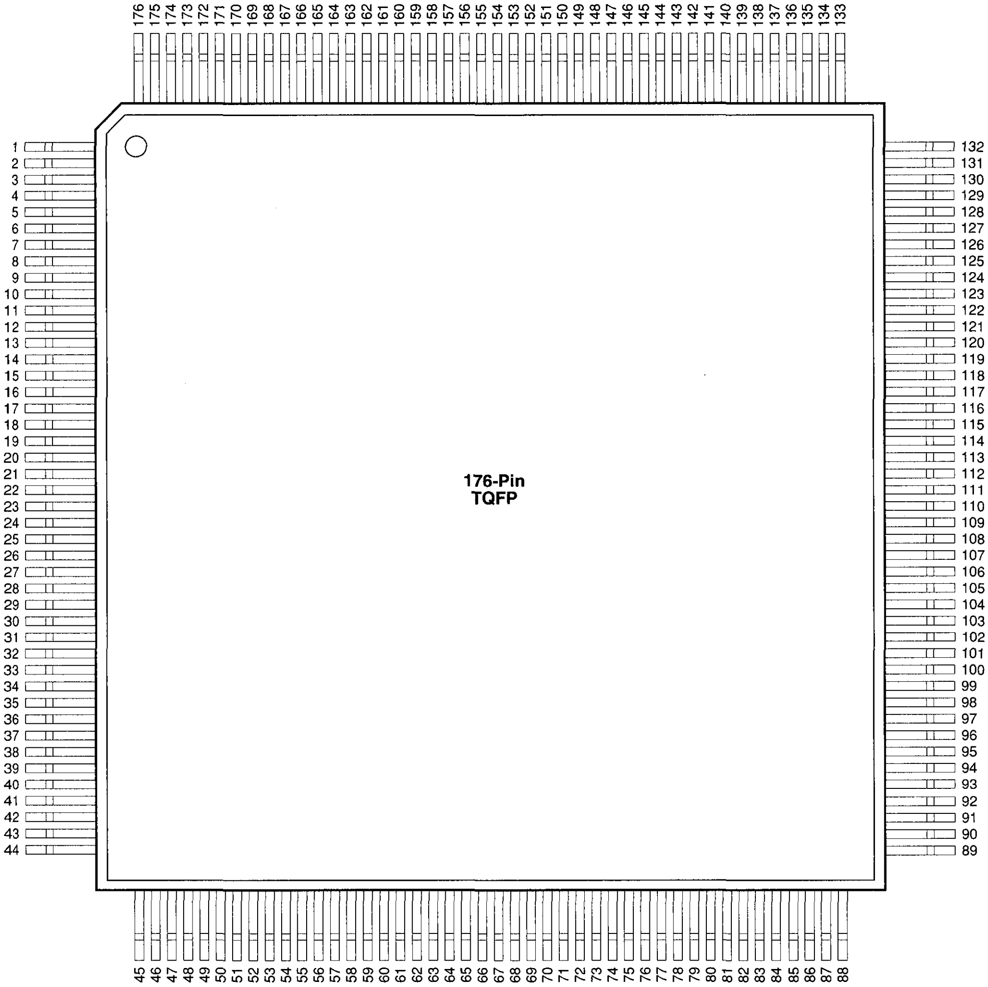
Pin Number	A1460 A14V60 Function	A14100 A14V100 Function
1	GND	GND
2	SDI, I/O	SDI, I/O
11	MODE	MODE
12	VCC	VCC
25	VCC	VCC
26	GND	GND
27	VCC	VCC
28	GND	GND
40	VCC	VCC
41	VCC	VCC
52	GND	GND
53	NC	I/O
60	VCC	VCC
65	NC	I/O
76	PRB, I/O	PRB, I/O
77	GND	GND
78	VCC	VCC
79	GND	GND
80	VCC	VCC
82	HCLK, I/O	HCLK, I/O
98	VCC	VCC
102	NC	I/O
104	IOPCL, I/O	IOPCL, I/O
105	GND	GND
114	VCC	VCC
115	VCC	VCC
116	NC	I/O
129	GND	GND
130	VCC	VCC
131	GND	GND
132	VCC	VCC
145	VCC	VCC
146	GND	GND
147	NC	I/O
148	VCC	VCC
156	IOCLK, I/O	IOCLK, I/O
157	GND	GND
158	NC	I/O
164	VCC	VCC
180	CLKA, I/O	CLKA, I/O
181	CLKB, I/O	CLKB, I/O
182	VCC	VCC
183	GND	GND
184	VCC	VCC
185	GND	GND
186	PRA, I/O	PRA, I/O
195	NC	I/O
201	VCC	VCC
205	NC	I/O
208	DCLK, I/O	DCLK, I/O

Notes:

1. *All unlisted pin numbers are user I/Os.*
2. *NC : Denotes No Connection*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

176-Pin TQFP (Top View)



176-Pin TQFP

Pin Number	A1440 A14V40 Function	A1460 A14V60 Function
1	GND	GND
2	SDI, I/O	SDI, I/O
10	MODE	MODE
11	VCC	VCC
20	NC	I/O
21	GND	GND
22	VCC	VCC
23	GND	GND
32	VCC	VCC
33	VCC	VCC
44	GND	GND
49	NC	I/O
51	NC	I/O
63	NC	I/O
64	PRB, I/O	PRB, I/O
65	GND	GND
66	VCC	VCC
67	VCC	VCC
69	HCLK, I/O	HCLK, I/O
82	NC	I/O
83	NC	I/O
88	IOPCL, I/O	IOPCL, I/O
89	GND	GND

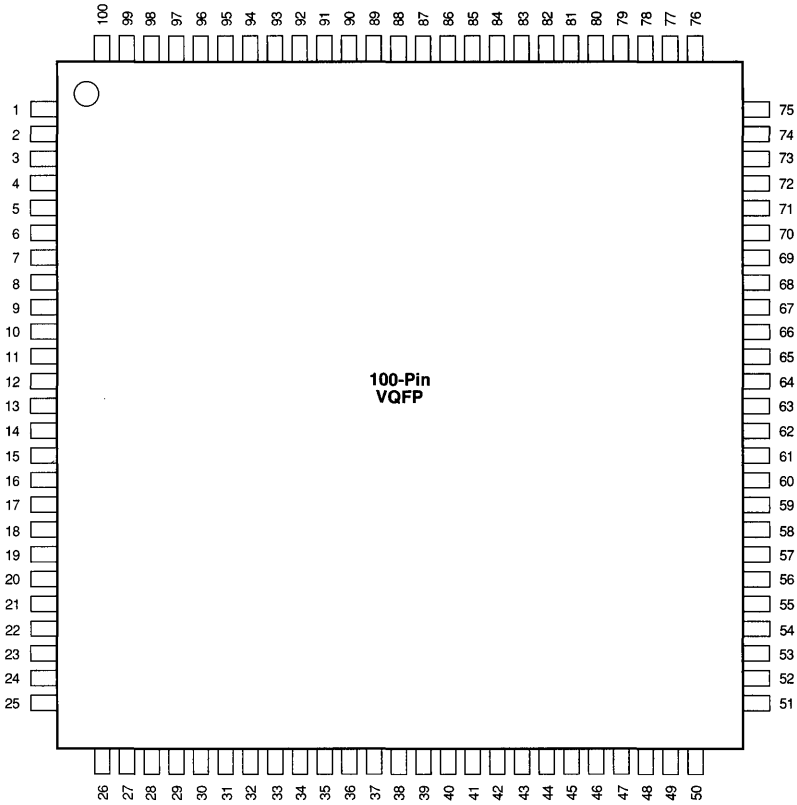
Pin Number	A1440 A14V40 Function	A1460 A14V60 Function
98	VCC	VCC
99	VCC	VCC
108	GND	GND
109	VCC	VCC
110	GND	GND
119	NC	I/O
121	NC	I/O
122	VCC	VCC
123	GND	GND
124	VCC	VCC
132	IOCLK, I/O	IOCLK, I/O
133	GND	GND
138	NC	I/O
152	CLKA, I/O	CLKA, I/O
153	CLKB, I/O	CLKB, I/O
154	VCC	VCC
155	GND	GND
156	VCC	VCC
157	PRA, I/O	PRA, I/O
158	NC	I/O
170	NC	I/O
176	DCLK, I/O	DCLK, I/O

Notes:

1. *All unlisted pin numbers are user I/Os.*
2. *NC : Denotes No Connection*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

100-Pin VQFP (Top View)



100-Pin VQFP

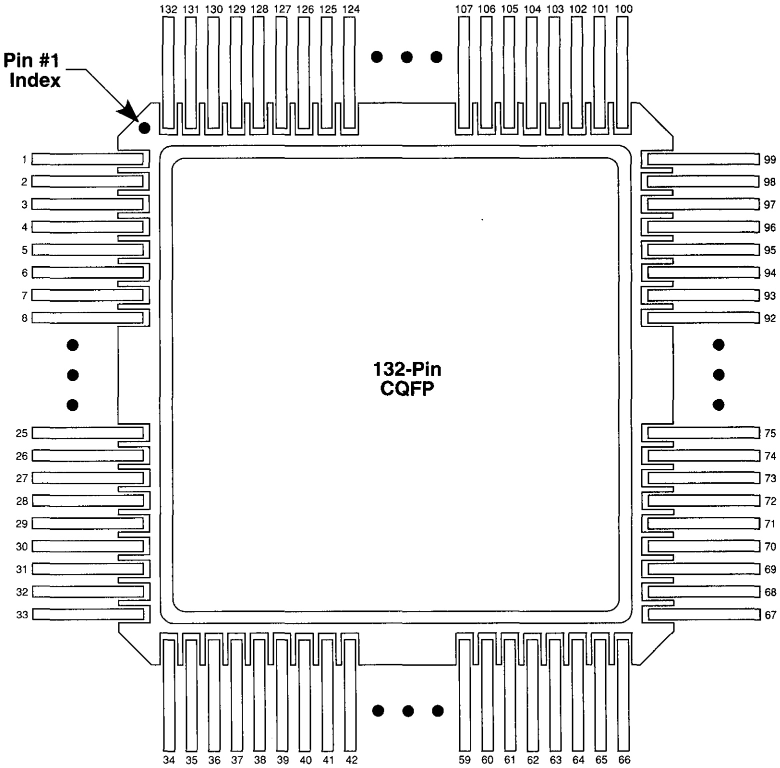
Pin Number	A1415 A14V15 Function	A1425 A14V25 Function	A1440 A14V40 Function
1	GND	GND	GND
2	SDI, I/O	SDI, I/O	SDI, I/O
7	MODE	MODE	MODE
8	VCC	VCC	VCC
9	GND	GND	GND
20	VCC	VCC	VCC
21	NC	I/O	I/O
34	PRB, I/O	PRB, I/O	PRB, I/O
35	VCC	VCC	VCC
36	GND	GND	GND
37	VCC	VCC	VCC
39	HCLK, I/O	HCLK, I/O	HCLK, I/O
50	IOPCL, I/O	IOPCL, I/O	IOPCL, I/O
51	GND	GND	GND
57	VCC	VCC	VCC
58	VCC	VCC	VCC
67	VCC	VCC	VCC
68	GND	GND	GND
69	GND	GND	GND
74	NC	I/O	I/O
75	IOCLK, I/O	IOCLK, I/O	IOCLK, I/O
87	CLKA, I/O	CLKA, I/O	CLKA, I/O
88	CLKB, I/O	CLKB, I/O	CLKB, I/O
89	VCC	VCC	VCC
90	VCC	VCC	VCC
91	GND	GND	GND
92	PRA, I/O	PRA, I/O	PRA, I/O
93	NC	I/O	I/O
100	DCLK, I/O	DCLK, I/O	DCLK, I/O

Notes:

1. *All unlisted pin numbers are user I/Os.*
2. *NC : Denotes No Connection*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Package Pin Assigments (continued)

132-Pin CQFP (Top View)



132-Pin CQFP

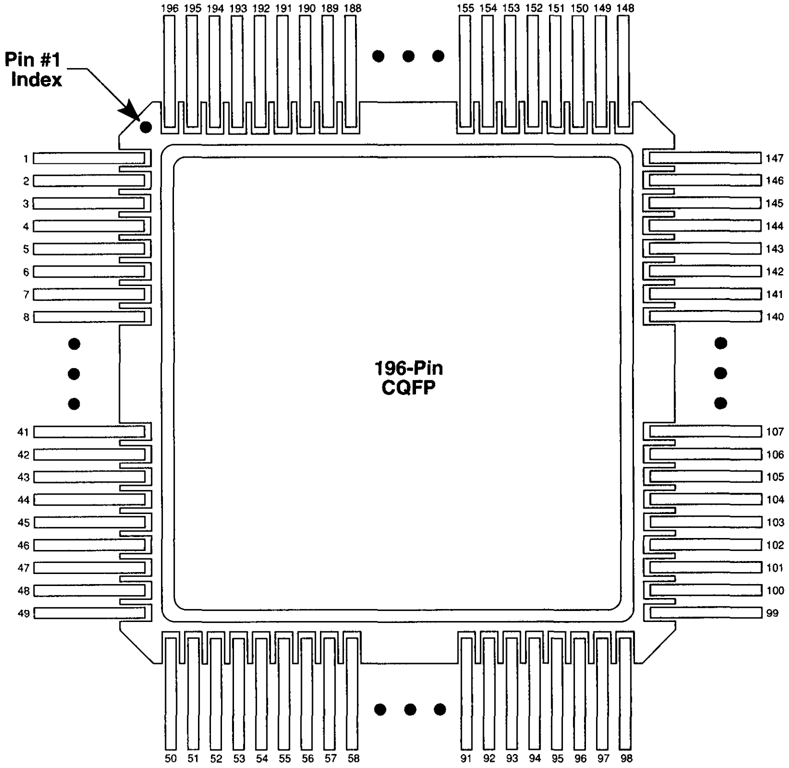
Pin Number	A1425 Function
1	NC
2	GND
3	SDI, I/O
9	MODE
10	GND
11	VCC
22	VCC
26	GND
27	VCC
34	NC
36	GND
42	GND
43	VCC
48	PRB, I/O
50	HCLK, I/O
58	GND
59	VCC
64	IOPCL, I/O
65	GND
66	NC
67	NC
74	GND
75	VCC
78	VCC
89	VCC
90	GND
91	VCC
92	GND
98	IOCLK, I/O
99	NC
100	NC
101	GND
106	GND
107	VCC
116	CLKA, I/O
117	CLKB, I/O
118	PRA, I/O
122	GND
123	VCC
131	DCLK, I/O
132	NC

Notes:

1. *All unlisted pin numbers are user I/Os.*
2. *NC: Denotes No Connection*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Package Pin Assigments (continued)

196-Pin CQFP (Top View)



196-Pin CQFP

Pin Number	A1460 Function
1	GND
2	SDI, I/O
11	MODE
12	VCC
13	GND
37	GND
38	VCC
39	VCC
51	GND
52	GND
59	VCC
64	GND
77	HCLK, I/O
79	PRB, I/O
86	GND
94	VCC
98	GND
100	IOPCL, I/O
101	GND

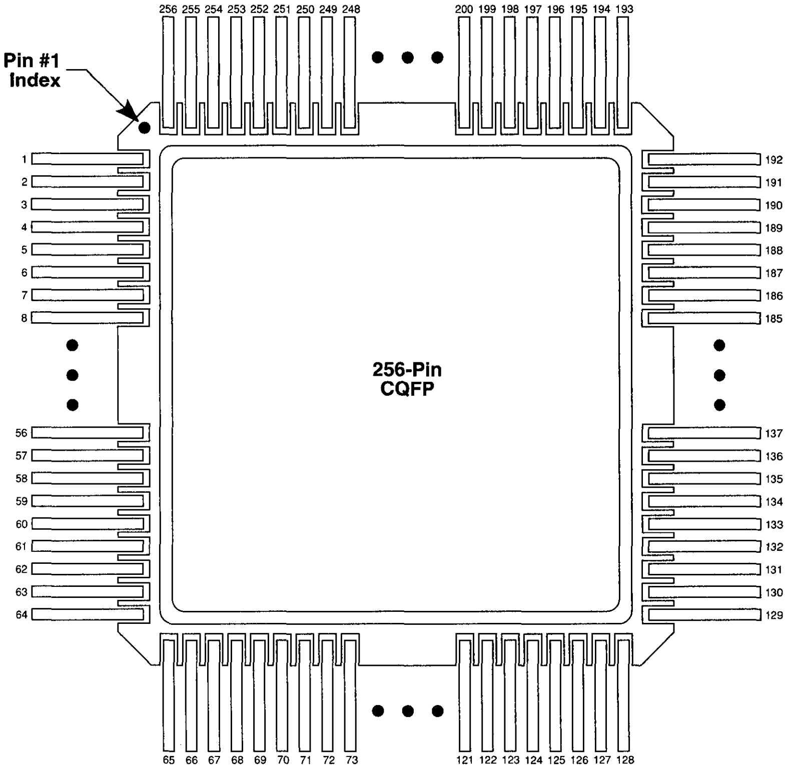
Pin Number	A1460 Function
110	VCC
111	VCC
112	GND
137	VCC
138	GND
139	GND
140	VCC
148	IOCLK, I/O
149	GND
155	VCC
162	GND
172	CLKA, I/O
173	CLKB, I/O
174	PRA, I/O
183	GND
189	VCC
193	GND
196	DCLK, I/O

Notes:

1. *All unlisted pin numbers are user I/Os.*
2. *NC : Denotes No Connection*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Package Pin Assigments (continued)

256-Pin CQFP (Top View)



256-Pin CQFP

Pin Number	A14100 Function
1	GND
2	SDI, I/O
11	MODE
28	VCC
29	GND
30	VCC
31	GND
46	VCC
59	GND
90	PRB, I/O
91	GND
92	VCC
93	GND
94	VCC
96	HCLK, I/O
110	GND
127	IOPCL, I/O
128	GND
141	VCC

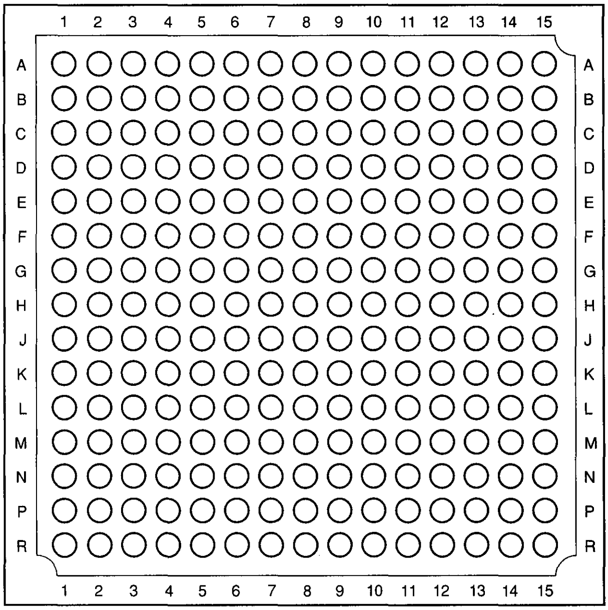
Pin Number	A14100 Function
158	GND
159	VCC
160	GND
161	VCC
174	VCC
175	GND
176	GND
188	IOCLK, I/O
189	GND
219	CLKA, I/O
220	CLKB, I/O
221	VCC
222	GND
223	VCC
224	GND
225	PRA, I/O
240	GND
256	DCLK, I/O

Notes:

1. *All unlisted pin numbers are user I/Os.*
2. *NC: Denotes No Connection*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

225-Pin BGA (Top View)



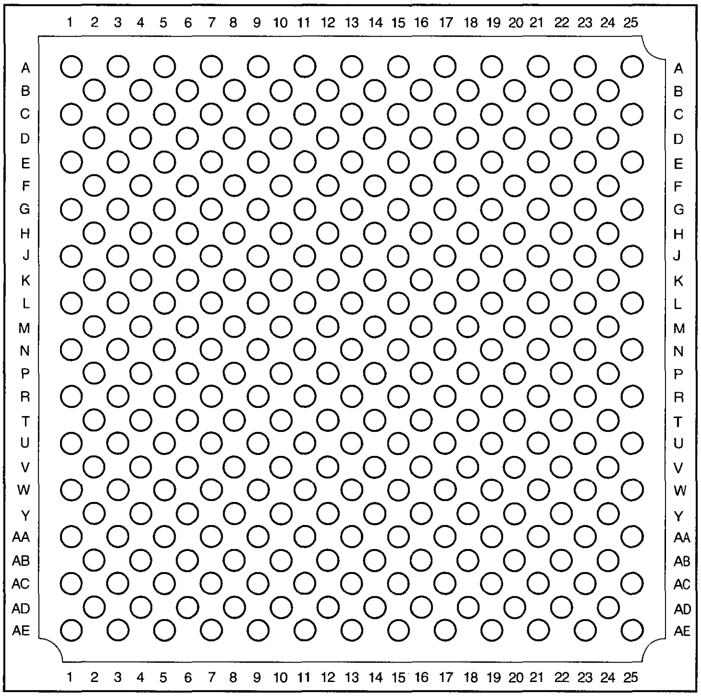
A1460 Function	Location
CLKA or I/O	C8
CLKB or I/O	B8
DCLK or I/O	B2
GND	A1, A15, D15, F8, G7, G8, G9, H6, H7, H8, H9, H10, J7, J8, J9, K8, P2, R15
HCLK or I/O	P9
IOCLK or I/O	B14
IOPCL or I/O	P14
MODE	D1
NC	A11, B5, B7, D8, D12, F6, F11, H1, H12, H14, K11, L1, L13, N8, P5, R1, R8, R11, R14
PRA OR I/O	A7
PRB or I/O	L7
SDI or I/O	D4
V_{CC}	A8, B12, D5, D14, E3, E8, E13, H2, H3, H11, H15, K4, L2, L12, M8, M15, P4, P8, R13

Notes:

1. *Unused I/O pins are designated as outputs by ALS and are driven low.*
2. *All unassigned pins are available for use as I/Os.*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

313-Pin BGA (Top View)



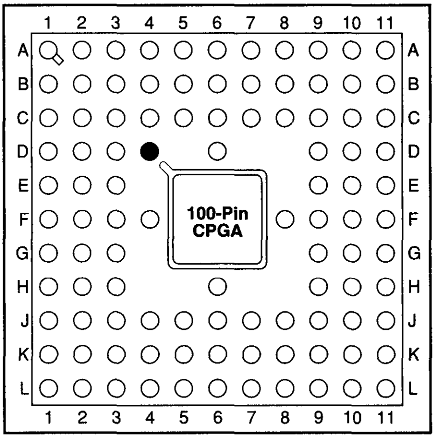
A14100 A14V100 Function	Location
CLKA or I/O	J13
CLKB or I/O	G13
DCLK or I/O	B2
GND	A1, A25, AD2, AE25, J21, L13, M12, M14, N11, N13, N15, P12, P14, R13
HCLK or I/O	T14
IOCLK or I/O	B24
IOPCL or I/O	AD24
MODE	G3
NC	A3, A13, A23, AA5, AA9, AA23, AB2, AB4, AB20, AC13, AC25, AD22, AE1, AE21, B14, C5, C25, D4, D24, E3, E21, F6, F10, F16, G1, G25, H18, H24, J1, J7, J25, K12, L15, L17, M6, N1, N5, N7, N21, N23, P20, R11, T6, T8, U9, U13, U21, V16, W7, Y20, Y24
PRA OR I/O	H12
PRB or I/O	AD12
SDI or I/O	C1
V_{CC}	AB18, AD6, AE13, C13, C19, E13, G9, H22, K8, K20, M16, N3, N9, N25, U5, W13, V2, V22, V24

Notes:

1. *Unused I/O pins are designated as outputs by ALS and are driven low.*
2. *All unassigned pins are available for use as I/Os.*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

100-Pin CPGA (Top View)



● Orientation Pin

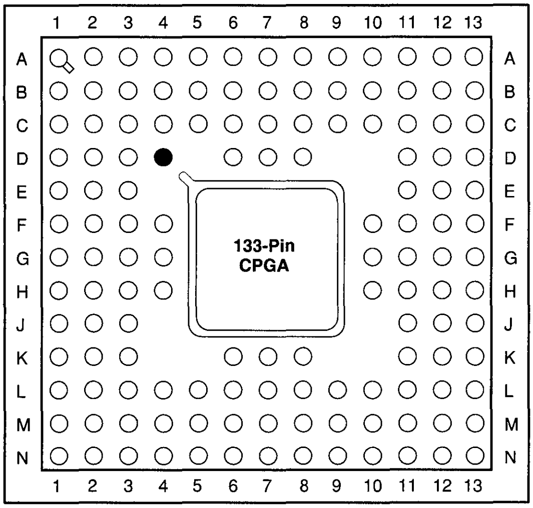
A1415 Function	Location
CLKA or I/O	C7
CLKB or I/O	D6
DCLK or I/O	C4
GND	C3, C6, C9, E9, F3, F9, J3, J6, J8, J9
HCLK or I/O	H6
IOCLK or I/O	C10
IOPCL or I/O	K9
MODE	C2
PRA OR I/O	A6
PRB or I/O	L3
SDI or I/O	B3
V_{CC}	B6, B10, E11, F2, F10, G2, K2, K6, K10

Notes:

1. *Unused I/O pins are designated as outputs by ALS and are driven low.*
2. *All unassigned pins are available for use as I/Os.*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

133-Pin CPGA (Top View)



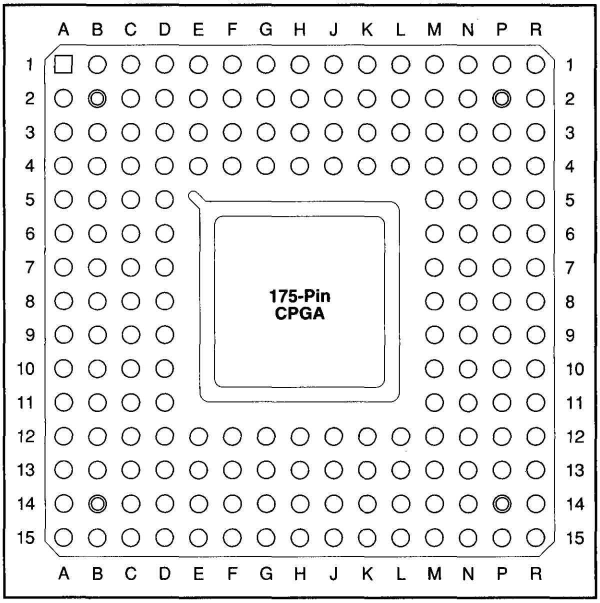
A1425 Function	Location
CLKA or I/O	D7
CLKB or I/O	B6
DCLK or I/O	D4
GND	A2, C3, C7, C11, C12, F10, G3, G11, L3, L7, L11, M3, N12
HCLK or I/O	K7
IOCLK or I/O	C10
IOPCL or I/O	L10
MODE	E3
NC	A1, A7, A13, G1, G13, N1, N7, N13
PRA OR I/O	A6
PRB or I/O	L6
SDI or I/O	C2
V_{CC}	B2, B7, B12, E11, G2, G12, J2, J12, M2, M7, M12

Notes:

1. *Unused I/O pins are designated as outputs by ALS and are driven low.*
2. *All unassigned pins are available for use as I/Os.*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

175-Pin CPGA (Top View)



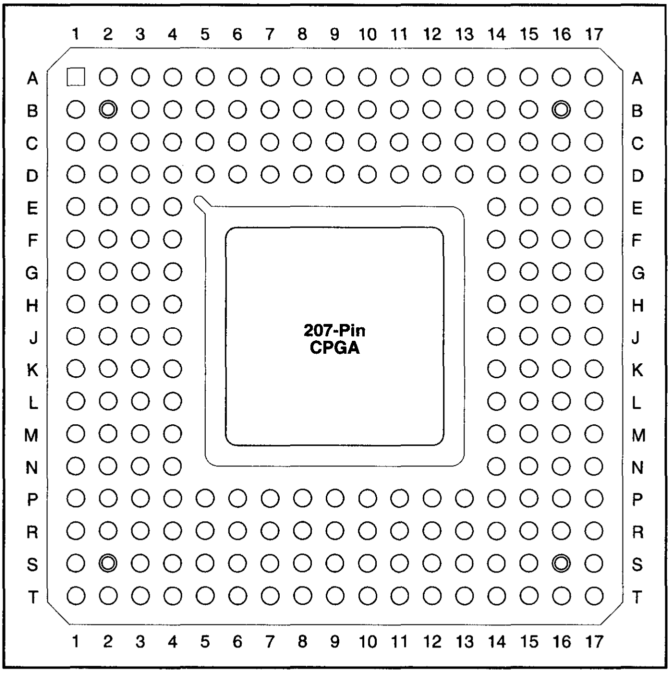
A1440 Function	Location
CLKA or I/O	C9
CLKB or I/O	A9
DCLK or I/O	D5
GND	D4, D8, D11, D12, E4, E14, H4, H12, L4, L12, M4, M8, M12
HCLK or I/O	R8
IOCLK or I/O	E12
IOPCL or I/O	P13
MODE	F3
NC	A1, A2, A15, B2, B3, P2, P14, R1, R2, R14, R15
PRA OR I/O	B8
PRB or I/O	R7
SDI or I/O	D3
V_{CC}	C3, C8, C13, E15, H3, H13, L1, L14, N3, N8, N13

Notes:

1. *Unused I/O pins are designated as outputs by ALS and are driven low.*
2. *All unassigned pins are available for use as I/Os.*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

207-Pin CPGA (Top View)



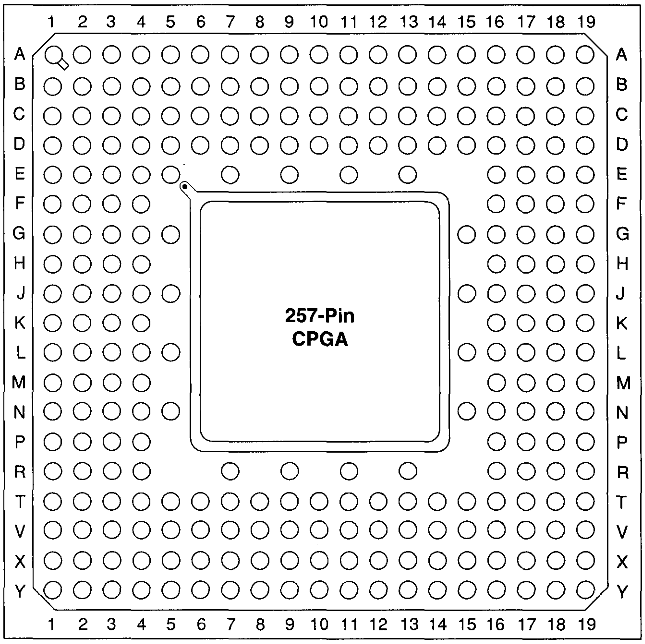
A1460 Function	Location
CLKA or I/O	K1
CLKB or I/O	J3
DCLK or I/O	E4
GND	C15, D4, D5, D9, D14, J4, J14, P3, P4, P7, P9, P14, R15
HCKL or I/O	J15
IOCLK or I/O	P5
IOPCL or I/O	N14
MODE	D7
NC	A1, A2, A16, A17, B1, B17, C1, C2, S1, S3, S17, T1, T2, T16, T17
PRA OR I/O	H1
PRB or I/O	K16
SDI or I/O	C3
V_{CC}	B2, B9, B16, D11, J2, J16, P12, S2, S9, S16, T5

Notes:

1. *Unused I/O pins are designated as outputs by ALS and are driven low.*
2. *All unassigned pins are available for use as I/Os.*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

257-Pin CPGA (Top View)



A14100 Function	Location
CLKA or I/O	L4
CLKB or I/O	L5
DCLK or I/O	E4
GND	B16, C4, D4, D10, D16, E11, J5, K4, K16, L15, R4, T4, T10, T16, T17, X7
HCLK or I/O	J16
IOCLK or I/O	T5
IOPCL or I/O	R16
MODE	A5
NC	E5
PRA OR I/O	J1
PRB or I/O	J17
SDI or I/O	B4
V_{CC}	C3, C10, C13, C17, K3, K17, V3, V7, V10, V17, X14

Notes:

1. *Unused I/O pins are designated as outputs by ALS and are driven low.*
2. *All unassigned pins are available for use as I/Os.*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Actel

Advanced Information

Accelerator Series FPGAs – ACT 3 PCI Compliant Family

PCI

Feature Set

- Up to 10,000 Gate Array equivalent gates
- Up to 250 MHz on-chip performance
- 7.5 ns Clock-to-Output
- Up to 1,153 dedicated flip-flops
- Up to 228 user-programmable I/O pins
- PCI compliant I/O drivers
- Four High-Speed, Low Skew clocks
- Highly predictable, synthesis friendly architecture supports high-level design methodologies
- 100% module utilization with automatic place and route tools
- Deterministic, user-controllable timing via DirectTime software tools

Actel enhanced the popular ACT 3 Accelerator Series of FPGAs to include PCI compliant I/O drivers. ACT 3 FPGAs are based upon Actel's proprietary PLICE antifuse technology and state-of-the-art 0.6-micron CMOS process. ACT 3 devices offer a high performance, PCI compliant programmable solution. The ACT 3 PCI Compliant family delivers 250 MHz on-chip operation and 7.5 nanosecond clock-to-output performance with capacities spanning from 4,000 to 10,000 gate array equivalent gates. The PCI compliant ACT 3 devices are denoted with a "P" designator and are shown in the chart below.

The ACT 3 PCI Compliant devices were specifically designed to be 100 percent compliant with PCI Local Bus Specification (version 2.1). Combining PCI-compliance with the industry's most synthesis friendly architecture provides the fastest PCI solution of any FPGA, regardless of whether you're designing a PCI interface from scratch or using a third-party synthesizable "core."

Actel's ACT 3 PCI Compliant devices provide a high capacity, synthesis friendly programmable solution to PCI applications. The following headings detail the pertinent PCI Local Bus Specifications along with the corresponding ACT 3 parameters. The section numbers in the notes denote the pertinent section in the PCI Local Bus Specification (version 2.1). ACT 3 devices comply 100% to the electrical and timing specifications detailed in the PCI specification. However, as with all programmable logic devices, the performance of the final product depends upon the user's design and optimization techniques. The electrical and timing specifications are specified as in version 2.1 of the PCI Specification. Used in conjunction with the PCI Local Bus Specification, ACT 3 devices can offer a cost effective, high performance PCI solution.

ACT 3 PCI-Compliant Devices

Device	A1440BP	A1460BP	A14100BP
Logic Gates	4,000	6,000	10,000
Logic Modules	564	848	1,377
Sequential Modules	288	432	697
Combinatorial Modules	276	416	680
Dedicated Flip-Flops[1]	568	768	1,153
User I/Os (maximum)	140	168	228
Packages (by pin count)			
PQFP	160	160, 208	
RQFP			208
TQFP	176	176	
BGA		225	313

Note:

1. *One flip-flop/S-Module, two flip-flops/IO-Module*

Electrical Specifications

The PCI bus specifies I/O drivers in terms of the DC and AC characteristics. However, since the PCI bus drivers spend a relatively large proportion of time transitioning from one power rail to the other, PCI drivers are primarily characterized by their V/I curves (Tables 1 and 2).

Output Drive Characteristics for 5.0 V Signaling

ACT 3 PCI device I/O drivers were designed specifically for high-performance PCI systems. Figures 1 and 2 show the typical output drive characteristics of the 5.0 V ACT 3 devices. ACT 3 output drivers are compliant with the PCI Local Bus Specification.

Table 1 • *DC Specification for 5.0V Signaling*[1]

			PCI			ACT 3	
Symbol	**Parameter**	**Condition**	**Minimum**	**Maximum**	**Units**	**Minimum**	**Maximum**
V_{CC}	Supply Voltage		4.75	5.25	V	4.75	5.25[2]
V_{IH}	Input High Voltage		2	Vcc + 0.5	V	2	Vcc + 0.5
V_{IL}	Input Low Voltage		–0.5	0.8	V	–0.3	0.8
I_{IH}	Input High Current	Vin = 2.7		70	uA	—	10
I_{IL}	Input Low Current	Vin=0.5		–70	uA	—	–10
V_{OH}	Output High Voltage	Iout = –2 mA	2.4		V	3.7	
V_{OL}	Output Low Voltage	Iout = 3 mA, 6 mA		0.55	V	—	0.33
C_{IN}	Input pin capacitance			10	pF	—	10
C_{CLK}	CLK pin capacitance		5	12	pF	—	10
L_{PIN}	Pin inductance			20	nH	—	< 8 nH[3]

Notes:

1. *PCI Local Bus Specification section 4.2.1.1.*
2. *Maximum rating for Vcc: –0.5V to 7.0V. Refer to Accelerator Series FPGAs ACT 3 Family data sheet.*
3. *Dependent upon the chosen package. PCI recommends QFP packaging to reduce pin inductance and capacitance.*

Table 2 • *AC Specifications for 5.0V Signaling*[1]

			PCI			ACT 3	
Symbol	**Parameter**	**Condition**	**Minimum**	**Maximum**	**Units**	**Minimum**	**Maximum**
I_{CL}	Low Clamp Current	–5 < Vin ≤ –1	–25 + (Vin +1) /0.015		mA	–50	–10
Slew (r)	Output rise slew rate	0.4V to 2.4V load	1	5	V/ns	1.8	2.8
Slew (f)	Output fall slew rate	2.4V to 0.4V load	1	5	V/ns	2.8	4.3

Note:

1. *PCI Local Bus Specification section 4.2.1.2.*

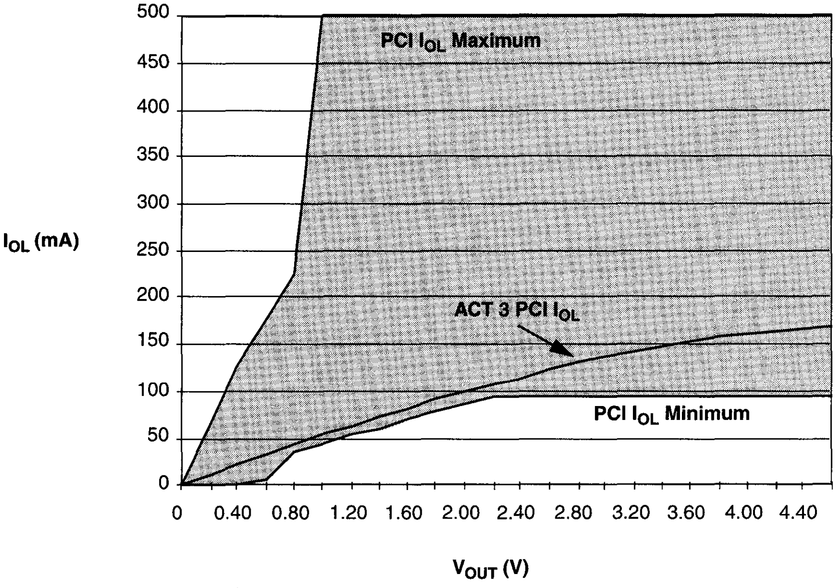


Figure 1 • *Typical Output Drive Characteristics (Based upon simulation data)*

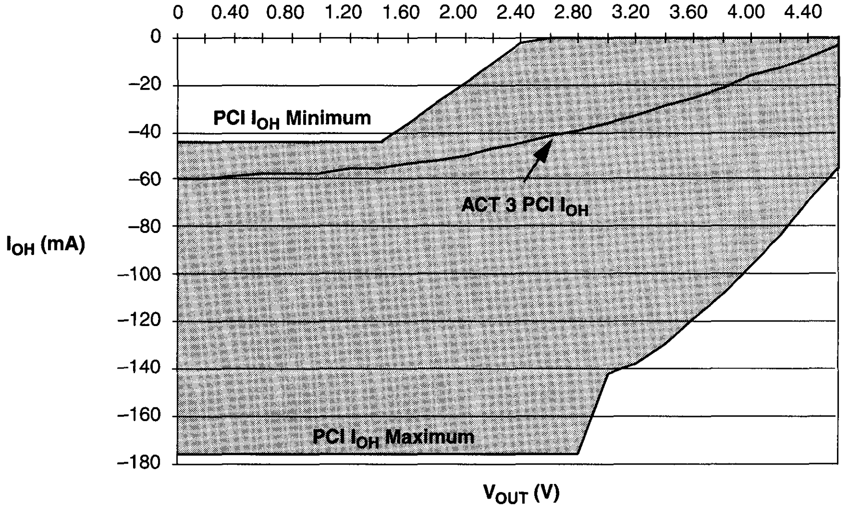


Figure 2 • *Typical Output Drive Characteristics (Based upon simulation data)*

System Timing Specification

The following heading lists the critical PCI timing parameters and the corresponding timing parameter for the ACT 3 PCI compliant devices. See Tables 3 and 4.

PCI Models

Actel will provide synthesizable VHDL and Verilog models for a PCI target interface, a PCI master interface and a PCI-PCI bridge interface. Consult your local Actel sales representative for more details.

Table 3 • *Clock Specification for 33 MHz PCI*[1]

		PCI			ACT 3	
Symbol	**Parameter**	**Minimum**	**Maximum**	**Units**	**Minimum**	**Maximum**
T_{CYC}	CLK Cycle Time	30	—	ns	4.0	—
T_{HIGH}	CLK High Time	11	—	ns	1.9	—
T_{LOW}	CLK Low Time	11	—	ns	1.9	—
—	CLK Skew	1	4	V/ns	—	4.0

Note:

[1] *PCI Local Bus Specification Section 4.2.3.1.*

Table 4 • *Timing Parameters for 33 MHz PCI*[1]

		PCI			ACT 3	
Symbol	**Parameter**	**Minimum**	**Maximum**	**Units**	**Minimum**	**Maximum**
T_{VAL}	CLK to Signal Valid—bussed Signals	2	11	ns	2.0	7.5
$T_{VAL(PTP)}$	CLK to Signal Valid—point to point	2	12	ns	2.0	7.5
T_{ON}	Float to active	2		ns	2.0	4.0
T_{OFF}	Active to Float		28	ns	—	7.4[2]
T_{SU}	Input Setup time to CLK—bussed signals	7		ns	1.5	—
$T_{SU(PTP)}$	Input Setup time to CLK—point to point	10, 12		ns	1.5	—
T_H	Input Hold to CLK	0		ns	—	0

Notes:

[1] *Based upon simulation data for internal timing parameters. PCI Local Bus Specification Section 4.2.3.2.*

[2] *Toff is system dependent. ACT 3 PCI devices have 7.4 ns turn-off time, reflection is typically an additional 10 ns.*

ACT™ 1 Series FPGAs

Features

- 5V and 3.3V Families fully compatible with JEDEC specifications
- Up to 2000 Gate Array Gates (6000 PLD equivalent gates)
- Replaces up to 50 TTL Packages
- Replaces up to twenty 20-Pin PAL® Packages
- Design Library with over 250 Macro Functions
- Gate Array Architecture Allows Completely Automatic Place and Route
- Up to 547 Programmable Logic Modules
- Up to 273 Flip-Flops
- Data Rates to 75 MHz
- Two In-Circuit Diagnostic Probe Pins Support Speed Analysis to 25 MHz
- Built-In High Speed Clock Distribution Network
- I/O Drive to 10 mA (5 V), 6 mA (3.3 V)
- Nonvolatile, User Programmable
- Fabricated in 1.0 micron CMOS technology

Description

The ACT™ 1 Series of field programmable gate arrays (FPGAs) offers a variety of package, speed, and application combinations. Devices are implemented in silicon gate, 1-micron two-level metal CMOS, and they employ Actel's PLICE® antifuse technology. The unique architecture offers gate array flexibility, high performance, and instant turnaround through user programming. Device utilization is typically 95 to 100 percent of available logic modules.

ACT 1 devices also provide system designers with unique on-chip diagnostic probe capabilities, allowing convenient testing and debugging. Additional features include an on-chip clock driver with a hardwired distribution network. The network provides efficient clock distribution with minimum skew.

The user-definable I/Os are capable of driving at both TTL and CMOS drive levels. Available packages include plastic and ceramic J-leaded chip carriers, ceramic and plastic quad flatpacks, and ceramic pin grid array.

A security fuse may be programmed to disable all further programming and to protect the design from being copied or reverse engineered.

Product Family Profile

Device	A1010B A10V10B	A1020B A10V20B
Capacity		
Gate Array Equivalent Gates	1,200	2,000
PLD Equivalent Gates	3,000	6,000
TTL Equivalent Packages	30	50
20-Pin PAL Equivalent Packages	12	20
Logic Modules	295	547
Flip-Flops (maximum)	147	273
Routing Resources		
Horizontal Tracks/Channel	22	22
Vertical Tracks/Column	13	13
PLICE Antifuse Elements	112,000	186,000
User I/Os (maximum)	57	69
Packages:	44 PLCC	44 PLCC
	68 PLCC	68 PLCC
		84 PLCC
	100 PQFP	100 PQFP
	80 VQFP	80 VQFP
	84 CPGA	84 CPGA
		84 CQFP
Performance		
5 V Data Rate (maximum)	75 MHz	75 MHz
3.3 V Data Rate (maximum)	55 MHz	55 MHz

Note: See Product Plan on page 1-148 for package availability.

The Designer and Designer Advantage™ Systems

The ACT 1 device family is supported by Actel's Designer and Designer Advantage Systems, allowing logic design implementation with minimum effort. The systems offer Microsoft® Windows™ and X Windows™ graphical user interfaces and integrate with the resident CAE system to provide a complete gate array design environment: schematic capture, simulation, fully automatic place and route, timing verification, and device programming. The systems also include the ACTmap™ VHDL optimization and synthesis tool and the ACTgen™ Macro Builder, a powerful macro function generator for counters, adders, and other structural blocks.

The systems are available for 386/486/Pentium™ PC and for HP™ and Sun™ workstations and for running Viewlogic®, Mentor Graphics®, Cadence™, OrCAD™, and Synopsys design environments.

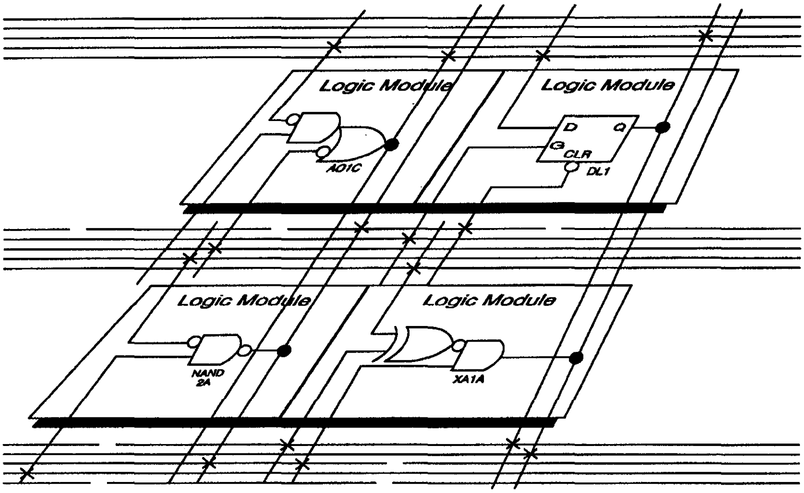


Figure 1 • Partial View of an ACT 1 Device

ACT 1 Device Structure

A partial view of an ACT 1 device (Figure 1) depicts four logic modules and distributed horizontal and vertical interconnect tracks. PLICE antifuses, located at intersections of the horizontal and vertical tracks, connect logic module inputs and outputs. During programming, these antifuses are addressed and programmed to make the connections required by the circuit application.

The ACT 1 Logic Module

The ACT 1 logic module is an 8-input, one-output logic circuit chosen for the wide range of functions it implements and for its efficient use of interconnect routing resources (Figure 2).

The logic module can implement the four basic logic functions (NAND, AND, OR, and NOR) in gates of two, three, or four inputs. Each function may have many versions, with different combinations of active-low inputs. The logic module can also implement a variety of D-latches, exclusivity functions, AND-ORs, and OR-ANDs. No dedicated hardwired latches or flip-flops are required in the array, since latches and flip-flops may be constructed from logic modules wherever needed in the application.

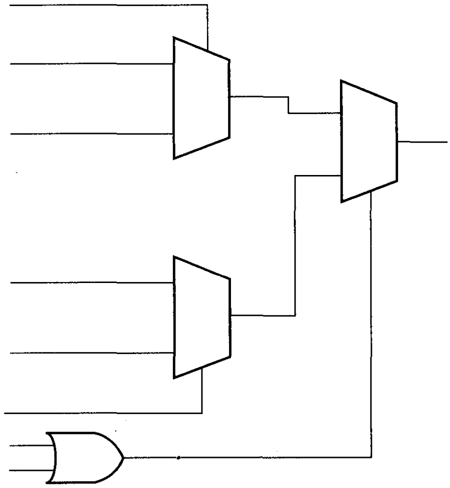


Figure 2 • ACT 1 Logic Module

I/O Buffers

Each I/O pin is available as an input, output, three-state, or bidirectional buffer. Input and output levels are compatible with standard TTL and CMOS specifications. Outputs sink or source 10 mA at TTL levels. See Electrical Specifications for additional I/O buffer specifications.

Device Organization

ACT 1 devices consist of a matrix of logic modules arranged in rows separated by wiring channels. This array is surrounded by a ring of peripheral circuits including I/O buffers, testability circuits, and diagnostic probe circuits providing real-time diagnostic capability. Between rows of logic modules are routing channels containing sets of segmented metal tracks with PLICE antifuses. Each channel has 22 signal tracks. Vertical routing is permitted via 13 vertical tracks per logic module column. The resulting network allows arbitrary and flexible interconnections between logic modules and I/O modules.

Probe Pin

ACT 1 devices have two independent diagnostic probe pins. These pins allow the user to observe any two internal signals by entering the appropriate net name in the diagnostic software. Signals may be viewed on a logic analyzer using Actel's Actionprobe® diagnostic tools. The probe pins can also be used as user-defined I/Os when debugging is finished.

ACT 1 Array Performance

Temperature and Voltage Effects

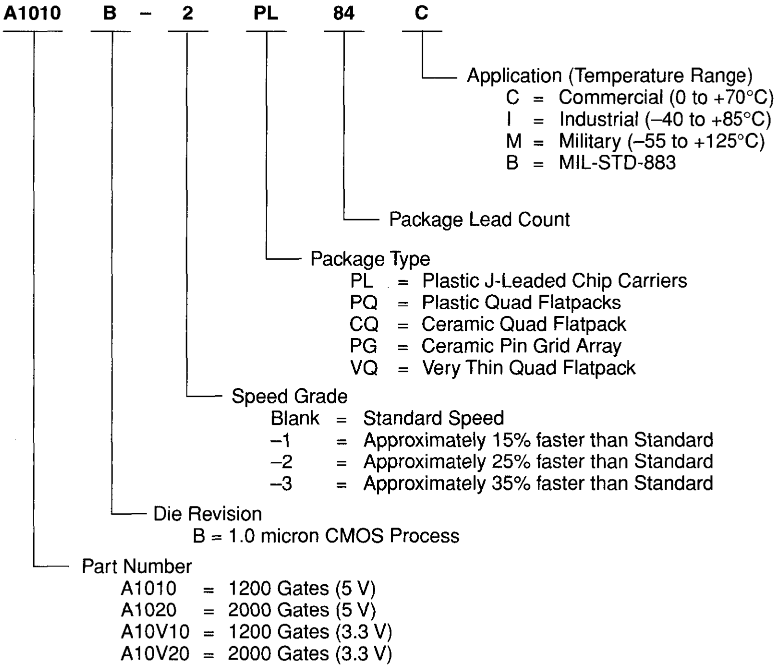
Worst-case delays for ACT 1 arrays are calculated in the same manner as for masked array products. A typical delay parameter is multiplied by a derating factor to account for temperature, voltage, and processing effects. However, in an ACT 1 array, temperature and voltage effects are less dramatic than with masked devices. The electrical characteristics of module interconnections on ACT 1 devices remain constant over voltage and temperature fluctuations.

As a result, the total derating factor from typical to worst-case for a standard speed ACT 1 array is only 1.19 to 1, compared to 2 to 1 for a masked gate array.

Logic Module Size

Logic module size also affects performance. A mask programmed gate array cell with four transistors usually implements only one logic level. In the more complex logic module (similar to the complexity of a gate array macro) of an ACT 1 array, implementation of multiple logic levels within a single module is possible. This eliminates interlevel wiring and associated RC delays. The effect is termed "net compression."

Ordering Information



A1010 B – 2 PL 84 C

Application (Temperature Range)
C = Commercial (0 to +70°C)
I = Industrial (–40 to +85°C)
M = Military (–55 to +125°C)
B = MIL-STD-883

Package Lead Count

Package Type
PL = Plastic J-Leaded Chip Carriers
PQ = Plastic Quad Flatpacks
CQ = Ceramic Quad Flatpack
PG = Ceramic Pin Grid Array
VQ = Very Thin Quad Flatpack

Speed Grade
Blank = Standard Speed
–1 = Approximately 15% faster than Standard
–2 = Approximately 25% faster than Standard
–3 = Approximately 35% faster than Standard

Die Revision
B = 1.0 micron CMOS Process

Part Number
A1010 = 1200 Gates (5 V)
A1020 = 2000 Gates (5 V)
A10V10 = 1200 Gates (3.3 V)
A10V20 = 2000 Gates (3.3 V)

Product Plan

	Speed Grade*				Application			
	Std	–1	–2	–3	C	I	M	B
A1010B Device								
44-pin Plastic Leaded Chip Carrier (PL)	✔	✔	✔	✔	✔	✔	—	—
68-pin Plastic Leaded Chip Carrier (PL)	✔	✔	✔	✔	✔	✔	—	—
100-pin Plastic Quad Flatpack (PQ)	✔	✔	✔	✔	✔	✔	—	—
80-pin Very Thin (1.0 mm) Quad Flatpack (VQ)	✔	✔	✔	✔	✔	—	—	—
84-pin Ceramic Pin Grid Array (PG)	✔	✔	—	—	✔	—	✔	✔
A1020B Device								
44-pin Plastic Leaded Chip Carrier (PL)	✔	✔	✔	✔	✔	✔	—	—
68-pin Plastic Leaded Chip Carrier (PL)	✔	✔	✔	✔	✔	✔	—	—
84-pin Plastic Leaded Chip Carrier (PL)	✔	✔	✔	✔	✔	✔	—	—
100-pin Plastic Quad Flatpack (PQ)	✔	✔	✔	✔	✔	✔	—	—
80-pin Very Thin (1.0 mm) Quad Flatpack (VQ)	✔	✔	✔	✔	✔	—	—	—
84-pin Ceramic Pin Grid Array (PG)	✔	✔	—	—	✔	—	✔	✔
84-pin Ceramic Quad Flatpack (CQ)	✔	✔	—	—	✔	—	✔	✔
A10V10B Device								
68-pin Plastic Leaded Chip Carrier (PL)	✔	—	—	—	✔	—	—	—
80-pin Very Thin (1.0 mm) Quad Flatpack (VQ)	✔	—	—	—	✔	—	—	—
A10V20B Device								
68-pin Plastic Leaded Chip Carrier (PL)	✔	—	—	—	✔	—	—	—
84-pin Plastic Leaded Chip Carrier (PL)	✔	—	—	—	✔	—	—	—
80-pin Very Thin (1.0 mm) Quad Flatpack (VQ)	✔	—	—	—	✔	—	—	—

Applications:			Availability:		* Speed Grade:	
C	=	Commercial	✔ =	Available	–1 =	Approx. 15% faster than Standard
I	=	Industrial	P =	Planned	–2 =	Approx. 25% faster than Standard
M	=	Military	— =	Not Planned	–3 =	Approx. 35% faster than Standard
B	=	MIL-STD-883				

Device Resources

			User I/Os				
Device	Logic Modules	Gates	44-pin	68-pin	80-pin	84-pin	100-pin
A1010B, A10V10B	295	1200	34	57	57	57	57
A1020B, A10V20B	547	2000	34	57	69	69	69

Pin Description

CLK **Clock (Input)**

TTL Clock input for global clock distribution network. The Clock input is buffered prior to clocking the logic modules. This pin can also be used as an I/O.

DCLK **Diagnostic Clock (Input)**

TTL Clock input for diagnostic probe and device programming. DCLK is active when the MODE pin is HIGH. This pin functions as an I/O when the MODE pin is LOW.

GND **Ground**

Input LOW supply voltage.

I/O **Input/Output (Input, Output)**

I/O pin functions as an input, output, three-state, or bidirectional buffer. Input and output levels are compatible with standard TTL and CMOS specifications. Unused I/O pins are automatically driven LOW by the ALS software.

MODE **Mode (Input)**

The MODE pin controls the use of multifunction pins (DCLK, PRA, PRB, SDI). When the MODE pin is HIGH, the special functions are active. When the MODE pin is LOW, the pins function as I/O. To provide Actionprobe capability, the MODE pin should be terminated to GND through a 10K resistor so that the MODE pin can be pulled high when required.

NC **No Connection**

This pin is not connected to circuitry within the device.

PRA **Probe A (Output)**

The Probe A pin is used to output data from any user-defined design node within the device. This independent diagnostic pin is used in conjunction with the Probe B pin to allow real-time diagnostic output of any signal path within the device. The Probe A pin can be used as a user-defined I/O when debugging has been completed. The pin's probe capabilities can be permanently disabled to protect the programmed design's confidentiality. PRA is active when the MODE pin is HIGH. This pin functions as an I/O when the MODE pin is LOW.

PRB **Probe B (Output)**

The Probe B pin is used to output data from any user-defined design node within the device. This independent diagnostic pin is used in conjunction with the Probe A pin to allow real-time diagnostic output of any signal path within the device. The Probe B pin can be used as a user-defined I/O when debugging has been completed. The pin's probe capabilities can be permanently disabled to protect the programmed design's confidentiality. PRB is active when the MODE pin is HIGH. This pin functions as an I/O when the MODE pin is LOW.

SDI **Serial Data Input (Input)**

Serial data input for diagnostic probe and device programming. SDI is active when the MODE pin is HIGH. This pin functions as an I/O when the MODE pin is LOW.

V_{CC} **Supply Voltage**

Input HIGH supply voltage.

Absolute Maximum Ratings[1]

Free air temperature range

Symbol	Parameter	Limits	Units
V_{CC}	DC Supply Voltage[2]	–0.5 to +7.0	Volts
V_I	Input Voltage	–0.5 to V_{CC} +0.5	Volts
V_O	Output Voltage	–0.5 to V_{CC} +0.5	Volts
I_{IO}	I/O Sink/Source Current[3]	±20	mA
T_{STG}	Storage Temperature	–65 to +150	°C

Notes:

1. *Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum rated conditions for extended periods may affect device reliability. Device should not be operated outside the Recommended Operating Conditions.*
2. *$V_{PP} = V_{CC}$, except during device programming.*
3. *Device inputs are normally high impedance and draw extremely low current. However, when input voltage is greater than V_{CC} + 0.5 V or less than GND – 0.5 V, the internal protection diode will be forward biased and can draw excessive current.*

Recommended Operating Conditions

Parameter	Commercial	Industrial	Military	Units
Temperature Range[1]	0 to +70	–40 to +85	–55 to +125	°C
Power Supply Tolerance	±5	±10	±10	%V_{CC}

Note:

1. *Ambient temperature (T_A) used for commercial and industrial; case temperature (T_C) used for military.*

Electrical Specifications (5V)

Symbol	Parameter	Commercial		Industrial		Military		Units
		Min.	Max.	Min.	Max.	Min.	Max.	
V_{OH}[1]	(I_{OH} = –10 mA)[2]	2.4						V
	(I_{OH} = –6 mA)	3.84						V
	(I_{OH} = –4 mA)			3.7		3.7		V
V_{OL}[1]	(I_{OL} = 10 mA)[2]		0.5					V
	(I_{OL} = 6 mA)		0.33		0.40		0.40	V
V_{IL}		–0.3	0.8	–0.3	0.8	–0.3	0.8	V
V_{IH}		2.0	V_{CC} + 0.3	2.0	V_{CC} + 0.3	2.0	V_{CC} + 0.3	V
Input Transition Time t_R, t_F[2]			500		500		500	ns
C_{IO} I/O Capacitance[2, 3]			10		10		10	pF
Standby Current, I_{CC}[4] (typical = 1 mA)			3		10		20	mA
Leakage Current[5]		–10	10	–10	10	–10	10	µA

Notes:

1. *Only one output tested at a time. V_{CC} = min.*
2. *Not tested, for information only.*
3. *Includes worst-case 84-pin PLCC package capacitance. V_{OUT} = 0 V, f = 1 MHz.*
4. *Typical standby current = 1 mA. All outputs unloaded. All inputs = V_{CC} or GND.*
5. *V_O, V_{IN} = V_{CC} or GND.*

Electrical Specifications (3.3V)

Parameter		Commercial		Units
		Min.	Max.	
V_{OH}[1]	(I_{OH} = –4 mA)	2.15		V
	(I_{OH} = –3.2 mA)	2.4		V
V_{OL}[1]	(I_{OL} = 6 mA)		0.4	V
V_{IL}		–0.3	0.8	V
V_{IH}		2.0	V_{CC} + 0.3	V
Input Transition Time t_R, t_F[2]			500	ns
C_{IO} I/O Capacitance[2, 3]			10	pF
Standby Current, I_{CC}[4] (typical = 0.3 mA)			0.75	mA
Leakage Current[5]		–10	10	µA

Notes:

1. *Only one output tested at a time. V_{CC} = min.*
2. *Not tested, for information only.*
3. *Includes worst-case 84-pin PLCC package capacitance. V_{OUT} = 0 V, f = 1 MHz.*
4. *Typical standby current = 0.3 mA. All outputs unloaded. All inputs = V_{CC} or GND.*
5. *V_O, V_{IN} = V_{CC} or GND*

Package Thermal Characteristics

The device junction to case thermal characteristics is θjc, and the junction to ambient air characteristics is θja. The thermal characteristics for θja are shown with two different air flow rates. Maximum junction temperature is 150°C.

A sample calculation of the maximum power dissipation for an 84-pin plastic leaded chip carrier at commercial temperature is as follows:

$$\frac{\text{Max junction temp.}(°\text{C}) - \text{Max commercial temp.}(°\text{C})}{\theta\text{ja}(°\text{C/W})} = \frac{150°\text{C} - 70°\text{C}}{37°\text{C/W}} = 2.2\text{ W}$$

Package Type	Pin Count	θjc	θja Still Air	θja 300 ft/min	Units
Plastic J-Leaded Chip Carrier	44	15	45	35	°C/W
	68	13	38	29	°C/W
	84	12	37	28	°C/W
Plastic Quad Flatpack	100	13	48	40	°C/W
Very Thin (1.0 mm) Quad Flatpack	80	12	43	35	°C/W
Ceramic Pin Grid Array	84	8	33	20	°C/W
Ceramic Quad Flatpack	84	5	40	30	°C/W

General Power Equation

$P = [I_{CC}\text{standby} + I_{CC}\text{active}] * V_{CC} + I_{OL} * V_{OL} * N + I_{OH} * (V_{CC} - V_{OH}) * M$

Where:

I_{CC}standby is the current flowing when no inputs or outputs are changing.

I_{CC}active is the current flowing due to CMOS switching.

I_{OL}, I_{OH} are TTL sink/source currents.

V_{OL}, V_{OH} are TTL level output voltages.

N equals the number of outputs driving TTL loads to V_{OL}.

M equals the number of outputs driving TTL loads to V_{OH}.

An accurate determination of N and M is problematical because their values depend on the family type, design details, and on the system I/O. The power can be divided into two components: static and active.

Static Power Component

Actel FPGAs have small static power components that result in lower power dissipation than PALs or PLDs. By integrating multiple PALs/PLDs into one FPGA, an even greater reduction in board-level power dissipation can be achieved.

The power due to standby current is typically a small component of the overall power. Standby power is calculated below for commercial, worst case conditions.

I_{CC}	V_{CC}	Power
3 mA	5.25 V	15.75 mW (max)
1 mA	5.25 V	5.25 mW (typ)
0.75 mA	3.60 V	2.70 mW (max)
0.30 mA	3.30 V	0.99 mW (typ)

Active Power Component

Power dissipation in CMOS devices is usually dominated by the active (dynamic) power dissipation. This component is frequency dependent, a function of the logic and the

external I/O. Active power dissipation results from charging internal chip capacitances of the interconnect, unprogrammed antifuses, module inputs, and module outputs, plus external capacitance due to PC board traces and load device inputs. An additional component of the active power dissipation is the totem-pole current in CMOS transistor pairs. The net effect can be associated with an equivalent capacitance that can be combined with frequency and voltage to represent active power dissipation.

Equivalent Capacitance

The power dissipated by a CMOS circuit can be expressed by the Equation 1.

$$\text{Power (uW)} = C_{EQ} * V_{CC2} * F \qquad (1)$$

Where:

C_{EQ} is the equivalent capacitance expressed in pF.

V_{CC} is the power supply in volts.

F is the switching frequency in MHz.

Equivalent capacitance is calculated by measuring I_{CC}active at a specified frequency and voltage for each circuit component of interest. Measurements have been made over a range of frequencies at a fixed value of V_{CC}. Equivalent capacitance is frequency independent so that the results may be used over a wide range of operating conditions. Equivalent capacitance values are shown below.

C_{EQ} Values for Actel FPGAs

	A10V10B A10V20B	A1010B A1020B
Modules (C_{EQM})	3.2	3.7
Input Buffers ($_{CEQI}$)	10.9	22.1
Output Buffers (C_{EQO})	11.6	31.2
Routed Array Clock Buffer Loads (C_{EQCR})	4.1	4.6

To calculate the active power dissipated from the complete design, the switching frequency of each part of the logic must be known. Equation 2 shows a piece-wise linear summation over all components.

$$\text{Power} = V_{CC}{}^{2} * [(m * C_{EQM} * f_m)_{\text{modules}} + (n * C_{EQI} * f_n)_{\text{inputs}} + (p * (C_{EQO} + C_L) * f_p)_{\text{outputs}} + 0.5 * (q_1 * C_{EQCR} * f_{q1})_{\text{routed\_Clk1}} + (r_1 * f_{q1})_{\text{routed\_Clk1}}] \qquad (2)$$

Where:

m = Number of logic modules switching at fm

n = Number of input buffers switching at fn

p = Number of output buffers switching at fp

q_1 = Number of clock loads on the first routed array clock (All families)

r_1 = Fixed capacitance due to first routed array clock (All families)

C_{EQM} = Equivalent capacitance of logic modules in pF

C_{EQI} = Equivalent capacitance of input buffers in pF

C_{EQO} = Equivalent capacitance of output buffers in pF

C_{EQCR} = Equivalent capacitance of routed array clock in pF

C_L = Output lead capacitance in pF

f_m = Average logic module switching rate in MHz

f_n = Average input buffer switching rate in MHz

f_p = Average output buffer switching rate in MHz

f_{q1} = Average first routed array clock rate in MHz (All families)

Fixed Capacitance Values for Actel FPGAs (pF)

Device Type	r_1 routed_Clk1
A1010B	41.4
A1020B	68.6
A10V10B	40
A10V20B	65

Determining Average Switching Frequency

To determine the switching frequency for a design, you must have a detailed understanding of the data input values to the circuit. The following guidelines are meant to represent worst-case scenarios so that they can be generally used to predict the upper limits of power dissipation. These guidelines are as follows:

Logic Modules (m)	90% of modules
Inputs switching (n)	#inputs/4
Outputs switching (p)	#outputs/4
First routed array clock loads (q_1)	40% of modules
Load capacitance (C_L)	35 pF
Average logic module switching rate (f_m)	F/10
Average input switching rate (f_n)	F/5
Average output switching rate (f_p)	F/10
Average first routed array clock rate (f_{q1})	F

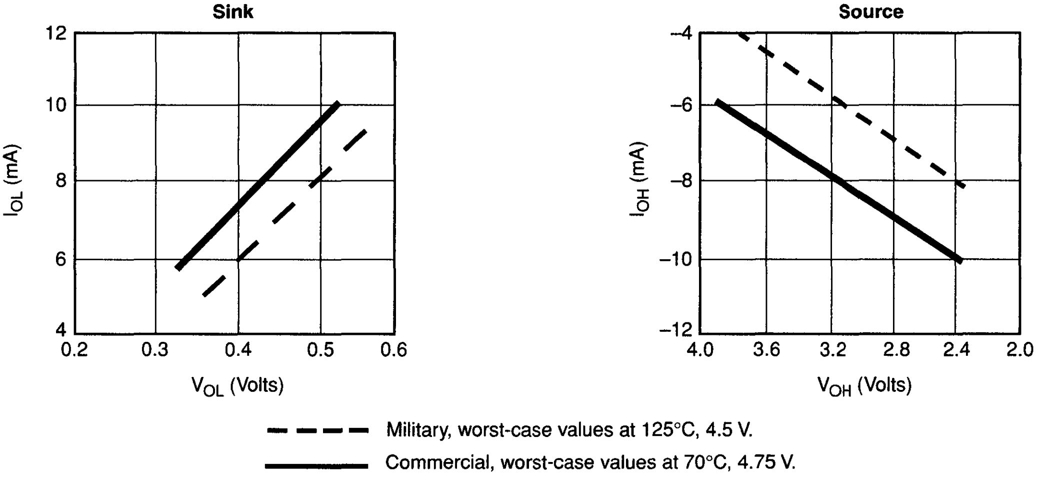
Functional Timing Tests

AC timing for logic module internal delays is determined after place and route. The DirectTime Analyzer utility displays actual timing parameters for circuit delays. ACT 1 devices are AC tested to a "binning" circuit specification.

The circuit consists of one input buffer + n logic modules + one output buffer (n = 16 for A1010B; n = 28 for A1020B). The logic modules are distributed along two sides of the device, as inverting or non-inverting buffers. The modules are connected through programmed antifuses with typical capacitive loading.

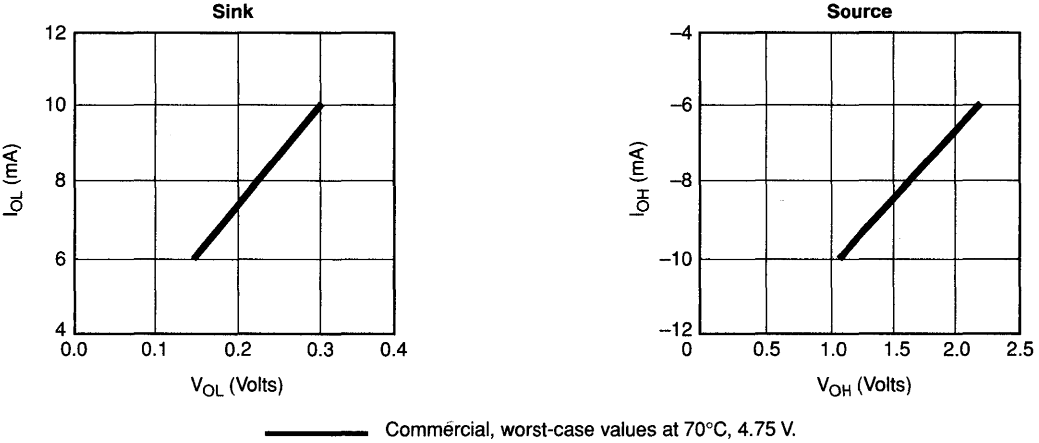
Propagation delay [$t_{PD} = (t_{PLH} + t_{PHL})/2$] is tested to the following AC test specifications.

Output Buffer Performance Derating (5V)



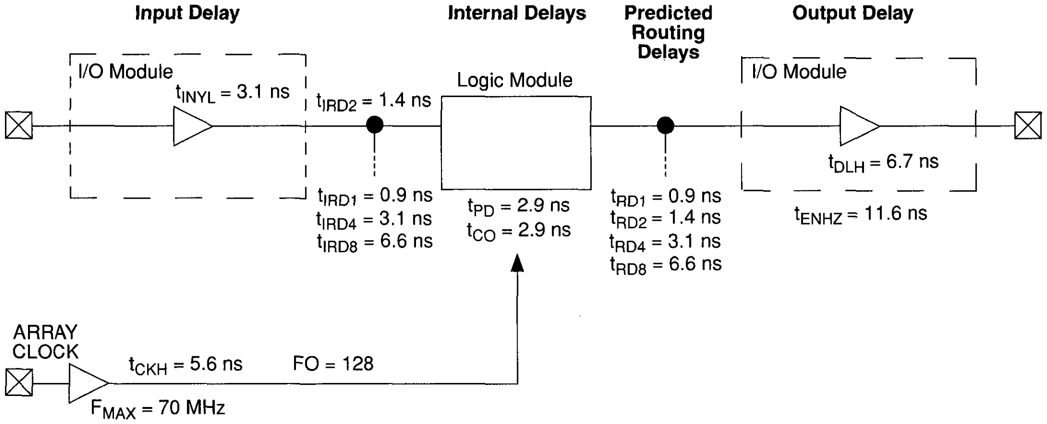
Note: The above curves are based on characterizations of sample devices and are not completely tested on all devices.

Output Buffer Performance Derating (3.3V)



Note: The above curves are based on characterizations of sample devices and are not completely tested on all devices.

ACT 1 Timing Module*



** Values shown for ACT 1 '–3 speed' devices at worst-case commercial conditions.*

Predictable Performance: Tight Delay Distributions

Propagation delay between logic modules depends on the resistive and capacitive loading of the routing tracks, the interconnect elements, and the module inputs being driven. Propagation delay increases as the length of routing tracks, the number of interconnect elements, or the number of inputs increases.

From a design perspective, the propagation delay can be statistically correlated or modeled by the fanout (number of loads) driven by a module. Higher fanout usually requires some paths to have longer routing tracks.

The ACT 1 family delivers a very tight fanout delay distribution. This tight distribution is achieved in two ways: by decreasing the delay of the interconnect elements and by decreasing the number of interconnect elements per path.

Actel's patented PLICE antifuse offers a very low resistive/capacitive interconnect. The ACT 1 family's antifuses, fabricated in 1.0 micron lithography, offer nominal levels of 200 ohms resistance and 7.5 femtofarad (fF) capacitance per antifuse.

The ACT 1 fanout distribution is also tight due to the low number of antifuses required for each interconnect path. The ACT 1 family's proprietary architecture limits the number of antifuses per path to a maximum of four, with 90% of interconnects using two antifuses.

Timing Characteristics

Timing characteristics for ACT 1 devices fall into three categories: family dependent, device dependent, and design dependent. The input and output buffer characteristics are common to all ACT 1 family members. Internal routing delays are device dependent. Design dependency means actual delays are not determined until after placement and routing of the user design is complete. Delay values may then be determined by using the DirectTime Analyzer utility or performing simulation with post-layout delays.

Critical Nets and Typical Nets

Propagation delays are expressed only for typical nets, which are used for initial design performance evaluation. Critical net delays can then be applied to the most time-critical paths. Critical nets are determined by net property assignment prior to placement and routing. Up to 6% of the nets in a design may be designated as critical, while 90% of the nets in a design are typical.

Long Tracks

Some nets in the design use long tracks. Long tracks are special routing resources that span multiple rows, columns, or modules. Long tracks employ three and sometimes four antifuse connections. This increases capacitance and resistance, resulting in longer net delays for macros connected to long tracks. Typically, up to 6% of nets in a fully utilized device require long tracks. Long tracks contribute approximately 5 ns to 10 ns delay. This additional delay is represented statistically in higher fanout (FO=8) routing delays in the data sheet specifications section.

Timing Derating

A best case timing derating factor of 0.45 is used to reflect best case processing. Note that this factor is relative to the "standard speed" timing parameters, and must be multiplied by the appropriate voltage and temperature derating factors for a given application.

Timing Derating Factor (Temperature and Voltage)

	Industrial		Military	
	Min.	Max.	Min.	Max.
(Commercial Minimum/Maximum Specification) x	0.69	1.11	0.67	1.23

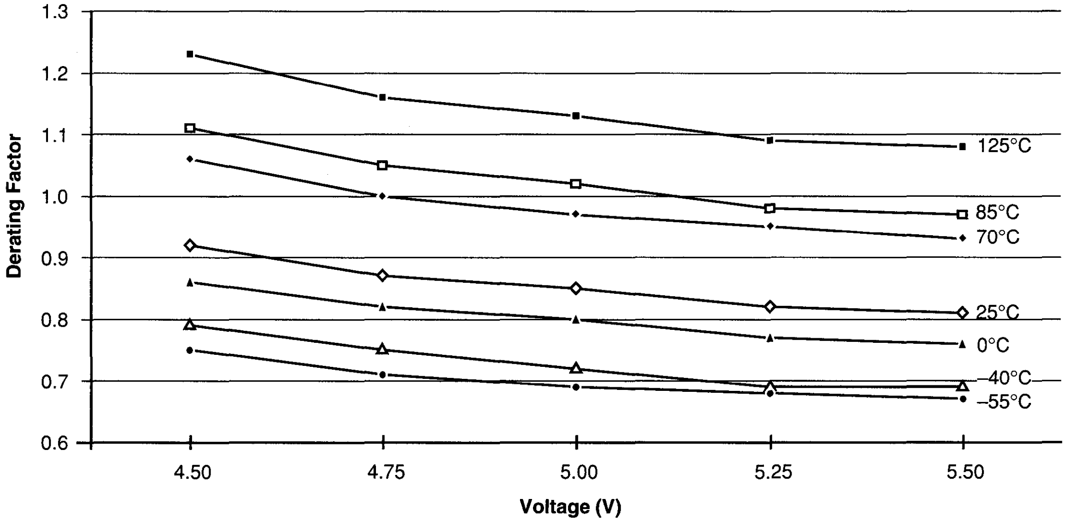
Timing Derating Factor for Designs at Typical Temperature (T_J = 25°C) and Voltage (5.0 V)

(Commercial Maximum Specification) x	0.85

Temperature and Voltage Derating Factors (normalized to Worst-Case Commercial, T_J = 4.75 V, 70°C)

	–55	–40	0	25	70	85	125
4.50	0.75	0.79	0.86	0.92	1.06	1.11	1.23
4.75	0.71	0.75	0.82	0.87	1.00	1.05	1.16
5.00	0.69	0.72	0.80	0.85	0.97	1.02	1.13
5.25	0.68	0.69	0.77	0.82	0.95	0.98	1.09
5.50	0.67	0.69	0.76	0.81	0.93	0.97	1.08

Junction Temperature and Voltage Derating Curves
(normalized to Worst-Case Commercial, T_J = 4.75 V, 70°C)

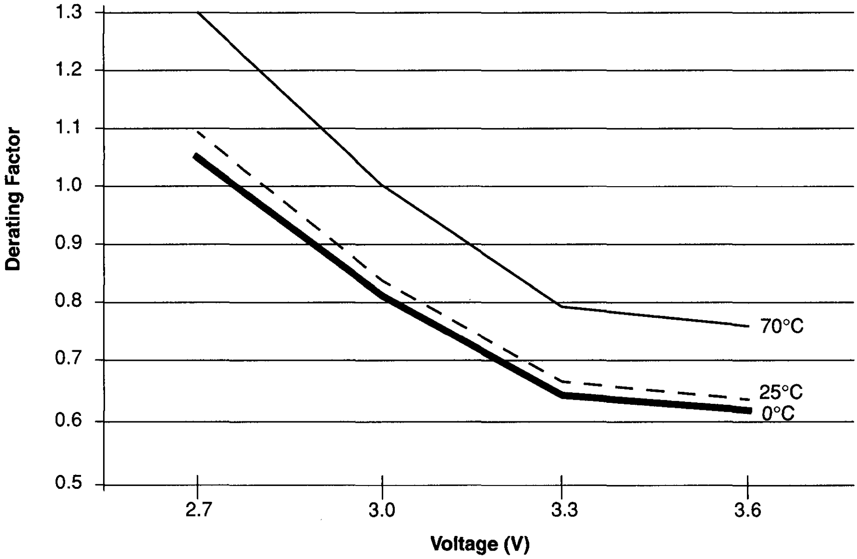


Note: This derating factor applies to all routing and propagation delays.

Temperature and Voltage Derating Factors (normalized to Worst-Case Commercial, T_J = 3.0 V, 70°C)

	0	25	70
2.7	1.05	1.09	1.30
3.0	0.81	0.84	1.00
3.3	0.64	0.67	0.79
3.6	0.62	0.64	0.76

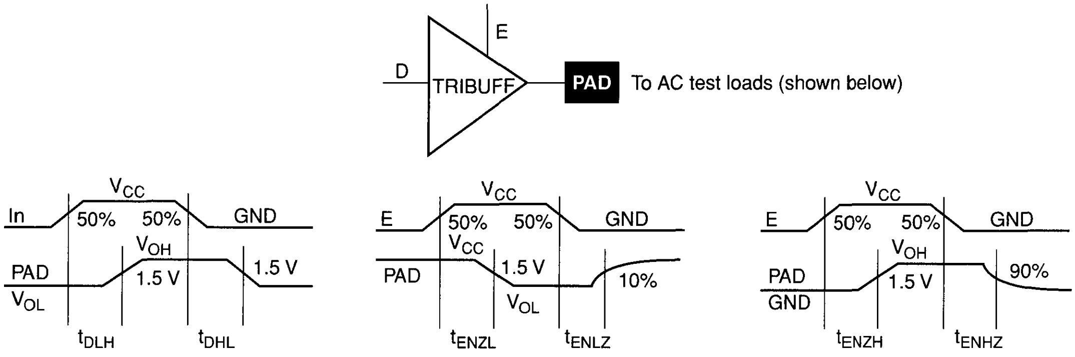
Junction Temperature and Voltage Derating Curves (normalized to Worst-Case Commercial, T_J = 3.0 V, 70°C)



Note: This derating factor applies to all routing and propagation delays.

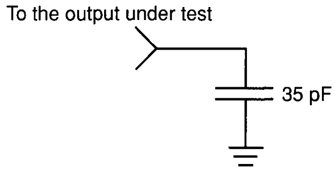
Parameter Measurement

Output Buffer Delays

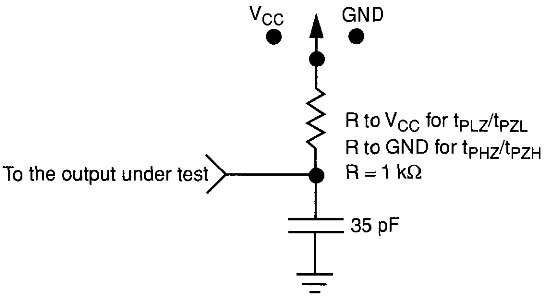


AC Test Loads

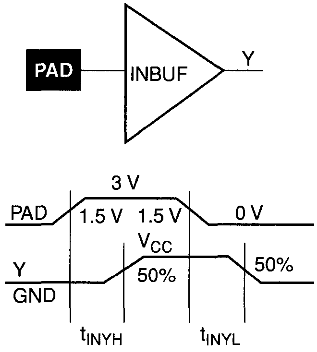
Load 1
(Used to measure propagation delay)



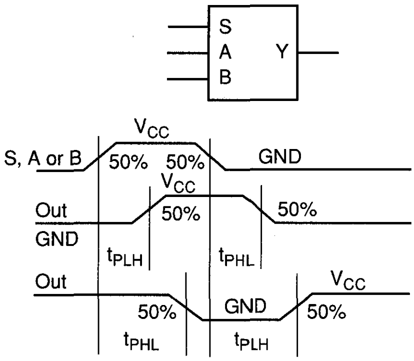
Load 2
(Used to measure rising/falling edges)



Input Buffer Delays

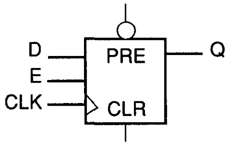


Module Delays

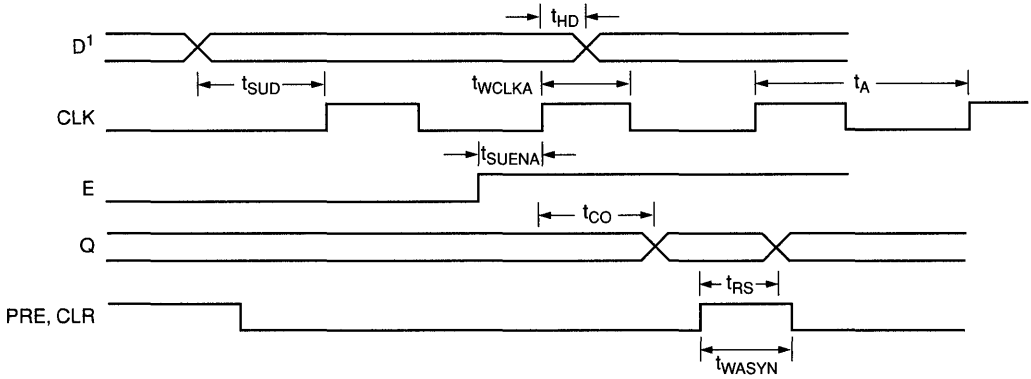


Sequential Timing Characteristics

Flip-Flops and Latches



(Positive edge triggered)



Note: *D represents all data functions involving A, B, S for multiplexed flip-flops.*

ACT 1 Timing Characteristics

(Worst-Case Commercial Conditions, V_{CC} = 4.75 V, T_J = 70°C)[1]

Logic Module Propagation Delays		'–3' Speed		'–2' Speed		'–1' Speed		'Std' Speed		3.3 V Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{PD1}	Single Module		2.9		3.4		3.8		4.5		6.5	ns
t_{PD2}	Dual Module Macros		6.8		7.8		8.8		10.4		15.1	ns
t_{CO}	Sequential Clk to Q		2.9		3.4		3.8		4.5		6.5	ns
t_{GO}	Latch G to Q		2.9		3.4		3.8		4.5		6.5	ns
t_{RS}	Flip-Flop (Latch) Reset to Q		2.9		3.4		3.8		4.5		6.5	ns
Predicted Routing Delays[2]												
t_{RD1}	FO=1 Routing Delay		0.9		1.1		1.2		1.4		2.0	ns
t_{RD2}	FO=2 Routing Delay		1.4		1.7		1.9		2.2		3.2	ns
t_{RD3}	FO=3 Routing Delay		2.1		2.5		2.8		3.3		4.8	ns
t_{RD4}	FO=4 Routing Delay		3.1		3.6		4.1		4.8		7.0	ns
t_{RD8}	FO=8 Routing Delay		6.6		7.7		8.7		10.2		14.8	ns
Sequential Timing Characteristics[3]												
t_{SUD}	Flip-Flop (Latch) Data Input Setup	5.5		6.4		7.2		8.5		10.0		ns
t_{HD}[4]	Flip-Flop (Latch) Data Input Hold	0.0		0.0		0.0		0.0		0.0		ns
t_{SUENA}	Flip-Flop (Latch) Enable Setup	5.5		6.4		7.2		8.5		10.0		ns
t_{HENA}	Flip-Flop (Latch) Enable Hold	0.0		0.0		0.0		0.0		0.0		ns
t_{WCLKA}	Flip-Flop (Latch) Clock Active Pulse Width	6.8		8.0		9.0		10.5		9.8		ns
t_{WASYN}	Flip-Flop (Latch) Asynchronous Pulse Width	6.8		8.0		9.0		10.5		9.8		ns
t_A	Flip-Flop Clock Input Period	14.2		16.7		18.9		22.3		20.0		ns
f_{MAX}	Flip-Flop (Latch) Clock Frequency (FO = 128)		70		60		53		45		50	MHz

Notes:

1. *V_{CC} = 3.0 V for 3.3V specifications.*
2. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*
3. *Setup times assume fanout of 3. Further testing information can be obtained from the DirectTime Analyzer utility.*
4. *The Hold Time for the DFME1A macro may be greater than 0 ns. Use the Designer 3.0 or later Timer to check the Hold Time for this macro.*

ACT 1 Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

Input Module Propagation Delays			'–3' Speed		'–2' Speed		'–1' Speed		'Std' Speed		3.3 V Speed		
Parameter	Description		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Units
t_{INYH}	Pad to Y High			3.1		3.5		4.0		4.7		6.8	ns
t_{INYL}	Pad to Y Low			3.1		3.5		4.0		4.7		6.8	ns
Input Module Predicted Routing Delays[1]													
t_{IRD1}	FO=1 Routing Delay			0.9		1.1		1.2		1.4		2.0	ns
t_{IRD2}	FO=2 Routing Delay			1.4		1.7		1.9		2.2		3.2	ns
t_{IRD3}	FO=3 Routing Delay			2.1		2.5		2.8		3.3		4.8	ns
t_{IRD4}	FO=4 Routing Delay			3.1		3.6		4.1		4.8		7.0	ns
t_{IRD8}	FO=8 Routing Delay			6.6		7.7		8.7		10.2		14.8	ns
Global Clock Network													
t_{CKH}	Input Low to High	FO = 16		4.9		5.6		6.4		7.5		6.7	
		FO = 128		5.6		6.4		7.3		8.6		7.9	ns
t_{CKL}	Input High to Low	FO = 16		6.4		7.4		8.4		9.9		8.8	
		FO = 128		7.0		8.1		9.2		10.8		10.0	ns
t_{PWH}	Minimum Pulse Width High	FO = 16	6.5		7.5		8.5		10.0		8.9		
		FO = 128	6.8		8.0		9.0		10.5		9.8		ns
t_{PWL}	Minimum Pulse Width Low	FO = 16	6.5		7.5		8.5		10.0		8.9		
		FO = 128	6.8		8.0		9.0		10.5		9.8		ns
t_{CKSW}	Maximum Skew	FO = 16		1.2		1.3		1.5		1.8		1.5	
		FO = 128		1.8		2.1		2.4		2.8		2.4	ns
t_P	Minimum Period	FO = 16	13.2		15.4		17.6		20.9		18.2		
		FO = 128	14.2		16.7		18.9		22.3		20		ns
f_{MAX}	Maximum Frequency	FO = 16		75		65		57		48		55	
		FO = 128		70		60		53		45		50	MHz

Note:

1. *These parameters should be used for estimating device performance. Optimization techniques may further reduce delays by 0 to 4 ns. Routing delays are for typical designs across worst-case operating conditions. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

ACT 1 Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

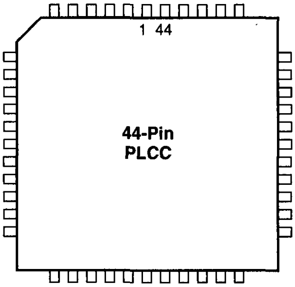
Output Module Timing		'–3' Speed		'–2' Speed		'–1' Speed		'Std' Speed		3.3 V Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
TTL Output Module Timing[1]												
t_{DLH}	Data to Pad High		6.7		7.6		8.7		10.3		15.0	ns
t_{DHL}	Data to Pad Low		7.5		8.6		9.8		11.5		16.7	ns
t_{ENZH}	Enable Pad Z to High		6.6		7.5		8.6		10.2		14.8	ns
t_{ENZL}	Enable Pad Z to Low		7.9		9.1		10.4		12.2		17.7	ns
t_{ENHZ}	Enable Pad High to Z		10.0		11.6		13.1		15.4		22.4	ns
t_{ENLZ}	Enable Pad Low to Z		9.0		10.4		11.8		13.9		20.2	ns
d_{TLH}	Delta Low to High		0.06		0.07		0.08		0.09		0.13	ns/pF
d_{THL}	Delta High to Low		0.08		0.09		0.10		0.12		0.17	ns/pF
CMOS Output Module Timing[1]												
t_{DLH}	Data to Pad High		7.9		9.2		10.4		12.2		17.7	ns
t_{DHL}	Data to Pad Low		6.4		7.2		8.2		9.8		14.2	ns
t_{ENZH}	Enable Pad Z to High		6.0		6.9		7.9		9.2		13.4	ns
t_{ENZL}	Enable Pad Z to Low		8.3		9.4		10.7		12.7		18.5	ns
t_{ENHZ}	Enable Pad High to Z		10.0		11.6		13.1		15.4		22.4	ns
t_{ENLZ}	Enable Pad Low to Z		9.0		10.4		11.8		13.9		20.2	ns
d_{TLH}	Delta Low to High		0.10		0.11		0.13		0.15		0.22	ns/pF
d_{THL}	Delta High to Low		0.06		0.07		0.08		0.09		0.13	ns/pF

Notes:

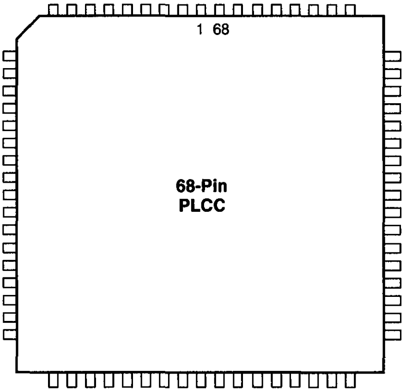
1. *Delays based on 35 pF loading.*
2. *SSO information can be found in the "Simultaneous Switching Output Limits for Actel FPGAs" application note on page 4-125.*

Package Pin Assignments

44-Pin PLCC



68-Pin PLCC



Signal	A1010B Function	A1020B Function
3	VCC	VCC
10	GND	GND
14	VCC	VCC
16	VCC	VCC
21	GND	GND
25	VCC	VCC
32	GND	GND
33	CLK, I/O	CLK, I/O
34	MODE	MODE
35	VCC	VCC
36	SDI, I/O	SDI, I/O
37	DCLK, I/O	DCLK, I/O
38	PRA, I/O	PRA, I/O
39	PRB, I/O	PRB, I/O
43	GND	GND

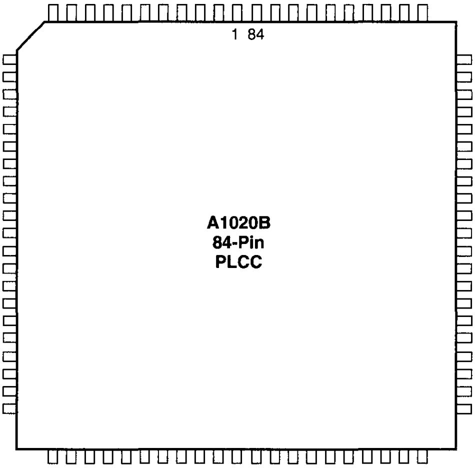
Signal	A1010B, A10V10B Function	A1020B, A10V20B Functions
4	VCC	VCC
14	GND	GND
15	GND	GND
21	VCC	VCC
25	VCC	VCC
32	GND	GND
38	VCC	VCC
49	GND	GND
52	CLK, I/O	CLK, I/O
54	MODE	MODE
55	VCC	VCC
56	SDI, I/O	SDI, I/O
57	DCLK, I/O	DCLK, I/O
58	PRA, I/O	PRA, I/O
59	PRB, I/O	PRB, I/O
66	GND	GND

Notes:

1. *NC: Denotes No Connection*
2. *All unlisted pin numbers are user I/Os.*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

84-Pin PLCC



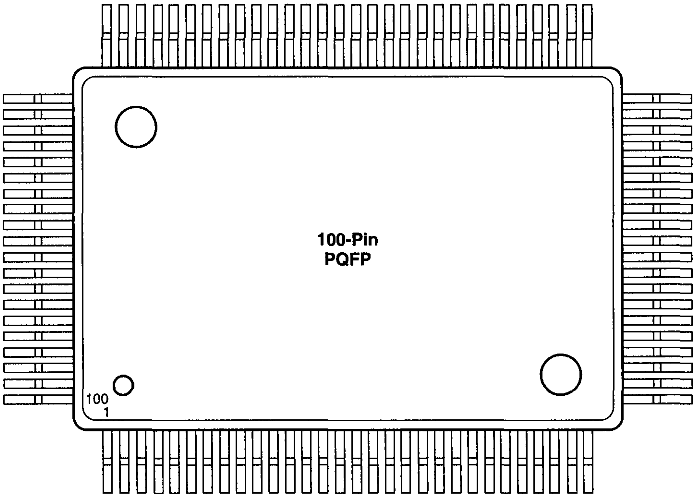
Signal	A1020B, A10V20B Function
4	VCC
12	NC
18	GND
19	GND
25	VCC
26	VCC
33	VCC
40	GND
46	VCC
60	GND
61	GND
64	CLK, I/O
66	MODE
67	VCC
68	VCC
72	SDI, I/O
73	DCLK, I/O
74	PRA, I/O
75	PRB, I/O
82	GND

Notes:

1. *NC: Denotes No Connection*
2. *All unlisted pin numbers are user I/Os.*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

100-Pin PQFP



Pin	A1010B Function	A1020B Function
1	NC	NC
2	NC	NC
3	NC	NC
4	NC	NC
5	NC	NC
6	PRB, I/O	PRB, I/O
13	GND	GND
19	VCC	VCC
27	NC	NC
28	NC	NC
29	NC	NC
30	NC	NC
31	NC	I/O
32	NC	I/O
33	NC	I/O
36	GND	GND
37	GND	GND
43	VCC	VCC
44	VCC	VCC
48	NC	I/O
49	NC	I/O
50	NC	I/O
51	NC	NC
52	NC	NC

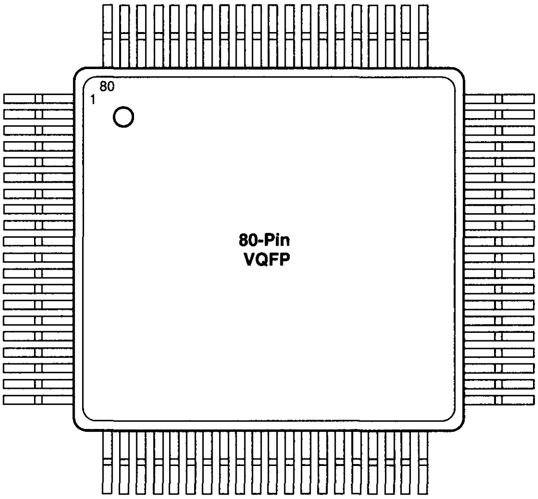
Pin	A1010B Function	A1020B Function
53	NC	NC
54	NC	NC
55	NC	NC
56	VCC	VCC
63	GND	GND
69	VCC	VCC
77	NC	NC
78	NC	NC
79	NC	NC
80	NC	I/O
81	NC	I/O
82	NC	I/O
86	GND	GND
87	GND	GND
90	CLK, I/O	CLK, I/O
92	MODE	MODE
93	VCC	VCC
94	VCC	VCC
95	NC	I/O
96	NC	I/O
97	NC	I/O
98	SDI, I/O	SDI, I/O
99	DCLK, I/O	DCLK, I/O
100	PRA, I/O	PRA, I/O

Notes:

1. *NC: Denotes No Connection*
2. *All unlisted pin numbers are user I/Os.*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

80-Pin VQFP



Pin	A1010B, A10V10B Function	A1020B, A10V20B Function
2	NC	I/O
3	NC	I/O
4	NC	I/O
7	GND	GND
13	VCC	VCC
17	NC	I/O
18	NC	I/O
19	NC	I/O
20	VCC	VCC
27	GND	GND
33	VCC	VCC
41	NC	I/O
42	NC	I/O
43	NC	I/O

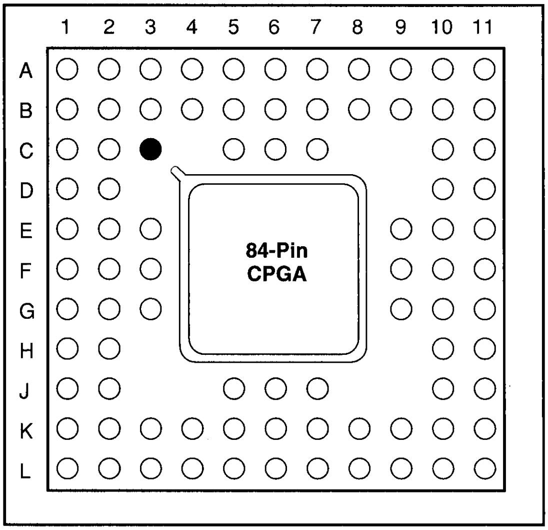
Pin	A1010B, A10V10B Function	A1020B, A10V20B Function
47	GND	GND
50	CLK, I/O	CLK, I/O
52	MODE	MODE
53	VCC	VCC
54	NC	I/O
55	NC	I/O
56	NC	I/O
57	SDI, I/O	SDI, I/O
58	DCLK, I/O	DCLK, I/O
59	PRA, I/O	PRA, I/O
60	NC	NC
61	PRB, I/O	PRB, I/O
68	GND	GND
74	VCC	VCC

Notes:

1. *NC: Denotes No Connection*
2. *All unlisted pin numbers are user I/Os.*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

84-Pin CPGA



● Orientation Pin (C3)

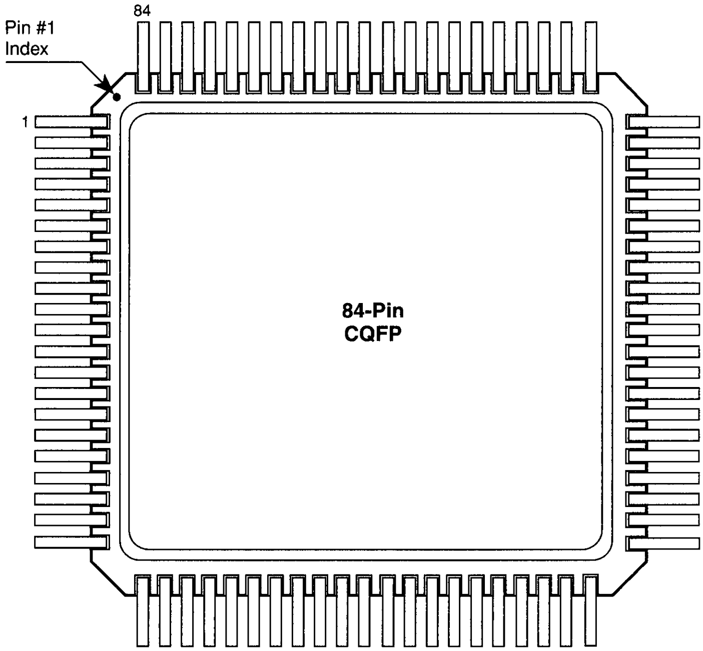
Pin	A1010B Function	A1020B Function
A11	PRA, I/O	PRA, I/O
B1	NC	I/O
B2	NC	NC
B5	VCC	VCC
B7	GND	GND
B10	PRB, I/O	PRB, I/O
B11	SDI, I/O	SDI,I/O
C1	NC	I/O
C2	NC	I/O
C10	DCLK, I/O	DCLK, I/O
C11	NC	I/O
D10	NC	I/O
D11	NC	I/O
E2	GND	GND
E3	GND	GND
E9	VCC	VCC
E10	VCC	VCC
E11	MODE	MODE
F1	VCC	VCC
F9	CLK, I/O	CLK, I/O
F10	GND	GND
G2	VCC	VCC
G10	GND	GND
J2	NC	I/O
J10	NC	I/O
K1	NC	I/O
K2	VCC	VCC
K5	GND	GND
K7	VCC	VCC
K10	NC	I/O
K11	NC	I/O
L1	NC	I/O

Notes:

1. *NC: Denotes No Connection*
2. *All unlisted pin numbers are user I/Os.*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

84-Pin CQFP



Pin	A1020B Function
1	NC
7	GND
8	GND
14	VCC
15	VCC
22	VCC
29	GND
35	VCC
49	GND
50	GND

Pin	A1020B Function
53	CLK, I/O
55	MODE
56	VCC
57	VCC
61	SDI, I/O
62	DCLK, I/O
63	PRA, I/O
64	PRB, I/O
71	GND
77	VCC

Notes:

1. *NC: Denotes No Connection*
2. *All unlisted pin numbers are user I/Os.*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage; otherwise it can be terminated directly to GND.*

Actel

ACT™ 2 Family FPGAs

Features

- Up to 8000 Gate Array Gates (20,000 PLD equivalent gates)
- Replaces up to 200 TTL Packages
- Replaces up to eighty 20-Pin PAL® Packages
- Design Library with over 500 Macro Functions
- Single-Module Sequential Functions
- Wide-Input Combinatorial Functions
- Up to 1232 Programmable Logic Modules
- Up to 998 Flip-Flops
- Datapath Performance at 105 MHz
- 16-Bit Accumulator Performance to 39 MHz
- Two In-Circuit Diagnostic Probe Pins Support Speed Analysis to 50 MHz
- Two High-Speed, Low-Skew Clock Networks
- I/O Drive to 10 mA
- Nonvolatile, User Programmable
- Logic Fully Tested Prior to Shipment
- 1.0-micron CMOS Technology

Product Family Profile

Device	A1225A	A1240A	A1280A
Capacity			
Gate Array Equivalent Gates	2,500	4,000	8,000
PLD Equivalent Gates	6,250	10,000	20,000
TTL Equivalent Packages	63	100	200
20-Pin PAL Equivalent Packages	25	40	80
Logic Modules	451	684	1,232
S-Modules	231	348	624
C-Modules	220	336	608
Flip-Flops (maximum)	382	568	998
Routing Resources			
Horizontal Tracks/Channel	36	36	36
Vertical Tracks/Channel	15	15	15
PLICE Antifuse Elements	250,000	400,000	750,000
User I/Os (maximum)	83	104	140
Packages[1]	100 CPGA 100 PQFP 100 VQFP 84 PLCC	132 CPGA 144 PQFP 176 TQFP 84 PLCC	176 CPGA 160 PQFP 176 TQFP 84 PLCC 172 CQFP
Performance[2]			
16-Bit Prescaled Counters	105 MHz	100 MHz	85 MHz
16-Bit Loadable Counters	70 MHz	69 MHz	67 MHz
16-Bit Accumulators	39 MHz	38 MHz	36 MHz

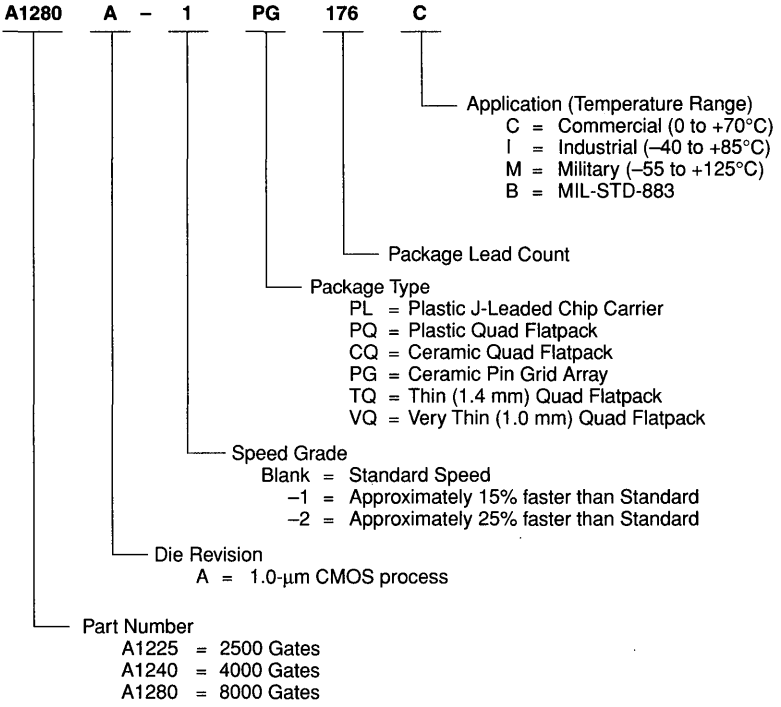
Notes:

1. *See product plan on page 1-171 for package availability.*
2. *Performance is based on '–2' speed devices at commercial worst-case operating conditions using PREP Benchmarks, Suite #1, Version 1.2, dated 3-28-93, any analysis is not endorsed by PREP.*

Description

The ACT™ 2 family represents Actel's second generation of field programmable gate arrays (FPGAs). The ACT 2 family presents a two-module architecture, consisting of C-modules and S-modules. These modules are optimized for both combinatorial and sequential designs. Based on Actel's patented channeled array architecture, the ACT 2 family provides significant enhancements to gate density and performance while maintaining downward compatibility with the ACT 1 design environment and upward compatibility with the ACT 3 design environment. The devices are implemented in silicon gate, 1.0-µm, two-level metal CMOS, and employ Actel's PLICE® antifuse technology. This revolutionary architecture offers gate array design flexibility, high performance, and fast time-to-production with user programming. The ACT 2 family is supported by the Designer and Designer Advantage Systems, which offers automatic pin assignment, validation of electrical and design rules, automatic placement and routing, timing analysis, user programming, and diagnostic probe capabilities. The systems are supported on the following platforms: 386/486™ PC, Sun™, and HP™ workstations. The systems provide CAE interfaces to the following design environments: Cadence, Viewlogic®, Mentor Graphics®, and OrCAD™.

Ordering Information



A1280 A – 1 PG 176 C

Application (Temperature Range)
C = Commercial (0 to +70°C)
I = Industrial (–40 to +85°C)
M = Military (–55 to +125°C)
B = MIL-STD-883

Package Lead Count

Package Type
PL = Plastic J-Leaded Chip Carrier
PQ = Plastic Quad Flatpack
CQ = Ceramic Quad Flatpack
PG = Ceramic Pin Grid Array
TQ = Thin (1.4 mm) Quad Flatpack
VQ = Very Thin (1.0 mm) Quad Flatpack

Speed Grade
Blank = Standard Speed
–1 = Approximately 15% faster than Standard
–2 = Approximately 25% faster than Standard

Die Revision
A = 1.0-µm CMOS process

Part Number
A1225 = 2500 Gates
A1240 = 4000 Gates
A1280 = 8000 Gates

Product Plan[1]

	Speed Grade*			Application			
	Std	–1	–2	C	I	M	B
A1225A Device							
100-pin Ceramic Pin Grid Array (PG)	✔	✔	✔	✔	—	—	—
100-pin Plastic Quad Flatpack (PQ)	✔	✔	✔	✔	✔	—	—
100-pin Very Thin (1.0 mm) Quad Flatpack (VQ)	✔	✔	✔	✔	—	—	—
84-pin Plastic Leaded Chip Carrier (PL)	✔	✔	✔	✔	✔	—	—
A1240A Device							
132-pin Ceramic Pin Grid Array (PG)	✔	✔	✔	✔	—	✔	✔
176-pin Thin (1.4 mm) Quad Flatpack (TQ)	✔	✔	✔	✔	—	—	—
144-pin Plastic Quad Flatpack (PQ)	✔	✔	✔	✔	✔	—	—
84-pin Plastic Leaded Chip Carrier (PL)	✔	✔	✔	✔	✔	—	—
A1280A Device							
176-pin Ceramic Pin Grid Array (PG)	✔	✔	✔	✔	—	✔	✔
176-pin Thin (1.4 mm) Quad Flatpack (TQ)	✔	✔	✔	✔	—	—	—
160-pin Plastic Quad Flatpack (PQ)	✔	✔	✔	✔	✔	—	—
172-pin Ceramic Quad Flatpack (CQ)	✔	✔	✔	✔	—	✔	✔

Applications:	C = Commercial	Availability:	✔ =	Available	* Speed Grade:	–1 =	Approx. 15% faster than Standard
	I = Industrial		P =	Planned		–2 =	Approx. 25% faster than Standard
	M = Military		— =	Not Planned			
	B = MIL-STD-883						

Note:

[1] *Please consult Actel representatives for current availability.*

Device Resources

			User I/Os									
			CPGA			PQFP			PLCC	CQFP	TQFP	VQFP
Device Series	Logic Modules	Gates	176-pin	132-pin	100-pin	160-pin	144-pin	100-pin	84-pin	172-pin	176-pin	100-pin
A1225A	451	2500	—	—	83	—	—	83	72	—	—	83
A1240A	684	4000	—	104	—	—	104	—	72	—	104	—
A1280A	1232	8000	140	—	—	125	—	—	72	140	140	—

Pin Description

CLKA Clock A (Input)

TTL Clock input for clock distribution networks. The Clock input is buffered prior to clocking the logic modules. This pin can also be used as an I/O.

CLKB Clock B (Input)

TTL Clock input for clock distribution networks. The Clock input is buffered prior to clocking the logic modules. This pin can also be used as an I/O.

DCLK Diagnostic Clock (Input)

TTL Clock input for diagnostic probe and device programming. DCLK is active when the MODE pin is HIGH. This pin functions as an I/O when the MODE pin is LOW.

GND Ground

LOW supply voltage.

I/O Input/Output (Input, Output)

The I/O pin functions as an input, output, three-state, or bidirectional buffer. Input and output levels are compatible with standard TTL and CMOS specifications. Unused I/O pins are automatically driven LOW by the ALS software.

MODE Mode (Input)

The MODE pin controls the use of multifunction pins (DCLK, PRA, PRB, SDI). When the MODE pin is HIGH, the special functions are active. When the MODE pin is LOW, the pins function as I/Os. To provide Actionprobe capability, the MODE pin should be terminated to GND through a 10K resistor so that the MODE pin can be pulled high when required.

NC No Connection

This pin is not connected to circuitry within the device.

PRA Probe A (Output)

The Probe A pin is used to output data from any user-defined design node within the device. This independent diagnostic pin is used in conjunction with the Probe B pin to allow real-time diagnostic output of any signal path within the device. The Probe A pin can be used as a user-defined I/O when debugging has been completed. The pin's probe capabilities can be permanently disabled to protect programmed design confidentiality. PRA is active when the MODE pin is HIGH. This pin functions as an I/O when the MODE pin is LOW.

PRB Probe B (Output)

The Probe B pin is used to output data from any user-defined design node within the device. This independent diagnostic pin is used in conjunction with the Probe A pin to allow real-time diagnostic output of any signal path within the device. The Probe B pin can be used as a user-defined I/O when debugging has been completed. The pin's probe capabilities can be permanently disabled to protect programmed design confidentiality. PRB is active when the MODE pin is HIGH. This pin functions as an I/O when the MODE pin is LOW.

SDI Serial Data Input (Input)

Serial data input for diagnostic probe and device programming. SDI is active when the MODE pin is HIGH. This pin functions as an I/O when the MODE pin is LOW.

V_{CC} 5 V Supply Voltage

HIGH supply voltage.

Absolute Maximum Ratings[1]

Free air temperature range

Symbol	Parameter	Limits	Units
V_{CC}	DC Supply Voltage	–0.5 to +7.0	V
V_I	Input Voltage	–0.5 to V_{CC} +0.5	V
V_O	Output Voltage	–0.5 to V_{CC} +0.5	V
I_{IO}	I/O Source/Sink Current[2]	±20	mA
T_{STG}	Storage Temperature	–65 to +150	°C

Notes:

1. *Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum rated conditions for extended periods may affect device reliability. Device should not be operated outside the Recommended Operating Conditions.*
2. *Device inputs are normally high impedance and draw extremely low current. However, when input voltage is greater than V_{CC} + 0.5 V or less than GND – 0.5 V, the internal protection diode will be forward biased and can draw excessive current.*

Recommended Operating Conditions

Parameter	Commercial	Industrial	Military	Units
Temperature Range[1]	0 to +70	–40 to +85	–55 to +125	°C
Power Supply Tolerance	±5	±10	±10	%V_{CC}

Notes:

1. *Ambient temperature (T_A) is used for commercial and industrial; case temperature (T_C) is used for military.*

Electrical Specifications

Symbol	Parameter	Commercial		Industrial		Military		Units
		Min.	Max.	Min.	Max.	Min.	Max.	
V_{OH}[1]	(I_{OH} = –10 mA)[2]	2.4						V
	(I_{OH} = –6 mA)	3.84						V
	(I_{OH} = –4 mA)			3.7		3.7		V
V_{OL}[1]	(I_{OL} = 10 mA)[2]		0.5					V
	(I_{OL} = 6 mA)		0.33		0.40		0.40	V
V_{IL}		–0.3	0.8	–0.3	0.8	–0.3	0.8	V
V_{IH}		2.0	V_{CC} + 0.3	2.0	V_{CC} + 0.3	2.0	V_{CC} + 0.3	V
Input Transition Time t_R, t_F[2]			500		500		500	ns
C_{IO} I/O Capacitance[2, 3]			10		10		10	pF
Standby Current, I_{CC}[4] (typical = 1 mA)			2		10		20	mA
Leakage Current[5]		–10	10	–10	10	–10	10	µA

Notes:

1. *Only one output tested at a time. V_{CC} = min.*
2. *Not tested, for information only.*
3. *Includes worst-case 176 CPGA package capacitance. V_{OUT} = 0 V, f = 1 MHz.*
4. *All outputs unloaded. All inputs = V_{CC} or GND, typical I_{CC} = 1 mA. I_{CC} limit includes I_{PP} and I_{SV} during normal operation.*
5. *V_{OUT}, V_{IN} = V_{CC} or GND.*

Package Thermal Characteristics

The device junction to case thermal characteristic is θjc, and the junction to ambient air characteristic is θja. The thermal characteristics for θja are shown with two different air flow rates.

Maximum junction temperature is 150°C.

A sample calculation of the absolute maximum power dissipation allowed for a PQFP 160-pin package at commercial temperature is as follows:

$$\frac{\text{Max. junction temp. (°C)} - \text{Max. commercial temp.}}{\theta\text{ja (°C/W)}} = \frac{150°\text{C} - 70°\text{C}}{33°\text{C/W}} = 2.4 \text{ W}$$

Package Type	Pin Count	θjc	θja Still Air	θja 300 ft/min	Units
Ceramic Pin Grid Array	100	5	35	17	°C/W
	132	5	30	15	°C/W
	176	8	23	12	°C/W
Ceramic Quad Flatpack	172	8	25	15	°C/W
Plastic Quad Flatpack[1]	100	13	48	40	°C/W
	144	15	40	32	°C/W
	160	15	38	30	°C/W
Plastic Leaded Chip Carrier[2]	84	12	37	28	°C/W
Very Thin Quad Flatpack[3]	100	12	43	35	°C/W
Thin Quad Flatpack[4]	176	15	32	25	°C/W

Notes: (Maximum Power in Still Air)

1. *Maximum Power Dissipation for PQFP packages are 1.9 Watts (100-pin), 2.3 Watts (144-pin), and 2.4 Watts (160-pin).*
2. *Maximum Power Dissipation for PLCC packages is 2.7 Watts.*
3. *Maximum Power Dissipation for VQFP packages is 2.3 Watts.*
4. *Maximum Power Dissipation for TQFP packages is 3.1 Watts.*

Power Dissipation

$$P = [I_{CC}\text{standby} + I_{CC}\text{active}] * V_{CC} + I_{OL} * V_{OL} * N + I_{OH} * (V_{CC} - V_{OH}) * M$$

Where:

I_{CC} standby is the current flowing when no inputs or outputs are changing.

I_{CC} active is the current flowing due to CMOS switching.

I_{OL}, I_{OH} are TTL sink/source currents.

V_{OL}, V_{OH} are TTL level output voltages.

N equals the number of outputs driving TTL loads to V_{OL}.

M equals the number of outputs driving TTL loads to V_{OH}.

An accurate determination of N and M is problematical because their values depend on the family type, design details, and on the system I/O. The power can be divided into two components: static and active.

Static Power Component

Actel FPGAs have small static power components that result in lower power dissipation than PALs or PLDs. By integrating multiple PALs/PLDs into one FPGA, an even greater reduction in board-level power dissipation can be achieved.

The power due to standby current is typically a small component of the overall power. Standby power is calculated below for commercial, worst case conditions.

I_{CC}	V_{CC}	Power
2 mA	5.25 V	10.5 mW

The static power dissipated by TTL loads depends on the number of outputs driving high or low and the DC load current. Again, this value is typically small. For instance, a 32-bit bus sinking 4 mA at 0.33 V will generate 42 mW with all outputs driving low, and 140 mW with all outputs driving high. The actual dissipation will average somewhere between as I/Os switch states with time.

Active Power Component

Power dissipation in CMOS devices is usually dominated by the active (dynamic) power dissipation. This component is frequency dependent, a function of the logic and the external I/O. Active power dissipation results from charging internal chip capacitances of the interconnect, unprogrammed antifuses, module inputs, and module outputs, plus external capacitance due to PC board traces and load device inputs. An additional component of the active power dissipation is the totem-pole current in CMOS transistor pairs. The net effect can be associated with an equivalent capacitance that can be combined with frequency and voltage to represent active power dissipation.

Equivalent Capacitance

The power dissipated by a CMOS circuit can be expressed by the Equation 1.

$$\text{Power (uW)} = C_{EQ} * V_{CC}^2 * F \qquad (1)$$

Where:

C_{EQ} is the equivalent capacitance expressed in pF.

V_{CC} is the power supply in volts.

F is the switching frequency in MHz.

Equivalent capacitance is calculated by measuring ICC active at a specified frequency and voltage for each circuit component of interest. Measurements have been made over a range of frequencies at a fixed value of VCC. Equivalent capacitance is frequency independent so that the results may be used over a wide range of operating conditions. Equivalent capacitance values are shown below.

C_{EQ} Values for Actel FPGAs

Modules (C_{EQM})	5.8
Input Buffers (C_{EQI})	12.9
Output Buffers (C_{EQO})	23.8
Routed Array Clock Buffer Loads (C_{EQCR})	3.9

To calculate the active power dissipated from the complete design, the switching frequency of each part of the logic must be known. Equation 2 shows a piece-wise linear summation over all components.

$$\text{Power} = V_{CC}^2 * [(m * C_{EQM} * f_m)_{modules} + (n * C_{EQI} * f_n)_{inputs} + (p * (C_{EQO} + C_L) * f_p)_{outputs} + 0.5 * (q_1 * C_{EQCR} * f_{q1})_{routed\_Clk1} + (r_1 * f_{q1})_{routed\_Clk1} + 0.5 * (q_2 * C_{EQCR} * f_{q2})_{routed\_Clk2} + (r_2 * f_{q2})_{routed\_Clk2}] \qquad (2)$$

Where:

m = Number of logic modules switching at fm

n = Number of input buffers switching at fn

p = Number of output buffers switching at fp

q1 = Number of clock loads on the first routed array clock

q2 = Number of clock loads on the second routed array clock

r_1 = Fixed capacitance due to first routed array clock

r_2 = Fixed capacitance due to second routed array clock

C_{EQM} = Equivalent capacitance of logic modules in pF

C_{EQI} = Equivalent capacitance of input buffers in pF

C_{EQO} = Equivalent capacitance of output buffers in pF

C_{EQCR} = Equivalent capacitance of routed array clock in pF

C_L = Output lead capacitance in pF

f_m = Average logic module switching rate in MHz

f_n = Average input buffer switching rate in MHz

f_p = Average output buffer switching rate in MHz

f_{q1} = Average first routed array clock rate in MHz

f_{q2} = Average second routed array clock rate in MHz

Fixed Capacitance Values for Actel FPGAs (pF)

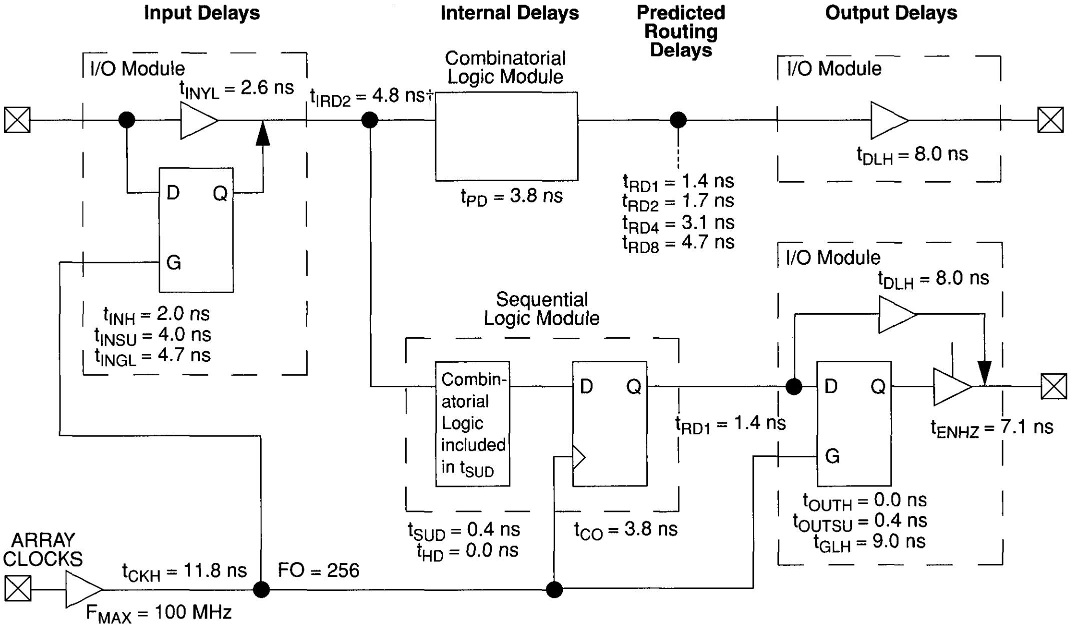
Device Type	r1 routed_Clk1	r2 routed_Clk2
A1225A	106	106.0
A1240A	134	134.2
A1280A	168	167.8

Determining Average Switching Frequency

To determine the switching frequency for a design, you must have a detailed understanding of the data input values to the circuit. The following guidelines are meant to represent worst-case scenarios so that they can be generally used to predict the upper limits of power dissipation. These guidelines are as follows:

Logic Modules (m)	80% of modules
Inputs switching (n)	# inputs/4
Outputs switching (p)	# outputs/4
First routed array clock loads (q_1)	40% of sequential modules
Second routed array clock loads (q_2)	40% of sequential modules
Load capacitance (C_L)	35 pF
Average logic module switching rate (f_m)	F/10
Average input switching rate (f_n)	F/5
Average output switching rate (f_p)	F/10
Average first routed array clock rate (f_{q1})	F
Average second routed array clock rate (f_{q2})	F/2

ACT 2 Timing Model*

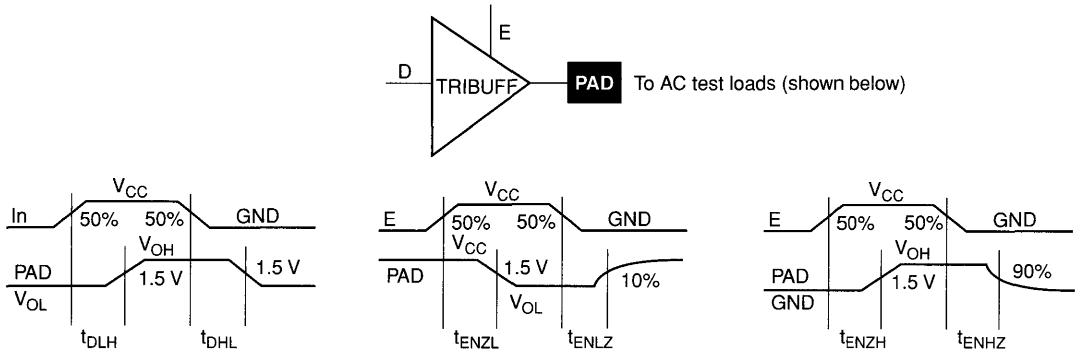


**Values shown for A1240A-2 at worst-case commercial conditions.*

† Input Module Predicted Routing Delay

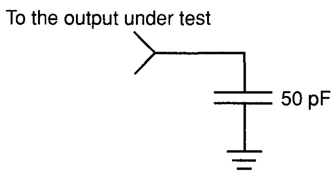
Parameter Measurement

Output Buffer Delays

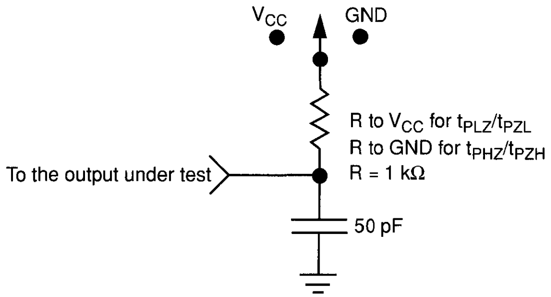


AC Test Loads

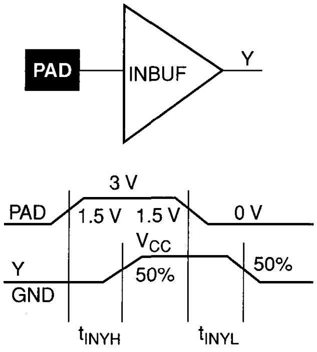
Load 1
(Used to measure propagation delay)



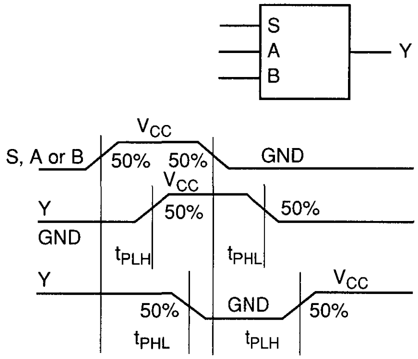
Load 2
(Used to measure rising/falling edges)



Input Buffer Delays

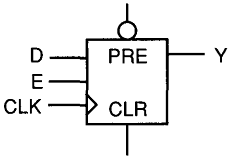


Module Delays

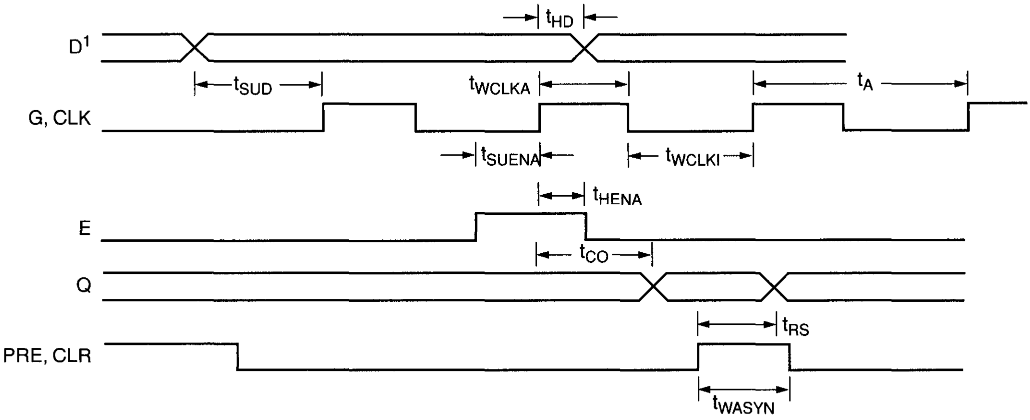


Sequential Module Timing Characteristics

Flip-Flops and Latches



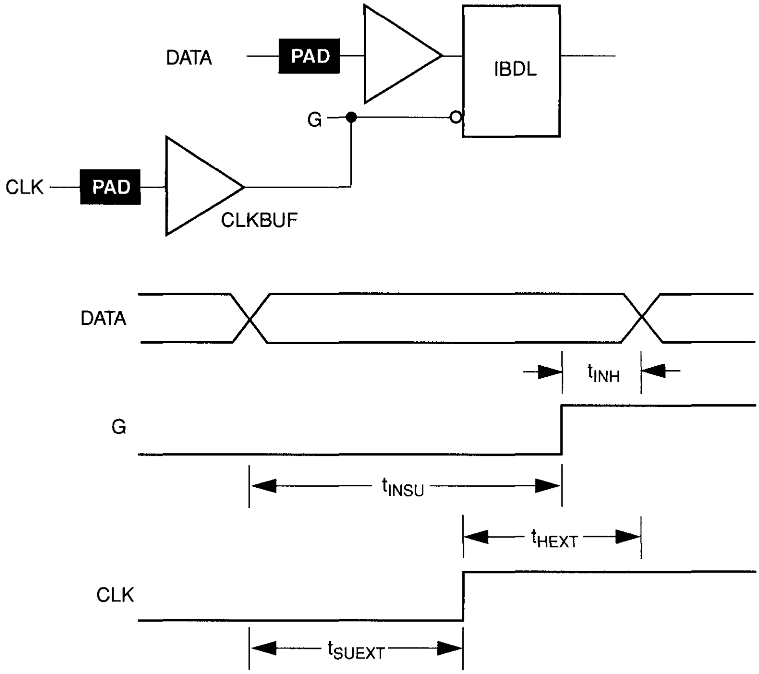
(Positive edge triggered)



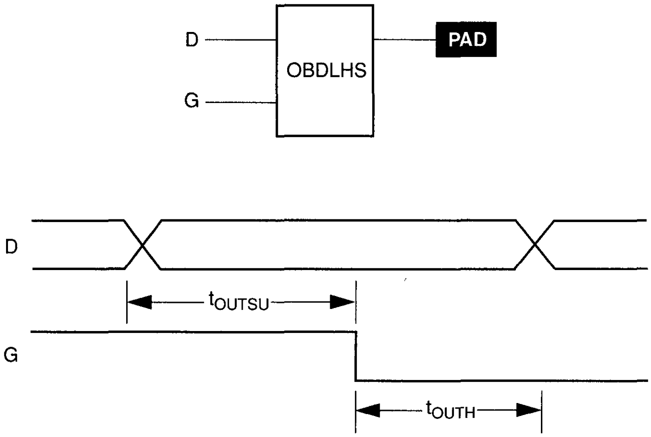
Note: D represents all data functions involving A, B, and S for multiplexed flip-flops.

Sequential Timing Characteristics (continued)

Input Buffer Latches



Output Buffer Latches



Timing Derating Factor (Temperature and Voltage)

	Industrial		Military	
	Min.	Max.	Min.	Max.
(Commercial Minimum/Maximum Specification) x	0.69	1.11	0.67	1.23

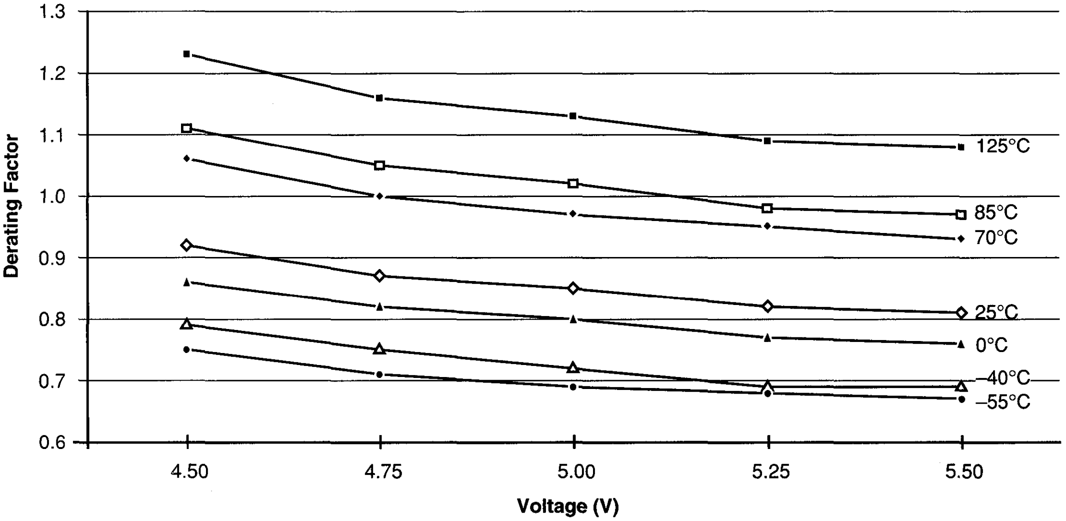
Timing Derating Factor for Designs at Typical Temperature (T_J = 25°C) and Voltage (5.0 V)

(Commercial Maximum Specification) x	0.85

Temperature and Voltage Derating Factors (normalized to Worst-Case Commercial, T_J = 4.75 V, 70°C)

	–55	–40	0	25	70	85	125
4.50	0.75	0.79	0.86	0.92	1.06	1.11	1.23
4.75	0.71	0.75	0.82	0.87	1.00	1.05	1.16
5.00	0.69	0.72	0.80	0.85	0.97	1.02	1.13
5.25	0.68	0.69	0.77	0.82	0.95	0.98	1.09
5.50	0.67	0.69	0.76	0.81	0.93	0.97	1.08

Junction Temperature and Voltage Derating Curves (normalized to Worst-Case Commercial, T_J = 4.75 V, 70°C)



Note: This derating factor applies to all routing and propagation delays.

A1225A Timing Characteristics

(Worst-Case Commercial Conditions, V_{CC} = 4.75 V, T_J = 70°C)

Logic Module Propagation Delays[1]		'–2' Speed		'–1' Speed		'Std' Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{PD1}	Single Module		3.8		4.3		5.0	ns
t_{CO}	Sequential Clk to Q		3.8		4.3		5.0	ns
t_{GO}	Latch G to Q		3.8		4.3		5.0	ns
t_{RS}	Flip-Flop (Latch) Reset to Q		3.8		4.3		5.0	ns
Predicted Routing Delays[2]								
t_{RD1}	FO=1 Routing Delay		1.1		1.2		1.4	ns
t_{RD2}	FO=2 Routing Delay		1.7		1.9		2.2	ns
t_{RD3}	FO=3 Routing Delay		2.3		2.6		3.0	ns
t_{RD4}	FO=4 Routing Delay		2.8		3.1		3.7	ns
t_{RD8}	FO=8 Routing Delay		4.4		4.9		5.8	ns
Sequential Timing Characteristics[3,4]								
t_{SUD}	Flip-Flop (Latch) Data Input Setup	0.4		0.4		0.5		ns
t_{HD}	Flip-Flop (Latch) Data Input Hold	0.0		0.0		0.0		ns
t_{SUENA}	Flip-Flop (Latch) Enable Setup	0.8		0.9		1.0		ns
t_{HENA}	Flip-Flop (Latch) Enable Hold	0.0		0.0		0.0		ns
t_{WCLKA}	Flip-Flop (Latch) Clock Active Pulse Width	4.5		5.0		6.0		ns
t_{WASYN}	Flip-Flop (Latch) Asynchronous Pulse Width	4.5		5.0		6.0		ns
t_A	Flip-Flop Clock Input Period	9.4		11.0		13.0		ns
t_{INH}	Input Buffer Latch Hold	0.0		0.0		0.0		ns
t_{INSU}	Input Buffer Latch Setup	0.4		0.4		0.5		ns
t_{OUTH}	Output Buffer Latch Hold	0.0		0.0		0.0		ns
t_{OUTSU}	Output Buffer Latch Setup	0.4		0.4		0.5		ns
f_{MAX}	Flip-Flop (Latch) Clock Frequency		105.0		90.0		75.0	MHz

Notes:

1. *For dual-module macros, use $t_{PD1} + t_{RD1} + t_{PDn}$, $t_{CO} + t_{RD1} + t_{PDn}$ or $t_{PD1} + t_{RD1} + t_{SUD}$, whichever is appropriate.*
2. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*
3. *Data applies to macros based on the S-module. Timing parameters for sequential macros constructed from C-modules can be obtained from the DirectTime Analyzer utility.*
4. *Setup and hold timing parameters for the Input Buffer Latch are defined with respect to the PAD and the D input. External setup/hold timing parameters must account for delay from an external PAD signal to the G inputs. Delay from an external PAD signal to the G input subtracts (adds) to the internal setup (hold) time.*

A1225A Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

Input Module Propagation Delays			‘–2 Speed		‘–1’ Speed		‘Std’ Speed		
Parameter	**Description**		**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{INYH}	Pad to Y High			2.9		3.3		3.8	ns
t_{INYL}	Pad to Y Low			2.6		3.0		3.5	ns
t_{INGH}	G to Y High			5.0		5.7		6.6	ns
t_{INGL}	G to Y Low			4.7		5.4		6.3	ns
Input Module Predicted Routing Delays[1]									
t_{IRD1}	FO=1 Routing Delay			4.1		4.6		5.4	ns
t_{IRD2}	FO=2 Routing Delay			4.6		5.2		6.1	ns
t_{IRD3}	FO=3 Routing Delay			5.3		6.0		7.1	ns
t_{IRD4}	FO=4 Routing Delay			5.7		6.4		7.6	ns
t_{IRD8}	FO=8 Routing Delay			7.4		8.3		9.8	ns
Global Clock Network									
t_{CKH}	Input Low to High	FO = 32 FO = 256		10.2 11.8		11.0 13.0		12.8 15.7	ns
t_{CKL}	Input High to Low	FO = 32 FO = 256		10.2 12.0		11.0 13.2		12.8 15.9	ns
t_{PWH}	Minimum Pulse Width High	FO = 32 FO = 256	3.4 3.8		4.1 4.5		4.5 5.0		ns
t_{PWL}	Minimum Pulse Width Low	FO = 32 FO = 256	3.4 3.8		4.1 4.5		4.5 5.0		ns
t_{CKSW}	Maximum Skew	FO = 32 FO = 256		0.7 3.5		0.7 3.5		0.7 3.5	ns
t_{SUEXT}	Input Latch External Setup	FO = 32 FO = 256	0.0 0.0		0.0 0.0		0.0 0.0		ns
t_{HEXT}	Input Latch External Hold	FO = 32 FO = 256	7.0 11.2		7.0 11.2		7.0 11.2		ns
t_P	Minimum Period	FO = 32 FO = 256	7.7 8.1		8.3 8.8		9.1 10.0		ns
f_{MAX}	Maximum Frequency	FO = 32 FO = 256		130.0 125.0		120.0 115.0		110.0 100.0	MHz

Note:

1. *These parameters should be used for estimating device performance. Optimization techniques may further reduce delays by 0 to 4 ns. Routing delays are for typical designs across worst-case operating conditions. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

A1225A Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

Output Module Timing		'–2 Speed		'–1' Speed		'Std' Speed		
Parameter	Description	Min.	Max.	Min.	Max.	Min.	Max.	Units
TTL Output Module Timing[1, 2]								
t_{DLH}	Data to Pad High		8.0		9.0		10.6	ns
t_{DHL}	Data to Pad Low		10.1		11.4		13.4	ns
t_{ENZH}	Enable Pad Z to High		8.9		10.0		11.8	ns
t_{ENZL}	Enable Pad Z to Low		11.6		13.2		15.5	ns
t_{ENHZ}	Enable Pad High to Z		7.1		8.0		9.4	ns
t_{ENLZ}	Enable Pad Low to Z		8.3		9.5		11.1	ns
t_{GLH}	G to Pad High		8.9		10.2		11.9	ns
t_{GHL}	G to Pad Low		11.2		12.7		14.9	ns
d_{TLH}	Delta Low to High		0.07		0.08		0.09	ns/pF
d_{THL}	Delta High to Low		0.12		0.13		0.16	ns/pF
CMOS Output Module Timing[1, 2]								
t_{DLH}	Data to Pad High		10.1		11.5		13.5	ns
t_{DHL}	Data to Pad Low		8.4		9.6		11.2	ns
t_{ENZH}	Enable Pad Z to High		8.9		10.0		11.8	ns
t_{ENZL}	Enable Pad Z to Low		11.6		13.2		15.5	ns
t_{ENHZ}	Enable Pad High to Z		7.1		8.0		9.4	ns
t_{ENLZ}	Enable Pad Low to Z		8.3		9.5		11.1	ns
t_{GLH}	G to Pad High		8.9		10.2		11.9	ns
t_{GHL}	G to Pad Low		11.2		12.7		14.9	ns
d_{TLH}	Delta Low to High		0.12		0.13		0.16	ns/pF
d_{THL}	Delta High to Low		0.09		0.10		0.12	ns/pF

Notes:

1. *Delays based on 50 pF loading.*
2. *SSO information can be found in the "Simultaneous Switching Output Limits for Actel FPGAs" application note on page 4-125.*

A1240A Timing Characteristics

(Worst-Case Commercial Conditions, V_{CC} = 4.75 V, T_J = 70°C)

Logic Module Propagation Delays[1]		'–2' Speed		'–1' Speed		'Std' Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{PD1}	Single Module		3.8		4.3		5.0	ns
t_{CO}	Sequential Clk to Q		3.8		4.3		5.0	ns
t_{GO}	Latch G to Q		3.8		4.3		5.0	ns
t_{RS}	Flip-Flop (Latch) Reset to Q		3.8		4.3		5.0	ns
Predicted Routing Delays[2]								
t_{RD1}	FO=1 Routing Delay		1.4		1.5		1.8	ns
t_{RD2}	FO=2 Routing Delay		1.7		2.0		2.3	ns
t_{RD3}	FO=3 Routing Delay		2.3		2.6		3.0	ns
t_{RD4}	FO=4 Routing Delay		3.1		3.5		4.1	ns
t_{RD8}	FO=8 Routing Delay		4.7		5.4		6.3	ns
Sequential Timing Characteristics[3, 4]								
t_{SUD}	Flip-Flop (Latch) Data Input Setup	0.4		0.4		0.5		ns
t_{HD}	Flip-Flop (Latch) Data Input Hold	0.0		0.0		0.0		ns
t_{SUENA}	Flip-Flop (Latch) Enable Setup	0.8		0.9		1.0		ns
t_{HENA}	Flip-Flop (Latch) Enable Hold	0.0		0.0		0.0		ns
t_{WCLKA}	Flip-Flop (Latch) Clock Active Pulse Width	4.5		6.0		6.5		ns
t_{WASYN}	Flip-Flop (Latch) Asynchronous Pulse Width	4.5		6.0		6.5		ns
t_A	Flip-Flop Clock Input Period	9.8		12.0		15.0		ns
t_{INH}	Input Buffer Latch Hold	0.0		0.0		0.0		ns
t_{INSU}	Input Buffer Latch Setup	0.4		0.4		0.5		ns
t_{OUTH}	Output Buffer Latch Hold	0.0		0.0		0.0		ns
t_{OUTSU}	Output Buffer Latch Setup	0.4		0.4		0.5		ns
f_{MAX}	Flip-Flop (Latch) Clock Frequency		100.0		80.0		66.0	MHz

Notes:

1. *For dual-module macros, use $t_{PD1} + t_{RD1} + t_{PDn}$, $t_{CO} + t_{RD1} + t_{PDn}$ or $t_{PD1} + t_{RD1} + t_{SUD}$, whichever is appropriate.*
2. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*
3. *Data applies to macros based on the S-module. Timing parameters for sequential macros constructed from C-modules can be obtained from the DirectTime Analyzer utility.*
4. *Setup and hold timing parameters for the Input Buffer Latch are defined with respect to the PAD and the D input. External setup/hold timing parameters must account for delay from an external PAD signal to the G inputs. Delay from an external PAD signal to the G input subtracts (adds) to the internal setup (hold) time.*

A1240A Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

Input Module Propagation Delays			'–2' Speed		'–1' Speed		'Std' Speed		
Parameter	**Description**		**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{INYH}	Pad to Y High			2.9		3.3		3.8	ns
t_{INYL}	Pad to Y Low			2.6		3.0		3.5	ns
t_{INGH}	G to Y High			5.0		5.7		6.6	ns
t_{INGL}	G to Y Low			4.7		5.4		6.3	ns
Input Module Predicted Routing Delays[1]									
t_{IRD1}	FO=1 Routing Delay			4.2		4.8		5.6	ns
t_{IRD2}	FO=2 Routing Delay			4.8		5.4		6.4	ns
t_{IRD3}	FO=3 Routing Delay			5.4		6.1		7.2	ns
t_{IRD4}	FO=4 Routing Delay			5.9		6.7		7.9	ns
t_{IRD8}	FO=8 Routing Delay			7.9		8.9		10.5	ns
Global Clock Network									
t_{CKH}	Input Low to High	FO = 32		10.2		11.0		12.8	
		FO = 256		11.8		13.0		15.7	ns
t_{CKL}	Input High to Low	FO = 32		10.2		11.0		12.8	
		FO = 256		12.0		13.2		15.9	ns
t_{PWH}	Minimum Pulse Width High	FO = 32	3.8		4.5		5.5		
		FO = 256	4.1		5.0		5.8		ns
t_{PWL}	Minimum Pulse Width Low	FO = 32	3.8		4.5		5.5		
		FO = 256	4.1		5.0		5.8		ns
t_{CKSW}	Maximum Skew	FO = 32		0.5		0.5		0.5	
		FO = 256		2.5		2.5		2.5	ns
t_{SUEXT}	Input Latch External Setup	FO = 32	0.0		0.0		0.0		
		FO = 256	0.0		0.0		0.0		ns
t_{HEXT}	Input Latch External Hold	FO = 32	7.0		7.0		7.0		
		FO = 256	11.2		11.2		11.2		ns
t_P	Minimum Period	FO = 32	8.1		9.1		11.1		
		FO = 256	8.8		10.0		11.7		ns
f_{MAX}	Maximum Frequency	FO = 32		125.0		110.0		90.0	
		FO = 256		115.0		100.0		85.0	MHz

Note:

1. *These parameters should be used for estimating device performance. Optimization techniques may further reduce delays by 0 to 4 ns. Routing delays are for typical designs across worst-case operating conditions. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

A1240A Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

Output Module Timing		'–2' Speed		'–1' Speed		'Std' Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
TTL Output Module Timing[1, 2]								
t_{DLH}	Data to Pad High		8.0		9.0		10.6	ns
t_{DHL}	Data to Pad Low		10.1		11.4		13.4	ns
t_{ENZH}	Enable Pad Z to High		8.9		10.0		11.8	ns
t_{ENZL}	Enable Pad Z to Low		11.7		13.2		15.5	ns
t_{ENHZ}	Enable Pad High to Z		7.1		8.0		9.4	ns
t_{ENLZ}	Enable Pad Low to Z		8.4		9.5		11.1	ns
t_{GLH}	G to Pad High		9.0		10.2		11.9	ns
t_{GHL}	G to Pad Low		11.2		12.7		14.9	ns
d_{TLH}	Delta Low to High		0.07		0.08		0.09	ns/pF
d_{THL}	Delta High to Low		0.12		0.13		0.16	ns/pF
CMOS Output Module Timing[1, 2]								
t_{DLH}	Data to Pad High		10.2		11.5		13.5	ns
t_{DHL}	Data to Pad Low		8.4		9.6		11.2	ns
t_{ENZH}	Enable Pad Z to High		8.9		10.0		11.8	ns
t_{ENZL}	Enable Pad Z to Low		11.7		13.2		15.5	ns
t_{ENHZ}	Enable Pad High to Z		7.1		8.0		9.4	ns
t_{ENLZ}	Enable Pad Low to Z		8.4		9.5		11.1	ns
t_{GLH}	G to Pad High		9.0		10.2		11.9	ns
t_{GHL}	G to Pad Low		11.2		12.7		14.9	ns
d_{TLH}	Delta Low to High		0.12		0.13		0.16	ns/pF
d_{THL}	Delta High to Low		0.09		0.10		0.12	ns/pF

Notes:

1. *Delays based on 50 pF loading.*
2. *SSO information can be found in the "Simultaneous Switching Output Limits for Actel FPGAs" application note on page 4-125.*

A1280A Timing Characteristics

(Worst-Case Commercial Conditions, V_{CC} = 4.75 V, T_J = 70°C)

Logic Module Propagation Delays[1]		'–2' Speed		'–1' Speed		'Std' Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{PD1}	Single Module		3.8		4.3		5.0	ns
t_{CO}	Sequential Clk to Q		3.8		4.3		5.0	ns
t_{GO}	Latch G to Q́		3.8		4.3		5.0	ns
t_{RS}	Flip-Flop (Latch) Reset to Q		3.8		4.3		5.0	ns
Predicted Routing Delays[2]								
t_{RD1}	FO=1 Routing Delay		1.7		2.0		2.3	ns
t_{RD2}	FO=2 Routing Delay		2.5		2.8		3.3	ns
t_{RD3}	FO=3 Routing Delay		3.0		3.4		4.0	ns
t_{RD4}	FO=4 Routing Delay		3.7		4.2		4.9	ns
t_{RD8}	FO=8 Routing Delay		6.7		7.5		8.8	ns
Sequential Timing Characteristics[3,4]								
t_{SUD}	Flip-Flop (Latch) Data Input Setup	0.4		0.4		0.5		ns
t_{HD}	Flip-Flop (Latch) Data Input Hold	0.0		0.0		0.0		ns
t_{SUENA}	Flip-Flop (Latch) Enable Setup	0.8		0.9		1.0		ns
t_{HENA}	Flip-Flop (Latch) Enable Hold	0.0		0.0		0.0		ns
t_{WCLKA}	Flip-Flop (Latch) Clock Active Pulse Width	5.5		6.0		7.0		ns
t_{WASYN}	Flip-Flop (Latch) Asynchronous Pulse Width	5.5		6.0		7.0		ns
t_A	Flip-Flop Clock Input Period	11.7		13.3		18.0		ns
t_{INH}	Input Buffer Latch Hold	0.0		0.0		0.0		ns
t_{INSU}	Input Buffer Latch Setup	0.4		0.4		0.5		ns
t_{OUTH}	Output Buffer Latch Hold	0.0		0.0		0.0		ns
t_{OUTSU}	Output Buffer Latch Setup	0.4		0.4		0.5		ns
f_{MAX}	Flip-Flop (Latch) Clock Frequency		85.0		75.0		50.0	MHz

Notes:

1. *For dual-module macros, use $t_{PD1} + t_{RD1} + t_{PDn}$, $t_{CO} + t_{RD1} + t_{PDn}$, or $t_{PD1} + t_{RD1} + t_{SUD}$, whichever is appropriate.*
2. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*
3. *Data applies to macros based on the S-module. Timing parameters for sequential macros constructed from C-modules can be obtained from the DirectTime Analyzer utility.*
4. *Setup and hold timing parameters for the Input Buffer Latch are defined with respect to the PAD and the D input. External setup/hold timing parameters must account for delay from an external PAD signal to the G inputs. Delay from an external PAD signal to the G input subtracts (adds) to the internal setup (hold) time.*

A1280A Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

Input Module Propagation Delays			'–2' Speed		'–1' Speed		'Std' Speed		
Parameter	**Description**		**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{INYH}	Pad to Y High			2.9		3.3		3.8	ns
t_{INYL}	Pad to Y Low			2.7		3.0		3.5	ns
t_{INGH}	G to Y High			5.0		5.7		6.6	ns
t_{INGL}	G to Y Low			4.8		5.4		6.3	ns
Input Module Predicted Routing Delays[1]									
t_{IRD1}	FO=1 Routing Delay			4.6		5.1		6.0	ns
t_{IRD2}	FO=2 Routing Delay			5.2		5.9		6.9	ns
t_{IRD3}	FO=3 Routing Delay			5.6		6.3		7.4	ns
t_{IRD4}	FO=4 Routing Delay			6.5		7.3		8.6	ns
t_{IRD8}	FO=8 Routing Delay			9.4		10.5		12.4	ns
Global Clock Network									
t_{CKH}	Input Low to High	FO = 32		10.2		11.0		12.8	
		FO = 384		13.1		14.6		17.2	ns
t_{CKL}	Input High to Low	FO = 32		10.2		11.0		12.8	
		FO = 384		13.3		14.9		17.5	ns
t_{PWH}	Minimum Pulse Width High	FO = 32	5.0		5.5		6.6		
		FO = 384	5.8		6.4		7.6		ns
t_{PWL}	Minimum Pulse Width Low	FO = 32	5.0		5.5		6.6		
		FO = 384	5.8		6.4		7.6		ns
t_{CKSW}	Maximum Skew	FO = 32		0.5		0.5		0.5	
		FO = 384		2.5		2.5		2.5	ns
t_{SUEXT}	Input Latch External Setup	FO = 32	0.0		0.0		0.0		
		FO = 384	0.0		0.0		0.0		ns
t_{HEXT}	Input Latch External Hold	FO = 32	7.0		7.0		7.0		
		FO = 384	11.2		11.2		11.2		ns
t_P	Minimum Period	FO = 32	9.6		11.2		13.3		
		FO = 384	10.6		12.6		15.3		ns
f_{MAX}	Maximum Frequency	FO = 32		105.0		90.0		75.0	
		FO = 384		95.0		80.0		65.0	MHz

Note:

1. *These parameters should be used for estimating device performance. Optimization techniques may further reduce delays by 0 to 4 ns. Routing delays are for typical designs across worst-case operating conditions. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

A1280A Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

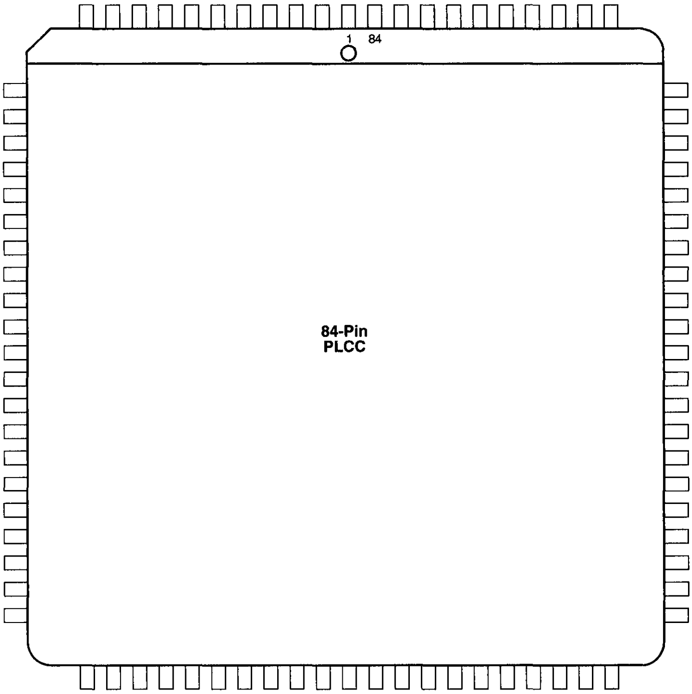
Output Module Timing		'–2' Speed		'–1' Speed		'Std' Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
TTL Output Module **Timing**[1, 2]								
t_{DLH}	Data to Pad High		8.1		9.0		10.6	ns
t_{DHL}	Data to Pad Low		10.2		11.4		13.4	ns
t_{ENZH}	Enable Pad Z to High		9.0		10.0		11.8	ns
t_{ENZL}	Enable Pad Z to Low		11.8		13.2		15.5	ns
t_{ENHZ}	Enable Pad High to Z		7.1		8.0		9.4	ns
t_{ENLZ}	Enable Pad Low to Z		8.4		9.5		11.1	ns
t_{GLH}	G to Pad High		9.0		10.2		11.9	ns
t_{GHL}	G to Pad Low		11.3		12.7		14.9	ns
d_{TLH}	Delta Low to High		0.07		0.08		0.09	ns/pF
d_{THL}	Delta High to Low		0.12		0.13		0.16	ns/pF
CMOS Output Module **Timing**[1, 2]								
t_{DLH}	Data to Pad High		10.3		11.5		13.5	ns
t_{DHL}	Data to Pad Low		8.5		9.6		11.2	ns
t_{ENZH}	Enable Pad Z to High		9.0		10.0		11.8	ns
t_{ENZL}	Enable Pad Z to Low		11.8		13.2		15.5	ns
t_{ENHZ}	Enable Pad High to Z		7.1		8.0		9.4	ns
t_{ENLZ}	Enable Pad Low to Z		8.4		9.5		11.1	ns
t_{GLH}	G to Pad High		9.0		10.2		11.9	ns
t_{GHL}	G to Pad Low		11.3		12.7		14.9	ns
d_{TLH}	Delta Low to High		0.12		0.13		0.16	ns/pF
d_{THL}	Delta High to Low		0.09		0.10		0.12	ns/pF

Notes::

1. *Delays based on 50 pF loading.*
2. *SSO information can be found in the "Simultaneous Switching Output Limits for Actel FPGAs" application note on page 4-125.*

Package Pin Assignments

84-Pin PLCC



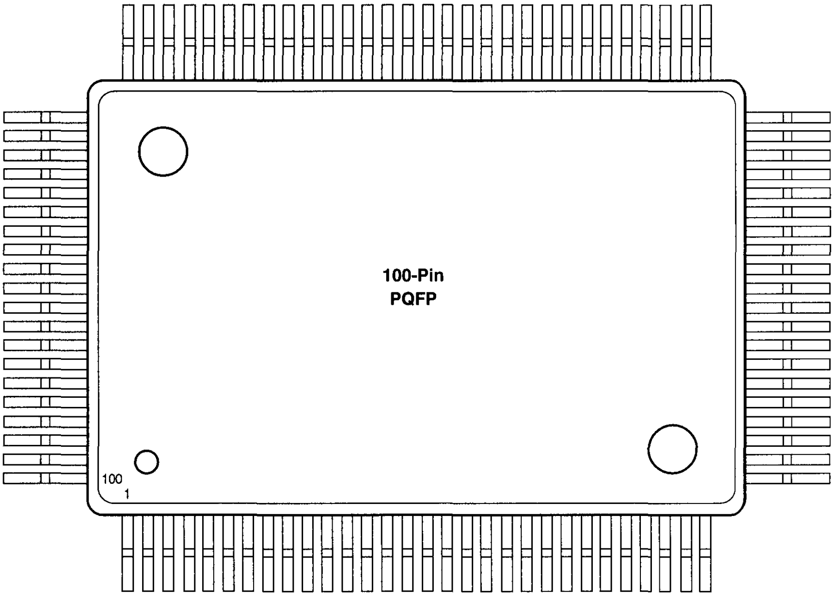
Signal	A1225A Function	A1240A Function	A1280A Function
2	CLKB, I/O	CLKB, I/O	CLKB, I/O
4	PRB, I/O	PRB, I/O	PRB, I/O
6	GND	GND	GND
10	DCLK, I/O	DCLK, I/O	DCLK, I/O
12	MODE	MODE	MODE
22	VCC	VCC	VCC
23	VCC	VCC	VCC
28	GND	GND	GND
43	VCC	VCC	VCC
49	GND	GND	GND
63	GND	GND	GND
64	VCC	VCC	VCC
65	VCC	VCC	VCC
70	GND	GND	GND
76	SDI, I/O	SDI, I/O	SDI, I/O
81	PRA, I/O	PRA, I/O	PRA, I/O
83	CLKA, I/O	CLKA, I/O	CLKA, I/O
84	VCC	VCC	VCC

Notes:

1. *All unlisted pin numbers are user I/Os.*
2. *MODE pin should be terminated to GND through a 10K resistor to enable Actionprobe usage, otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

100-Pin PQFP



Pin Number	A1225A Function
2	DCLK, I/O
4	MODE
9	GND
16	VCC
17	VCC
22	GND
34	GND
40	VCC
46	GND
57	GND
64	GND
65	VCC

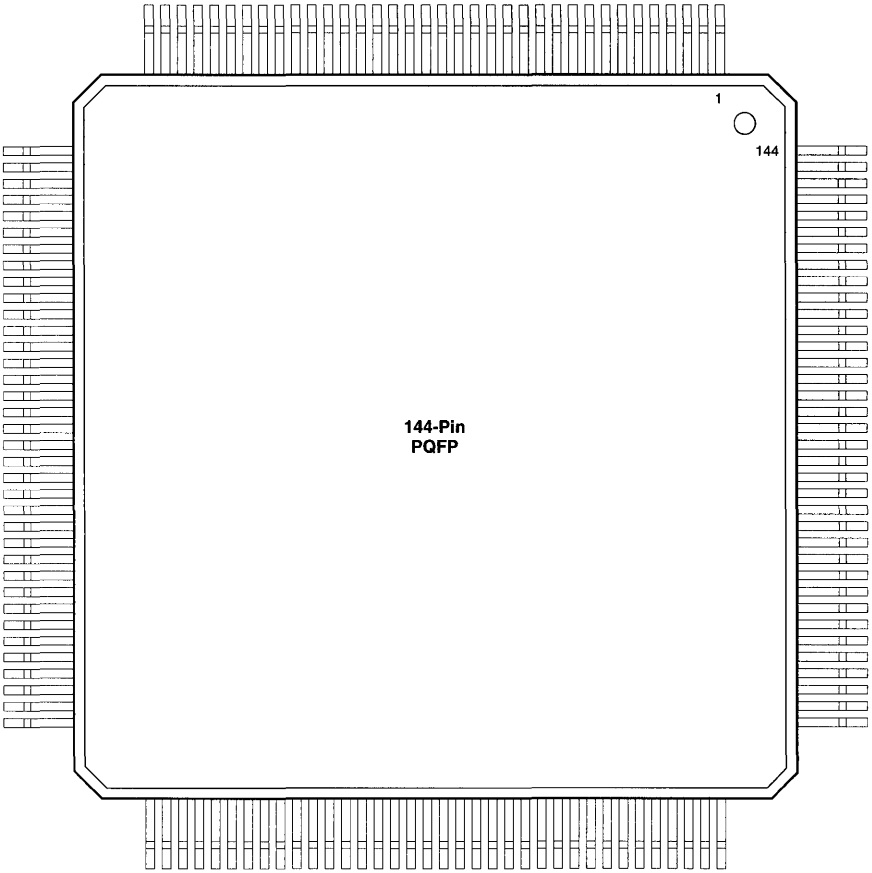
Pin Number	A1225A Function
66	VCC
67	VCC
72	GND
79	SDI, I/O
84	GND
87	PRA, I/O
89	CLKA, I/O
90	VCC
92	CLKB, I/O
94	PRB, I/O
96	GND

Notes:

1. *All unlisted pin numbers are user I/Os.*
2. *MODE pin should be terminated to GND through a 10K resistor to enable Actionprobe usage, otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

144-Pin PQFP



144-Pin PQFP

Pin Number	A1240A Function
2	MODE
9	GND
10	GND
11	GND
18	VCC
19	VCC
20	VCC
21	VCC
28	GND
29	GND
30	GND
44	GND
45	GND
46	GND
54	VCC
55	VCC
56	VCC
64	GND
65	GND
79	GND
80	GND
81	GND
88	GND

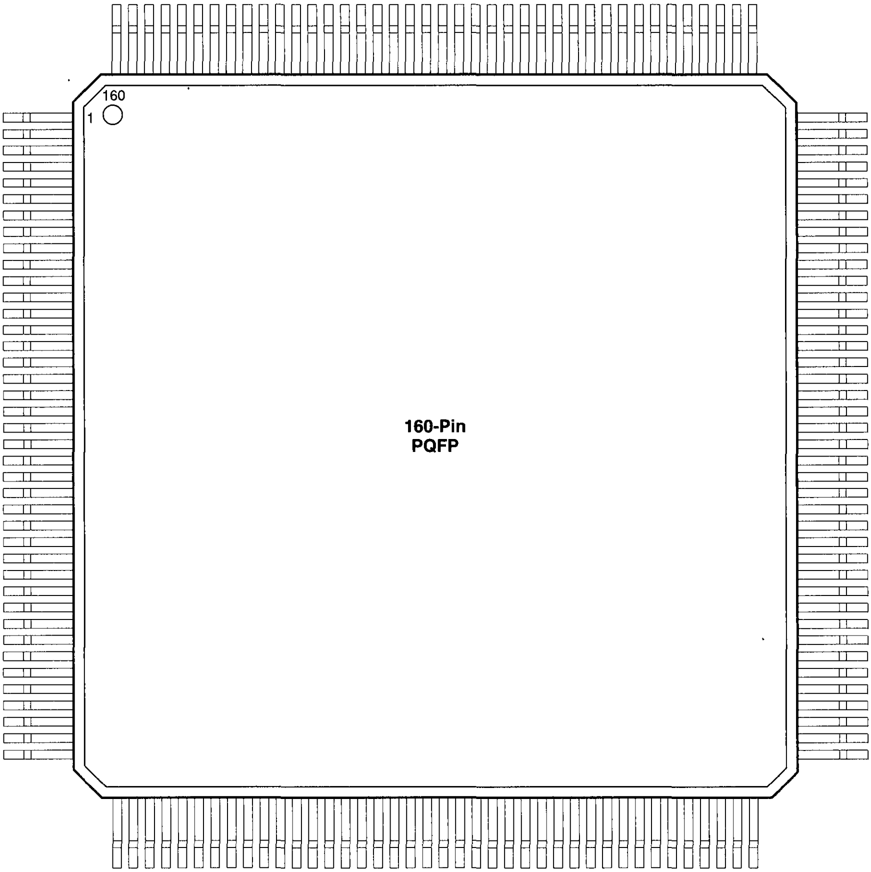
Pin Number	A1240A Function
89	VCC
90	VCC
91	VCC
92	VCC
93	VCC
100	GND
101	GND
102	GND
110	SDI, I/O
116	GND
117	GND
118	GND
123	PRA, I/O
125	CLKA, I/O
126	VCC
127	VCC
128	VCC
130	CLKB, I/O
132	PRB, I/O
136	GND
137	GND
138	GND
144	DCLK, I/O

Notes:

1. *All unlisted pin numbers are user I/Os.*
2. *MODE pin should be terminated to GND through a 10K resistor to enable Actionprobe usage, otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

160-Pin PQFP



160-Pin PQFP

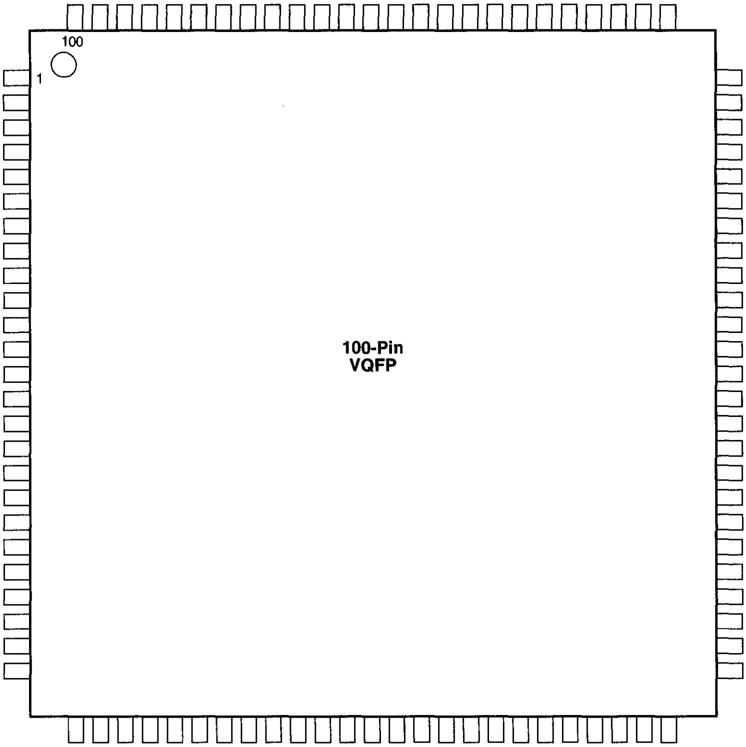
Pin Number	A1280A Function
2	DCLK, I/O
6	VCC
11	GND
16	PRB, I/O
18	CLKB, I/O
20	VCC
21	CLKA, I/O
23	PRA, I/O
30	GND
35	VCC
38	SDI, I/O
40	GND
44	GND
49	GND
54	VCC
57	VCC
58	VCC
59	GND
60	VCC
61	GND
64	GND
69	GND
80	GND
86	VCC
89	GND
98	VCC
99	GND
109	GND
114	VCC
120	GND
125	GND
130	GND
135	VCC
138	VCC
139	VCC
140	GND
145	GND
150	VCC
155	GND
159	MODE
160	GND

Notes:

1. *All unlisted pin numbers are user I/Os.*
2. *MODE pin should be terminated to GND through a 10K resistor to enable Actionprobe usage, otherwise it can be terminated directly to GND.*

Package Pin Assigments (continued)

100-Pin VQFP



100-Pin VQFP

Pin Number	A1225A Function
2	MODE
7	GND
14	VCC
15	VCC
20	GND
32	GND
38	VCC
44	GND
55	GND
62	GND
63	VCC
64	VCC

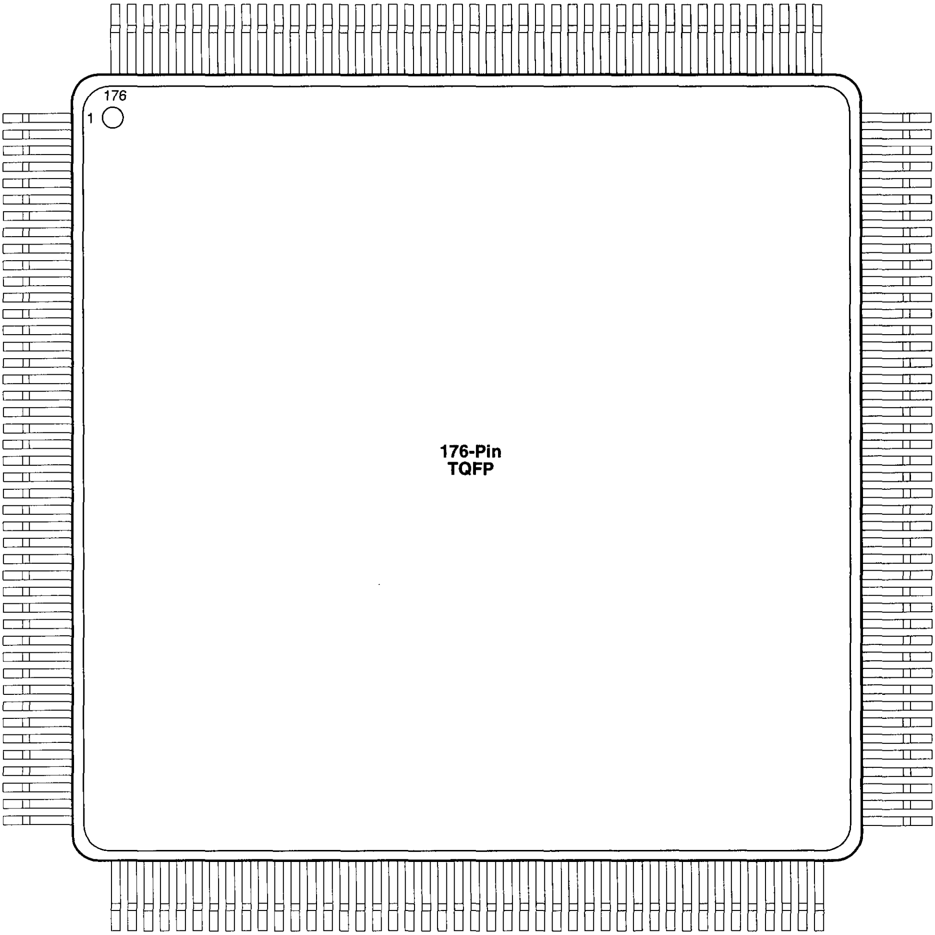
Pin Number	A1225A Function
65	VCC
70	GND
77	SDI, I/O
82	GND
85	PRA, I/O
87	CLKA, I/O
88	VCC
90	CLKB, I/O
92	PRB, I/O
94	GND
100	DCLK, I/O

Notes:

1. *All unlisted pin numbers are user I/Os.*
2. *MODE pin should be terminated to GND through a 10K resistor to enable Actionprobe usage, otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

176-Pin TQFP



176-Pin TQFP

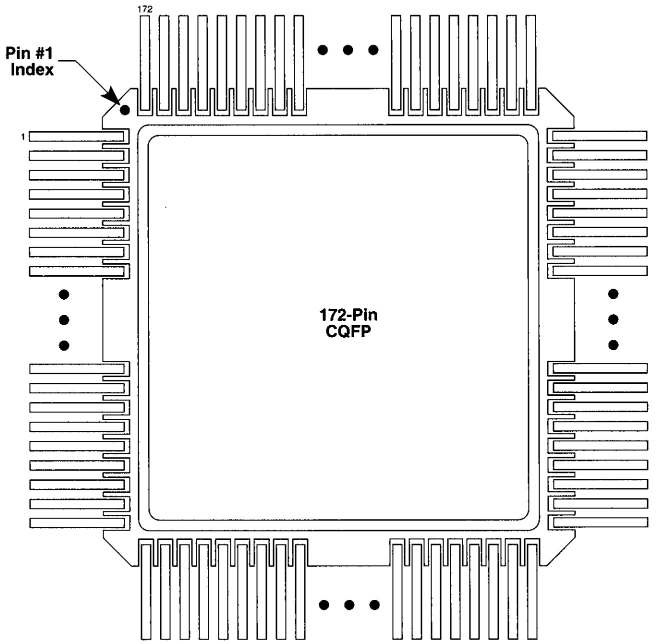
Pin Number	A1240A Function	A1280A Function
1	GND	GND
2	MODE	MODE
8	NC	NC
10	NC	I/O
11	NC	I/O
13	NC	VCC
18	GND	GND
19	NC	I/O
20	NC	I/O
22	NC	I/O
23	GND	GND
24	NC	VCC
25	VCC	VCC
26	NC	I/O
27	NC	I/O
28	VCC	VCC
29	NC	I/O
33	NC	NC
37	NC	I/O
38	NC	NC
45	GND	GND
52	NC	VCC
54	NC	I/O
55	NC	I/O
57	NC	NC
61	NC	I/O
64	NC	I/O
66	NC	I/O
67	GND	GND
68	VCC	VCC
74	NC	I/O
77	NC	NC
78	NC	I/O
80	NC	I/O
82	NC	VCC
86	NC	I/O
89	GND	GND
96	NC	I/O
97	NC	I/O
101	NC	NC
103	NC	I/O
106	GND	GND
107	NC	I/O
108	NC	I/O
109	GND	GND
110	VCC	VCC
111	GND	GND
112	VCC	VCC
113	VCC	VCC
114	NC	I/O
115	NC	I/O
116	NC	VCC
121	NC	NC
124	NC	I/O
125	NC	I/O
126	NC	NC
133	GND	GND
135	SDI, I/O	SDI, I/O
136	NC	I/O
140	NC	VCC
143	NC	I/O
144	NC	I/O
145	NC	NC
147	NC	I/O
151	NC	I/O
152	PRA, I/O	PRA, I/O
154	CLKA, I/O	CLKA, I/O
155	VCC	VCC
156	GND	GND
158	CLKB, I/O	CLKB, I/O
160	PRB, I/O	PRB, I/O
161	NC	I/O
165	NC	NC
166	NC	I/O
168	NC	I/O
170	NC	VCC
173	NC	I/O
175	DCLK, I/O	DCLK, I/O

Notes:

1. *NC: Denotes No Connection*
2. *All unlisted pin numbers are user I/Os.*
3. *MODE pin should be terminated to GND through a 10K resistor to enable Actionprobe usage, otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

172-Pin CQFP



Pin Number	A1280A Function
1	MODE
7	GND
12	VCC
17	GND
22	GND
23	VCC
24	VCC
27	VCC
32	GND
37	GND
50	VCC
55	GND
65	GND
66	VCC
75	GND
80	VCC
98	GND
103	GND
106	GND

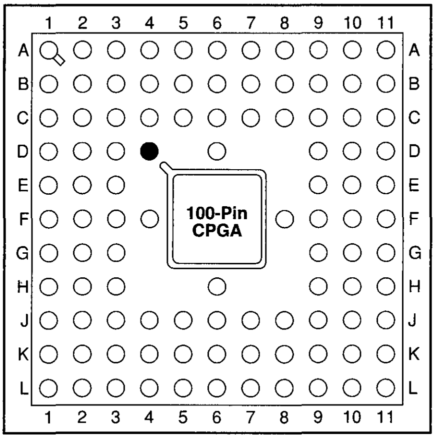
Pin Number	A1280A Function
107	VCC
108	GND
109	VCC
110	VCC
113	VCC
118	GND
123	GND
131	SDI, I/O
136	VCC
141	GND
148	PRA, I/O
150	CLKA, I/O
151	VCC
152	GND
154	CLKB, I/O
156	PRB, I/O
161	GND
166	VCC
171	DCLK, I/O

Notes:

1. *All unlisted pin numbers are user I/Os.*
2. *MODE pin should be terminated to GND through a 10K resistor to enable Actionprobe usage, otherwise it can be terminated directly to GND.*

Package Pin Assigments (continued)

100-Pin CPGA



● Orientation Pin

Pin Number	A1225A Function
A4	PRB, I/O
A7	PRA, I/O
B6	VCC
C2	MODE
C3	DCLK, I/O
C5	GND
C6	CLKA, I/O
C7	GND
C8	SDI, I/O
D6	CLKB, I/O
D10	GND
E3	GND

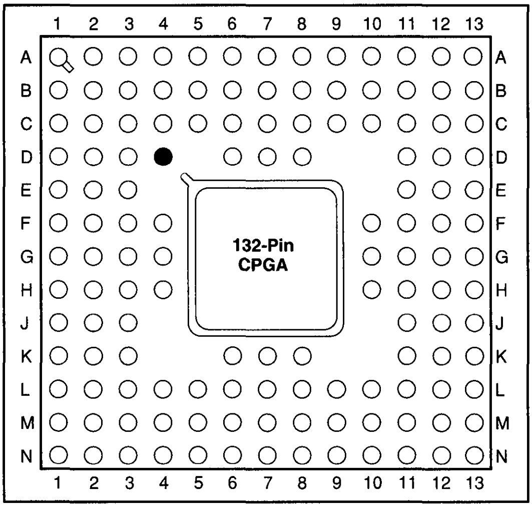
Pin Number	A1225A Function
E11	VCC
F3	VCC
F9	VCC
F10	VCC
F11	GND
G1	VCC
G3	GND
G9	GND
J5	GND
J7	GND
K6	VCC

Note:

1. *All unlisted pin numbers are user I/Os.*
2. *MODE pin should be terminated to GND through a 10K resistor to enable Actionprobe usage, otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

132-Pin CPGA



● Orientation Pin

Pin Number	A1240A Function
A1	MODE
B5	GND
B6	CLKB, I/O
B7	CLKA, I/O
B8	PRA, I/O
B9	GND
B12	SDI, I/O
C3	DCLK, I/O
C5	GND
C6	PRB, I/O
C7	VCC
C9	GND
D7	VCC
E3	GND
E11	GND
E12	GND
F4	GND

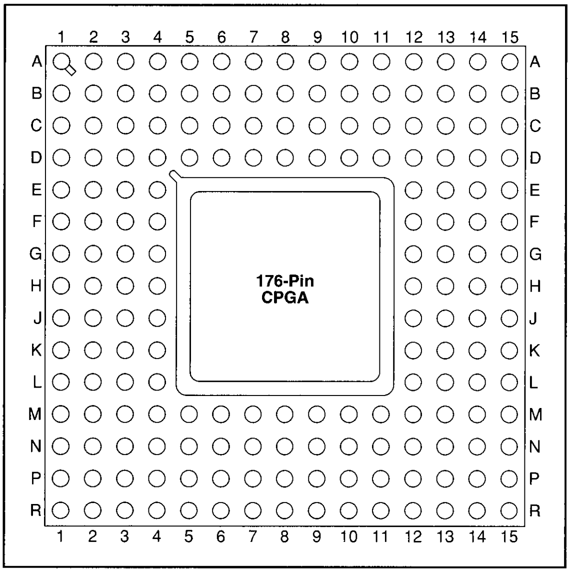
Pin Number	A1240A Function
G2	VCC
G3	VCC
G4	VCC
G10	VCC
G11	VCC
G12	VCC
G13	VCC
H13	GND
J2	GND
J3	GND
J11	GND
K7	VCC
K12	GND
L5	GND
L7	VCC
L9	GND
M9	GND

Notes:

1. *All unlisted pin numbers are user I/Os.*
2. *MODE pin should be terminated to GND through a 10K resistor to enable Actionprobe usage, otherwise it can be terminated directly to GND.*

Package Pin Assignments (continued)

176-Pin CPGA



Pin Number	A1280A Function
A3	CLKA, I/O
B3	DCLK, I/O
B8	CLKB, I/O
B14	SDI, I/O
C3	MODE
C8	GND
C9	PRA, I/O
D4	GND
D5	VCC
D6	GND
D7	PRB, I/O
D8	VCC
D10	GND
D11	VCC
D12	GND
E4	GND
E12	GND
F4	VCC
F12	GND
G4	GND
G12	VCC

Pin Number	A1280A Function
H2	VCC
H3	VCC
H4	GND
H12	GND
H13	VCC
H14	VCC
J4	VCC
J12	GND
J13	GND
J14	VCC
K4	GND
K12	GND
L4	GND
M4	GND
M5	VCC
M6	GND
M8	GND
M10	GND
M11	VCC
M12	GND
N8	VCC

Notes:

1. *All unlisted pin numbers are user I/Os.*
2. *MODE pin should be terminated to GND through a 10K resistor to enable Actionprobe usage, otherwise it can be terminated directly to GND.*

Actel

Military Field Programmable Gate Arrays

Features

- Highly Predictable Performance with 100 Percent Automatic Placement and Routing
- Device Sizes from 1200 to 10,000 gates (up to 25,000 PLD equivalent gates)
- Up to 4, Fast, Low-Skew Clock Networks
- Up to 228 User-Programmable I/O Pins
- More Than 500 Macro Functions
- Replaces up to 250 TTL Packages
- Replaces up to 100 20-pin PAL Packages
- Up to 1153 Dedicated Flip-Flops
- I/O Drive to 10 mA
- Devices Available to DESC SMD
- CQFP and CPGA Packaging
- Nonvolatile, User Programmable
- Logic Fully Tested Prior to Shipment

ACT 3 Features

- Highest-Performance, Highest-Capacity FPGA Family
- System Performance to 60 MHz over Military Temperature
- Low-Power 0.8-micron CMOS Technology

1200XL Features

- Pin for Pin Compatible with ACT 2
- System Performance to 50 MHz over Military Temperature
- Low-Power 0.6-micron CMOS Technology

ACT 2 Features

- Best-Value, High-Capacity FPGA Family
- System Performance to 40 MHz over Military Temperature
- Low-Power 1.0-micron CMOS Technology

ACT 1 Features

- Lowest-Cost FPGA Family
- System Performance to 20 MHz over Military Temperature
- Low-Power 1.0-micron CMOS Technology

Product Family Profile

Family	ACT 3			1200XL	ACT 2		ACT 1	
Device	A1425A	A1460A	A14100A	A1280XL	A1240A	A1280A	A1010B	A1020B
Capacity								
Gate Array Equivalent Gates	2,500	6,000	10,000	8,000	4,000	8,000	1,200	2,000
PLD Equivalent Gates	6,250	15,000	25,000	20,000	10,000	20,000	3,000	6,000
TTL Equivalent Packages (40 gates)	60	150	250	200	100	200	30	50
20-Pin PAL Equivalent Packages (100 gates)	25	60	100	80	40	80	12	20
Logic Modules	310	848	1377	1,232	684	1232	295	547
S-Modules	160	432	697	624	348	624	—	—
C-Modules	150	416	680	608	336	608	295	547
Flip-Flops (maximum)	435	976	1493	998	568	998	147	273
User I/Os (maximum)	100	168	228	140	104	140	57	69
Packages[1] (by pin count)								
CPGA	133	207	257	176	132	176	84	84
CQFP	132	196	256	172	—	172	—	84
Performance								
System Speed (maximum)	60 MHz	60 MHz	60 MHz	50 MHz	40 MHz	40 MHz	20 MHz	20 MHz

Note:

1. See Product Plan on page 1-209 for package availability.

High-Reliability, Low-Risk Solution

Actel builds the most reliable field programmable gate arrays (FPGAs) in the industry, with overall antifuse reliability ratings of less than 10 Failures-In-Time (FITs), corresponding to a useful life of more than 40 years. Actel FPGAs have been production proven, with more than five million devices shipped and more than one trillion antifuses manufactured. Actel devices are fully tested prior to shipment, with an outgoing defect level of only 122 ppm. (Further reliability data is available in the "Actel Device Reliability Report.")

100 Percent Tested

Device functionality is fully tested before shipment and during device programming. Routing tracks, logic modules, and programming, debug, and test circuits are 100 percent tested before shipment. Antifuse integrity also is tested before shipment. Programming algorithms are tested when a device is programmed using Actel's Activator® 2 or Activator 2S programming stations.

Benefits

No Cost Risk—Once you have a Designer/Designer Advantage™ System, Actel's CAE software and programming package, you can produce as many chips as you like for just the cost of the device itself, with no NRE charges to eat up your development budget every time you want to try out a new design.

No Time Risk—After entering your design, placement and routing is automatic, and programming the device takes only about 5 to 15 minutes for an average design. You save time in the design entry process by using tools that are familiar to you. The Designer/Designer Advantage System software interfaces with popular CAE packages such as Cadence, Mentor Graphics, OrCAD, and Viewlogic, running on platforms such as HP, Sun, and PC. In addition, synthesis capability is provided with support of synthesis tools from Synopsys, IST, Exemplar, and DATA I/O.

No Reliability Risk—The PLICE® antifuse is a one-time programmable, nonvolatile connection. Since Actel devices are permanently programmed, no downloading from EPROM or SRAM storage is required. Inadvertent erasure is impossible, and there is no need to reload the program after power disruptions. Fabrication using a low-power CMOS process means cooler junction temperatures. Actel's non-PLD architecture delivers lower dynamic operating current. Our reliability tests show a very low failure rate of 66 FITs at 90°C junction temperature with no degradation in AC performance. Special stress testing at wafer test eliminates infant mortalities prior to packaging.

No Security Risk—Reverse engineering of programmed Actel devices from optical or electrical data is extremely difficult. Programmed antifuses cannot be identified from a photograph or by using a SEM. The antifuse map cannot be deciphered either electrically or by microprobing. Each device has a silicon signature that identifies its origins, down to the wafer lot and fabrication facility.

No Testing Risk—Unprogrammed Actel parts are fully tested at the factory. This includes the logic modules, interconnect tracks, and I/Os. AC performance is ensured by special speed path tests, and programming circuitry is verified on test antifuses. During the programming process, an algorithm is run to ensure that all antifuses are correctly programmed. In addition, Actel's Actionprobe® diagnostic tools allow 100 percent observability of all internal nodes to check and debug your design.

Actel FPGA Description

The Actel families of FPGAs offer a variety of packages, speed/performance characteristics, and processing levels for use in all high-reliability and military applications. Devices are implemented in a silicon gate, two-level metal CMOS process, utilizing Actel's PLICE antifuse technology. This unique architecture offers gate array flexibility, high performance, and quick turnaround through user programming. Device utilization is typically 95 percent of available logic modules.

Actel devices also provide system designers with on-chip diagnostic probe/debug capability, allowing the user to observe 100 percent of the nodes within the design, even while the device is operating in-system. All Actel devices include on-chip clock drivers and a hard-wired distribution network.

User-definable I/Os are capable of driving at both TTL and CMOS drive levels. Available packages for the military are the Ceramic Quad Flat Pack (CQFP) and the Ceramic Pin Grid Array (CPGA). See Product Plan on page 1-209 for details.

All Actel FPGAs are supported by the Actel Designer Series, which offers automatic or user-definable pin assignment, validation of electrical and design rules, automatic placement and routing, timing analysis, user programming, and debug/diagnostic probe capabilities. The Designer Series fully supports schematic capture and backannotated simulation through design kits for Cadence, Mentor Graphics, OrCAD, and Viewlogic. Synthesis is supported with kits for use with synthesis tools from Synopsys, IST, Exemplar, and DATA I/O.

Also available is the ACTmap™ VHDL optimization and synthesis tool that provides logic synthesis and optimization from PAL language or VHDL description inputs. An FPGA macro generator (ACTgen Macro Builder) is provided, allowing the user easily to create higher-level functions such as counters and adders. Finally, ChipEdit is a graphical/visual design tool that allows the user to modify the automatic place and route results.

ACT 3 Description

The ACT 3 family is the third-generation Actel FPGA family. This family offers the highest-performance and highest-capacity devices, ranging from 2,500 to 10,000 gates, with system performance to 60 MHz over the military temperature range. The devices have four clock distribution networks, including dedicated array and I/O clocks. In addition, the ACT 3 family offers the highest I/O-to-gate ratio available. ACT 3 devices are manufactured using 0.8 micron CMOS technology.

1200XL Description

The 1200XL family is pin and functionally compatible with the ACT 2 family, and is design compatible with all other Actel families. The 1200XL offers significant performance enhancements in comparison with the ACT 2 family, with system performance to 50 MHz over the military temperature range, without increased costs. 1200XL devices are manufactured using 0.6 micron CMOS technology.

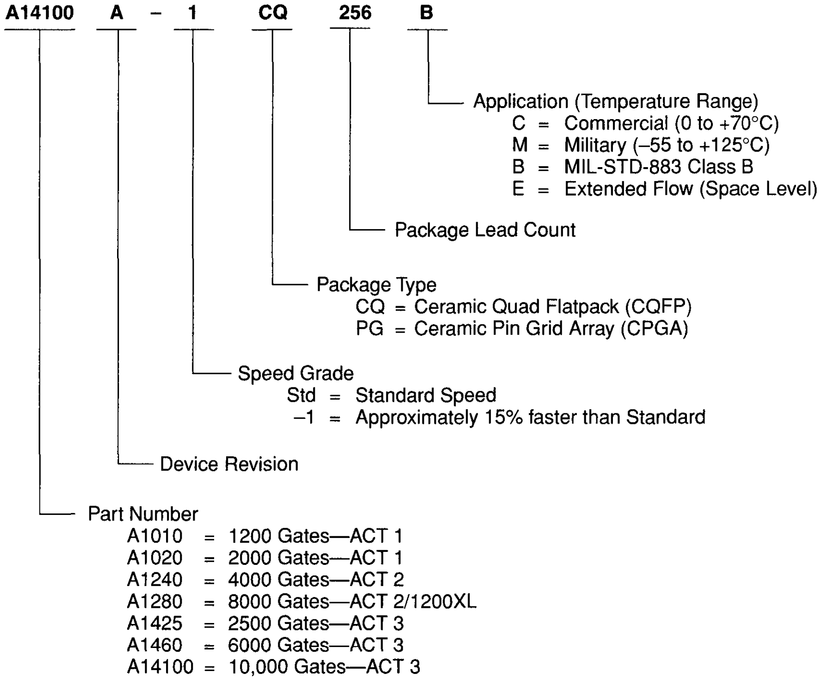
ACT 2 Description

The ACT 2 family is the second-generation Actel FPGA family. This family offers the best-value, high-capacity devices, ranging from 4,000 to 8,000 gates, with system performance to 40 MHz over the military temperature range. The devices have two routed array clock distribution networks. ACT 2 devices are manufactured using 1.0 micron CMOS technology.

ACT 1 Description

The ACT 1 family is the first Actel FPGA family and the first antifuse-based FPGA. This family offers the lowest-cost logic integration, with devices ranging from 1,200 to 2,000 gates, with system performance to 20 MHz over the military temperature range. The devices have one routed array clock distribution network. ACT 1 devices are manufactured using 1.0 micron CMOS technology.

Military Device Ordering Information



A14100 A – 1 CQ 256 B

Application (Temperature Range)
C = Commercial (0 to +70°C)
M = Military (–55 to +125°C)
B = MIL-STD-883 Class B
E = Extended Flow (Space Level)

Package Lead Count

Package Type
CQ = Ceramic Quad Flatpack (CQFP)
PG = Ceramic Pin Grid Array (CPGA)

Speed Grade
Std = Standard Speed
–1 = Approximately 15% faster than Standard

Device Revision

Part Number
A1010 = 1200 Gates—ACT 1
A1020 = 2000 Gates—ACT 1
A1240 = 4000 Gates—ACT 2
A1280 = 8000 Gates—ACT 2/1200XL
A1425 = 2500 Gates—ACT 3
A1460 = 6000 Gates—ACT 3
A14100 = 10,000 Gates—ACT 3

DESC SMD/Actel Part Number Cross Reference

Actel Part Number	DESC SMD	DESC SMD
(Gold Leads)	(Gold Leads)	(Solder Dipped)
A1010B-PG84B	5962-9096403MXC	5962-9096403MXA
A1010B-1PG84B	5962-9096404MXC	5962-9096404MXA
A1020B-PG84B	5962-9096503MUC	5962-9096503MUA
A1020B-1PG84B	5962-9096504MUC	5962-9096504MUA
A1020B-CQ84B	5962-9096503MTC	5962-9096503MTA
A1020B-1CQ84B	5962-9096504MTC	5962-9096504MTA
A1240A-PG132B	5962-9322101MXC	5962-9322101MXA
A1240A-1PG132B	5962-9322102MXC	5962-9322102MXA
A1280A-PG176B	5962-9215601MXC	5962-9215601MXA
A1280A-1PG176B	5962-9215602MXC	5962-9215602MXA
A1280A-CQ172B	5962-9215601MYC	5962-9215601MYA
A1280A-1CQ172B	5962-9215602MYC	5962-9215602MYA
A1425A-PG133B	5962-9552001MXC	5962-9552001MXA
A1425A-1PG133B	5962-9552002MXC	5962-9552002MXA
A1425A-CQ132B	5962-9552001MYC	5962-9552001MYA
A1425A-1CQ132B	5962-9552002MYC	5962-9552002MYA
A1460A-PG207B	5962-9550801MXC	5962-9550801MXA
A1460A-1PG207B	5962-9550802MXC	5962-9550802MXA
A1460A-CQ196B	5962-9550801MYC	5962-9550801MYA
A1460A-1CQ196B	5962-9550802MYC	5962-9550802MYA
A14100A-PG257B	5962-9552101MXC	5962-9552101MXA
A14100A-1PG257B	5962-9552102MXC	5962-9552102MXA
A14100A-CQ256B	5962-9552101MYC	5962-9552101MYA
A14100A-1CQ256B	5962-9552102MYC	5962-9552102MYA

Product Plan

	Speed Grade		Application			
ACT 3 Family	**Std**	**–1**	**C**	**M**	**B**	**E**
A1425A Device						
132-pin Ceramic Quad Flatpack (CQFP)	✓	✓	✓	✓	✓	—
133-pin Ceramic Pin Grid Array (CPGA)	✓	✓	✓	✓	✓	—
A1460A Device						
196-pin Ceramic Quad Flatpack (CQFP)	✓	✓	✓	✓	✓	—
207-pin Ceramic Pin Grid Array (CPGA)	✓	✓	✓	✓	✓	—
A14100A Device						
256-pin Ceramic Quad Flatpack (CQFP)	✓	✓	✓	✓	✓	—
257-pin Ceramic Pin Grid Array (CPGA)	✓	✓	✓	✓	✓	—
	Speed Grade		**Application**			
1200XL Family	**Std**	**–1**	**C**	**M**	**B**	**E**
A1280XL Device						
172-pin Ceramic Quad Flatpack (CQFP)	P	P	P	P	P	—
176-pin Ceramic Pin Grid Array (CPGA)	P	P	P	P	P	—
	Speed Grade		**Application**			
ACT 2 Family	**Std**	**–1**	**C**	**M**	**B**	**E**
A1240A Device						
132-pin Ceramic Pin Grid Array (CPGA)	✓	✓	✓	✓	✓	—
A1280A Device						
172-pin Ceramic Quad Flatpack (CQFP)	✓	✓	✓	✓	✓	✓
176-pin Ceramic Pin Grid Array (CPGA)	✓	✓	✓	✓	✓	✓
	Speed Grade		**Application**			
ACT 1 Family	**Std**	**–1**	**C**	**M**	**B**	**E**
A1010B Device						
84-pin Ceramic Pin Grid Array (CPGA)	✓	✓	✓	✓	✓	—
A1020B Device						
84-pin Ceramic Quad Flatpack (CQFP)	✓	✓	✓	✓	✓	✓
84-pin Ceramic Pin Grid Array (CPGA)	✓	✓	✓	✓	✓	✓

Applications: C = Commercial
M = Military
B = MIL-STD-883
E = Extended Flow

Availability: ✓ = Available Now
P = Planned
— = Not Planned

Speed Grade:–1=Approx. 15% faster than Standard

ACT 3 Device Resources

FPGA Device Type	Logic Modules	Gate Array Equivalent Gates	User I/Os					
			CQFP			CPGA		
			132-pin	196-pin	256-pin	133-pin	207-pin	257-pin
A1425A	310	2500	100	—	—	100	—	—
A1460A	848	6000	—	168	—	—	168	—
A14100A	1377	10,000	—	—	228	—	—	228

1200XL Device Resources

FPGA Device Type	Logic Modules	Gate Array Equivalent Gates	User I/Os	
			CQFP	CPGA
			172-pin	176-pin
A1280XL	1232	8000	140	140

ACT 2 Device Resources

FPGA Device Type	Logic Modules	Gate Array Equivalent Gates	User I/Os		
			CQFP	CPGA	
			172-pin	132-pin	176-pin
A1240A	684	4000	—	104	—
A1280A	1232	8000	140	—	140

ACT 1 Device Resources

FPGA Device Type	Logic Modules	Gate Array Equivalent Gates	User I/Os	
			CQFP	CPGA
			84-pin	84-pin
A1010B	295	1200	—	57
A1020B	547	2000	69	69

Pin Description

CLK **Clock (Input)**

ACT 1 only. TTL Clock input for global clock distribution network. The Clock input is buffered prior to clocking the logic modules. This pin can also be used as an I/O.

CLKA **Clock A (Input)**

ACT 3, 1200XL, and ACT 2 only. TTL Clock input for global clock distribution networks. The Clock input is buffered prior to clocking the logic modules. This pin can also be used as an I/O.

CLKB **Clock B (Input)**

ACT 3, 1200XL, and ACT 2 only. TTL Clock input for global clock distribution networks. The Clock input is buffered prior to clocking the logic modules. This pin can also be used as an I/O.

DCLK **Diagnostic Clock (Input)**

TTL Clock input for diagnostic probe and device programming. DCLK is active when the MODE pin is HIGH. This pin functions as an I/O when the MODE pin is LOW.

GND **Ground**

LOW supply voltage.

HCLK **Dedicated (Hard-wired) Array Clock (Input)**

ACT 3 only. TTL Clock input for sequential modules. This input is directly wired to each S-module and offers clock speeds independent of the number of S-modules being driven. This pin can also be used as an I/O.

I/O **Input/Output (Input, Output)**

I/O pin functions as an input, output, tristate, or bidirectional buffer. Input and output levels are compatible with standard TTL and CMOS specifications. Unused I/O pins are automatically driven LOW.

IOCLK **Dedicated (Hard-wired) I/O Clock (Input)**

ACT 3 only. TTL Clock input for I/O modules. This input is directly wired to each I/O module and offers clock speeds independent of the number of I/O modules being driven. This pin can also be used as an I/O.

IOPCL **Dedicated (Hard-wired) I/O Preset/Clear (Input)**

ACT 3 only. TTL input for I/O preset or clear. This global input is directly wired to the preset and clear inputs of all I/O registers. This pin functions as an I/O when no I/O preset or clear macros are used.

MODE **Mode (Input)**

The MODE pin controls the use of diagnostic pins (DCLK, PRA, PRB, SDI). When the MODE pin is HIGH, the special functions are active. When the MODE pin is LOW, the pins function as I/Os. When the MODE pin is LOW, the pins function as I/Os. To provide Actionprobe capability, the MODE pin should be terminated to GND through a 10K resistor so that the MODE pin can be pulled high when required.

NC **No Connection**

This pin is not connected to circuitry within the device.

PRA **Probe A (Output)**

The Probe A pin is used to output data from any user-defined design node within the device. This independent diagnostic pin can be used in conjunction with the Probe B pin to allow real-time diagnostic output of any signal path within the device. The Probe A pin can be used as a user-defined I/O when debugging has been completed. The pin's probe capabilities can be permanently disabled to protect programmed design confidentiality. PRA is accessible when the MODE pin is HIGH. This pin functions as an I/O when the MODE pin is LOW.

PRB **Probe B (Output)**

The Probe B pin is used to output data from any user-defined design node within the device. This independent diagnostic pin can be used in conjunction with the Probe A pin to allow real-time diagnostic output of any signal path within the device. The Probe B pin can be used as a user-defined I/O when debugging has been completed. The pin's probe capabilities can be permanently disabled to protect programmed design confidentiality. PRB is accessible when the MODE pin is HIGH. This pin functions as an I/O when the MODE pin is LOW.

SDI **Serial Data Input (Input)**

Serial data input for diagnostic probe and device programming. SDI is active when the MODE pin is HIGH. This pin functions as an I/O when the MODE pin is LOW.

V_{CC} **5V Supply Voltage**

HIGH supply voltage.

Actel Military Product Flow

Step	Screen	833—Class B 833 Method	833—Class B Requirement
1.0	Internal Visual	2010, Test Condition B	100%
2.0	Temperature Cycling	1010, Test Condition C	100%
3.0	Constant Acceleration	2001, Test Condition E (min), Y1, Orientation Only	100%
4.0	Seal a.Fine b.Gross	1014	 100% 100%
5.0	Visual Inspection	2009	100%
6.0	Pre-burn-in Electrical Parameters	In accordance with Actel applicable device specification	100%
7.0	Burn-in Test	1015 Condition D 160 hours @ 125°C Min.	100%
8.0	Interim (Post-burn-in) Electrical Parameters	In accordance with Actel applicable device specification	100%
9.0	Percent Defective Allowable	5%	All Lots
10.0	Final Electrical Test	In accordance with Actel applicable device specification	
	a. Static Tests (1) 25°C (Subgroup 1, Table I, 5005) (2) –55°C and +125°C (Subgroups 2, 3, Table I, 5005)		100%
	b. Dynamic and Functional Tests (1) 25°C (Subgroup 7, Table I, 5005) (2) –55°C and +125°C (Subgroups 8A and 8B, Table I, 5005)		100%
	c. Switching Tests at 25°C (Subgroup 9, Table I, 5005)		100%
11.0	Qualification or Quality Confirmation Inspection Test Sample Selection (Group A)	5005	All Lots
12.0	External Visual	2009	100%

Actel Extended Flow[1, 2]

	Screen	Method	Require- ment
1.	Wafer Lot Acceptance[3]	5007 with step coverage waiver	All Lots
2.	Destructive In-Line Bond Pull[4]	2011, condition D	Sample
3.	Internal Visual	2010, condition A	100%
4.	Serialization		100%
5.	Temperature Cycling	1010, condition C	100%
6.	Constant Acceleration	2001, condition E (min), Y_1 orientation only	100%
7.	Visual Inspection	2009	100%
8.	Particle Impact Noise Detection	2020, condition A	100%
9.	Radiographic	2012	100%
10.	Pre-burn-in Test	In accordance with Actel applicable device specification	100%
11.	Burn-in Test	1015, condition D, 240 hours @ 125°C minimum	100%
12.	Interim (Post-burn-in) Electrical Parameters	In accordance with Actel applicable device specification	100%
13.	Reverse Bias Burn-in	1015, condition C, 72 hours @ 150°C minimum	100%
14.	Interim (Post-burn-in) Electrical Parameters	In accordance with Actel applicable device specification	100%
15.	Percent Defective Allowable (PDA) Calculation	5%, 3% functional parameters @ 25°C	All Lots
16.	Final Electrical Test	In accordance with Actel applicable device specification	100%
	a. Static Tests		100%
	(1) 25°C (Subgroup 1, Table1)	5005	
	(2) –55°C and +125°C (Subgroups 2, 3, Table 1)	5005	
	b. Dynamic and Functional Tests		100%
	(1) 25°C (Subgroup 7, Table 15)	5005	
	(2) –55°C and +125°C (Subgroups 5 and 6, 8a and b, Table 1)	5005	
	c. Switching Tests at 25°C (Subgroup 9, Table I, 5005)	5005	100%
17.	Seal a.Fine b.Gross	1014	100%
18.	Qualification or Quality Conformance Inspection Test Sample Selection	5005	Group A & Group B
19	External Visual	2009	100%

Notes:

1. *Actel offers the Extended Flow in order to satisfy those customers that require additional screening beyond the requirements of MIL-STD-883, Class B. Actel is compliant to the requirements of MIL-STD-883, Paragraph 1.2.1, and MIL-I-38535, Appendix A. Actel is offering this extended flow incorporating the majority of the screening procedures as outlined in Method 5004 of MIL-STD-883 Class S. The exceptions to Method 5004 are shown in notes 2 to 4 below.*
2. *Method 5004 requires a 100 percent Radiation latch-up testing to Method 1020. Actel will not be performing any radiation testing, and this requirement must be waived in its entirety.*
3. *Wafer lot acceptance is performed to Method 5007; however the step coverage requirement as specified in Method 2018 must be waived.*
4. *Method 5004 requires a 100 percent, nondestructive bond pull to Method 2023. Actel substitutes a destructive bond pull to Method 2011, condition D on a sample basis only.*

Absolute Maximum Ratings[1]

Free air temperature range

Symbol	Parameter	Limits	Units
V_{CC}	DC Supply Voltage[2, 3, 4]	–0.5 to +7.0	V
V_I	Input Voltage	–0.5 to V_{CC} +0.5	V
V_O	Output Voltage	–0.5 to V_{CC} +0.5	V
I_{IO}	I/O Source Sink Current[5]	±20	mA
T_{STG}	Storage Temperature	–65 to +150	°C

Notes:

1. *Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum rated conditions for extended periods may affect device reliability. Device should not be operated outside the recommended operating conditions.*
2. *$V_{PP} = V_{CC}$, except during device programming.*
3. *$V_{SV} = V_{CC}$, except during device programming.*
4. *V_{KS} = GND, except during device programming.*
5. *Device inputs are normally high impedance and draw extremely low current. However, when input voltage is greater than V_{CC} + 0.5 V or less than GND – 0.5 V, the internal protection diode will be forward biased and can draw excessive current.*

Recommended Operating Conditions

Parameter	Commercial	Military	Units
Temperature Range[1]	0 to +70	–55 to +125	°C
Power Supply Tolerance	±5	±10	%V_{CC}

Note:

1. *Ambient temperature (T_A) is used for commercial and industrial; case temperature (T_C) is used for military.*

Package Thermal Characteristics

The device junction to case thermal characteristic is θjc, and the junction to ambient air characteristic is θja. The thermal characteristics for θja are shown with two different air flow rates.

Maximum junction temperature is 150°C.

A sample calculation of the absolute maximum power dissipation allowed for a CPGA 176-pin package at military temperature is as follows:

$$\frac{\text{Max. junction temp. (°C)} - \text{Max. military temp.}}{\theta\text{ja (°C/W)}} = \frac{150°\text{C} - 125°\text{C}}{23°\text{C/W}} = 1.1\text{ W}$$

Package Type	Pin Count	θjc	θja Still Air	θja 300 ft/min	Units
Ceramic Pin Grid Array	84	20	33	20	°C/W
	132	20	26	16	°C/W
	133	20	37	24	°C/W
	176	20	23	12	°C/W
	207	20	22	14	°C/W
	257	20	21	13	°C/W
Ceramic Quad Flatpack	84	13	40	25	°C/W
	132	13	55	30	°C/W
	172	13	25	15	°C/W
	196	13	36	24	°C/W
	256	13	30	18	°C/W

Electrical Specifications

			Commercial		Military		
Symbol	Parameter	Test Condition	Min.	Max.	Min.	Max.	Units
V_{OH}[1, 2]	HIGH Level Output	I_{OH} = –4 mA (CMOS)			3.7		V
		I_{OH} = –6 mA (CMOS)	3.84				V
V_{OL}[1, 2]	LOW Level Output	I_{OL} = +6 mA (CMOS)		0.33		0.4	V
V_{IH}	HIGH Level Input	TTL Inputs	2.0	V_{CC} + 0.3	2.0	V_{CC} + 0.3	V
V_{IL}	LOW Level Input	TTL Inputs	–0.3	0.8	–0.3	0.8	V
I_{IN}	Input Leakage	$V_I = V_{CC}$ or GND	–10	+10	–10	+10	μA
I_{OZ}	3-state Output Leakage	$V_O = V_{CC}$ or GND	–10	+10	–10	+10	μA
C_{IO}	I/O Capacitance[3, 4]			10		10	pF
$I_{CC(S)}$	Standby V_{CC} Supply Current	$V_I = V_{CC}$ or GND, I_O = 0 mA					
		ACT 1		3		20	mA
		ACT 2/3/1200XL		2		20	mA
$I_{CC(D)}$	Dynamic V_{CC} Supply Current	See "Power Dissipation" Section					

Notes:

1. *Actel devices can drive and receive either CMOS or TTL signal levels. No assignment of I/Os as TTL or CMOS is required.*
2. *Tested one output at a time, V_{CC} = min.*
3. *Not tested; for information only.*
4. *V_{OUT} = 0V, f = 1 MHz*

General Power Equation

$$P = [I_{CC}\text{standby} + I_{CC}\text{active}] * V_{CC} + I_{OL} * V_{OL} * N + I_{OH} * (V_{CC} - V_{OH}) * M$$

Where:

I_{CC}standby is the current flowing when no inputs or outputs are changing.

I_{CC}active is the current flowing due to CMOS switching.

I_{OL}, I_{OH} are TTL sink/source currents.

V_{OL}, V_{OH} are TTL level output voltages.

N equals the number of outputs driving TTL loads to V_{OL}.

M equals the number of outputs driving TTL loads to V_{OH}.

An accurate determination of N and M is problematical because their values depend on the family type, on design details, and on the system I/O. The power can be divided into two components—static and active.

Static Power Component

Actel FPGAs have small static power components that result in power dissipation lower than that of PALs or PLDs. By integrating multiple PALs or PLDs into one FPGA, an even greater reduction in board-level power dissipation can be achieved.

The power due to standby current is typically a small component of the overall power. Standby power is calculated below for commercial, worst-case conditions.

Family	I_{CC}	V_{CC}	Power
ACT 1	3 mA	5.25 V	15.8 mW
1200XL	2mA	5.25V	10.5mW
ACT 2	2 mA	5.25 V	10.5 mW
ACT 3	2 mA	5.25 V	10.5 mW

The static power dissipated by TTL loads depends on the number of outputs driving high or low and the DC load current. Again, this value is typically small. For instance, a 32-bit bus sinking 4 mA at 0.33 V will generate 42 mW with all outputs driving low, and 140 mW with all outputs driving high.

Active Power Component

Power dissipation in CMOS devices is usually dominated by the active (dynamic) power dissipation. This component is frequency dependent, a function of the logic and the external I/O. Active power dissipation results from charging internal chip capacitances of the interconnect, unprogrammed antifuses, module inputs, and module outputs, plus external capacitance due to PC board traces and load device inputs. An additional component of the active power dissipation is the totem-pole current in CMOS transistor pairs. The net effect can be associated with an equivalent capacitance that can be combined with frequency and voltage to represent active power dissipation.

Equivalent Capacitance

The power dissipated by a CMOS circuit can be expressed by the equation 1

$$\text{Power (uW)} = C_{EQ} * V_{CC}^2 * F \qquad (1)$$

Where:

C_{EQ} is the equivalent capacitance expressed in pF.

V_{CC} is the power supply in volts.

F is the switching frequency in MHz.

Equivalent capacitance is calculated by measuring I_{CC}active at a specified frequency and voltage for each circuit component of interest. Measurements are made over a range of frequencies at a fixed value of V_{CC}. Equivalent capacitance is frequency independent so that the results can be used over a wide range of operating conditions. Equivalent capacitance values are shown below.

CEQ Values for Actel FPGAs

	ACT 1	1200XL	ACT 2	ACT 3
Modules (C_{EQM})	3.7	5.2	5.8	6.7
Input Buffers (C_{EQI})	22.1	11.6	12.9	7.2
Output Buffers (C_{EQO})	31.2	23.8	23.8	10.4
Routed Array Clock Buffer Loads (C_{EQCR})	4.6	3.5	3.9	1.6
Dedicated Clock Buffer Loads (C_{EQCD})	n/a	n/a	n/a	0.7
I/O Clock Buffer Loads (C_{EQCI})	n/a	n/a	n/a	0.9

To calculate the active power dissipated from the complete design, the switching frequency of each part of the logic must be known. Equation 2 shows a piecewise linear summation over all components, it applies to all ACT 1, 1200XL, ACT 2, and ACT 3 devices. Since the ACT 1 family has only one routed array clock, the terms labeled routed_Clk2, dedicated_Clk, and IO_Clk do not apply. Similarly, the ACT 2 family has two routed array clocks, and the dedicated_Clk and IO_Clk terms do not apply. For ACT 3 devices, all terms will apply.

$$\text{Power} = V_{CC}^2 * [(m * C_{EQM} * f_m)_{\text{modules}} + (n * C_{EQI} * f_n)_{\text{inputs}} + (p * (C_{EQO} + C_L) * f_p)_{\text{outputs}} + 0.5 * (q_1 * C_{EQCR} * f_{q1})_{\text{routed\_Clk1}} + (r_1 * f_{q1})_{\text{routed\_Clk1}} + 0.5 * (q_2 * C_{EQCR} * f_{q2})_{\text{routed\_Clk2}} + (r_2 * f_{q2})_{\text{routed\_Clk2}} + 0.5 * (s_1 * C_{EQCD} * f_{s1})_{\text{dedicated\_Clk}} + (s_2 * C_{EQCI} * f_{s2})_{\text{IO\_Clk}}] \qquad (2)$$

Where:

m = Number of logic modules switching at fm

n = Number of input buffers switching at fn

p = Number of output buffers switching at fp

q_1 = Number of clock loads on the first routed array clock (all families)

q_2 = Number of clock loads on the second routed array clock (ACT 2, 1200XL, ACT 3 only)

r_1 = Fixed capacitance due to first routed array clock (all families)

r_2 = Fixed capacitance due to second routed array clock (ACT 2, 1200XL, ACT 3 only)

s_1 = Fixed number of clock loads on the dedicated array clock (ACT 3 only)

s_2 = Fixed number of clock loads on the dedicated I/O clock (ACT 3 only)

C_{EQM} = Equivalent capacitance of logic modules in pF

C_{EQI} = Equivalent capacitance of input buffers in pF

C_{EQO} = Equivalent capacitance of output buffers in pF

C_{EQCR} = Equivalent capacitance of routed array clock in pF

C_{EQCD} = Equivalent capacitance of dedicated array clock in pF

C_{EQCI} = Equivalent capacitance of dedicated I/O clock in pF

C_L = Output lead capacitance in pF

f_m = Average logic module switching rate in MHz

f_n = Average input buffer switching rate in MHz

f_p = Average output buffer switching rate in MHz

f_{q1} = Average first routed array clock rate in MHz (all families)

f_{q2} = Average second routed array clock rate in MHz (ACT 2, 1200XL, ACT 3 only)

f_{s1} = Average dedicated array clock rate in MHz (ACT 3 only)

f_{s2} = Average dedicated I/O clock rate in MHz (ACT 3 only)

Fixed Capacitance Values for Actel FPGAs (pF)

	r_1	r_2
Device Type	routed_Clk1	routed_Clk2
A1010B	41	n/a
A1020B	69	n/a
A1240A	134	134
A1280A	168	168
A1280XL	168	168
A1425A	75	75
A1460A	165	165
A14100A	195	195

Fixed Clock Loads (s_1/s_2—ACT 3 Only)

	s_1	s_2
	Clock Loads on	Clock Loads on
	Dedicated	Dedicated
Device Type	Array Clock	I/O Clock
A1425A	160	100
A1460A	432	168
A14100A	697	228

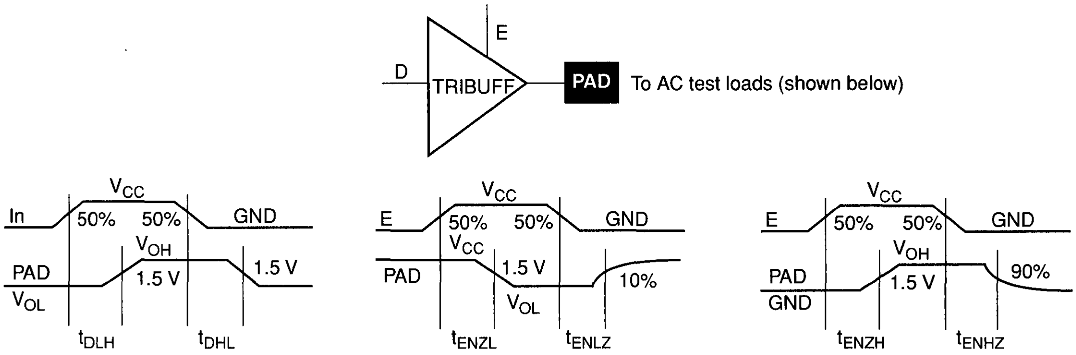
Determining Average Switching Frequency

To determine the switching frequency for a design, you must have a detailed understanding of the data values input to the circuit. The guidelines in the table below are meant to represent worst-case scenarios so that they can be generally used to predict the upper limits of power dissipation.

Type	ACT 1	ACT 2/1200XL	ACT 3
Logic modules (m)	90% of modules	80% of modules	80% of modules
Input switching (n)	# inputs/4	# inputs/4	# inputs/4
Outputs switching (p)	#outputs/4	#outputs/4	#outputs/4
First routed array clock loads (q_1)	40% of modules	40% of sequential modules	40% of sequential modules
Second routed array clock loads (q_2)	n/a	40% of sequential modules	40% of sequential modules
Load capacitance (C_L)	35 pF	35 pF	35 pF
Average logic module switching rate (f_m)	F/10	F/10	F/10
Average input switching rate (f_n)	F/5	F/5	F/5
Average output switching rate (f_p)	F/10	F/10	F/10
Average first routed array clock rate (f_{q1})	F	F	F/2
Average second routed array clock rate (f_{q2})	n/a	F/2	F/2
Average dedicated array clock rate (f_{s1})	n/a	n/a	F
Average dedicated I/O clock rate (f_{s2})	n/a	n/a	F

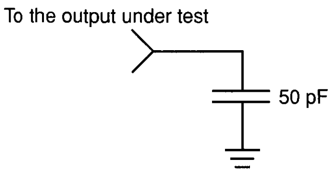
Parameter Measurement

Output Buffer Delays

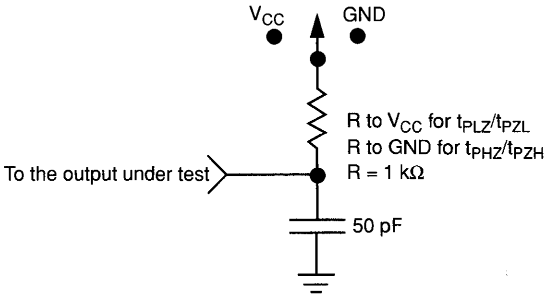


AC Test Load

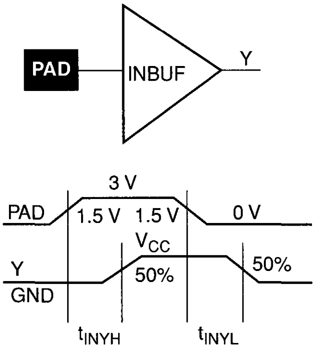
Load 1
(Used to measure propagation delay)



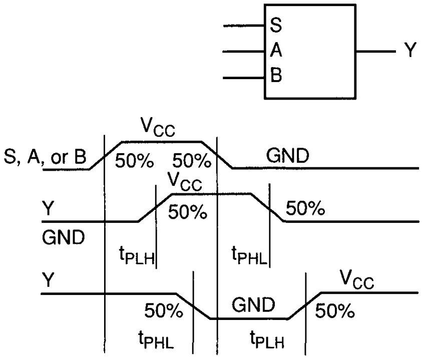
Load 2
(Used to measure rising/falling edges)



Input Buffer Delays

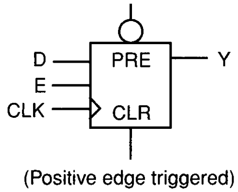


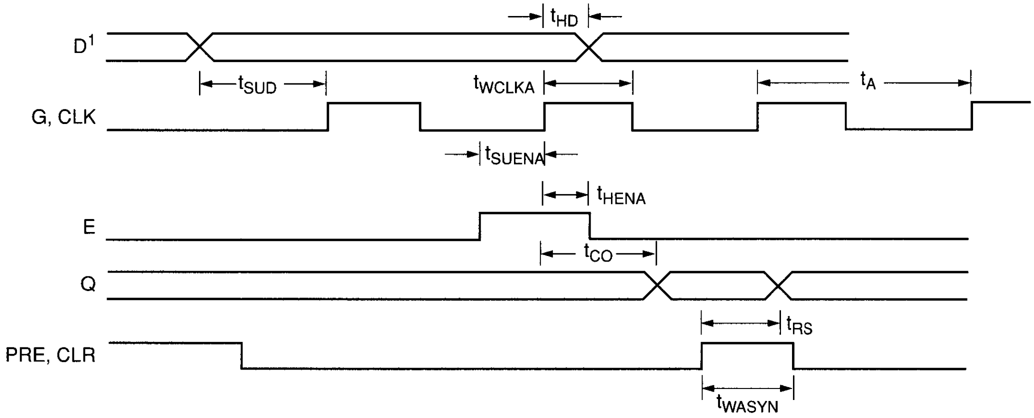
Combinatorial Macro Delays



Sequential Timing Characteristics

Flip-Flops and Latches (ACT 1, ACT 2, and 1200XL)



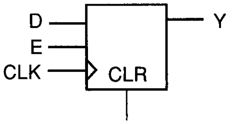


Note:

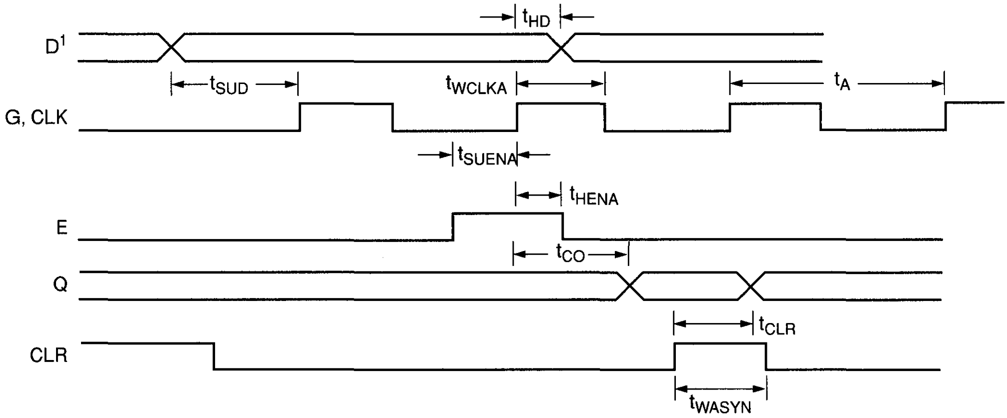
1. *D represents all data functions involving A, B, and S for multiplexed flip-flops.*

Sequential Timing Characteristics (continued)

Flip-Flops and Latches (ACT 3)



(Positive edge triggered)

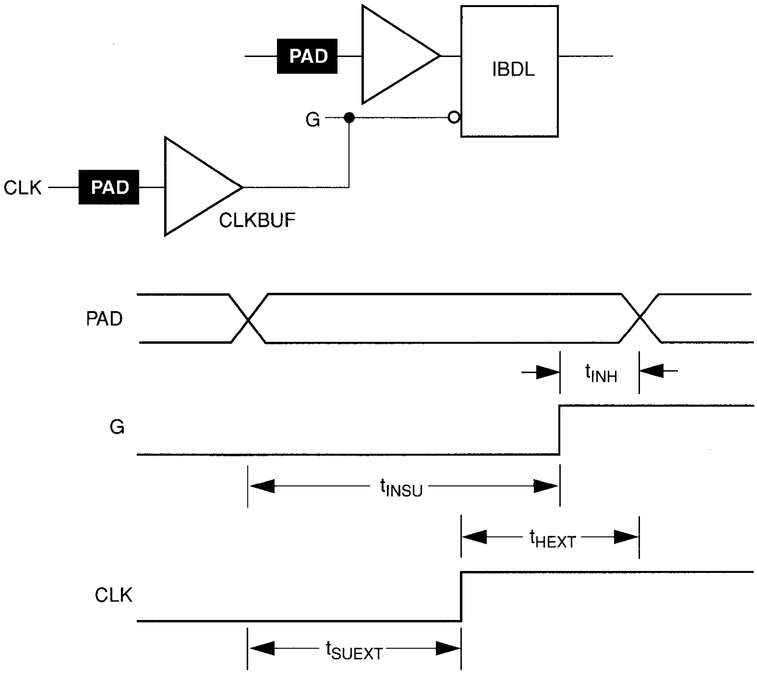


Note:

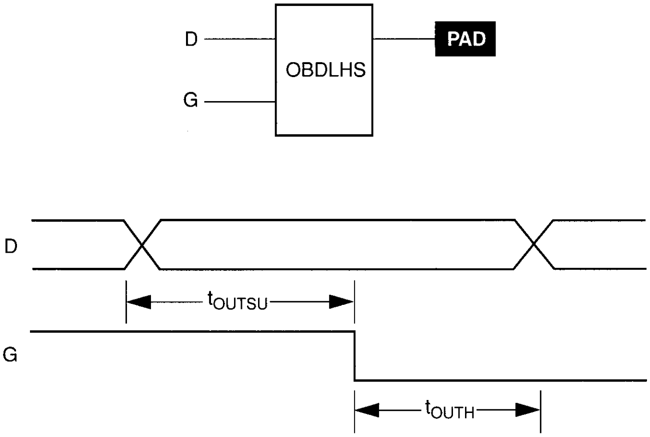
1. *D represents all data functions involving A, B, and S for multiplexed flip-flops.*

Sequential Timing Characteristics (continued)

Input Buffer Latches (ACT 2 and 1200XL)



Output Buffer Latches (ACT 2 and 1200XL)



ACT 1 Timing Characteristics

(Worst-Case Military Conditions)

Logic Module Propagation Delays		–1 Speed		Std Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{PD1}	Single Module		4.7		5.5	ns
t_{PD2}	Dual Module Macros		10.8		12.7	ns
t_{CO}	Sequential Clk to Q		4.7		5.5	ns
t_{GO}	Latch G to Q		4.7		5.5	ns
t_{RS}	Flip-Flop (Latch) Reset to Q		4.7		5.5	ns
Predicted Routing Delays[1]						
t_{RD1}	FO=1 Routing Delay		1.5		1.7	ns
t_{RD2}	FO=2 Routing Delay		2.3		2.7	ns
t_{RD3}	FO=3 Routing Delay		3.4		4.0	ns
t_{RD4}	FO=4 Routing Delay		5.0		5.9	ns
t_{RD8}	FO=8 Routing Delay		10.6		12.5	ns
Sequential Timing Characteristics[2]						
t_{SUD}	Flip-Flop (Latch) Data Input Setup	8.8		10.4		ns
t_{HD}	Flip-Flop (Latch) Data Input Hold	0.0		0.0		ns
t_{SUENA}	Flip-Flop (Latch) Enable Setup	8.8		10.4		ns
t_{HENA}	Flip-Flop (Latch) Enable Hold	0.0		0.0		ns
t_{WCLKA}	Flip-Flop (Latch) Clock Active Pulse Width	10.9		12.9		ns
t_{WASYN}	Flip-Flop (Latch) Asynchronous Pulse Width	10.9		12.9		ns
t_A	Flip-Flop Clock Input Period	23.2		27.3		ns
f_{MAX}	Flip-Flop (Latch) Clock Frequency		44		37	MHz

Notes:

1. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*
2. *Setup times assume fanout of 3. Further derating information can be obtained from the DirectTime Analyzer utility.*

ACT 1 Timing Characteristics (continued)

(Worst-Case Military Conditions)

Input Module Propagation Delays			–1 Speed		Std Speed		
Parameter	Description		Min.	Max.	Min.	Max.	Units
t_{INYH}	Pad to Y High			4.9		5.8	ns
t_{INYL}	Pad to Y Low			4.9		5.8	ns
Input Module Predicted Routing Delays[1]							
t_{IRD1}	FO=1 Routing Delay			1.5		1.7	ns
t_{IRD2}	FO=2 Routing Delay			2.3		2.7	ns
t_{IRD3}	FO=3 Routing Delay			3.4		4.0	ns
t_{IRD4}	FO=4 Routing Delay			5.0		5.9	ns
t_{IRD8}	FO=8 Routing Delay			10.6		12.5	ns
Global Clock Network							
t_{CKH}	Input Low to High	FO = 16 FO = 128		7.8 8.9		9.2 10.5	ns
t_{CKL}	Input High to Low	FO = 16 FO = 128		10.3 11.2		12.1 13.2	ns
t_{PWH}	Minimum Pulse Width High	FO = 16 FO = 128	10.4 10.9		12.2 12.9		ns
t_{PWL}	Minimum Pulse Width Low	FO = 16 FO = 128	10.4 10.9		12.2 12.9		ns
t_{CKSW}	Maximum Skew	FO = 16 FO = 128		1.9 2.9		2.2 3.4	ns
t_P	Minimum Period	FO = 16 FO = 128	21.7 23.2		25.6 27.3		ns
f_{MAX}	Maximum Frequency	FO = 16 FO = 128		46 44		40 37	MHz

Note:

1. *These parameters should be used for estimating device performance. Routing delays are for typical designs across worst-case operating conditions. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

ACT 1 Timing Characteristics (continued)

(Worst-Case Military Conditions)

Output Module Timing		–1 Speed		Std Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
TTL Output Module Timing[1]						
t_{DLH}	Data to Pad High		12.1		14.2	ns
t_{DHL}	Data to Pad Low		13.8		16.3	ns
t_{ENZH}	Enable Pad Z to High		12.0		14.1	ns
t_{ENZL}	Enable Pad Z to Low		14.6		17.1	ns
t_{ENHZ}	Enable Pad High to Z		16.0		18.8	ns
t_{ENLZ}	Enable Pad Low to Z		14.5		17.0	ns
d_{TLH}	Delta Low to High		0.09		0.11	ns/pF
d_{THL}	Delta High to Low		0.12		0.15	ns/pF
CMOS Output Module Timing[1]						
t_{DLH}	Data to Pad High		15.1		17.7	ns
t_{DHL}	Data to Pad Low		11.5		13.6	ns
t_{ENZH}	Enable Pad Z to High		12.0		14.1	ns
t_{ENZL}	Enable Pad Z to Low		14.6		17.1	ns
t_{ENHZ}	Enable Pad High to Z		16.0		18.8	ns
t_{ENLZ}	Enable Pad Low to Z		14.5		17.0	ns
d_{TLH}	Delta Low to High		0.16		0.18	ns/pF
d_{THL}	Delta High to Low		0.09		0.11	ns/pF

Notes:

1. *Delays based on 50 pF loading.*
2. *SSO information can be found in the "Simultaneously Switching Output Limits for Actel FPGAs" application note on page 4-125.*

A1240A Timing Characteristics

(Worst-Case Military Conditions)

Logic Module Propagation Delays[1]		–1 Speed		Std Speed		
Parameter	Description	Min.	Max.	Min.	Max.	Units
t_{PD1}	Single Module		5.2		6.1	ns
t_{CO}	Sequential Clk to Q		5.2		6.1	ns
t_{GO}	Latch G to Q		5.2		6.1	ns
t_{RS}	Flip-Flop (Latch) Reset to Q		5.2		6.1	ns
Predicted Routing Delays[2]						
t_{RD1}	FO=1 Routing Delay		1.9		2.2	ns
t_{RD2}	FO=2 Routing Delay		2.4		2.8	ns
t_{RD3}	FO=3 Routing Delay		3.1		3.7	ns
t_{RD4}	FO=4 Routing Delay		4.3		5.0	ns
t_{RD8}	FO=8 Routing Delay		6.6		7.7	ns
Sequential Timing Characteristics[3, 4]						
t_{SUD}	Flip-Flop (Latch) Data Input Setup	0.5		0.5		ns
t_{HD}	Flip-Flop (Latch) Data Input Hold	0.0		0.0		ns
t_{SUENA}	Flip-Flop (Latch) Enable Setup	1.3		1.3		ns
t_{HENA}	Flip-Flop (Latch) Enable Hold	0.0		0.0		ns
t_{WCLKA}	Flip-Flop (Latch) Clock Active Pulse Width	7.4		8.1		ns
t_{WASYN}	Flip-Flop (Latch) Asynchronous Pulse Width	7.4		8.1		ns
t_A	Flip-Flop Clock Input Period	14.8		18.6		ns
t_{INH}	Input Buffer Latch Hold	2.5		2.5		ns
t_{INSU}	Input Buffer Latch Setup	–3.5		–3.5		ns
t_{OUTH}	Output Buffer Latch Hold	0.0		0.0		ns
t_{OUTSU}	Output Buffer Latch Setup	0.5		0.5		ns
f_{MAX}	Flip-Flop (Latch) Clock Frequency		63		54	MHz

Notes:

1. *For dual-module macros, use* $t_{PD1} + t_{RD1} + t_{PDn}$, $t_{CO} + t_{RD1} + t_{PDn}$, *or* $t_{PD1} + t_{RD1} + t_{SUD}$, *whichever is appropriate.*
2. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*
3. *Data applies to macros based on the S-module. Timing parameters for sequential macros constructed from C-modules can be obtained from the DirectTime Analyzer utility.*
4. *Setup and hold timing parameters for the Input Buffer Latch are defined with respect to the PAD and the D input. External setup/hold timing parameters must account for delay from an external PAD signal to the G inputs. Delay from an external PAD signal to the G input subtracts (adds) to the internal setup (hold) time.*

A1240A Timing Characteristics (continued)

(Worst-Case Military Conditions)

Input Module Propagation Delays			–1 Speed		Std Speed		
Parameter	**Description**		**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{INYH}	Pad to Y High			4.0		4.7	ns
t_{INYL}	Pad to Y Low			3.6		4.3	ns
t_{INGH}	G to Y High			6.9		8.1	ns
t_{INGL}	G to Y Low			6.6		7.7	ns
Input Module Predicted Routing Delays[1]							
t_{IRD1}	FO=1 Routing Delay			5.8		6.9	ns
t_{IRD2}	FO=2 Routing Delay			6.7		7.8	ns
t_{IRD3}	FO=3 Routing Delay			7.5		8.8	ns
t_{IRD4}	FO=4 Routing Delay			8.2		9.7	ns
t_{IRD8}	FO=8 Routing Delay			10.9		12.9	ns
Global Clock Network							
t_{CKH}	Input Low to High	FO = 32 FO = 256		13.3 16.3		15.7 19.2	ns
t_{CKL}	Input High to Low	FO = 32 FO = 256		13.3 16.5		15.7 19.5	ns
t_{PWH}	Minimum Pulse Width High	FO = 32 FO = 256	5.7 6.0		6.7 7.1		ns
t_{PWL}	Minimum Pulse Width Low	FO = 32 FO = 256	5.7 6.0		6.7 7.1		ns
t_{CKSW}	Maximum Skew	FO = 32 FO = 256		0.6 3.1		0.6 3.1	ns
t_{SUEXT}	Input Latch External Setup	FO = 32 FO = 256	0.0 0.0		0.0 0.0		ns
t_{HEXT}	Input Latch External Hold	FO = 32 FO = 256	8.6 13.8		8.6 13.8		ns
t_P	Minimum Period	FO = 32 FO = 256	11.5 12.2		13.5 14.3		ns
f_{MAX}	Maximum Frequency	FO = 32 FO = 256		87 82		74 70	MHz

Note:

1. *These parameters should be used for estimating device performance. Optimization techniques may further reduce delays by 0 to 4 ns. Routing delays are for typical designs across worst-case operating conditions. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

A1240A Timing Characteristics (continued)

(Worst-Case Military Conditions)

Output Module Timing		–1 Speed		Std Speed		
Parameter	Description	Min.	Max.	Min.	Max.	Units
TTL Output Module Timing[1]						
t_{DLH}	Data to Pad High		11.0		13.0	ns
t_{DHL}	Data to Pad Low		13.9		16.4	ns
t_{ENZH}	Enable Pad Z to High		12.3		14.4	ns
t_{ENZL}	Enable Pad Z to Low		16.1		19.0	ns
t_{ENHZ}	Enable Pad High to Z		9.8		11.5	ns
t_{ENLZ}	Enable Pad Low to Z		11.5		13.6	ns
t_{GLH}	G to Pad High		12.4		14.6	ns
t_{GHL}	G to Pad Low		15.5		18.2	ns
d_{TLH}	Delta Low to High		0.09		0.11	ns/pF
d_{THL}	Delta High to Low		0.17		0.20	ns/pF
CMOS Output Module Timing[1]						
t_{DLH}	Data to Pad High		14.0		16.5	ns
t_{DHL}	Data to Pad Low		11.7		13.7	ns
t_{ENZH}	Enable Pad Z to High		12.3		14.4	ns
t_{ENZL}	Enable Pad Z to Low		16.1		19.0	ns
t_{ENHZ}	Enable Pad High to Z		9.8		11.5	ns
t_{ENLZ}	Enable Pad Low to Z		11.5		13.6	ns
t_{GLH}	G to Pad High		12.4		14.6	ns
t_{GHL}	G to Pad Low		15.5		18.2	ns
d_{TLH}	Delta Low to High		0.17		0.20	ns/pF
d_{THL}	Delta High to Low		0.12		0.15	ns/pF

Notes:

1. *Delays based on 50 pF loading.*
2. *SSO information can be found in the "Simultaneously Switching Output Limits for Actel FPGAs" application note on page 4-125.*

A1280A Timing Characteristics

(Worst-Case Military Conditions)

Logic Module Propagation Delays[1]		–1 Speed		Std Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{PD1}	Single Module		5.2		6.1	ns
t_{CO}	Sequential Clk to Q		5.2		6.1	ns
t_{GO}	Latch G to Q		5.2		6.1	ns
t_{RS}	Flip-Flop (Latch) Reset to Q		5.2		6.1	ns
Predicted Routing Delays[2]						
t_{RD1}	FO=1 Routing Delay		2.4		2.8	ns
t_{RD2}	FO=2 Routing Delay		3.4		4.0	ns
t_{RD3}	FO=3 Routing Delay		4.2		4.9	ns
t_{RD4}	FO=4 Routing Delay		5.1		6.0	ns
t_{RD8}	FO=8 Routing Delay		9.2		10.8	ns
Sequential Timing Characteristics[3, 4]						
t_{SUD}	Flip-Flop (Latch) Data Input Setup	0.5		0.5		ns
t_{HD}	Flip-Flop (Latch) Data Input Hold	0.0		0.0		ns
t_{SUENA}	Flip-Flop (Latch) Enable Setup	1.3		1.3		ns
t_{HENA}	Flip-Flop (Latch) Enable Hold	0.0		0.0		ns
t_{WCLKA}	Flip-Flop (Latch) Clock Active Pulse Width	7.4		8.6		ns
t_{WASYN}	Flip-Flop (Latch) Asynchronous Pulse Width	7.4		8.6		ns
t_A	Flip-Flop Clock Input Period	16.4		22.1		ns
t_{INH}	Input Buffer Latch Hold	2.5		2.5		ns
t_{INSU}	Input Buffer Latch Setup	–3.5		–3.5		ns
t_{OUTH}	Output Buffer Latch Hold	0.0		0.0		ns
t_{OUTSU}	Output Buffer Latch Setup	0.5		0.5		ns
f_{MAX}	Flip-Flop (Latch) Clock Frequency		60		41	MHz

Notes:

1. *For dual-module macros, use $t_{PD1} + t_{RD1} + t_{PDn}$, $t_{CO} + t_{RD1} + t_{PDn}$, or $t_{PD1} + t_{RD1} + t_{SUD}$, whichever is appropriate.*
2. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*
3. *Data applies to macros based on the S-module. Timing parameters for sequential macros constructed from C-modules can be obtained from the DirectTime Analyzer utility.*
4. *Setup and hold timing parameters for the input buffer latch are defined with respect to the PAD and the D input. External setup/hold timing parameters must account for delay from an external PAD signal to the G inputs. Delay from an external PAD signal to the G input subtracts (adds) to the internal setup (hold) time.*

A1280A Timing Characteristics (continued)

(Worst-Case Military Conditions)

Input Module Propagation Delays			–1 Speed		Std Speed		
Parameter	**Description**		**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{INYH}	Pad to Y High			4.0		4.7	ns
t_{INYL}	Pad to Y Low			3.6		4.3	ns
t_{INGH}	G to Y High			6.9		8.1	ns
t_{INGL}	G to Y Low			6.6		7.7	ns
Input Module Predicted Routing Delays[1]							
t_{RD1}	FO=1 Routing Delay			6.2		7.3	ns
t_{RD2}	FO=2 Routing Delay			7.2		8.4	ns
t_{RD3}	FO=3 Routing Delay			7.7		9.1	ns
t_{RD4}	FO=4 Routing Delay			8.9		10.5	ns
t_{RD8}	FO=8 Routing Delay			12.9		15.2	ns
Global Clock Network							
t_{CKH}	Input Low to High	FO = 32 FO = 384		13.3 17.9		15.7 21.1	ns
t_{CKL}	Input High to Low	FO = 32 FO = 384		13.3 18.2		15.7 21.4	ns
t_{PWH}	Minimum Pulse Width High	FO = 32 FO = 384	6.9 7.9		8.1 9.3		ns
t_{PWL}	Minimum Pulse Width Low	FO = 32 FO = 384	6.9 7.9		8.1 9.3		ns
t_{CKSW}	Maximum Skew	FO = 32 FO = 384		0.6 3.1		0.6 3.1	ns
t_{SUEXT}	Input Latch External Setup	FO = 32 FO = 384	0.0 0.0		0.0 0.0		ns
t_{HEXT}	Input Latch External Hold	FO = 32 FO = 384	8.6 13.8		8.6 13.8		ns
t_P	Minimum Period	FO = 32 FO = 384	13.7 16.0		16.2 18.9		ns
f_{MAX}	Maximum Frequency	FO = 32 FO = 384		73 63		62 53	MHz

Note:

[1] *These parameters should be used for estimating device performance. Optimization techniques may further reduce delays by 0 to 4 ns. Routing delays are for typical designs across worst-case operating conditions. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

A1280A Timing Characteristics (continued)

(Worst-Case Military Conditions)

Output Module Timing		–1 Speed		Std Speed		
Parameter	Description	Min.	Max.	Min.	Max.	Units
TTL Output Module Timing[1]						
t_{DLH}	Data to Pad High		11.0		13.0	ns
t_{DHL}	Data to Pad Low		13.9		16.4	ns
t_{ENZH}	Enable Pad Z to High		12.3		14.4	ns
t_{ENZL}	Enable Pad Z to Low		16.1		19.0	ns
t_{ENHZ}	Enable Pad High to Z		9.8		11.5	ns
t_{ENLZ}	Enable Pad Low to Z		11.5		13.6	ns
t_{GLH}	G to Pad High		12.4		14.6	ns
t_{GHL}	G to Pad Low		15.5		18.2	ns
d_{TLH}	Delta Low to High		0.09		0.11	ns/pF
d_{THL}	Delta High to Low		0.17		0.20	ns/pF
CMOS Output Module Timing[1]						
t_{DLH}	Data to Pad High		14.0		16.5	ns
t_{DHL}	Data to Pad Low		11.7		13.7	ns
t_{ENZH}	Enable Pad Z to High		12.3		14.4	ns
t_{ENZL}	Enable Pad Z to Low		16.1		19.0	ns
t_{ENHZ}	Enable Pad High to Z		9.8		11.5	ns
t_{ENLZ}	Enable Pad Low to Z		11.5		13.6	ns
t_{GLH}	G to Pad High		12.4		14.6	ns
t_{GHL}	G to Pad Low		15.5		18.2	ns
d_{TLH}	Delta Low to High		0.17		0.20	ns/pF
d_{THL}	Delta High to Low		0.12		0.15	ns/pF

Notes:

1. *Delays based on 50 pF loading.*
2. *SSO information can be found in the "Simultaneously Switching Output Limits for Actel FPGAs" application note on page 4-125.*

A1280XL Timing Characteristics

(Worst-Case Military Conditions)

		Preliminary Information				
Logic Module Propagation Delays[1]		**–1 Speed**		**Std Speed**		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{PD1}	Single Module		3.7		4.3	ns
t_{CO}	Sequential Clk to Q		3.7		4.3	ns
t_{GO}	Latch G to Q		3.7		4.3	ns
t_{RS}	Flip-Flop (Latch) Reset to Q		3.7		4.3	ns
Predicted Routing Delays[2]						
t_{RD1}	FO=1 Routing Delay		1.7		2.1	ns
t_{RD2}	FO=2 Routing Delay		2.5		3.0	ns
t_{RD3}	FO=3 Routing Delay		3.1		3.6	ns
t_{RD4}	FO=4 Routing Delay		3.7		4.3	ns
t_{RD8}	FO=8 Routing Delay		7.0		8.3	ns
Sequential Timing Characteristics[3, 4]						
t_{SUD}	Flip-Flop (Latch) Data Input Setup	0.4		0.5		ns
t_{HD}	Flip-Flop (Latch) Data Input Hold	0.0		0.0		ns
t_{SUENA}	Flip-Flop (Latch) Enable Setup	1.1		1.2		ns
t_{HENA}	Flip-Flop (Latch) Enable Hold	0.0		0.0		ns
t_{WCLKA}	Flip-Flop (Latch) Clock Active Pulse Width	5.3		6.1		ns
t_{WASYN}	Flip-Flop (Latch) Asynchronous Pulse Width	5.3		6.1		ns
t_A	Flip-Flop Clock Input Period	10.7		12.3		ns
t_{INH}	Input Buffer Latch Hold	0.0		0.0		ns
t_{INSU}	Input Buffer Latch Setup	0.4		0.4		ns
t_{OUTH}	Output Buffer Latch Hold	0.0		0.0		ns
t_{OUTSU}	Output Buffer Latch Setup	0.4		0.4		ns
f_{MAX}	Flip-Flop (Latch) Clock Frequency		90		75	MHz

Notes:

1. *For dual-module macros, use $t_{PD1} + t_{RD1} + t_{PDn}$, $t_{CO} + t_{RD1} + t_{PDm}$, or $t_{PD1} + t_{RD1} + t_{SUD}$, whichever is appropriate.*
2. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*
3. *Data applies to macros based on the S-module. Timing parameters for sequential macros constructed from C-modules can be obtained from the DirectTime Analyzer utility.*
4. *Setup and hold timing parameters for the input buffer latch are defined with respect to the PAD and the D input. External setup/hold timing parameters must account for delay from an external PAD signal to the G inputs. Delay from an external PAD signal to the G input subtracts (adds) to the internal setup (hold) time.*

A1280XL Timing Characteristics (continued)

(Worst-Case Military Conditions)

			Preliminary Information				
Input Module Propagation Delays			**–1 Speed**		**Std Speed**		
Parameter	**Description**		**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{INYH}	Pad to Y High			1.5		1.7	ns
t_{INYL}	Pad to Y Low			1.7		2.1	ns
t_{INGH}	G to Y High			2.8		3.3	ns
t_{INGL}	G to Y Low			3.7		4.3	ns
Input Module Predicted Routing Delays[1]							
t_{RD1}	FO=1 Routing Delay			4.6		5.3	ns
t_{RD2}	FO=2 Routing Delay			5.2		6.1	ns
t_{RD3}	FO=3 Routing Delay			5.5		6.5	ns
t_{RD4}	FO=4 Routing Delay			6.4		7.5	ns
t_{RD8}	FO=8 Routing Delay			9.2		10.8	ns
Global Clock Network							
t_{CKH}	Input Low to High	FO = 32 FO = 384		7.1 8.0		8.4 9.5	ns
t_{CKL}	Input High to Low	FO = 32 FO = 384		7.0 8.0		8.3 9.5	ns
t_{PWH}	Minimum Pulse Width High	FO = 32 FO = 384	4.3 4.8		5.3 5.7		ns
t_{PWL}	Minimum Pulse Width Low	FO = 32 FO = 384	4.3 4.8		5.3 5.7		ns
t_{CKSW}	Maximum Skew	FO = 32 FO = 384		1.1 1.1		1.2 1.2	ns
t_{SUEXT}	Input Latch External Setup	FO = 32 FO = 384	0.0 0.0		0.0 0.0		ns
t_{HEXT}	Input Latch External Hold	FO = 32 FO = 384	3.6 4.6		4.2 5.3		ns
t_P	Minimum Period	FO = 32 FO = 384	9.1 9.8		10.7 11.8		ns
f_{MAX}	Maximum Frequency	FO = 32 FO = 384		110 100		90 85	MHz

Note:

1. *These parameters should be used for estimating device performance. Optimization techniques may further reduce delays by 0 to 4 ns. Routing delays are for typical designs across worst-case operating conditions. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

A1280XL Timing Characteristics (continued)

(Worst-Case Military Conditions)

		Preliminary Information				
Output Module Timing		**–1 Speed**		**Std Speed**		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
TTL Output Module Timing[1]						
t_{DLH}	Data to Pad High		5.3		6.2	ns
t_{DHL}	Data to Pad Low		5.7		6.6	ns
t_{ENZH}	Enable Pad Z to High		5.3		6.2	ns
t_{ENZL}	Enable Pad Z to Low		5.8		6.8	ns
t_{ENHZ}	Enable Pad High to Z		7.5		8.9	ns
t_{ENLZ}	Enable Pad Low to Z		7.5		8.9	ns
t_{GLH}	G to Pad High		5.9		6.9	ns
t_{GHL}	G to Pad Low		6.6		7.8	ns
d_{TLH}	Delta Low to High		0.05		0.06	ns/pF
d_{THL}	Delta High to Low		0.05		0.09	ns/pF
CMOS Output Module Timing[1]						
t_{DLH}	Data to Pad High		6.6		7.9	ns
t_{DHL}	Data to Pad Low		4.7		5.5	ns
t_{ENZH}	Enable Pad Z to High		5.3		6.2	ns
t_{ENZL}	Enable Pad Z to Low		5.8		6.8	ns
t_{ENHZ}	Enable Pad High to Z		7.5		8.9	ns
t_{ENLZ}	Enable Pad Low to Z		7.5		8.9	ns
t_{GLH}	G to Pad High		5.9		6.9	ns
t_{GHL}	G to Pad Low		6.6		7.8	ns
d_{TLH}	Delta Low to High		0.07		0.09	ns/pF
d_{THL}	Delta High to Low		0.06		0.09	ns/pF

Notes:

1. *Delays based on 50 pF loading.*
2. *SSO information can be found in the "Simultaneously Switching Output Limits for Actel FPGAs" application note on page 4-125.*

A1425A Timing Characteristics

(Worst-Case Military Conditions)

Logic Module Propagation Delays[1]		–1 Speed		Std Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{PD}	Internal Array Module		3.0		3.5	ns
t_{CO}	Sequential Clock to Q		3.0		3.5	ns
t_{CLR}	Asynchronous Clear to Q		3.0		3.5	ns
Predicted Routing Delays[2]						
t_{RD1}	FO=1 Routing Delay		1.3		1.5	ns
t_{RD2}	FO=2 Routing Delay		1.9		2.1	ns
t_{RD3}	FO=3 Routing Delay		2.1		2.5	ns
t_{RD4}	FO=4 Routing Delay		2.6		2.9	ns
t_{RD8}	FO=8 Routing Delay		4.2		4.9	ns
Logic Module Sequential Timing						
t_{SUD}	Flip-Flop (Latch) Data Input Setup	0.9		1.0		ns
t_{HD}	Flip-Flop (Latch) Data Input Hold	0.0		0.0		ns
t_{SUENA}	Flip-Flop (Latch) Enable Setup	0.9		1.0		ns
t_{HENA}	Flip-Flop (Latch) Enable Hold	0.0		0.0		ns
t_{WASYN}	Asynchronous Pulse Width	3.8		4.4		ns
t_{WCLKA}	Flip-Flop Clock Pulse Width	3.8		4.4		ns
t_A	Flip-Flop Clock Input Period	7.9		9.3		ns
f_{MAX}	Flip-Flop Clock Frequency		125		100	MHz

Notes:

1. *For dual-module macros, use $t_{PD} + t_{RD1} + t_{PDn}$, $t_{CO} + t_{RD1} + t_{PDn}$, or $t_{PD1} + t_{RD1} + t_{SUD}$, whichever is appropriate.*
2. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

A1425A Timing Characteristics (continued)

(Worst-Case Military Conditions)

I/O Module Input Propagation Delays		–1 Speed		Std Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{INY}	Input Data Pad to Y		4.2		4.9	ns
t_{ICKY}	Input Reg IOCLK Pad to Y		7.0		8.2	ns
t_{OCKY}	Output Reg IOCLK Pad to Y		7.0		8.2	ns
t_{ICLRY}	Input Asynchronous Clear to Y		7.0		8.2	ns
t_{OCLRY}	Output Asynchronous Clear to Y		7.0		8.2	ns
Predicted Input Routing Delays[1]						
t_{IRD1}	FO=1 Routing Delay		1.3		1.5	ns
t_{IRD2}	FO=2 Routing Delay		1.9		2.1	ns
t_{IRD3}	FO=3 Routing Delay		2.1		2.5	ns
t_{IRD4}	FO=4 Routing Delay		2.6		2.9	ns
t_{IRD8}	FO=8 Routing Delay		4.2		4.9	ns
I/O Module Sequential Timing						
t_{INH}	Input F-F Data Hold (w.r.t. IOCLK Pad)	0.0		0.0		ns
t_{INSU}	Input F-F Data Setup (w.r.t. IOCLK Pad)	2.1		2.4		ns
t_{IDEH}	Input Data Enable Hold (w.r.t. IOCLK Pad)	0.0		0.0		ns
t_{IDESU}	Input Data Enable Setup (w.r.t. IOCLK Pad)	8.7		10.0		ns
t_{OUTH}	Output F-F Data Hold (w.r.t. IOCLK Pad)	1.1		1.2		ns
t_{OUTSU}	Output F-F Data Setup (w.r.t. IOCLK Pad)	1.1		1.2		ns
t_{ODEH}	Output Data Enable Hold (w.r.t. IOCLK Pad)	0.5		0.6		ns
t_{ODESU}	Output Data Enable Setup (w.r.t. IOCLK Pad)	2.0		2.4		ns

Note:

1. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

A1425A Timing Characteristics (continued)

(Worst-Case Military Conditions)

I/O Module – TTL Output Timing[1]		–1 Speed		Std Speed		
Parameter	Description	Min.	Max.	Min.	Max.	Units
t_{DHS}	Data to Pad, High Slew		7.5		8.9	ns
t_{DLS}	Data to Pad, Low Slew		11.9		14.0	ns
t_{ENZHS}	Enable to Pad, Z to H/L, Hi Slew		6.0		7.0	ns
t_{ENZLS}	Enable to Pad, Z to H/L, Lo Slew		10.9		12.8	ns
t_{ENHSZ}	Enable to Pad, H/L to Z, Hi Slew		9.9		11.6	ns
t_{ENLSZ}	Enable to Pad, H/L to Z, Lo Slew		9.9		11.6	ns
t_{CKHS}	IOCLK Pad to Pad H/L, Hi Slew		10.5		11.6	ns
t_{CKLS}	IOCLK Pad to Pad H/L, Lo Slew		15.7		17.4	ns
d_{TLHHS}	Delta Low to High, Hi Slew		0.04		0.04	ns/pF
d_{TLHLS}	Delta Low to High, Lo Slew		0.07		0.08	ns/pF
d_{THLHS}	Delta High to Low, Hi Slew		0.05		0.06	ns/pF
d_{THLLS}	Delta High to Low, Lo Slew		0.07		0.08	ns/pF
I/O Module – CMOS Output Timing[1]						
t_{DHS}	Data to Pad, High Slew		9.2		10.8	ns
t_{DLS}	Data to Pad, Low Slew		17.3		20.3	ns
t_{ENZHS}	Enable to Pad, Z to H/L, Hi Slew		7.7		9.1	ns
t_{ENZLS}	Enable to Pad, Z to H/L, Lo Slew		13.1		15.5	ns
t_{ENHSZ}	Enable to Pad, H/L to Z, Hi Slew		9.9		11.6	ns
t_{ENLSZ}	Enable to Pad, H/L to Z, Lo Slew		10.5		11.6	ns
t_{CKHS}	IOCLK Pad to Pad H/L, Hi Slew		12.5		13.7	ns
t_{CKLS}	IOCLK Pad to Pad H/L, Lo Slew		18.1		20.1	ns
d_{TLHHS}	Delta Low to High, Hi Slew		0.06		0.07	ns/pF
d_{TLHLS}	Delta Low to High, Lo Slew		0.11		0.13	ns/pF
d_{THLHS}	Delta High to Low, Hi Slew		0.04		0.05	ns/pF
d_{THLLS}	Delta High to Low, Lo Slew		0.05		0.06	ns/pF

Notes:

1. *Delays based on 35 pF loading.*
2. *SSO information can be found in the "Simultaneously Switching Output Limits for Actel FPGAs" application note on page 4-125.*

A1425A Timing Characteristics (continued)

(Worst-Case Military Conditions)

Dedicated (Hard-Wired) I/O Clock Network		–1 Speed		Std Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{IOCKH}	Input Low to High (Pad to I/O Module Input)		3.0		3.5	ns
t_{IOPWH}	Minimum Pulse Width High	3.9		4.4		ns
t_{IOPWL}	Minimum Pulse Width Low	3.9		4.4		ns
t_{IOSAPW}	Minimum Asynchronous Pulse Width	3.9		4.4		ns
t_{IOCKSW}	Maximum Skew		0.5		0.5	ns
t_{IOP}	Minimum Period	7.9		9.3		ns
f_{IOMAX}	Maximum Frequency		125		100	MHz
Dedicated (Hard-Wired) Array Clock Network						
t_{HCKH}	Input Low to High (Pad to S-Module Input)		4.6		5.3	ns
t_{HCKL}	Input High to Low (Pad to S-Module Input)		4.6		5.3	ns
t_{HPWH}	Minimum Pulse Width High	3.9		4.4		ns
t_{HPWL}	Minimum Pulse Width Low	3.9		4.4		ns
t_{HCKSW}	Maximum Skew		0.4		0.4	ns
t_{HP}	Minimum Period	7.9		9.3		ns
f_{HMAX}	Maximum Frequency		125		100	MHz
Routed Array Clock Networks						
t_{RCKH}	Input Low to High (FO=64)		5.5		6.4	ns
t_{RCKL}	Input High to Low (FO=64)		6.0		7.0	ns
t_{RPWH}	Min. Pulse Width High (FO=64)	4.9		5.7		ns
t_{RPWL}	Min. Pulse Width Low (FO=64)	4.9		5.7		ns
t_{RCKSW}	Maximum Skew (FO=128)		1.1		1.2	ns
t_{RP}	Minimum Period (FO=64)	10.1		11.6		ns
f_{RMAX}	Maximum Frequency (FO=64)		100		85	MHz
Clock-to-Clock Skews						
$t_{IOHCKSW}$	I/O Clock to H-Clock Skew	0.0	3.0	0.0	3.0	ns
$t_{IORCKSW}$	I/O Clock to R-Clock Skew	0.0	3.0	0.0	3.0	ns
t_{HRCKSW}	H-Clock to R-Clock Skew					
	(FO = 64)	0.0	1.0	0.0	1.0	ns
	(FO = 50% max.)	0.0	3.0	0.0	3.0	ns

Note:

1. *Delays based on 35 pF loading.*

A1460A Timing Characteristics

(Worst-Case Military Conditions)

Logic Module Propagation Delays[1]		–1 Speed		Std Speed		
Parameter	Description	Min.	Max.	Min.	Max.	Units
t_{PD}	Internal Array Module		3.0		3.5	ns
t_{CO}	Sequential Clock to Q		3.0		3.5	ns
t_{CLR}	Asynchronous Clear to Q		3.0		3.5	ns
Predicted Routing Delays[2]						
t_{RD1}	FO=1 Routing Delay		1.3		1.5	ns
t_{RD2}	FO=2 Routing Delay		1.9		2.1	ns
t_{RD3}	FO=3 Routing Delay		2.1		2.5	ns
t_{RD4}	FO=4 Routing Delay		2.6		2.9	ns
t_{RD8}	FO=8 Routing Delay		4.2		4.9	ns
Logic Module Sequential Timing						
t_{SUD}	Flip-Flop (Latch) Data Input Setup	0.9		1.0		ns
t_{HD}	Flip-Flop (Latch) Data Input Hold	0.0		0.0		ns
t_{SUENA}	Flip-Flop (Latch) Enable Setup	0.9		1.0		ns
t_{HENA}	Flip-Flop (Latch) Enable Hold	0.0		0.0		ns
t_{WASYN}	Asynchronous Pulse Width	4.8		5.6		ns
t_{WCLKA}	Flip-Flop Clock Pulse Width	4.8		5.6		ns
t_A	Flip-Flop Clock Input Period	9.9		11.6		ns
f_{MAX}	Flip-Flop Clock Frequency		100		85	MHz

Notes:

1. *For dual-module macros, use* $t_{PD} + t_{RD1} + t_{PDn}$, $t_{CO} + t_{RD1} + t_{PDn}$, *or* $t_{PD1} + t_{RD1} + t_{SUD}$, *whichever is appropriate.*
2. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

A1460A Timing Characteristics (continued)

(Worst-Case Military Conditions)

I/O Module Input Propagation Delays		–1 Speed		Std Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{INY}	Input Data Pad to Y		4.2		4.9	ns
t_{ICKY}	Input Reg IOCLK Pad to Y		7.0		8.2	ns
t_{OCKY}	Output Reg IOCLK Pad to Y		7.0		8.2	ns
t_{ICLRY}	Input Asynchronous Clear to Y		7.0		8.2	ns
t_{OCLRY}	Output Asynchronous Clear to Y		7.0		8.2	ns
Predicted Input Routing Delays[1]						
t_{IRD1}	FO=1 Routing Delay		1.3		1.5	ns
t_{IRD2}	FO=2 Routing Delay		1.9		2.1	ns
t_{IRD3}	FO=3 Routing Delay		2.1		2.5	ns
t_{IRD4}	FO=4 Routing Delay		2.6		2.9	ns
t_{IRD8}	FO=8 Routing Delay		4.2		4.9	ns
I/O Module Sequential Timing						
t_{INH}	Input F-F Data Hold (w.r.t. IOCLK Pad)	0.0		0.0		ns
t_{INSU}	Input F-F Data Setup (w.r.t. IOCLK Pad)	2.1		2.4		ns
t_{IDEH}	Input Data Enable Hold (w.r.t. IOCLK Pad)	0.0		0.0		ns
t_{IDESU}	Input Data Enable Setup (w.r.t. IOCLK Pad)	8.7		10.0		ns
t_{OUTH}	Output F-F Data Hold (w.r.t. IOCLK Pad)	1.1		1.2		ns
t_{OUTSU}	Output F-F Data Setup (w.r.t. IOCLK Pad)	1.1		1.2		ns
t_{ODEH}	Output Data Enable Hold (w.r.t. IOCLK Pad)	0.5		0.6		ns
t_{ODESU}	Output Data Enable Setup (w.r.t. IOCLK Pad)	2.0		2.4		ns

Note:

1. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

A1460A Timing Characteristics (continued)

(Worst-Case Military Conditions)

I/O Module – TTL Output Timing[1]		–1 Speed		Std Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{DHS}	Data to Pad, High Slew		7.5		8.9	ns
t_{DLS}	Data to Pad, Low Slew		11.9		14.0	ns
t_{ENZHS}	Enable to Pad, Z to H/L, Hi Slew		6.0		7.0	ns
t_{ENZLS}	Enable to Pad, Z to H/L, Lo Slew		10.9		12.8	ns
t_{ENHSZ}	Enable to Pad, H/L to Z, Hi Slew		11.5		13.5	ns
t_{ENLSZ}	Enable to Pad, H/L to Z, Lo Slew		10.9		12.8	ns
t_{CKHS}	IOCLK Pad to Pad H/L, Hi Slew		11.6		13.4	ns
t_{CKLS}	IOCLK Pad to Pad H/L, Lo Slew		17.8		19.8	ns
d_{TLHHS}	Delta Low to High, Hi Slew		0.04		0.04	ns/pF
d_{TLHLS}	Delta Low to High, Lo Slew		0.07		0.08	ns/pF
d_{THLHS}	Delta High to Low, Hi Slew		0.05		0.06	ns/pF
d_{THLLS}	Delta High to Low, Lo Slew		0.07		0.08	ns/pF
I/O Module – CMOS Output Timing[1]						
t_{DHS}	Data to Pad, High Slew		9.2		10.8	ns
t_{DLS}	Data to Pad, Low Slew		17.3		20.3	ns
t_{ENZHS}	Enable to Pad, Z to H/L, Hi Slew		7.7		9.1	ns
t_{ENZLS}	Enable to Pad, Z to H/L, Lo Slew		13.1		15.5	ns
t_{ENHSZ}	Enable to Pad, H/L to Z, Hi Slew		10.9		12.8	ns
t_{ENLSZ}	Enable to Pad, H/L to Z, Lo Slew		10.9		12.8	ns
t_{CKHS}	IOCLK Pad to Pad H/L, Hi Slew		14.1		16.0	ns
t_{CKLS}	IOCLK Pad to Pad H/L, Lo Slew		20.2		22.4	ns
d_{TLHHS}	Delta Low to High, Hi Slew		0.06		0.07	ns/pF
d_{TLHLS}	Delta Low to High, Lo Slew		0.11		0.13	ns/pF
d_{THLHS}	Delta High to Low, Hi Slew		0.04		0.05	ns/pF
d_{THLLS}	Delta High to Low, Lo Slew		0.05		0.06	ns/pF

Notes:

1. *Delays based on 35 pF loading.*
2. *SSO information can be found in the "Simultaneously Switching Output Limits for Actel FPGAs" application note on page 4-125.*

A1460A Timing Characteristics (continued)

(Worst-Case Military Conditions)

Dedicated (Hard-Wired) I/O Clock Network		–1 Speed		Std Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{IOCKH}	Input Low to High (Pad to I/O Module Input)		3.5		4.1	ns
t_{IOPWH}	Minimum Pulse Width High	4.8		5.7		ns
t_{IOPWL}	Minimum Pulse Width Low	4.8		5.7		ns
t_{IOSAPW}	Minimum Asynchronous Pulse Width	3.9		4.4		ns
t_{IOCKSW}	Maximum Skew		0.9		1.0	ns
t_{IOP}	Minimum Period	9.9		11.6		ns
f_{IOMAX}	Maximum Frequency		100		85	MHz
Dedicated (Hard-Wired) Array Clock Network						
t_{HCKH}	Input Low to High (Pad to S-Module Input)		5.5		6.4	ns
t_{HCKL}	Input High to Low (Pad to S-Module Input)		5.5		6.4	ns
t_{HPWH}	Minimum Pulse Width High	4.8		5.7		ns
t_{HPWL}	Minimum Pulse Width Low	4.8		5.7		ns
t_{HCKSW}	Maximum Skew		0.9		1.0	ns
t_{HP}	Minimum Period	9.9		11.6		ns
f_{HMAX}	Maximum Frequency		100		85	MHz
Routed Array Clock Networks						
t_{RCKH}	Input Low to High (FO=256)		9.0		10.5	ns
t_{RCKL}	Input High to Low (FO=256)		9.0		10.5	ns
t_{RPWH}	Min. Pulse Width High (FO=256)	6.3		7.1		ns
t_{RPWL}	Min. Pulse Width Low (FO=256)	6.3		7.1		ns
t_{RCKSW}	Maximum Skew (FO=128)		1.9		2.1	ns
t_{RP}	Minimum Period (FO=256)	12.9		14.5		ns
f_{RMAX}	Maximum Frequency (FO=256)		75		65	MHz
Clock-to-Clock Skews						
$t_{IOHCKSW}$	I/O Clock to H-Clock Skew	0.0	3.0	0.0	3.0	ns
$t_{IORCKSW}$	I/O Clock to R-Clock Skew	0.0	5.0	0.0	5.0	ns
t_{HRCKSW}	H-Clock to R-Clock Skew					
	(FO = 64)	0.0	1.0	0.0	1.0	ns
	(FO = 50% max.)	0.0	3.0	0.0	3.0	ns

Note:

1. *Delays based on 35 pF loading.*

A14100A Timing Characteristics

(Worst-Case Military Conditions)

Logic Module Propagation Delays[1]		–1 Speed		Std Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{PD}	Internal Array Module		3.0		3.5	ns
t_{CO}	Sequential Clock to Q		3.0		3.5	ns
t_{CLR}	Asynchronous Clear to Q		3.0		3.5	ns
Predicted Routing Delays[2]						
t_{RD1}	FO=1 Routing Delay		1.3		1.5	ns
t_{RD2}	FO=2 Routing Delay		1.9		2.1	ns
t_{RD3}	FO=3 Routing Delay		2.1		2.5	ns
t_{RD4}	FO=4 Routing Delay		2.6		2.9	ns
t_{RD8}	FO=8 Routing Delay		4.2		4.9	ns
Logic Module Sequential Timing						
t_{SUD}	Flip-Flop (Latch) Data Input Setup	1.0		1.0		ns
t_{HD}	Flip-Flop (Latch) Data Input Hold	0.6		0.6		ns
t_{SUENA}	Flip-Flop (Latch) Enable Setup	1.0		1.0		ns
t_{HENA}	Flip-Flop (Latch) Enable Hold	0.6		0.6		ns
t_{WASYN}	Asynchronous Pulse Width	4.8		5.6		ns
t_{WCLKA}	Flip-Flop Clock Pulse Width	4.8		5.6		ns
t_A	Flip-Flop Clock Input Period	9.9		11.6		ns
f_{MAX}	Flip-Flop Clock Frequency		100		85	MHz

Notes:

[1] *For dual-module macros, use* $t_{PD} + t_{RD1} + t_{PDn}$, $t_{CO} + t_{RD1} + t_{PDn}$, *or* $t_{PD1} + t_{RD1} + t_{SUD}$, *whichever is appropriate.*

[2] *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

A14100A Timing Characteristics (continued)

(Worst-Case Military Conditions)

I/O Module Input Propagation Delays		–1 Speed		Std Speed		
Parameter	Description	Min.	Max.	Min.	Max.	Units
t_{INY}	Input Data Pad to Y		4.2		4.9	ns
t_{ICKY}	Input Reg IOCLK Pad to Y		7.0		8.2	ns
t_{OCKY}	Output Reg IOCLK Pad to Y		7.0		8.2	ns
t_{ICLRY}	Input Asynchronous Clear to Y		7.0		8.2	ns
t_{OCLRY}	Output Asynchronous Clear to Y		7.0		8.2	ns
Predicted Input Routing Delays[1]						
t_{IRD1}	FO=1 Routing Delay		1.3		1.5	ns
t_{IRD2}	FO=2 Routing Delay		1.9		2.1	ns
t_{IRD3}	FO=3 Routing Delay		2.1		2.5	ns
t_{IRD4}	FO=4 Routing Delay		2.6		2.9	ns
t_{IRD8}	FO=8 Routing Delay		4.2		4.9	ns
I/O Module Sequential Timing						
t_{INH}	Input F-F Data Hold (w.r.t. IOCLK Pad)	0.0		0.0		ns
t_{INSU}	Input F-F Data Setup (w.r.t. IOCLK Pad)	2.1		2.4		ns
t_{IDEH}	Input Data Enable Hold (w.r.t. IOCLK Pad)	0.0		0.0		ns
t_{IDESU}	Input Data Enable Setup (w.r.t. IOCLK Pad)	8.7		10.0		ns
t_{OUTH}	Output F-F Data Hold (w.r.t. IOCLK Pad)	1.2		1.2		ns
t_{OUTSU}	Output F-F Data Setup (w.r.t. IOCLK Pad)	1.2		1.2		ns
t_{ODEH}	Output Data Enable Hold (w.r.t. IOCLK Pad)	0.6		0.6		ns
t_{ODESU}	Output Data Enable Setup (w.r.t. IOCLK Pad)	2.4		2.4		ns

Note:

[1] *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Post-route timing analysis or simulation is required to determine actual worst-case performance. Post-route timing is based on actual routing delay measurements performed on the device prior to shipment.*

A14100A Timing Characteristics (continued)

(Worst-Case Military Conditions)

I/O Module – TTL Output Timing[1]		–1 Speed		Std Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{DHS}	Data to Pad, High Slew		7.5		8.9	ns
t_{DLS}	Data to Pad, Low Slew		11.9		14.0	ns
t_{ENZHS}	Enable to Pad, Z to H/L, Hi Slew		6.0		7.0	ns
t_{ENZLS}	Enable to Pad, Z to H/L, Lo Slew		10.9		12.8	ns
t_{ENHSZ}	Enable to Pad, H/L to Z, Hi Slew		11.9		14.0	ns
t_{ENLSZ}	Enable to Pad, H/L to Z, Lo Slew		10.9		12.8	ns
t_{CKHS}	IOCLK Pad to Pad H/L, Hi Slew		12.2		14.0	ns
t_{CKLS}	IOCLK Pad to Pad H/L, Lo Slew		17.8		17.8	ns
d_{TLHHS}	Delta Low to High, Hi Slew		0.04		0.04	ns/pF
d_{TLHLS}	Delta Low to High, Lo Slew		0.07		0.08	ns/pF
d_{THLHS}	Delta High to Low, Hi Slew		0.05		0.06	ns/pF
d_{THLLS}	Delta High to Low, Lo Slew		0.07		0.08	ns/pF
I/O Module – CMOS Output Timing[1]						
t_{DHS}	Data to Pad, High Slew		9.2		10.8	ns
t_{DLS}	Data to Pad, Low Slew		17.3		20.3	ns
t_{ENZHS}	Enable to Pad, Z to H/L, Hi Slew		7.7		9.1	ns
t_{ENZLS}	Enable to Pad, Z to H/L, Lo Slew		13.1		15.5	ns
t_{ENHSZ}	Enable to Pad, H/L to Z, Hi Slew		11.6		14.0	ns
t_{ENLSZ}	Enable to Pad, H/L to Z, Lo Slew		10.9		12.8	ns
t_{CKHS}	IOCLK Pad to Pad H/L, Hi Slew		14.4		16.0	ns
t_{CKLS}	IOCLK Pad to Pad H/L, Lo Slew		20.2		22.4	ns
d_{TLHHS}	Delta Low to High, Hi Slew		0.06		0.07	ns/pF
d_{TLHLS}	Delta Low to High, Lo Slew		0.11		0.13	ns/pF
d_{THLHS}	Delta High to Low, Hi Slew		0.04		0.05	ns/pF
d_{THLLS}	Delta High to Low, Lo Slew		0.05		0.06	ns/pF

Notes:

1. *Delays based on 35 pF loading.*
2. *SSO information can be found in the "Simultaneously Switching Output Limits for Actel FPGAs" application note on page 4-125.*

A14100A Timing Characteristics (continued)

(Worst-Case Military Conditions)

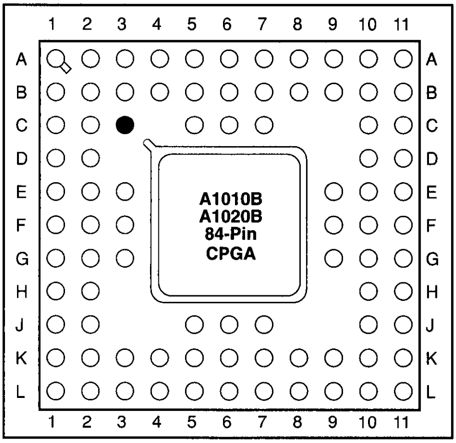
Dedicated (Hard-Wired) I/O Clock Network		–1 Speed		Std Speed		
Parameter	**Description**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**
t_{IOCKH}	Input Low to High (Pad to I/O Module Input)		3.5		4.1	ns
t_{IOPWH}	Minimum Pulse Width High	4.8		5.7		ns
t_{IOPWL}	Minimum Pulse Width Low	4.8		5.7		ns
t_{IOSAPW}	Minimum Asynchronous Pulse Width	3.9		4.4		ns
t_{IOCKSW}	Maximum Skew		0.9		1.0	ns
t_{IOP}	Minimum Period	9.9		11.6		ns
f_{IOMAX}	Maximum Frequency		100		85	MHz
Dedicated (Hard-Wired) Array Clock Network						
t_{HCKH}	Input Low to High (Pad to S-Module Input)		5.5		6.4	ns
t_{HCKL}	Input High to Low (Pad to S-Module Input)		5.5		6.4	ns
t_{HPWH}	Minimum Pulse Width High	4.8		5.7		ns
t_{HPWL}	Minimum Pulse Width Low	4.8		5.7		ns
t_{HCKSW}	Maximum Skew		0.9		1.0	ns
t_{HP}	Minimum Period	9.9		11.6		ns
f_{HMAX}	Maximum Frequency		100		85	MHz
Routed Array Clock Networks						
t_{RCKH}	Input Low to High (FO=256)		9.0		10.5	ns
t_{RCKL}	Input High to Low (FO=256)		9.0		10.5	ns
t_{RPWH}	Min. Pulse Width High (FO=256)	6.3		7.1		ns
t_{RPWL}	Min. Pulse Width Low (FO=256)	6.3		7.1		ns
t_{RCKSW}	Maximum Skew (FO=128)		1.9		2.1	ns
t_{RP}	Minimum Period (FO=256)	12.9		14.5		ns
f_{RMAX}	Maximum Frequency (FO=256)		75		65	MHz
Clock-to-Clock Skews						
$t_{IOHCKSW}$	I/O Clock to H-Clock Skew	0.0	3.5	0.0	3.5	ns
$t_{IORCKSW}$	I/O Clock to R-Clock Skew	0.0	5.0	0.0	5.0	ns
t_{HRCKSW}	H-Clock to R-Clock Skew					
	(FO = 64)	0.0	1.0	0.0	1.0	ns
	(FO = 50% max.)	0.0	3.0	0.0	3.0	

Note:

1. *Delays based on 35 pF loading.*

Package Pin Assignments

84-Pin CPGA (Top View)



● Orientation Pin (C3)

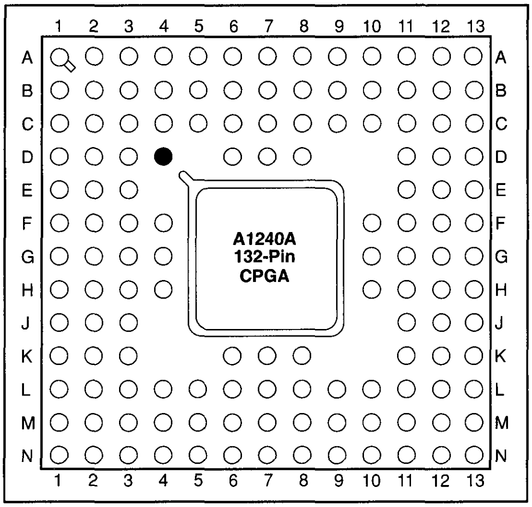
Signal	A1010B Devices	A1020B Devices
CLK or I/O	F9	F9
DCLK or I/O	C10	C10
GND	B7, E2, E3, F10, G10, K5	B7, E2, E3, F10, G10, K5
MODE	E11	E11
N/C (No Connection)	B1, B2, C1, C2, C11, D10, D11, J2, J10, K1, K10, K11, L1	B2
PRA or I/O	A11	A11
PRB or I/O	B10	B10
SDI or I/O	B11	B11
V_{CC}	B5, E9, E10, F1, G2, K2, K7	B5, E9, E10, F1, G2, K2, K7

Notes:

1. *V_{PP} must be terminated to V_{CC}, except during device programming.*
2. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage, otherwise it can be terminated directly to GND.*
3. *Unused I/O pins are designated as outputs by ALS and are driven low.*
4. *All unassigned pins are available for use as I/Os*
5. *The V_{PP}, V_{KS}, and V_{SV} pin names have been modified to reflect the normal system state (V_{CC} or GND) for these programming pins.*

Package Pin Assignments (continued)

132-Pin CPGA (Top View)



● Orientation Pin

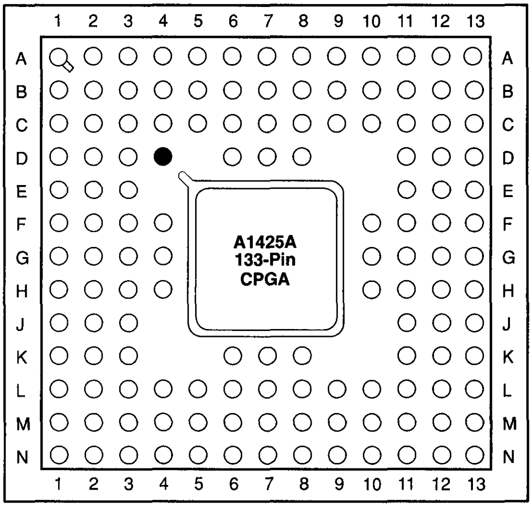
Signal	Location
CLKA or I/O	B7
CLKB or I/O	B6
DCLK or I/O	C3
GND	B5, B9, C5, C9, E3, E11, E12, F4, H13, J2, J3, J11, K12, L5, L9, M9
MODE	A1
PRA or I/O	B8
PRB or I/O	C6
SDI or I/O	B12
V_{CC}	C7, D7, G2, G3, G4, G10, G11, G12, G13, K7, L7

Notes:

1. *Unused I/O pins are designated as outputs by ALS and are driven low.*
2. *All unassigned pins are available for use as I/Os.*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage, otherwise it can be terminated directly to GND.*
4. *$V_{PP} = V_{CC}$, except during device programming.*
5. *$V_{SV} = V_{CC}$, except during device programming.*
6. *V_{KS} = GND, except during device programming.*
7. *The V_{PP}, V_{KS}, and V_{SV} pin names have been modified to reflect the normal system state (V_{CC} or GND) for these programming pins.*

Package Pin Assignments (continued)

133-Pin CPGA (Top View)



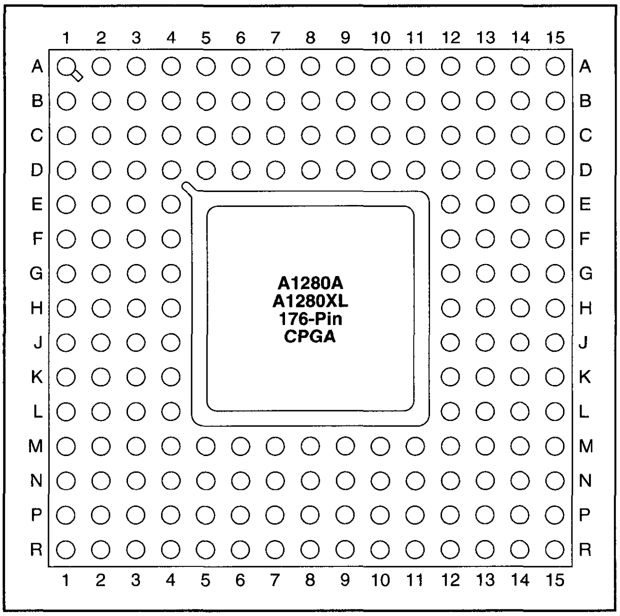
Signal	Location
CLKA or I/O	D7
CLKB or I/O	B6
DCLK or I/O	D4
GND	A2, C3, C7, C11, C12, F10, G3, G11, L3, L7, L11, M3, N12
HCLKA or I/O	K7
IOCLK or I/O	C10
IOPCL or I/O	L10
MODE	E3
NC	A1, A7, A13, G1, G13, N1, N7, N13
PRA or I/O	A6
PRB or I/O	L6
SDI or I/O	C2
V_{CC}	B2, B7, B12, E11, G2, G12, J2, J12, M2, M7, M12

Notes:

1. *Unused I/O pins are designated as outputs by ALS and are driven low.*
2. *All unassigned pins are available for use as I/Os.*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage, otherwise it can be terminated directly to GND.*
4. *$V_{PP} = V_{CC}$, except during device programming.*
5. *$V_{SV} = V_{CC}$, except during device programming.*
6. *V_{KS} = GND, except during device programming.*
7. *The V_{PP}, V_{KS}, and V_{SV} pin names have been modified to reflect the normal system state (V_{CC} or GND) for these programming pins.*

Package Pin Assignments (continued)

176-Pin CPGA (Top View)



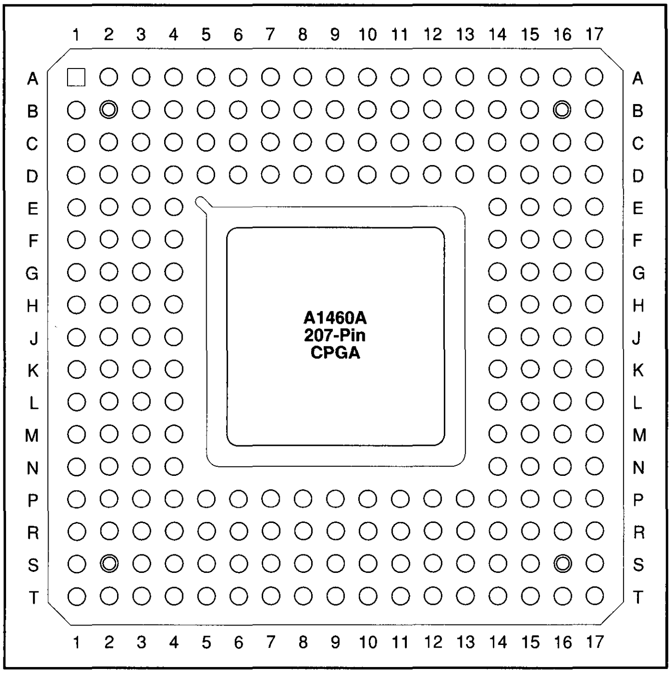
Signal	Location
CLKA or I/O	A9
CLKB or I/O	B8
DCLK or I/O	B3
GND	C8, D4, D6, D10, D12, E4, E12, F12, G4, H4, H12, J12, J13, K4, K12, L4, M4, M6, M8, M10, M12
MODE	C3
PRA or I/O	C9
PRB or I/O	D7
SDI or I/O	B14
V_{CC}	D5, D8, D11, F4, G12, H2, H3, H13, H14, J4, J14, M5, M11, N8

Notes:

1. *Unused I/O pins are designated as outputs by ALS and are driven low.*
2. *All unassigned pins are available for use as I/Os.*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage, otherwise it can be terminated directly to GND.*
4. *$V_{PP} = V_{CC}$, except during device programming.*
5. *$V_{SV} = V_{CC}$, except during device programming.*
6. *V_{KS} = GND, except during device programming.*
7. *The V_{PP}, V_{KS}, and V_{SV} pin names have been modified to reflect the normal system state (V_{CC} or GND) for these programming pins.*

Package Pin Assignments (continued)

207-Pin CPGA (Top View)



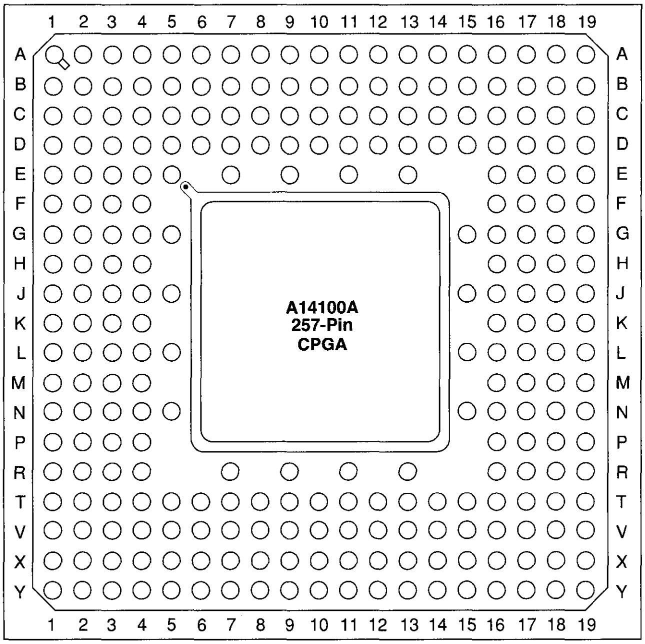
Signal	Location
CLKA or I/O	K1
CLKB or I/O	J3
DCLK or I/O	E4
GND	C15, D4, D5, D9, D14, J4, J14, P3, P4, P7, P9, P14, R15
HCKL or I/O	J15
IOCLK or I/O	P5
IOPCL or I/O	N14
MODE	D7
NC	A1, A2, A16, A17, B1, B17, C1, C2, S1, S3, S17, T1, T2, T16, T17
PRA OR I/O	H1
PRB or I/O	K16
SDI or I/O	C3
V_{CC}	B2, B9, B16, D11, J2, J16, P12, S2, S9, S16, T5

Notes:

1. *Unused I/O pins are designated as outputs by ALS and are driven low.*
2. *All unassigned pins are available for use as I/Os.*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage, otherwise it can be terminated directly to GND.*
4. *V_{PP} = V_{CC}, except during device programming.*
5. *V_{SV} = V_{CC}, except during device programming.*
6. *V_{KS} = GND, except during device programming.*
7. *The V_{PP}, V_{KS}, and V_{SV} pin names have been modified to reflect the normal system state (V_{CC} or GND) for these programming pins..*

Package Pin Assignments (continued)

257-Pin CPGA (Top View)



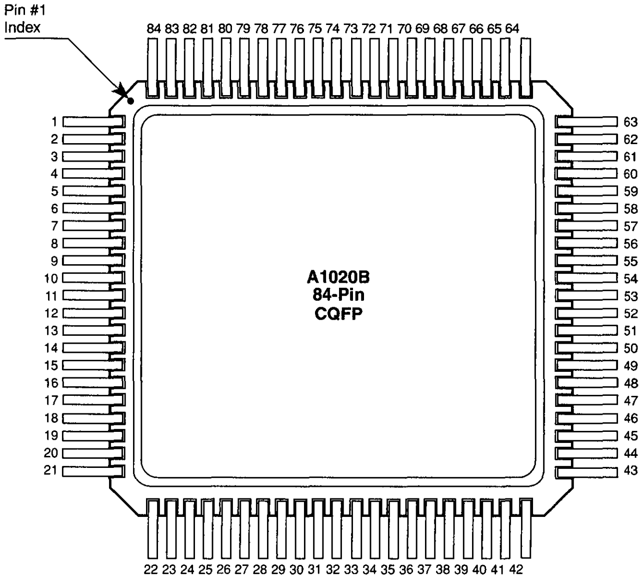
Signal	Location
CLKA or I/O	L4
CLKB or I/O	L5
DCLK or I/O	E4
GND	B16, C4, D4, D10, D16, E11, J5, K4, K16, L15, R4, T4, T10, T16, T17, X7
HCLK or I/O	J16
IOCLK or I/O	T5
IOPCL or I/O	R16
MODE	A5
NC	E5
PRA OR I/O	J1
PRB or I/O	J17
SDI or I/O	B4
V_{CC}	C3, C10, C13, C17, K3, K17, V3, V7, V10, V17, X14

Notes:

1. *Unused I/O pins are designated as outputs by ALS and are driven low.*
2. *All unassigned pins are available for use as I/Os.*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage, otherwise it can be terminated directly to GND.*
4. *$V_{PP} = V_{CC}$, except during device programming.*
5. *$V_{SV} = V_{CC}$, except during device programming.*
6. *V_{KS} = GND, except during device programming.*
7. *The V_{PP}, V_{KS}, and V_{SV} pin names have been modified to reflect the normal system state (V_{CC} or GND) for these programming pins.*

Package Pin Assignments (continued)

84-Pin CQFP (Top View)



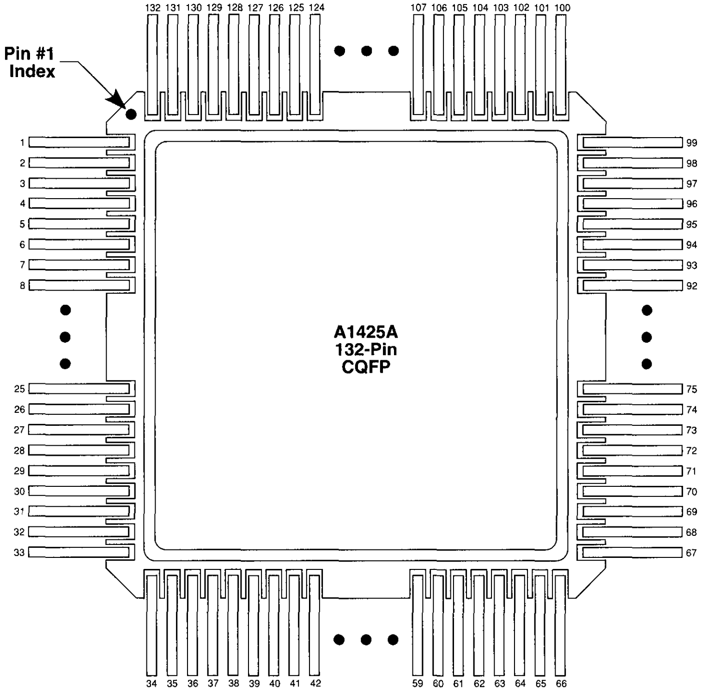
Signal	Location
CLKA or I/O	53
DCLK or I/O	62
GND	7, 8, 29, 49, 50, 71
MODE	55
N/C (No Connection)	1
PRA or I/O	63
PRB or I/O	64
SDI or I/O	61
V_{CC}	14, 15, 22, 35, 56, 57, 77

Notes:

1. *V_{PP} must be terminated to V_{CC}, except during device programming.*
2. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage, otherwise it can be terminated directly to GND.*
3. *Unused I/O pins are designated as outputs by ALS and are driven low.*
4. *All unassigned pins are available for use as I/Os*
5. *The V_{PP}, V_{KS}, and V_{SV} pin names have been modified to reflect the normal system state (V_{CC} or GND) for these programming pins.*

Package Pin Assignments (continued)

132-Pin CQFP (Top View)



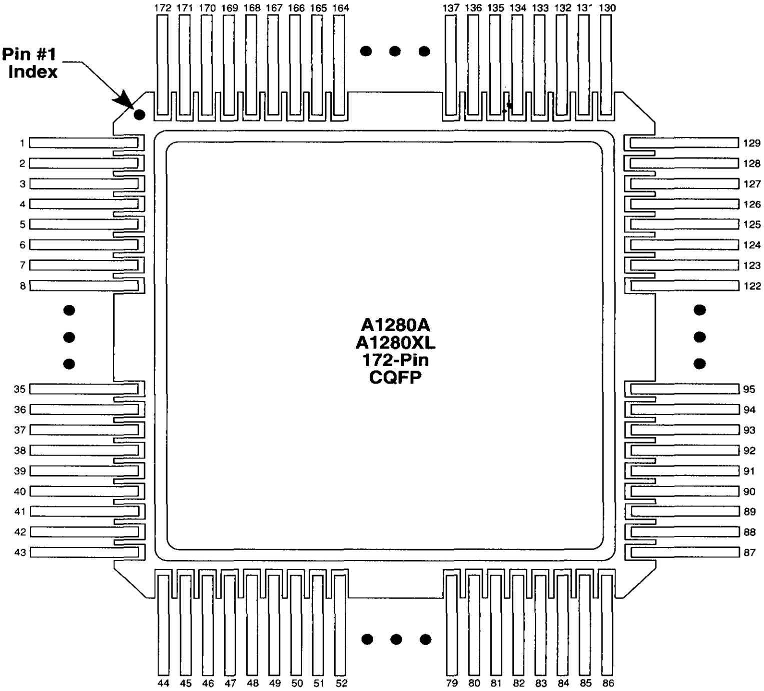
Signal	Pin Number
CLKA or I/O	116
CLKB or I/O	117
DCLK or I/O	131
GND	2, 10, 26, 36, 42, 58, 65, 74, 90, 92, 101, 106, 122
HCLK or I/O	50
IOCLK or I/O	98
IOPCL or I/O	64
MODE	9
NC	1, 34, 66, 67, 99, 100, 132
PRA or I/O	118
PRB or I/O	48
SDI or I/O	3
V_{CC}	11, 22, 27, 43, 59, 75, 78, 89, 91, 107, 123

Notes:

1. *Unused I/O pins are designated as outputs by ALS and are driven low.*
2. *All unassigned pins are available for use as I/Os.*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage, otherwise it can be terminated directly to GND.*
4. *$V_{PP} = V_{CC}$, except during device programming.*
5. *$V_{SV} = V_{CC}$, except during device programming.*
6. *V_{KS} = GND, except during device programming.*
7. *The V_{PP}, V_{KS}, and V_{SV} pin names have been modified to reflect the normal system state (V_{CC} or GND) for these programming pins.*

Package Pin Assignments (continued)

172-Pin CQFP (Top View)



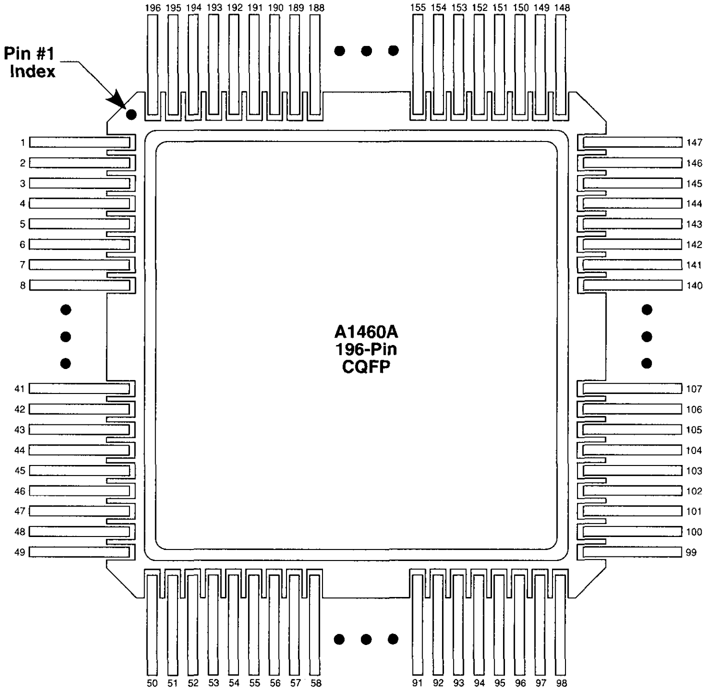
Signal	Pin Number
CLKA or I/O	150
CLKB or I/O	154
DCLK or I/O	171
GND	7, 17, 22, 32, 37, 55, 65, 75, 98, 103, 106, 108, 118, 123, 141, 152, 161
MODE	1
PRA or I/O	148
PRB or I/O	156
SDI or I/O	131
V_{CC}	12, 23, 24, 27, 50, 66, 80, 107, 109, 110, 113, 136, 151, 166

Notes:

1. *Unused I/O pins are designated as outputs by ALS and are driven low.*
2. *All unassigned pins are available for use as I/Os.*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage, otherwise it can be terminated directly to GND.*
4. *$V_{PP} = V_{CC}$, except during device programming.*
5. *$V_{SV} = V_{CC}$, except during device programming.*
6. *V_{KS} = GND, except during device programming.*
7. *The V_{PP}, V_{KS}, and V_{SV} pin names have been modified to reflect the normal system state (V_{CC} or GND) for these programming pins.*

Package Pin Assignments (continued)

196-Pin CQFP (Top View)



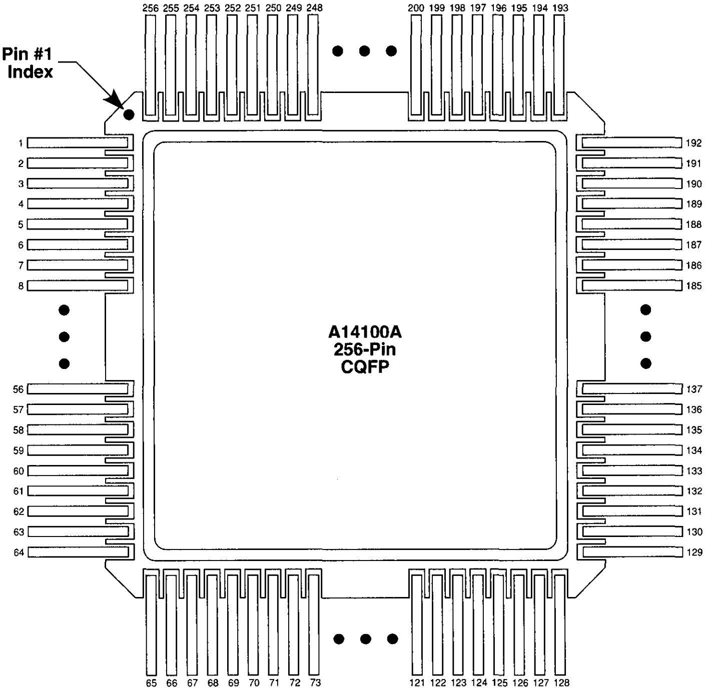
Signal	Pin Number
CLKA or I/O	172
CLKB or I/O	173
DCLK or I/O	196
GND	1, 13, 37, 51, 52, 64, 86, 98, 101, 112, 138, 139, 149, 162, 183, 193
HCLK or I/O	77
IOCLK or I/O	148
IOPCL or I/O	100
MODE	11
PRA or I/O	174
PRB or I/O	79
SDI or I/O	2
V_{CC}	12, 38, 39, 59, 94, 110, 111, 137, 140, 155, 189

Notes:

1. *Unused I/O pins are designated as outputs by ALS and are driven low.*
2. *All unassigned pins are available for use as I/Os.*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage, otherwise it can be terminated directly to GND.*
4. *$V_{PP} = V_{CC}$, except during device programming.*
5. *$V_{SV} = V_{CC}$, except during device programming.*
6. *V_{KS} = GND, except during device programming.*
7. *The V_{PP}, V_{KS}, and V_{SV} pin names have been modified to reflect the normal system state (V_{CC} or GND) for these programming pins.*

Package Pin Assignments (continued)

256-Pin CQFP (Top View)



Signal	Pin Number
CLKA or I/O	219
CLKB or I/O	220
DCLK or I/O	256
GND	1, 29, 31, 59, 91, 93, 110, 128, 158, 160, 175, 176, 189, 222, 224, 240
HCLK or I/O	96
IOCLK or I/O	188
IOPCL or I/O	127
MODE	11
PRA or I/O	225
PRB or I/O	90
SDI or I/O	2
V_{CC}	28, 30, 46, 92, 94, 141, 159, 161, 174, 221, 223

Notes:

1. *Unused I/O pins are designated as outputs by ALS and are driven low.*
2. *All unassigned pins are available for use as I/Os.*
3. *MODE should be terminated to GND through a 10K resistor to enable Actionprobe usage, otherwise it can be terminated directly to GND.*
4. *$V_{PP} = V_{CC}$, except during device programming.*
5. *$V_{SV} = V_{CC}$, except during device programming.*
6. *V_{KS} = GND, except during device programming.*
7. *The V_{PP}, V_{KS}, and V_{SV} pin names have been modified to reflect the normal system state (V_{CC} or GND) for these programming pins.*

Actel
Mask Programmed Gate Arrays

Features

- Mask Programmed versions of Actel Field Programmable Gate Arrays (FPGAs)
- Significant cost reduction for medium- to high-volume applications
- Pin-for-pin compatible with Actel FPGAs
- PCI Local Bus Revision 2 Compliant
- Automatic translation from Actel FPGA netlist to MPGA
- Test vectors generated from customer simulation vectors
- Short lead times for prototype and production devices
- MPGA available for all ACT 1, ACT 2, 1200XL, ACT 3, and 3200DX devices
- Device sizes from 1,200 to 10,000 gates
- Up to 175 user I/Os
- Available in commercial or industrial temperature ranges
- PLCC, PQFP, VQFP, and TQFP packages available
- Meets all internal worst-case FPGA performance specifications
- Lower I/O capacitance than FPGA
- Lower power dissipation than FPGA

Description

The Actel Mask Programmed Gate Array (MPGA) products are masked versions of the popular Actel FPGA families. These semi-custom devices offer the customer a design path that provides significant cost reduction without significant risk or engineering effort. For medium- to high-volume applications in which the design is fixed, the Actel FPGA used for prototyping and initial production can be replaced by the corresponding MPGA device.

Product Family Profile

	Capacity				Available Packages			
MPGA Device Type	Gate Array Equivalent Gates	PLD Equivalent Gates	Flip-Flops (Maximum)	User I/Os (Maximum)	PLCC	PQFP	VQFP	TQFP
M1010	1,200	3,000	147	57	44, 68-pin	100-pin	80-pin	—
M1020	2,000	6,000	273	69	44, 68, 84-pin	100-pin	80-pin	—
M1225	2,500	6,250	382	83	84-pin	100-pin	100-pin	—
M1240	4,000	10,000	568	104	84-pin	144-pin	—	176-pin
M1280	8,000	20,000	998	140	84-pin	160-pin	—	176-pin
M1415	1,500	3,750	312	80	84-pin	100-pin	100-pin	—
M1425	2,500	6,250	435	100	84-pin	100, 160-pin	100-pin	—
M1440	4,000	10,000	706	140	84-pin	160-pin	100-pin	176-pin
M1460	6,000	15,000	976	167	—	160, 208-pin	—	176-pin
M14100	10,000	25,000	1153	175	—	208-pin	—	—
M3265	6,500	1,600	747	126	84-pin	160-pin		176-pin
M32100	10,000	25,000	1031	152	84-pin	160, 208-pin	—	176-pin
M32140	14,000	35,000	1410	176	—	160, 208-pin	—	176-pin
M32200	20,000	50,000	1822	202	—	208, 240-pin	—	—
M32300	30,000	75,000	2804	250	—	240-pin	—	—
M32400	40,000	100,000	3759	288	—	240-pin	—	—

The granular, regular structure of the Actel antifuse-based FPGA products enables easy conversion to MPGA. Actel provides all required engineering services to convert the customer design from FPGA to MPGA, using proprietary software to automatically convert the FPGA logic design into the MPGA device. Test vector generation is made easy by software that converts the customer's third-party simulation vectors into the final vectors used to test the device in production.

All Actel MPGA devices are pin-for-pin compatible with the corresponding FPGA, and therefore no board redesign is required. MPGA devices meet all worst-case timing specifications of the FPGA devices. MPGA devices are available for all plastic packaged devices from ACT 1, ACT 2, 1200XL, ACT 3, and 3200DX families. See the “Product Plan” on page 1-260 for a detailed list of available device and package combinations.

Actel FPGA to MPGA Design Flow

Actel’s three families of FPGA devices offer a wide selection of device sizes, package choices, performance characteristics, and price points. The FPGA families provide the ideal prototyping tool \and are cost-effective for low- to medium-volume applications. As volumes increase, a cost-reduction path becomes a key factor to ensure continued success and profitability of the end product. Once the design has stabilized and volumes are increasing, a choice can be made to convert the design to an MPGA. Since the MPGA product is pin-for-pin compatible with the FPGA, no board redesign is required, and the MPGA can directly replace the FPGA.

A typical design process uses the FPGA device as the prototyping and initial production product of choice and converts to the MPGA as volumes warrant. Figure 1 shows the design process for Actel FPGA and MPGA devices. This option gives you the flexibility to adjust volumes as the demand for the end product changes. Since the MPGA is a semicustom device, all production is built to your order. If the design is already completed in the FPGA, any demand upsides can be satisfied by temporarily switching production back to the FPGA. Since Actel FPGAs are standard off-the-shelf devices, additional product requirements can be met within a short lead time.

The Actel FPGA devices offer the easiest and fastest way to bring a new product to market, and the three FPGA families offer a wide selection of low-cost, high-performance devices. The addition of the MPGA devices offers a simple, low-risk cost-reduction path as production volumes increase.

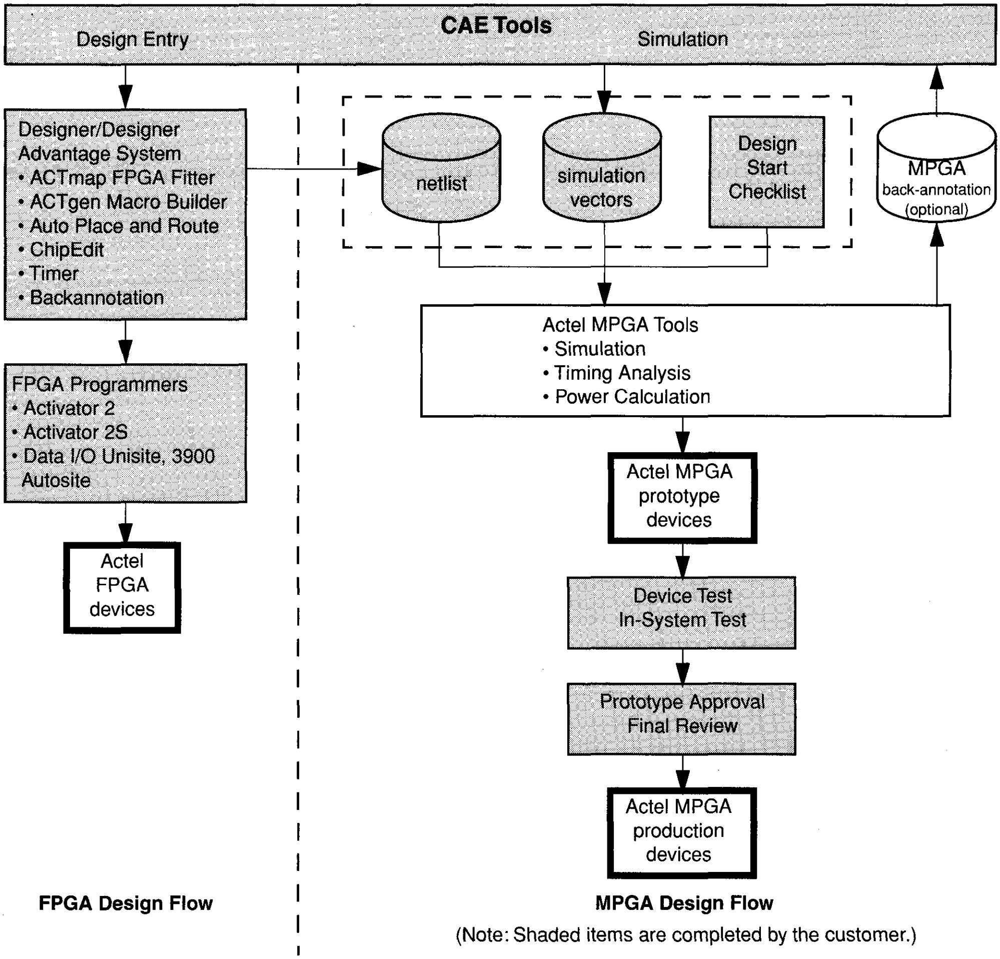


Figure 1* • *Actel Device Design Flow

Product Plan

	Availability	Application	
ACT 1 Family		**Commercial**	**Industrial**
M1010 Device			
44-pin Plastic Leaded Chip Carrier (PLCC)	✔	✔	✔
68-pin Plastic Leaded Chip Carrier (PLCC)	✔	✔	✔
80-pin Very Thin Plastic Quad Flatpack (VQFP)	✔	✔	—
100-pin Plastic Quad Flatpack (PQFP)	✔	✔	✔
M1020 Device			
44-pin Plastic Leaded Chip Carrier (PLCC)	✔	✔	✔
68-pin Plastic Leaded Chip Carrier (PLCC)	✔	✔	✔
80-pin Very Thin Plastic Quad Flatpack (VQFP)	✔	✔	—
84-pin Plastic Leaded Chip Carrier (PLCC)	✔	✔	✔
100-pin Plastic Quad Flatpack (PQFP)	✔	✔	✔
ACT 2/1200XL Family			
M1225 Device			
84-pin Plastic Leaded Chip Carrier (PLCC)	✔	✔	✔
100-pin Plastic Quad Flatpack (PQFP)	✔	✔	✔
100-pin Thin Plastic Quad Flatpack (TQFP)	✔	✔	—
M1240 Device			
84-pin Plastic Leaded Chip Carrier (PLCC)	✔	✔	✔
144-pin Plastic Quad Flatpack (PQFP)	✔	✔	✔
176-pin Thin Plastic Quad Flatpack (TQFP)	✔	✔	—
M1280 Device			
84-pin Plastic Leaded Chip Carrier (PLCC)	✔	✔	✔
160-pin Plastic Quad Flatpack (PQFP)	✔	✔	✔
176-pin Thin Plastic Quad Flatpack (TQFP)	✔	✔	—
ACT 3 Family			
M1415 Device			
84-pin Plastic Leaded Chip Carrier (PLCC)	✔	✔	✔
100-pin Plastic Quad Flatpack (PQFP)	✔	✔	✔
100-pin Very Thin Plastic Quad Flatpack (VQFP)	✔	✔	—
M1425 Device			
84-pin Plastic Leaded Chip Carrier (PLCC)	✔	✔	✔
100-pin Plastic Quad Flatpack (PQFP)	✔	✔	✔
100-pin Very Thin Plastic Quad Flatpack (VQFP)	✔	✔	—
160-pin Plastic Quad Flatpack (PQFP)	✔	✔	✔
M1440 Device			
84-pin Plastic Leaded Chip Carrier (PLCC)	✔	✔	✔
100-pin Very Thin Plastic Quad Flatpack (VQFP)	✔	✔	—
160-pin Plastic Quad Flatpack (PQFP)	✔	✔	✔
176-pin Thin Plastic Quad Flatpack (TQFP)	✔	✔	—
M1460 Device			
160-pin Plastic Quad Flatpack (PQFP)	✔	✔	✔
176-pin Thin Plastic Quad Flatpack (TQFP)	✔	✔	—
208-pin Plastic Quad Flatpack (PQFP)	✔	✔	✔
M14100 Device			
208-pin Plastic Quad Flatpack (PQFP)	✔	✔	✔

Product Plan (continued)

	Availability	Application	
3200DX Family			
M3265 Device			
84-pin Plastic Leaded Chip Carrier (PLCC)	✓	✓	✓
160-pin Plastic Quad Flatpack (PQFP)	✓	✓	✓
176-pin Thin Plastic Quad Flatpack (TQFP)	✓	✓	✓
M32100 Device			
84-pin Plastic Leaded Chip Carrier (PLCC)	P	P	P
160-pin Plastic Quad Flatpack (PQFP)	P	P	P
208-pin Plastic Quad Flatpack (PQFP)	P	P	P
176-pin Thin Plastic Quad Flatpack (TQFP)	P	P	P
M32140 Device			
160-pin Plastic Quad Flatpack (PQFP)	✓	✓	✓
208-pin Plastic Quad Flatpack (PQFP)	✓	✓	✓
176-pin Thin Plastic Quad Flatpack (TQFP)	✓	✓	✓
M32200 Device			
208-pin Plastic Quad Flatpack (PQFP)	P	P	P
240-pin Plastic Quad Flatpack (PQFP)	P	P	P
176-pin This Plastic Quad Flatpack (TQFP)	P	P	P
M32300 Device			
240-pin Plastic Quad Flatpack (PQFP)	P	P	P
M32400 Device			
240-pin Plastic Quad Flatpack (PQFP)	P	P	P

Availability:	✓	=	Available
	P	=	Planned
	—	=	Not Planned

ACT 1 Device Resources

		User I/Os				
		PLCC			PQFP	VQFP
MPGA Device Type	Gate Array Equivalent Gates	44-pin	68-pin	84-pin	100-pin	80-pin
M1010	1200	34	57	57	57	57
M1020	2000	34	57	69	69	69

ACT 2/1200XL Device Resources

		User I/Os					
		PLCC	PQFP			VQFP	TQFP
MPGA Device Type	Gate Array Equivalent Gates	84-pin	100-pin	144-pin	160-pin	100-pin	176-pin
M1225	2500	72	83	—	—	83	—
M1240	4000	72	—	104	—	—	104
M1280	8000	72	—	—	125	—	140

ACT 3 Device Resources

MPGA Device Type	Gate Array Equivalent Gates	User I/Os					
		PLCC	PQFP			VQFP	TQFP
		84-pin	100-pin	160-pin	208-pin	100-pin	176-pin
M1415	1500	70	80	—	—	80	—
M1425	2500	70	80	100	—	83	—
M1440	4000	70	—	131	—	83	140
M1460	6000	—	—	131	167	—	151
M14100	10000	—	—	—	175	—	—

3200DX Device Resources

MPGA Device Type	Gate Array Equivalent Gates	User I/Os				
		PLCC	PQFP			TQFP
		84-pin	160-pin	208-pin	240-pin	176-pin
M3265	6500	72	125	—	—	126
M32100	10000	72	125	156	—	151
M32140	14000	—	125	176	—	151
M32200	20000	—	—	176	TBD	—
M32300	30000	—	—	—	TBD	—
M32400	40000	—	—	—	TBD	—

Ordering Information

M14100 - RQ 208 C

Application (Temperature Range)
C = Commercial (0 to +70°C)
I = Industrial (–40 to +85°C)

Package Lead Count

Package Type
PL = Plastic Leaded Chip Carrier (PLCC)
PQ = Plastic Quad Flatpack (PQFP)
TQ = Thin (1.4 mm) Plastic Quad Flatpack (TQFP)
VQ = Very Thin (1.0 mm) Plastic Quad Flatpack (VQFP)
RQ = Power Plastic Quad Flatpack (RQFP)

Part Number
M1010= 1200 Gates—ACT 1
M1020= 2000 Gates—ACT 1
M1225= 2500 Gates—ACT 2/1200XL
M1240= 4000 Gates—ACT 2/1200XL
M1280= 8000 Gates—ACT 2/1200XL
M1415= 1500 Gates—ACT 3
M1425= 2500 Gates—ACT 3
M1440= 4000 Gates—ACT 3
M1460= 6000 Gates—ACT 3
M14100=10000 Gates—ACT 3
M3265= 6500 Gates—3200DX
M32100=10000 Gates—3200DX
M32140=14000 Gates—3200DX
M32200=20000 Gates—3200DX
M32300=30000 Gates—3200DX
M32400=40000 Gates—3200DX

Absolute Maximum Ratings[1]

Free air temperature range

Symbol	Parameter	Limits	Units
V_{CC}	DC Supply Voltage	–0.3 to +7.0	V
V_I	Input Voltage	–0.3 to V_{CC} +0.3	V
V_O	Output Voltage	–0.3 to V_{CC} +0.3	V
I_{IO}	I/O Source Sink Current	±20	mA
T_{STG}	Storage Temperature	–55 to +125	°C

Note:

[1] *Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum rated conditions for extended periods may affect device reliability. Device should not be operated outside the Recommended Operating Conditions.*

Recommended Operating Conditions

Parameter	Commercial	Industrial	Units
Temperature Range	0 to +70	–40 to +85	°C
Power Supply Tolerance	±5	±10	%V_{CC}

Electrical Specifications

			Commercial		Industrial		
Symbol	Parameter	Test Condition	Min.	Max.	Min.	Max.	Units
V_{OH}[1,2]	HIGH Level Output	I_{OH} = –6 mA (CMOS)	3.7		3.7		V
		I_{OH} = –8 mA (TTL)[3]	2.4		2.4		V
V_{OL}[1,2]	LOW Level Output	I_{OL} = +6 mA (CMOS)		0.4		0.4	V
		I_{OL} = +8 mA (TTL)[3]		0.4		0.4	V
V_{IH}	HIGH Level Input	TTL Inputs	2.0	V_{CC} + 0.3	2.0	V_{CC} + 0.3	V
V_{IL}	LOW Level Input	TTL Inputs	–0.3	0.8	–0.3	0.8	V
I_{IN}	Input Leakage	V_I = V_{CC} or GND	–1	+1	–1	+1	µA
I_{OZ}	3-state Output Leakage	V_O = V_{CC} or GND	–10	+10	–10	+10	µA
C_{IO}	I/O Capacitance[3]			10		10	pF
$I_{CC(S)}$	Standby Supply Current	V_I = V_{CC} or GND, I_O = 0 mA		100		500	µA

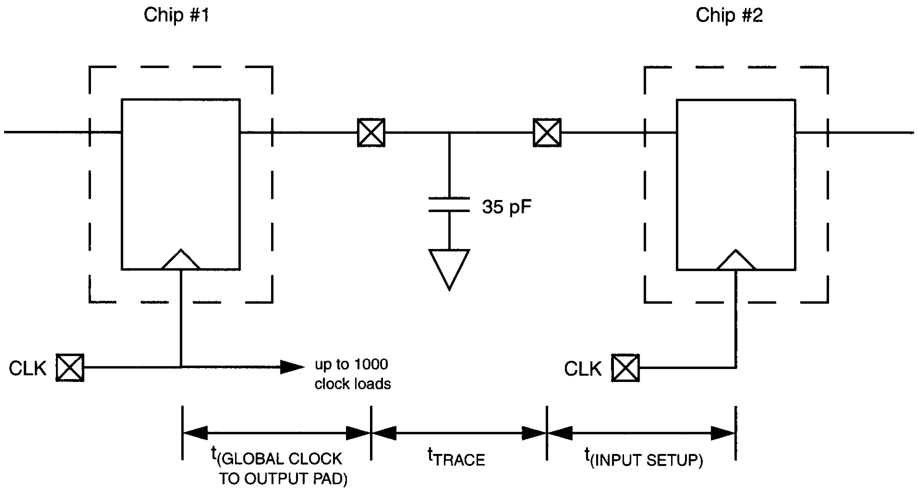
Notes:

[1] *Actel devices can drive and receive either CMOS or TTL signal levels. No assignment of I/Os as TTL or CMOS is required.*

[2] *Tested one output at a time, V_{CC} = min.*

[3] *Not tested, for information only.*

Chip-to-Chip Performance



Chip-to-Chip Performance (Worst-Case Commercial)					
	t(GLOBAL CLOCK TO OUPUT PAD)	tTRACE	t(INPUT SETUP)	Total	MHz
Actel MPGA	12.7	1.0	3.1	16.8	60

Pin Description

Package pin assignments for an FPGA design are directly transferred to the equivalent MPGA package because all I/O and power pins are located in identical positions. While the conversion of package pin assignments is transparent in the end product, there are two small functional differences to note between the device types. First, dedicated FPGA global and debugging pins are general purpose MPGA I/O pins. Also, dedicated FPGA programming voltage pins are Vcc or ground pins on an MPGA. Refer to Table 1 for a complete cross-reference of pin descriptions between the FPGA and MPGA.

Table 1 • FPGA-to-MPGA Pin Cross-Reference

FPGA Pin Description		MPGA Pin Description
CLK Clock (ACT 1 only) TTL Clock input for ACT 1 global clock distribution network. This pin can also be used as an I/O.	→	**No Change** If desired, TTL Clock input signals may be moved to any MPGA I/O location.
CLKA Clock A (ACT 3, 3200DX, 1200XL, and ACT 2 only) TTL Clock input for clock distribution networks. This pin can also be used as an I/O.	→	**No Change** If desired, TTL Clock input signals may be moved to any MPGA I/O location.
CLKB Clock B (ACT 3, 3200DX, 1200XL, and ACT 2 only) TTL Clock input for clock distribution networks. This pin can also be used as an I/O.	→	**No Change** If desired, TTL Clock input signals may be moved to any MPGA I/O location.
DCLK Diagnostic Clock TTL Clock input for diagnostic probe and device programming. Function is controlled by the MODE pin.	→	**I/O** This pin is used as an I/O only. It is not used for diagnostic probe or device programming functions on an MPGA.
GND Ground LOW supply voltage.	→	**Ground** LOW supply voltage.
HCLK Dedicated (Hard-wired) Array Clock (ACT 3 only) TTL Clock input for ACT 3 sequential modules. This pin can also be used as an I/O.	→	**No Change** If desired, TTL Clock input signals may be moved to any MPGA I/O location.
I/O Input/Output The I/O pin functions as an input, output, three-state, or bidirectional buffer. Unused pins are automatically driven LOW by the Designer software.	→	**I/O** User-defined MPGA I/O pins function identically to their FPGA counterparts. However, unused pins are NC (no connection) pins.
IOCLK Dedicated (Hard-wired) I/O Clock (ACT 3 only) TTL Clock input for ACT 3 I/O modules. This pin can also be used as an I/O.	→	**No Change** If desired, TTL Clock input signals may be moved to any MPGA I/O location.
IOPCL Dedicated (Hard-wired) I/O Preset/Clear (ACT 3 only) TTL input for ACT 3 I/O preset or clear. This pin can also be used as an I/O.	→	**No Change** If desired, this input signal may be moved to any MPGA I/O location.
MODE Mode The MODE pin controls the use of diagnostic pins (DCLK, PRA, PRB, SDI). When the MODE pin is HIGH, the special functions are active. When the MODE pin is LOW, the pins function as I/Os.	→	**TEST (No Connection)** This pin is reserved for parametric testing and should be connected to ground (LOW supply voltage).

Table 1 • FPGA-to-MPGA Pin Cross-Reference (continued)

FPGA Pin Description		MPGA Pin Description
NC No Connection This pin is not connected to circuitry within the device.	→	**NC No Connection** This pin is not connected to circuitry within the device.
PRA Probe A The Probe A pin is used for FPGA diagnostics. Function is controlled by the MODE pin.	→	**I/O** This pin is used as an I/O only. It is not used for diagnostic probe or device programming functions on an MPGA.
PRB Probe B The Probe B pin is used for FPGA diagnostics. Function is controlled by the MODE pin.	→	**I/O** This pin is used as an I/O only. It is not used for diagnostic probe or device programming functions on an MPGA.
QCLKA/B,C,D Quadrant Clock (Input/Output) (3200DX only) These four pins are the quadrant clock inputs. When not used as a register control signal, these pins can function as general purpose I/O.	→	**No Change** If desired, TTL Clock input signals may be moved to any MPGA location.
SDI Serial Data Input Serial data input for diagnostic probe and device programming. Function is controlled by the MODE pin.	→	**I/O** This pin is used as an I/O only. It is not used for diagnostic probe or device programming functions on an MPGA.
TCK Test Clock (3200DX only) Clock signal to shift the JTAG data into the device. This pin functions as an I/O when the JTAG fuse is not programmed	→	**No Change**
TDI Test Data In (3200DX only) Serial data input or JTAG instructions and data. Data is shifted in on the rising edge of TCLK. This pin functions as an I/O when the JTAG fuse is not programmed.	→	**No Change**
TDO Test Data Out (3200DX only) Serial data output for JTAG instructions and test data. This pin functions as an I/O when the JTAG fuse is not programmed.	→	**No Change**
TMS Test Mode Select (3200DX only) Serial data input for JTAG test mode. Data is shifted in on the rising edge of TCLK. This pin functions as an I/O when the JTAG fuse is not programmed.	→	**No Change**
V_{CC} **Supply Voltage** HIGH supply voltage.	→	V_{CC} HIGH supply voltage.

MPGA Architecture

The Actel MPGA is built using a "sea-of-gates" architecture. A solid, regularly ordered array of transistors is overlaid with a multilevel metal interconnect. Surrounding this logic core is an array of programmable power and I/O pads. Separate grids provide power and ground supplies for the core logic and I/O cells.

The highly dense structure of Actel MPGAs provides for a cost-effective solution while maintaining the high performance of each particular design. This architecture reduces die size for low cost while minimizing gate length and shortening routing paths for excellent system performance. The robust power supply grids provide high I/O current drive without sacrificing high noise immunity. Since Actel FPGAs use a similar gate array architecture, design migration is a straightforward, simple process. Because of the advanced technology employed by the MPGA, the internal and external performance of each design is virtually assured to be preserved or improved after migration. To simplify migration further, the I/O pads are carefully arranged to allow FPGA pin assignments to be directly transferred to the full line of MPGA packages. For more information about the ease of design migration from Actel FPGAs to MPGAs, see the application note "Designing for Migration to Actel MPGAs."

Power Dissipation

The power dissipation for an Actel MPGA is composed of two parts: static power and active power. The static power is a product of the standby supply current (Icc) and the DC supply voltage (Vcc). Specifications for Icc and Vcc are located in the "Electrical Specifications" section of this data sheet. The active power is a product of equivalent capacitance, square of the DC supply voltage, and average switching frequency of the circuit. It is expressed in the formula

$$\text{Power } (\mu W) = C_{EQ} \bullet V_{CC}^2 \bullet f$$

where

C_{EQ} is the equivalent capacitance in picofarads (pF)

V_{CC} is the DC supply voltage in volts (V)

f is the switching frequency in megahertz (MHz)

Upon receipt of the "Design Start Checklist" and associated materials, Actel calculates the MPGA active power dissipation for each design based on this formula. This calculation is immediately relayed to you so that you can update system power specifications accordingly. Typically, power dissipation of an Actel design is significantly lower for the MPGA version versus the FPGA version.

Timing Characteristics

The timing characteristics for Actel MPGA devices are consistent across family and device types. Typical I/O buffer, internal logic cell, and internal routing delays are common to all MPGA devices. The advanced technology of the devices ensures converted designs meet or exceed FPGA performance. Refer to the MPGA Timing Model diagram and Timing Characteristics chart for detailed timing and delay estimates.

Timing Derating

Timing derating factors due to temperature, voltage, and process variations are summarized in the following tables and graphs. Use these derating factors to determine device performance at any particular condition within the electrical and environmental specifications.

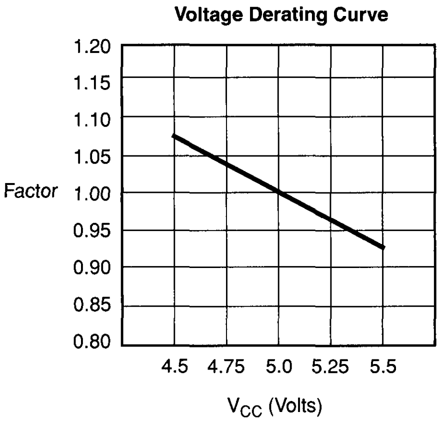
MPGA devices are manufactured in a CMOS process. Therefore, device performance varies according to temperature, voltage, and process variations. Minimum timing parameters reflect maximum operating voltage, minimum operating temperature, and best-case processing. Maximum timing parameters reflect minimum operating voltage, maximum operating temperature, and worst-case processing.

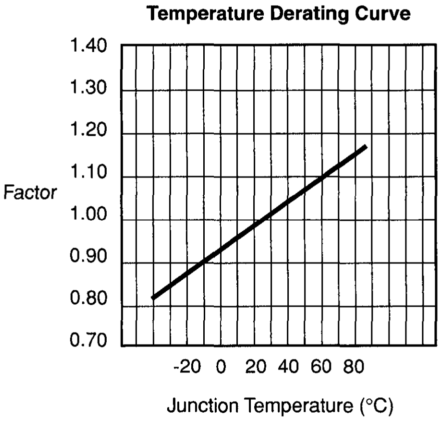
Timing Derating Factor, Temperature and Voltage

	Industrial	
	Minimum	Maximum
(Commercial Minimum/Maximum Specification) x	0.85	1.07

Timing Derating Factor for Designs at Typical Temperature (T_J = 25°C) and Voltage (V_{CC} = 5.0 V)

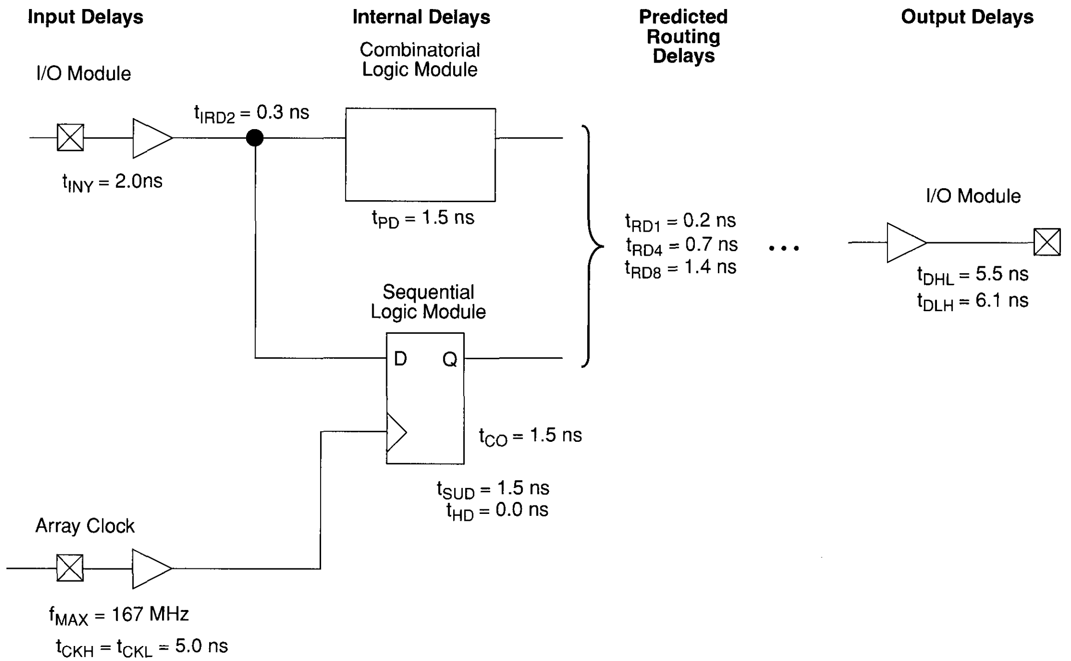
(Commercial Maximum Specification) x	0.86



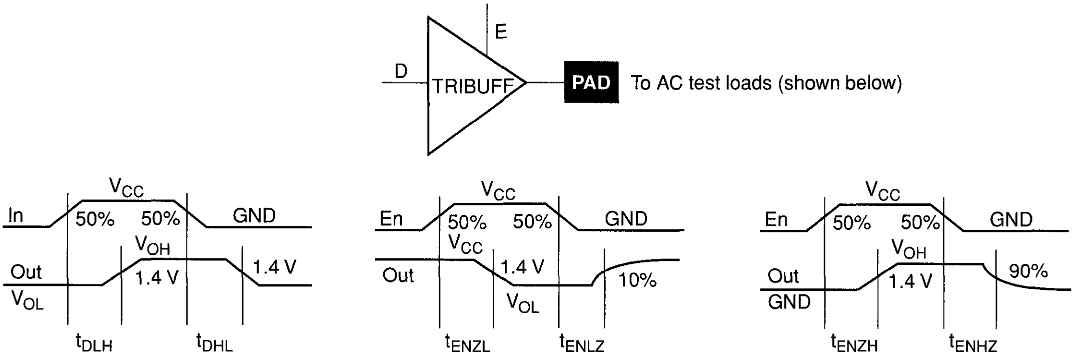


Note: This derating factor applies to all routing and propagation delays.

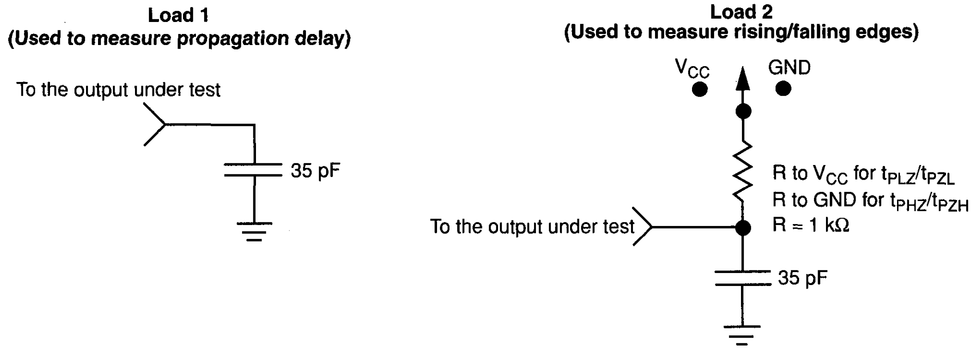
MPGA Timing Model



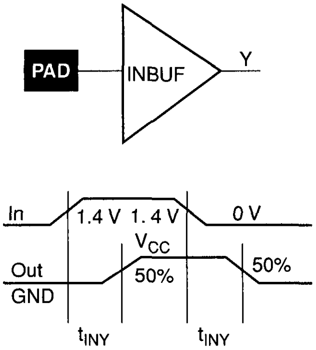
Output Buffer Delays



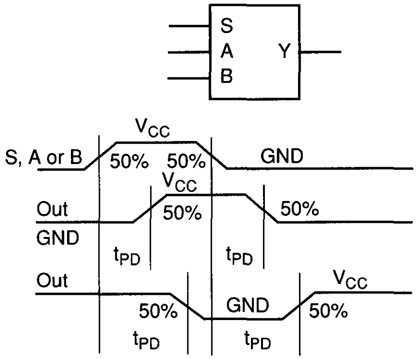
AC Test Loads


Load 1
(Used to measure propagation delay)
To the output under test
35 pF
Load 2
(Used to measure rising/falling edges)
V_{CC}
GND
R to V_{CC} for t_{PLZ}/t_{PZL}
R to GND for t_{PHZ}/t_{PZH}
R = 1 kΩ
To the output under test
35 pF

Input Buffer Delays

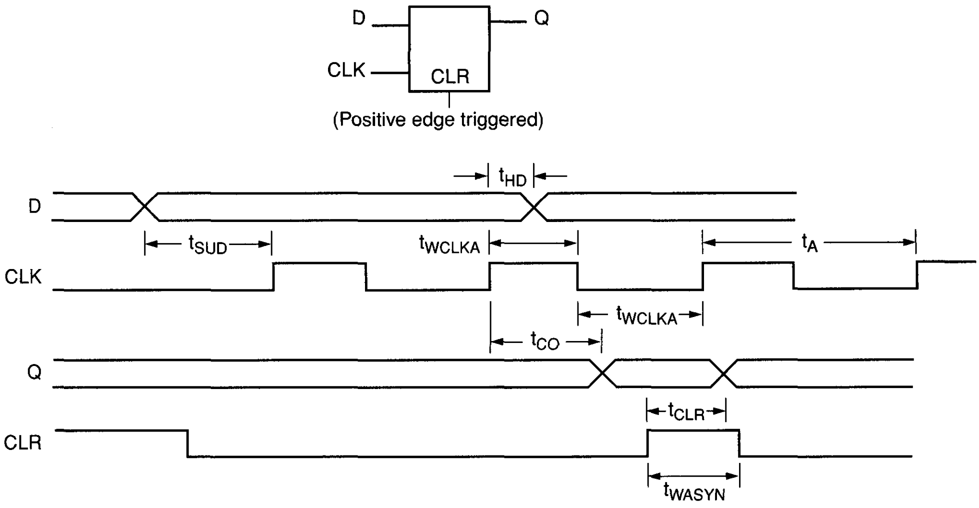


Module Delays



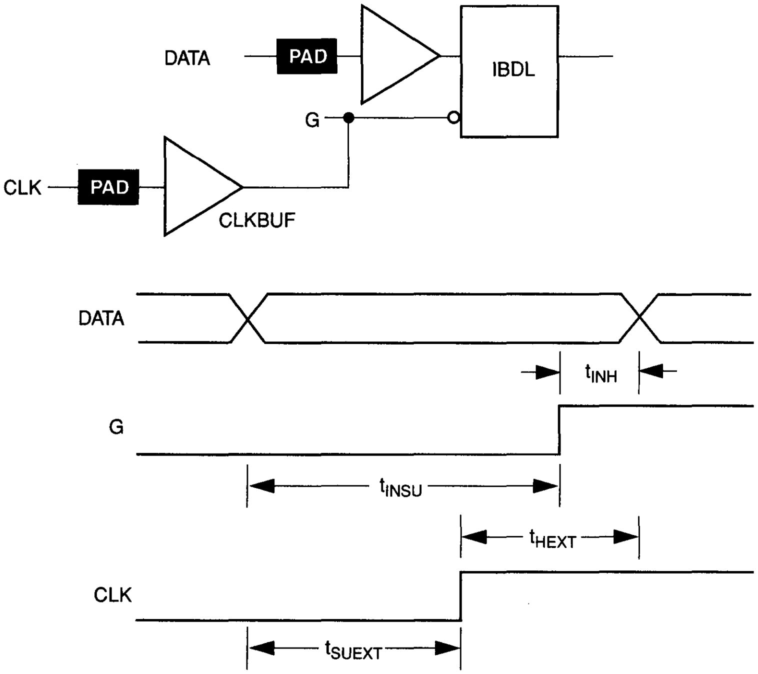
Sequential Module Timing Characteristics

Flip-Flops

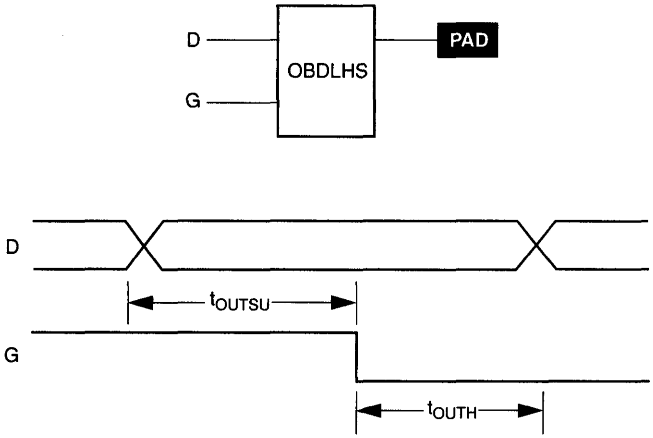


Sequential Timing Characteristics (continued)

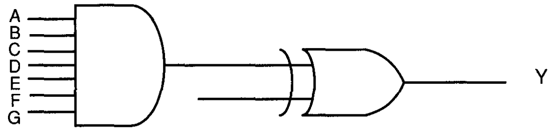
Input Buffer Latches (3200DX only)

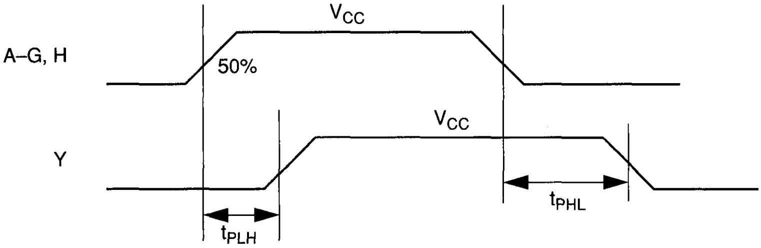


Output Buffer Latches (3200DX only)

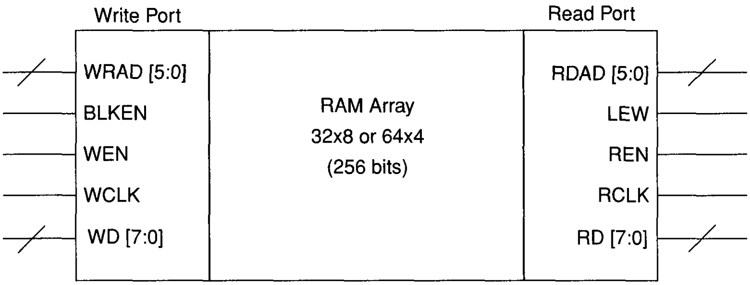


Decode Module Timing (3200DX only)



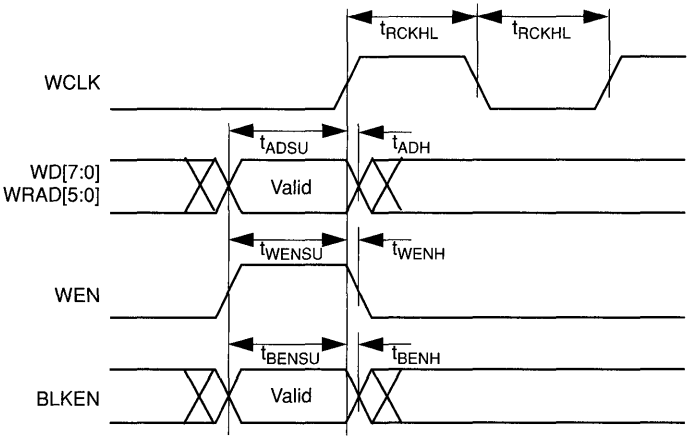


SRAM Timing Characteristic (3200DX only)



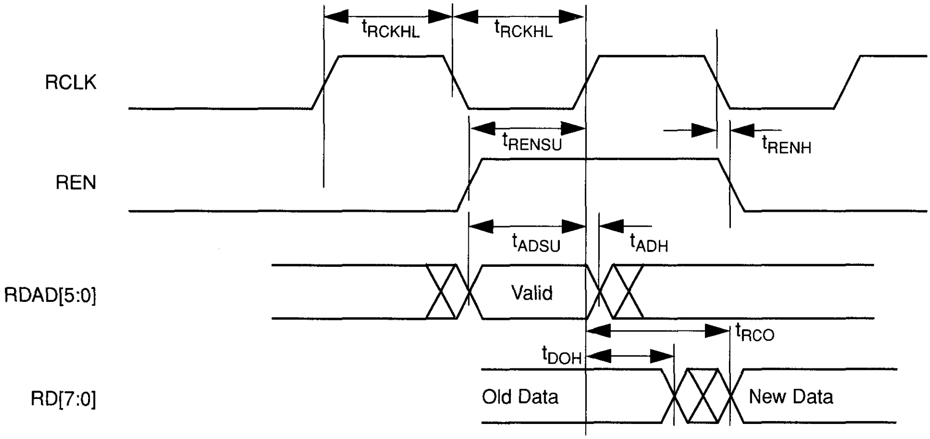
Dual-Port SRAM Timing Waveforms

SRAM Write Operation (3200DX only)



Note: *Identical timing for falling-edge clock.*

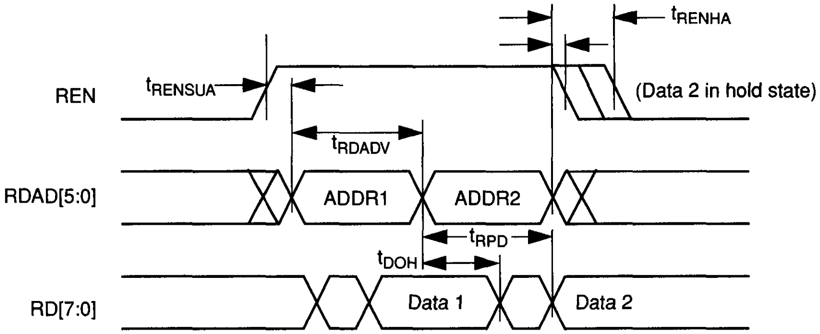
SRAM Synchronous Read Operation (3200DX only)



Note: *Identical timing for falling-edge clock*

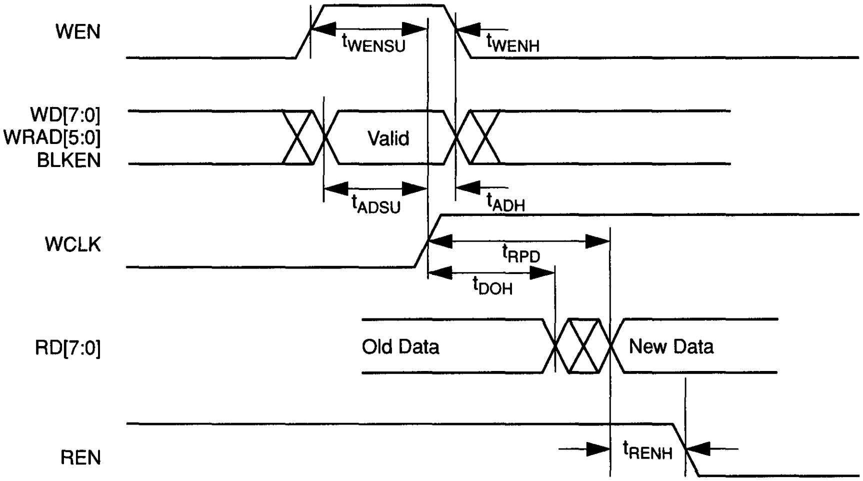
SRAM Asynchronous Read Operation—Type 1 (3200DX only)

((Read Address Controlled)



SRAM Asynchronous Read Operation—Type 2 (3200DX only)

(Write Address Controlled)



MPGA Timing Characteristics

(Worst-Case Commercial Conditions, V_{CC} = 4.75 V, T_J = 70°C)

		Preliminary Information		
Logic Module Propagation Delays				
Parameter	**Description**	**Min.**	**Max.**	**Units**
t_{PD}	Internal Array Module		1.5	ns
t_{CO}	Sequential Clock to Q		1.5	ns
t_{CLR}	Asynchronous Clear to Q		1.5	ns
Predicted Routing Delays[1]				
t_{RD1}	FO=1 Routing Delay		0.2	ns
t_{RD2}	FO=2 Routing Delay		0.3	ns
t_{RD3}	FO=3 Routing Delay		0.5	ns
t_{RD4}	FO=4 Routing Delay		0.7	ns
t_{RD8}	FO=8 Routing Delay		1.4	ns
Logic Module Sequential Timing				
t_{SUD}	Flip-Flop Data Input Setup	1.5		ns
t_{HD}	Flip-Flop Data Input Hold	0.0		ns
t_{SUD}	Latch Data Input Setup	1.5		ns
t_{HD}	Latch Data Input Hold	0.0		ns
t_{WASYN}	Asynchronous Pulse Width	2.0		ns
t_{WCLKA}	Flip-Flop Clock Pulse Width	2.0		ns
t_A	Flip-Flop Clock Input Period	8.0		ns
f_{MAX}	Flip-Flop Clock Frequency		125	MHz
I/O Module Input Propagation Delay				
t_{INY}	Input Data Pad to Y		2.0	ns
Predicted Input Routing Delays[1]				
t_{IRD1}	FO=1 Routing Delay		0.2	ns
t_{IRD2}	FO=2 Routing Delay		0.3	ns
t_{IRD3}	FO=3 Routing Delay		0.5	ns
t_{IRD4}	FO=4 Routing Delay		0.7	ns
t_{IRD8}	FO=8 Routing Delay		1.4	ns

Note:

1. *Routing delays are for typical designs across worst-case operating conditions. These parameters should be used for estimating device performance. Postroute timing analysis or simulation is required to determine actual worst-case performance. Postroute timing is based on actual routing delay measurements performed on the device prior to shipment.*

MPGA Timing Characteristics (continued)

(Worst-Case Commercial Conditions)

		Preliminary Information		
I/O Module – TTL Output Timing[1]				
Parameter	**Description**	**Min.**	**Max.**	**Units**
t_{DHL}	Data to Pad, High to Low		6.8	ns
t_{DLH}	Data to Pad, Low to High		3.9	ns
t_{ENZH}	Enable to Pad, Z to High		4.5	ns
t_{ENZL}	Enable to Pad, Z to Low		6.8	ns
t_{ENHZ}	Enable to Pad, High to Z		3.8	ns
t_{ENLZ}	Enable to Pad, Low to Z		2.0	ns
d_{TLH}	Delta Low to High		0.05	ns/pF
d_{THL}	Delta High to Low		0.09	ns/pF
I/O Module – CMOS Output Timing[1]				
t_{DHL}	Data to Pad, High to Low		5.5	ns
t_{DLH}	Data to Pad, Low to High		6.1	ns
t_{ENZH}	Enable to Pad, Z to High		6.7	ns
t_{ENZL}	Enable to Pad, Z to Low		5.6	ns
t_{ENHZ}	Enable to Pad, High to Z		3.8	ns
t_{ENLZ}	Enable to Pad, Low to Z		2.0	ns
d_{TLH}	Delta Low to High		0.09	ns/pF
d_{THL}	Delta High to Low		0.07	ns/pF
Global Clock Networks (for Fanout = 1000)				
t_{CKH}	Input Low to High		5.0	ns
t_{CKL}	Input High to Low		5.0	ns
t_{PWH}	Min. Pulse Width High	2.9		ns
t_{PWL}	Min. Pulse Width Low	2.9		ns
t_{CKSW}	Maximum Skew		0.4	ns
t_P	Minimum Period	6.0		ns
f_{MAX}	Maximum Frequency		167	MHz

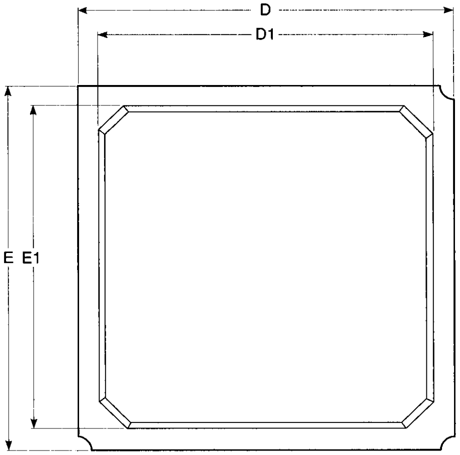
Note:

1. Delays based on 35pF loading.

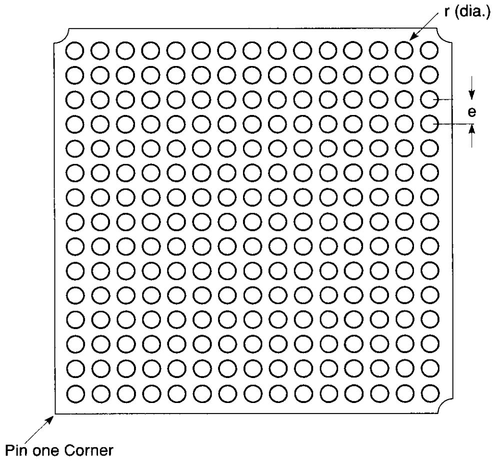
Package Mechanical Drawings

Ball Grid Array (BGA)

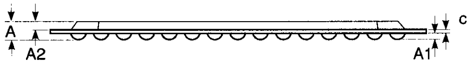
Top View



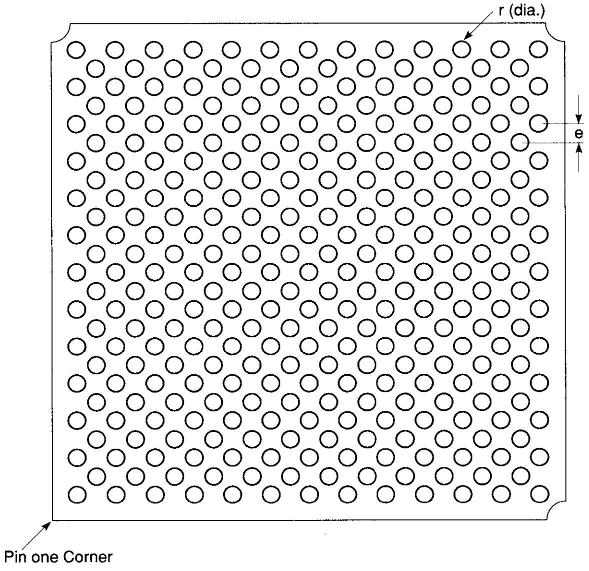
Bottom View
225 BGA



Side View



Bottom View
313 BGA



Ball Grid Array (BGA)

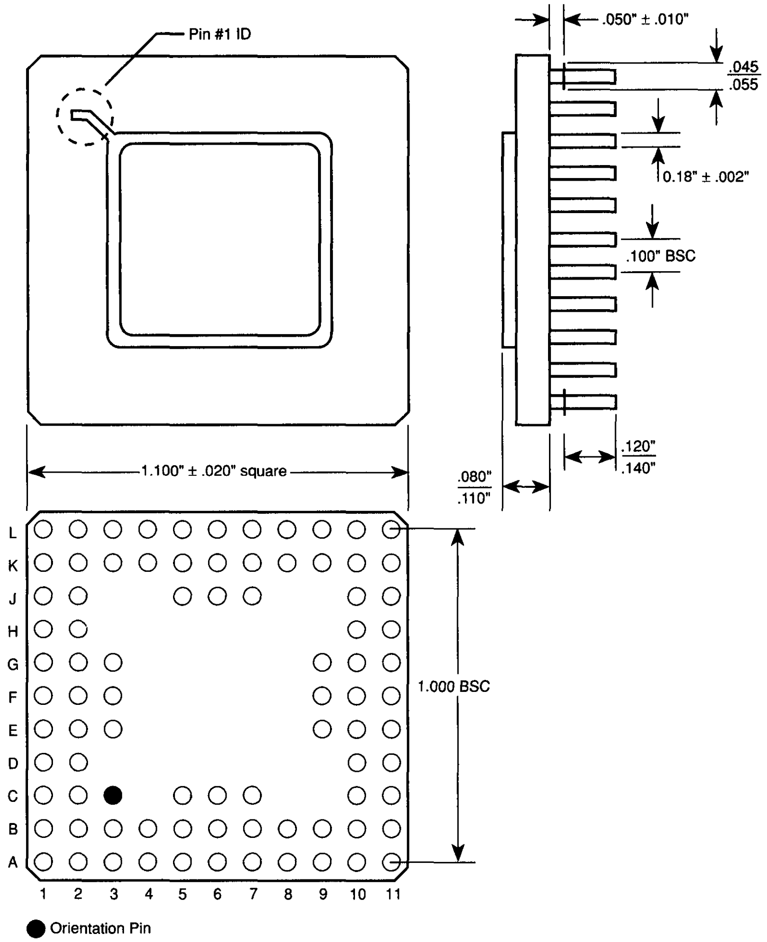
Package Name	BGA 225 MO-151		BGA 313 MO-151	
Dimensions	**Min**	**Max**	**Min**	**Max**
A2	1.12	1.22	1.12	1.22
A1	0.50	0.70	0.50	0.70
c	0.36 REF.		0.56 REF.	
A	1.92	2.32	2.12	2.52
D/E	26.80	27.20	34.80	35.20
D1/E1	24.00	24.70	30.00	30.70
e	1.50 BSC		1.27 BSC	
r (diameter)	0.60	0.90	0.60	0.90

Notes:

1. *All dimensions are in millimeters unless otherwise specified.*
2. *BSC—Basic Spacing between Centers.*

Ceramic Pin Grid Array

84-Pin CPGA

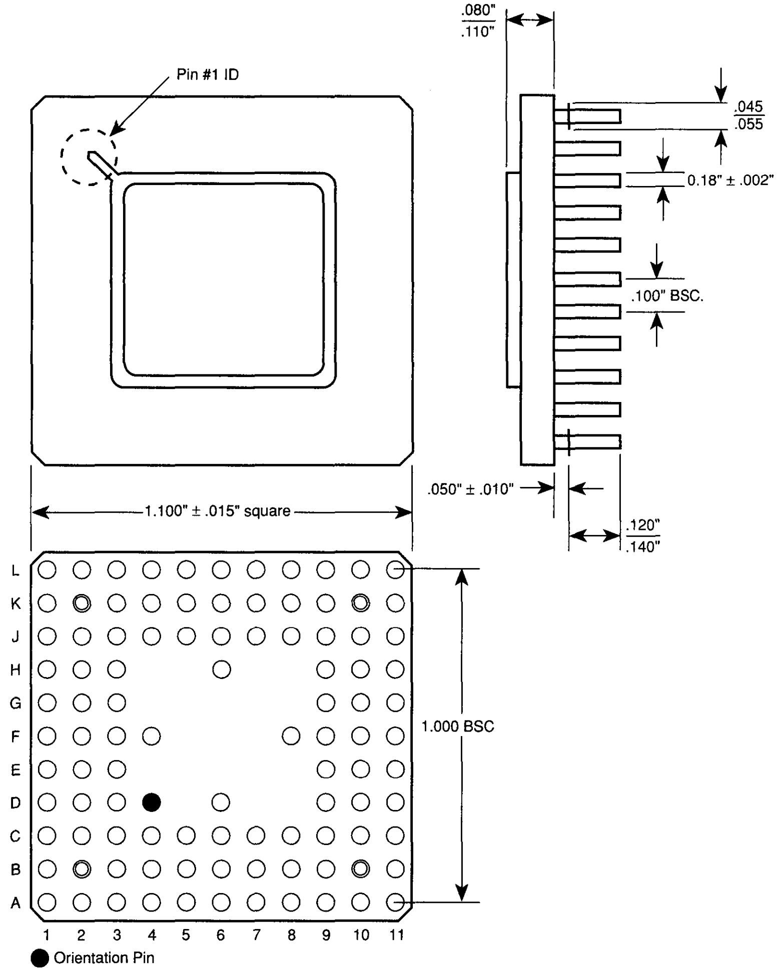


Notes:

1. *All dimensions are in inches unless otherwise stated.*
2. *BSC—Basic Spacing between Centers.*

Ceramic Pin Grid Array (continued)

100-Pin CPGA (continued)

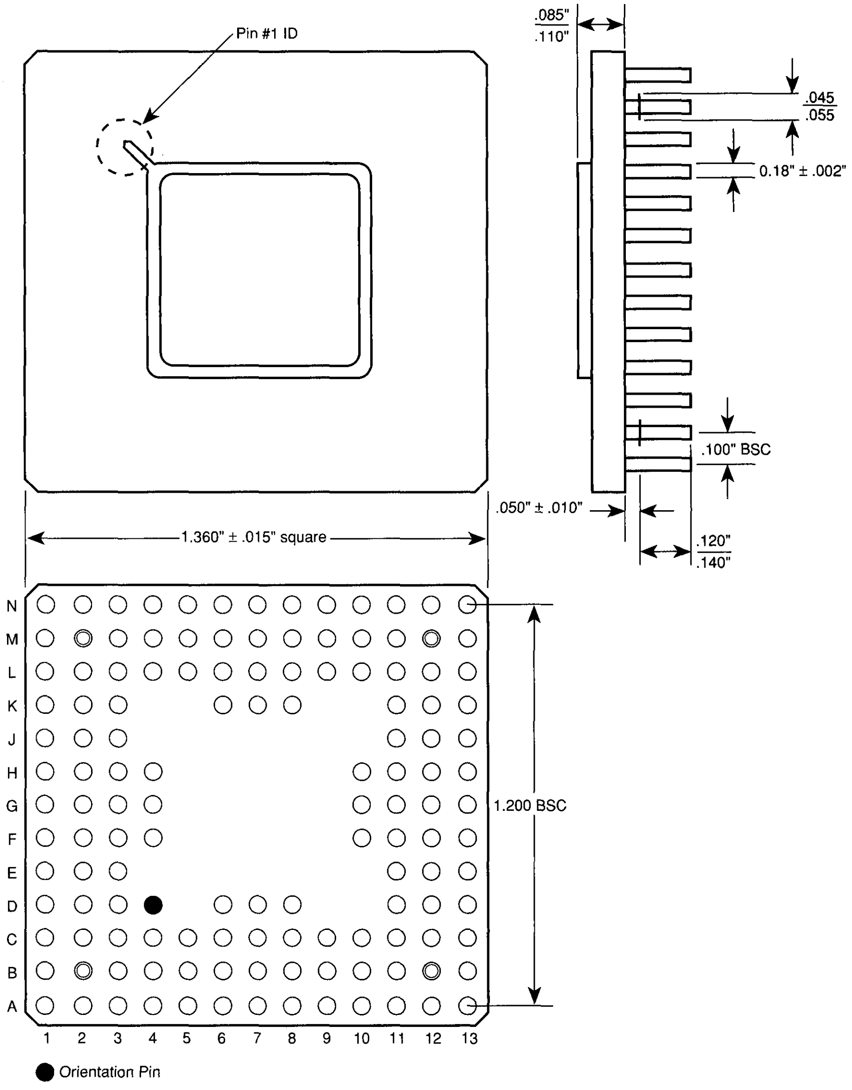


Notes:

1. *All dimensions are in inches unless otherwise stated.*
2. *BSC—Basic Spacing between Centers.*

Ceramic Pin Grid Array (continued)

132-Pin CPGA

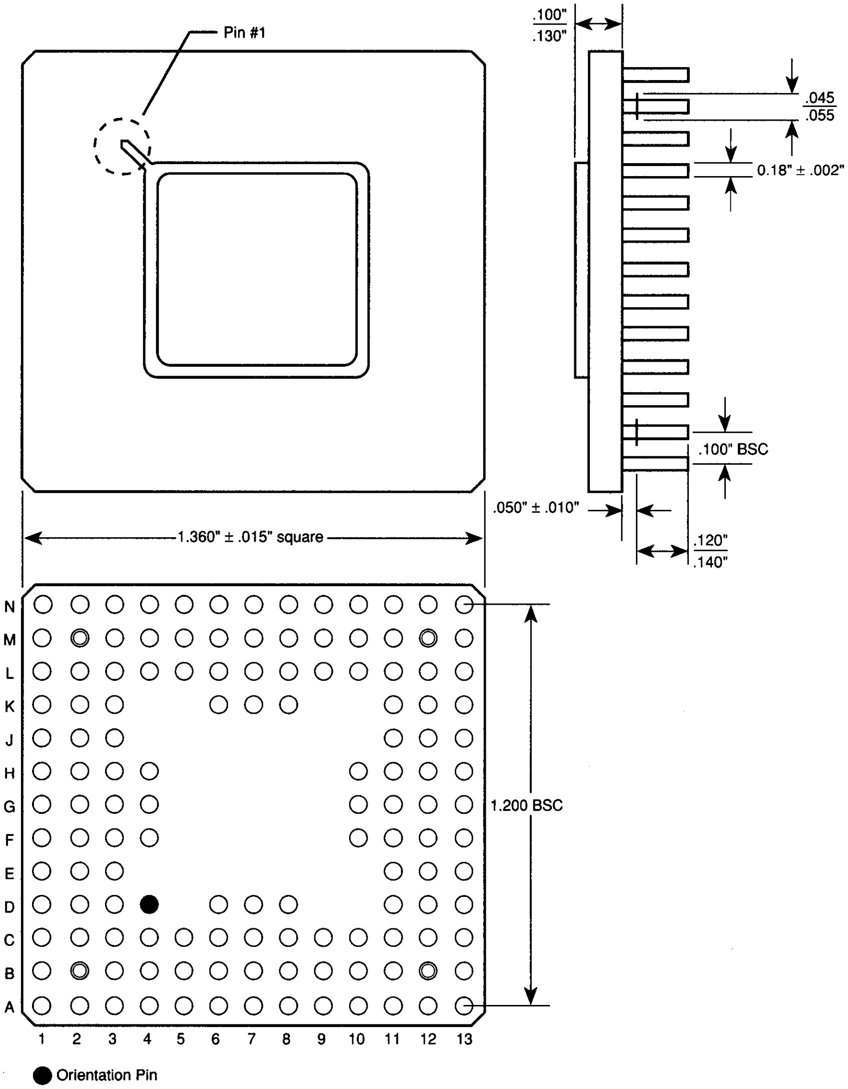


Notes:

1. *All dimensions are in inches unless otherwise stated.*
2. *BSC—Basic Spacing between Centers.*

Ceramic Pin Grid Array (continued)

133-Pin CPGA

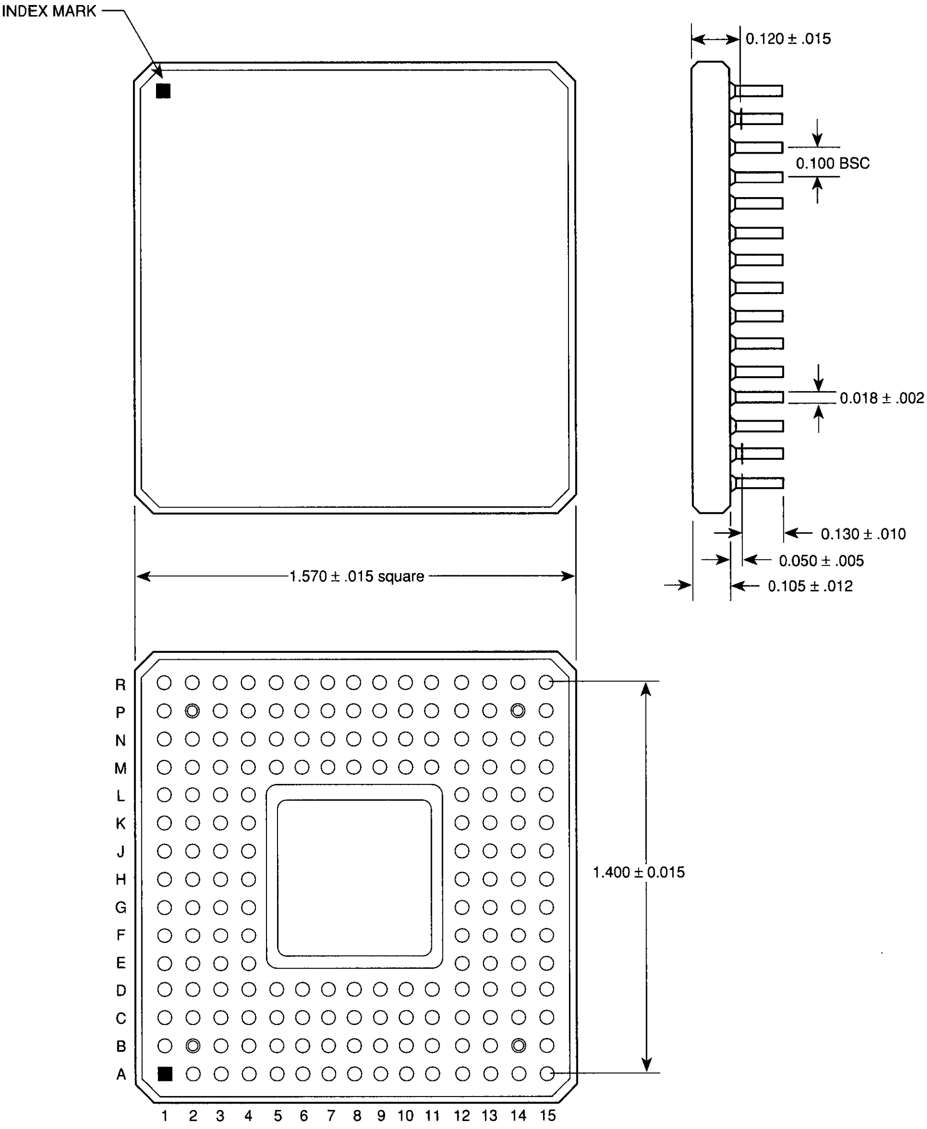


Notes:

1. *All dimensions are in inches unless otherwise stated.*
2. *BSC—Basic Spacing between Centers.*

Ceramic Pin Grid Array (continued)

175-Pin CPGA

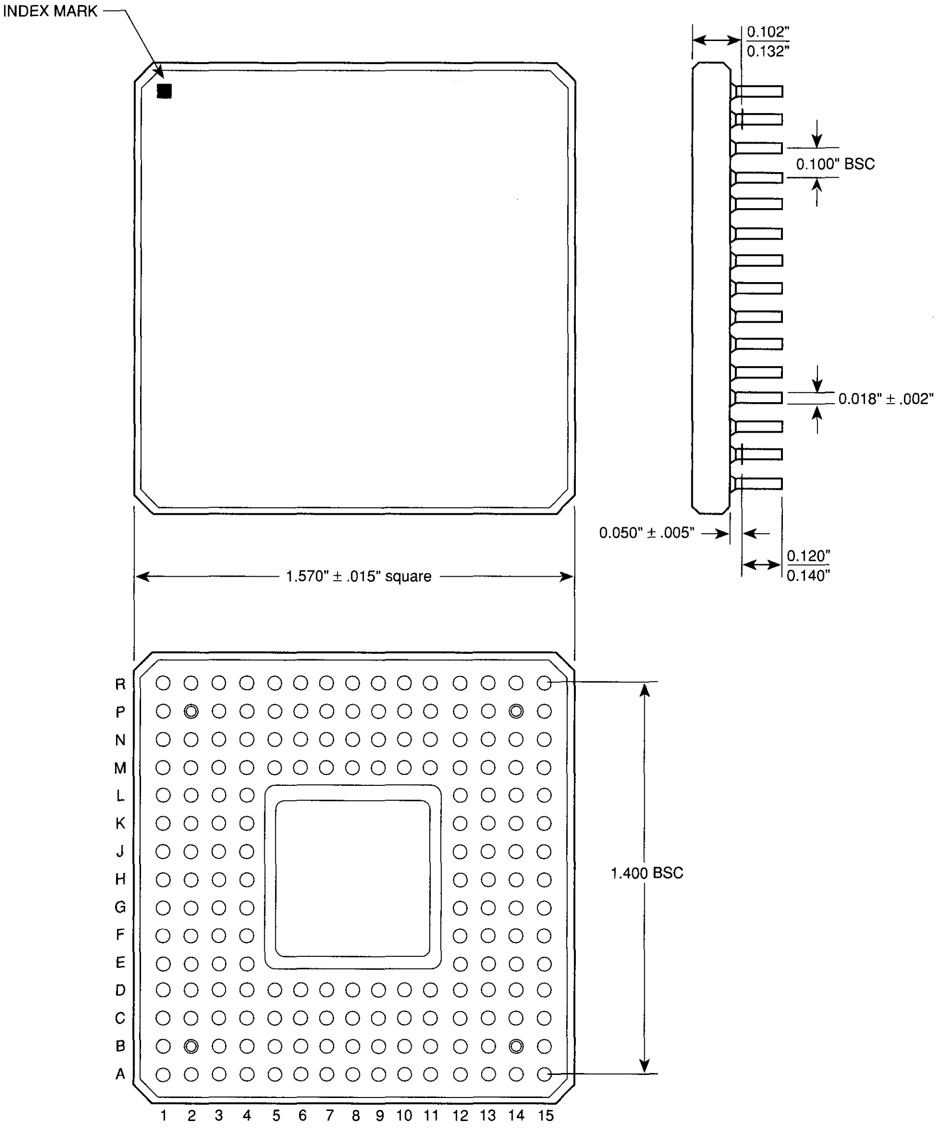


Notes:

1. *All dimensions are in inches unless otherwise stated.*
2. *BSC—Basic Spacing between Centers.*

Ceramic Pin Grid Array (continued)

176-Pin CPGA

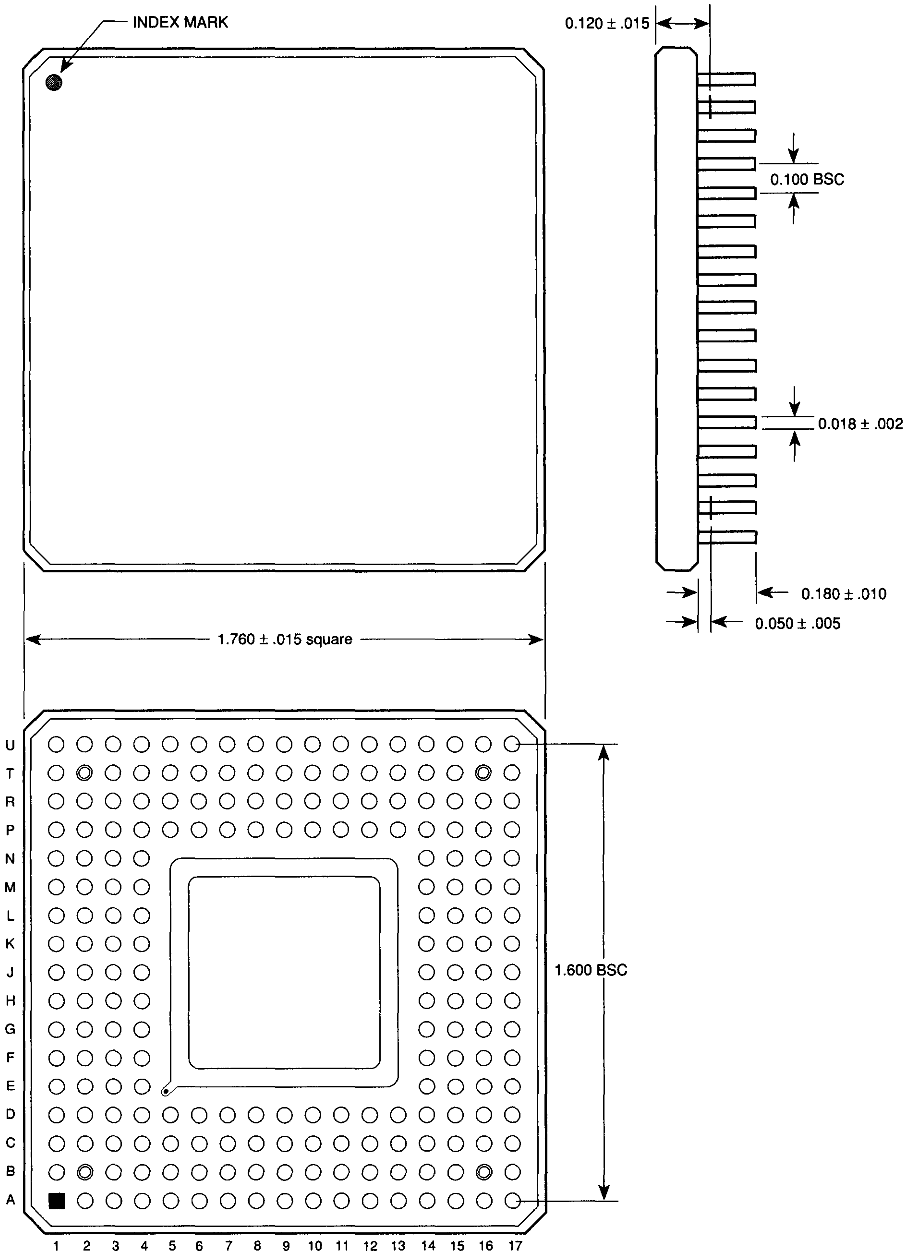


Notes:

1. *All dimensions are in inches unless otherwise stated.*
2. *BSC—Basic Spacing between Centers.*

Ceramic Pin Grid Array (continued)

207-Pin CPGA

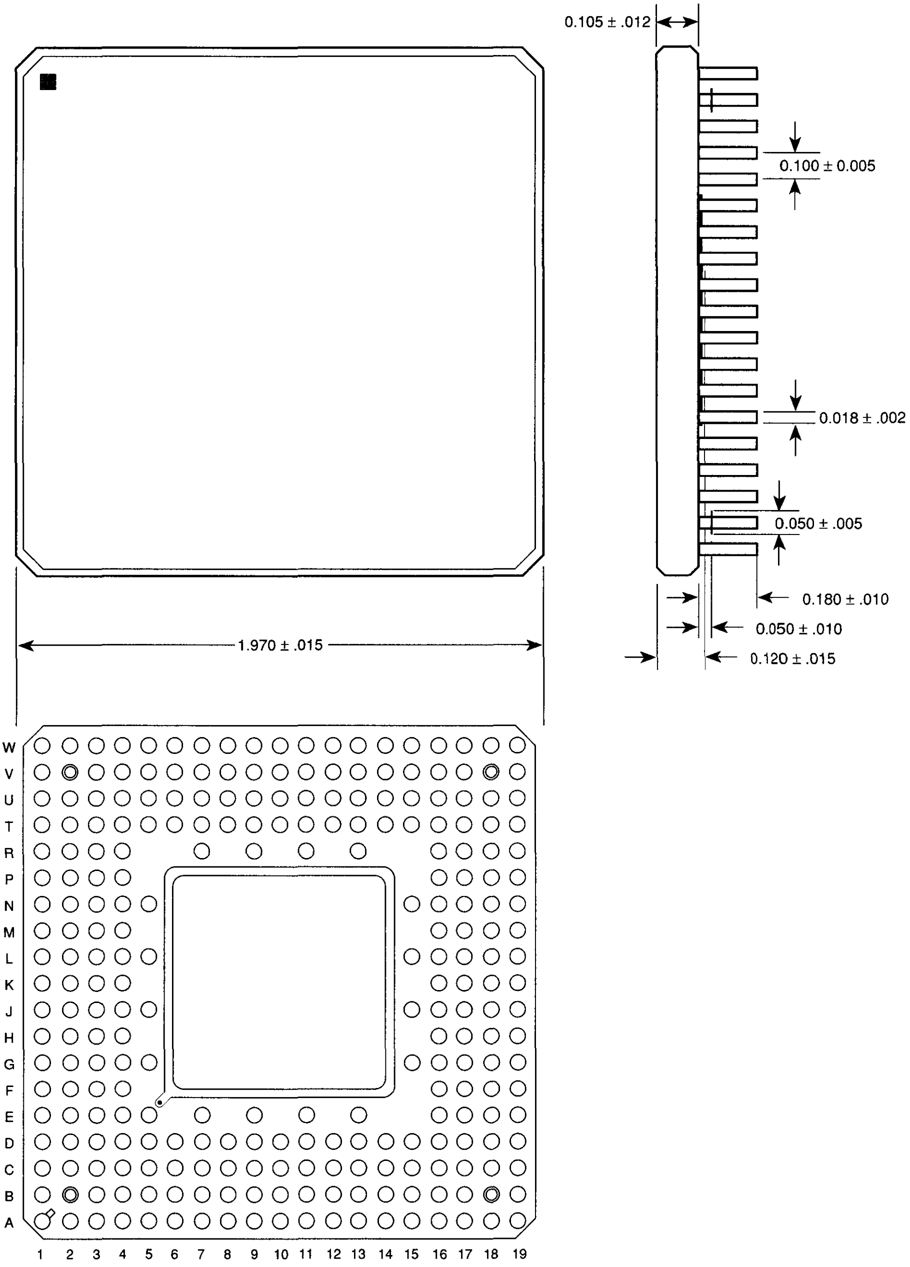


Notes:

1. *All dimensions are in inches unless otherwise stated.*
2. *BSC—Basic Spacing between Centers.*

Ceramic Pin Grid Array (continued)

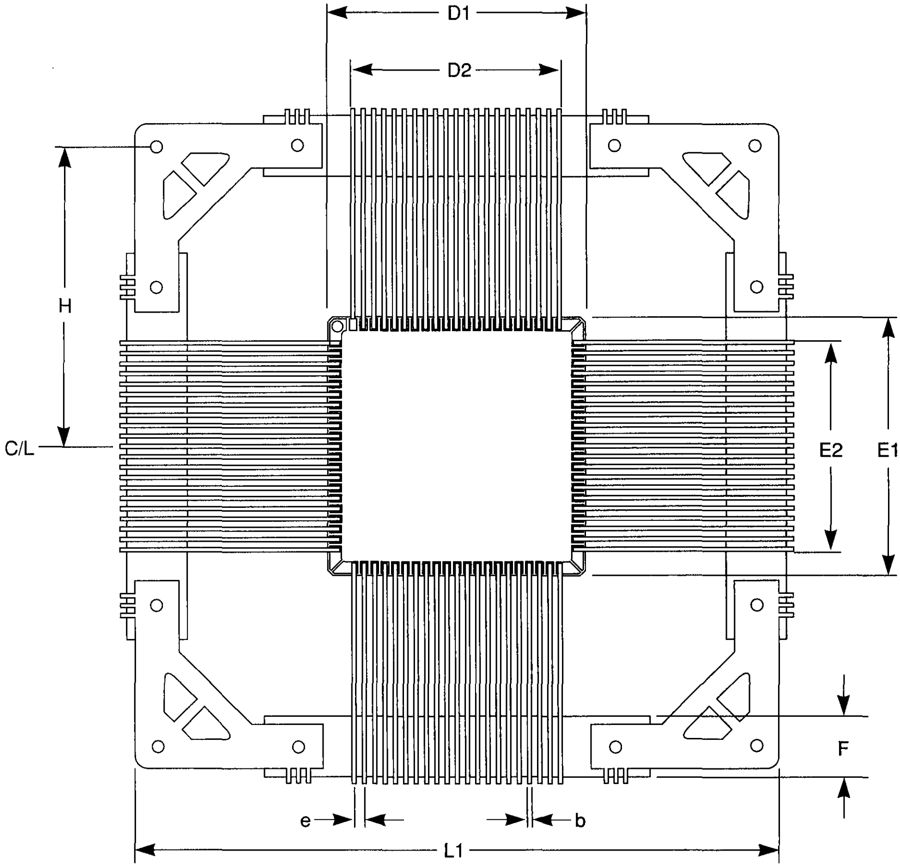
257-Pin CPGA

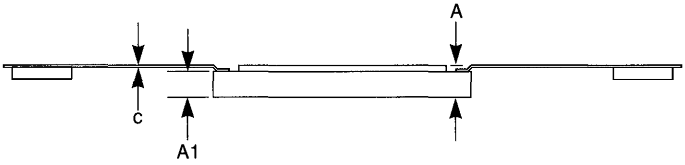


Notes:

1. *All dimensions are in inches unless otherwise stated.*
2. *BSC—Basic Spacing between Centers.*

Ceramic Quad Flatpack (CQFP)





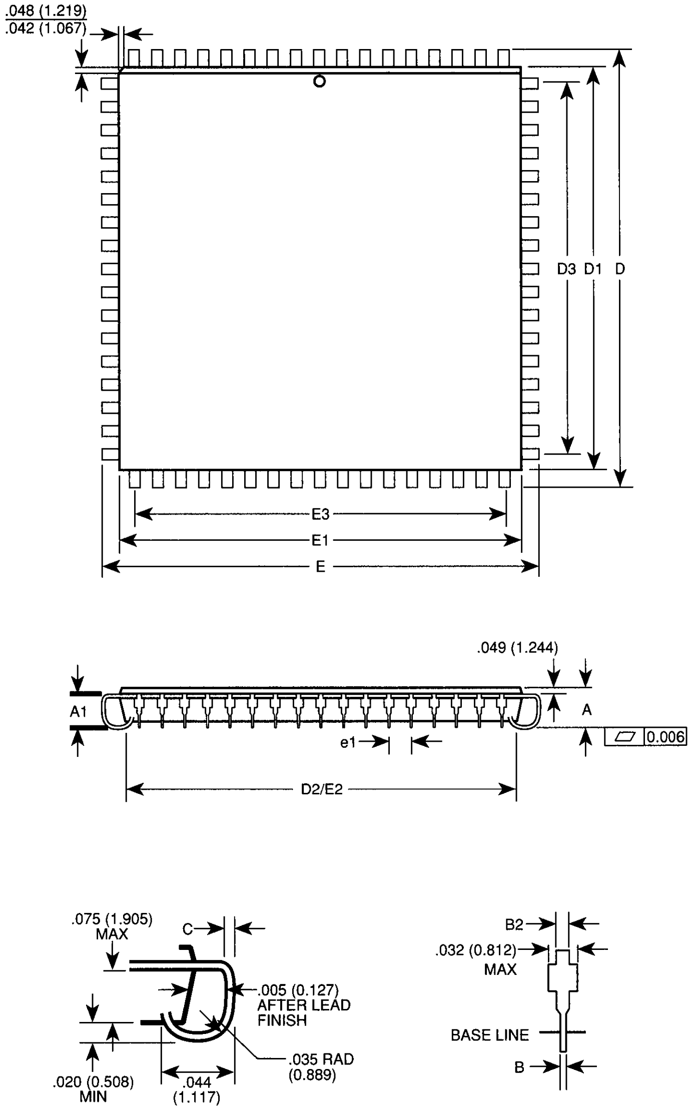
Cermamic Quad Flatpack (CQFP)

JEDEC	CQ84 MO-90		CQ132 MO-113		CQ172 MO-113		CQ196 MO-113		CQ256 MO-134	
Symbol	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
A		0.130	0.086	0.140	0.086	0.140	0.086	0.140	2.16	4.55
A1		0.105	0.078	0.125	0.078	0.125	0.078	0.125	2.00	4.00
b	0.006	0.012	0.007	0.010	0.007	0.010	0.007	0.010	0.19	0.25
c	0.004	0.008	0.004	0.008	0.004	0.008	0.004	0.008	0.11	0.20
D1/E1	0.635	0.660	0.935	0.965	1.165	1.195	1.325	1.365	25.64	36.36
D2/E2	.500 BSC		.800 BSC		1.050 BSC		1.200 BSC		31.5 BSC	
e	.025 BSC		.025 BSC		.025 BSC		.025 BSC		0.50 BSC	
F	0.130	0.150	0.325	0.375	0.175	0.225	0.175	0.200	6.85	8.65
H	.730 BSC		1.150 BSC		1.150 BSC		1.150 BSC		35.00 BSC	
L1	1.595	1.615	2.485	2.505	2.485	2.505	2.485	2.505	74.60	75.40

Note:

1. *All dimensions are in inches except CQ256, which is in millimeters.*
2. *BSC—Basic Spacing between Centers.*

Plastic Leaded Chip Carrier (PLCC)



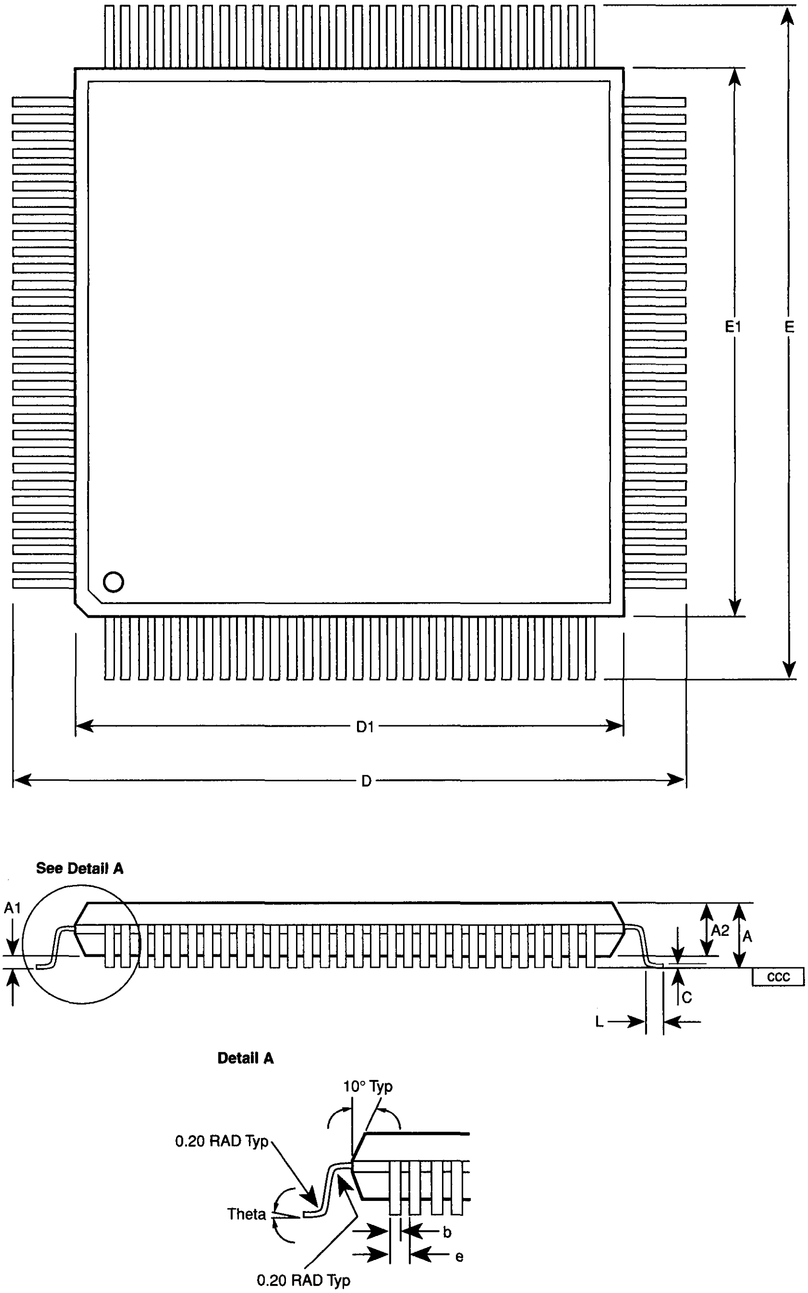
Plastic Leaded Chip Carrier Packages (PLCC)

Jedec Equiv	PLCC 44 MS007 AB VAR		PLCC 68 MS007 AD VAR		PLCC 84 MS007 AE VAR	
Dimension	Min	Max	Min	Max	Min	Max
A	0.155	0.175	0.155	0.175	0.155	0.175
A1	0.090	0.130	0.090	0.130	0.090	0.130
B	0.013	0.027	0.013	0.027	0.013	0.027
B2	0.026	0.032	0.026	0.032	0.026	0.032
C	0.007	0.013	0.005	0.011	0.005	0.011
D/E	0.670	0.710	0.970	1.010	1.170	1.210
D1/E1	0.640	0.660	0.940	0.960	1.140	1.160
D2/E2	0.590	0.630	0.890	0.930	1.090	1.130
D3/E3	0.50 nominal		0.80 nominal		1.00 nominal	
e1	0.050 BSC		0.050 BSC		0.050 BSC	

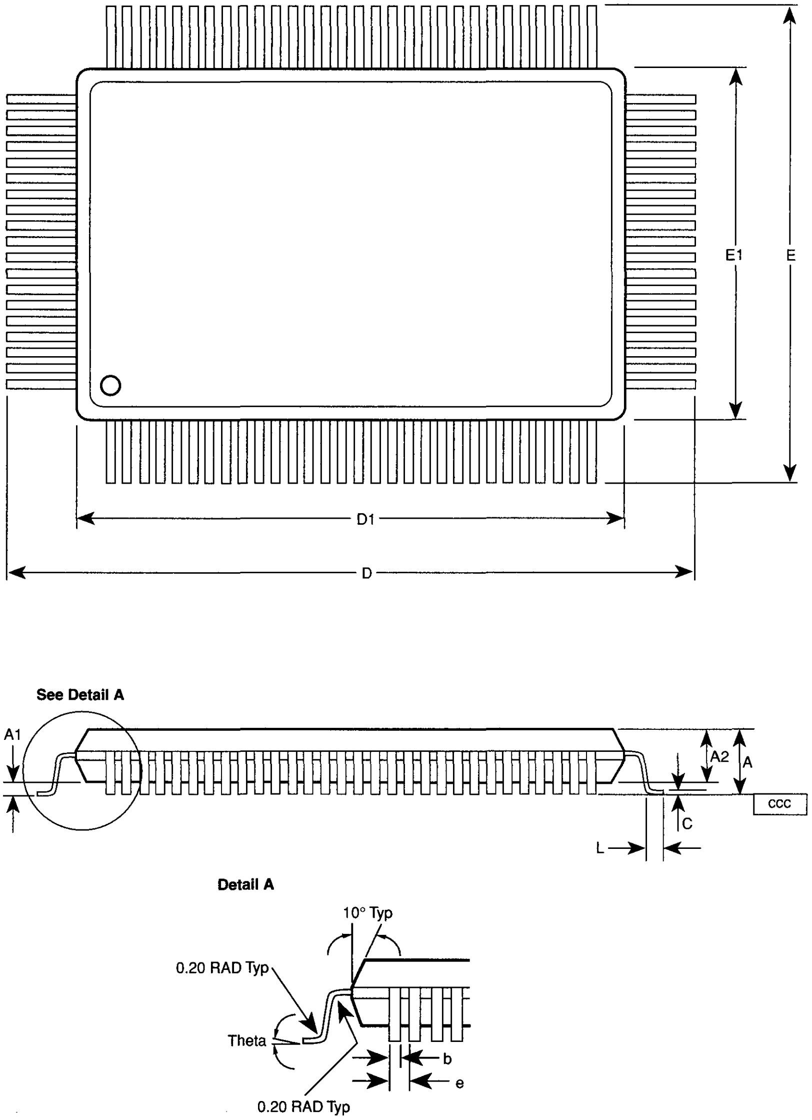
Notes:

1. *All dimensions are in inches.*
2. *BSC—Basic Spacing between Centers.*

Plastic Quad Flatpack (PQFP, RQFP, TQFP, VQFP)



Plastic Quad Flatpack Rectangular Package (PQ100)



Plastic Quad Flat Packages (PQFP)

Jedec Equiv	PQFP 100 MO-108		PQFP 144 MO-108		ACT 3 PQFP 160 MO-108		1200XL, 3200DX PQFP 160 MO-112		PQFP 208 MO-143	
Dimension	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**
A		3.40		4.07		4.07		4.07		4.10
A1	0.25		0.25		0.25		0.25		0.25	
A2	2.55	3.05	3.20	3.60	3.20	3.60	3.17	3.67	3.20	3.60
b	0.22	0.38	0.22	0.38	0.22	0.38	0.22	0.38	0.17	0.27
c	0.13	0.23	0.13	0.23	0.13	0.23	0.13	0.23	0.09	0.20
D	23.20 BSC		31.20 BSC		31.20 BSC		31.90 BSC		30.60 BSC	
D1	20.00 BSC		28.00 BSC		28.00 BSC		28.00 BSC		28.00 BSC	
E	17.20 BSC		31.20 BSC		31.20 BSC		31.90 BSC		30.60 BSC	
E1	14.00 BSC		28.00 BSC		28.00 BSC		28.00 BSC		28.00 BSC	
e	0.65 BSC		0.65 BSC		0.65 BSC		0.65 BSC		0.50 BSC	
L	0.73	1.03	0.73	1.03	0.73	1.03	0.65	0.95	0.50	0.75
ccc		0.10		0.08		0.08		0.10		0.08
Theta	0	7 deg	0	7 deg	0	7 deg	0	7 deg	0	7 deg

Notes:

1. *All dimensions are in millimeters.*
2. *BSC—Basic Spacing between Centers.*

Plastic Quad Flat Packages (RQFP)

Jedec Equiv	RQFP 208 MO-143		RQFP 240 MO-143	
Dimension	**Min**	**Max**	**Min**	**Max**
A		4.10		4.10
A1	0.25		0.25	
A2	3.20	3.60	3.20	3.60
b	0.17	0.27	0.17	0.27
c	0.09	0.20	0.09	0.20
D/E	30.60 BSC		34.60 BSC	
D1/E1	28.00 BSC		32.00 BSC	
e	0.50 BSC		0.50 BSC	
L	0.50	0.75	0.50	0.75
ccc		0.08		0.08
Theta	0	7 deg	0	7 deg

Notes:

1. *All dimensions are in millimeters.*
2. *BSC—Basic Spacing between Centers.*

Thin Quad Flatpacks (TQFP and VQFP)

Jedec Equiv	TQFP 176 MO-136		VQFP 80 MO-136		VQFP 100 MO-136	
Dimension	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**
A		1.60		1.20		1.20
A1	0.05	0.15	0.05	0.15	0.05	0.15
A2	1.35	1.45	0.95	1.05	0.95	1.05
b	0.17	0.27	0.22	0.38	0.17	0.27
c	0.09	0.20	0.09	0.20	0.09	0.20
D/E	26.00 BSC		16.00 BSC		16.00 BSC	
D1/E1	24.00 BSC		14.00 BSC		14.00 BSC	
e	0.50 BSC		0.65 BSC		0.50 BSC	
L	0.45	0.75	0.45	0.75	0.45	0.75
ccc		0.08		0.10		0.08
Theta	0	7 deg	0	7 deg	0	7 deg

Notes:

1. *All dimensions are in millimeters.*
2. *BSC—Basic Spacing between Centers.*

Socket Recommendation for Actel FPGA Packages

Prototyping Sockets Resold by Actel

These sockets have been engineered and manufactured by the respective original manufacturer. Actel resells small quantities of these sockets to Actel customers in order to provide a more complete prototyping solution. Actel also offers quick-turn design software and Actionprobe diagnostic tools to speed designs to market.

Package	Lead Count	Actel Part Number	Original Manufacturer	Original Manufacturer Part Number
PQFP	100	SY-PQ100	Yamaichi	IC149-100-014-S5
	144	SA-PQ144	AMP	824222-1
		SY-PQ144	Yamaichi	IC149-144K11453-0
	160	SA-PQ160	AMP	824220-1
		SY-PQ160	Yamaichi	IC140-160-023-S5
	208	SA-PQ208	AMP	824160-2
		SY-PQ208	Yamaichi	IC198-208-2101
RQFP	208	SA-PQ208	AMP	824160-2
		SY-PQ208	Yamaichi	IC198-208-2101
VQFP	80	SA-VQ80	AMP	824073-1
		SY-VQ80	Yamaichi	IC198-080-2001
CQFP	132	SY-CQ196	Yamaichi	Custom
	172	SY-CQ196	Yamaichi	Custom
	196	SY-CQ196	Yamaichi	Custom

Socket Sources for Actel FPGA Packages

Actel has compiled this list of know suppliers for the convenience of our customers. This is simply a list of suppliers that we are aware of rather than a list of recommended sockets, as we have not tested them for reliability. For information on these sockets, contact the manufacturers directly.

Surface Mount Sockets

		Prototype/Production	
Package	**Lead Count**	**Source**	**Part Number**
PLCC	44	AMP	821979-3 or 822035-3
		Methode	213-044-602
	68	AMP	822029-3 or 822073-3
		Methode	213-068-602
	84	AMP	821151 or 822180-1
		Methode	213-084-602
PQFP	100	Yamaichi	IC149-100-014-S5
	144	AMP	824222-1
		Yamaichi	IC149-144K11453-0
	160	AMP	824220-1
		Yamaichi	IC149-160-023-S5
	208	AMP	824160-2
		Yamaichi	IC198-208-2101
RQFP	208	AMP	824160-2
		Yamaichi	IC198-208-2101
VQFP	80	AMP	824073-1
		Yamaichi	IC198-080-2001
	100	Yamaichi	

CONTACTS:	AMP	(408) 725-4975
	Enplas	(415) 572-1683
	Methode	(408) 262-3812
	Mil-Max	(516) 922-6000
	McKenze	(510) 651-2700
	Nepenthe	(415) 496-6666
	Wells	(408) 559-8118
	Yamaichi	(408) 456-0797

Through Hole Sockets

		Prototype/Production		Zero Insertion	
Package	**Lead Count**	**Source**	**Part Number**	**Source**	**Part Number**
PLCC	44	Amp	821575-3	Nepenthe	PC1-044050-016
		Methode	213-044-401	Yamaichi	IC51-0444-400
	68	Amp	821574-3	Nepenthe	PC1-068050-001
		Methode	213-068-401	Yamaichi	IC51-0684-390-1
	84	Amp	821573-3	Nepenthe	PC1-084050-003
		Methode	213-084-401	Yamaichi	IC51-0844-401-1
PQFP	100			Yamaichi	IC51-1004-814-2
	144	Amp	822114-3 and 822115-3	Yamaichi	IC51-1014-KS10418
	160	Amp	822114-4 and 822115-4	Yamaichi	IC51-1604-845-1
	208			Yamaichi	IC51-1052-KS12897
RQFP	208			Yamaichi	IC51-1052-KS12897
TQFP	176			Yamaichi	IC51-1764-1505
VQFP	80			Yamaichi	IC51-0804-795
BGA	225			Enplas	BGA-225-1.5-01
	313			Enplas	BGA-313-841-1.27-01
CQFP	84 w/o tie bar			Wells	619-1000311-001
	84 w/ tie bar			Wells	619-1001611-001
	132			Actel	Custom
	172			Actel	Custom
	196			Actel	Custom
	256			TBD	TBD
PGA	85	Mil-Max	510-91-085-11-041	Yamaichi	NP35-11207-KS8108
		McKenze	PGA-85H-012B-1-1107		
	100/101	McKenze	PGA-101M-012B-1-11B5	Yamaichi	NP89-12110-KS11922
	132/133	McKenze	PGA-133H-003B-1-13GOR		
	175/176	Mil-Max	510-91-176-15-061	Yamaichi	NP89-22508-KS11957
		McKenze	PGA-177M-003B-1-1552		
	207			Amp	916227-6
	257			Amp	916229-6

Software Development Systems

Section 2: Software Development Systems

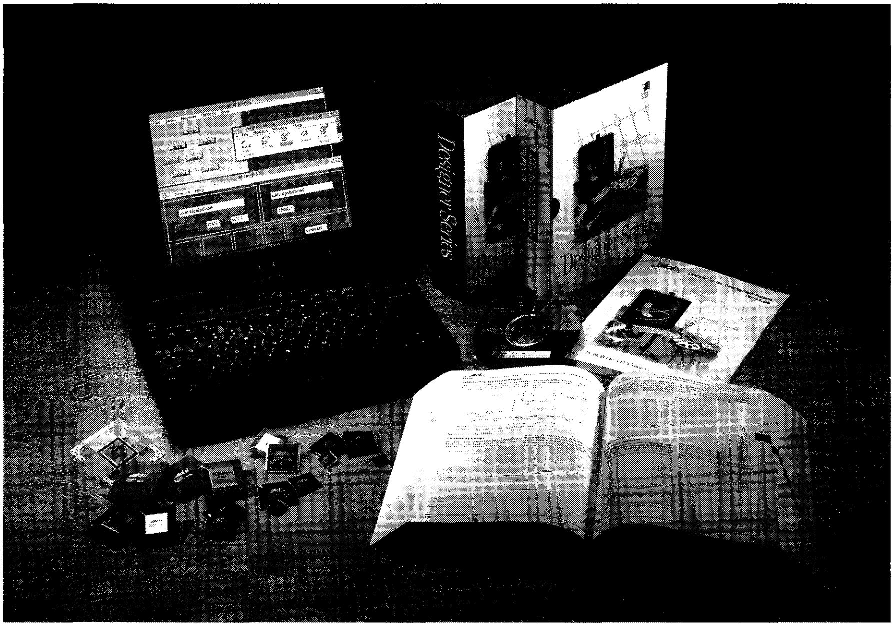
Introduction to Software

Field programmable gate arrays (FPGAs) offer many benefits when you need custom logic. FPGAs' on-site programmability allows you to move from concept to a working part in a very short time. There are no expensive nonrecurring engineering charges, and FPGAs allow you to make design changes at any time, even after you've released your product to market. FPGAs are only half the picture, though; the development system you use to design them is the other half.

An effective development system must satisfy several basic requirements:

- It must be easy to learn and use. You don't have the time to learn a complex new design tool.
- It must fit seamlessly into your existing design flow. You need an FPGA development system that works within your existing design methodology—no matter if you use synthesis, schematic capture, or some combination of the two.
- It must help you use the FPGAs' full capabilities. You shouldn't have to waste precious time becoming an FPGA expert. The development system should provide the expertise.
- It must provide support for synthesis methodologies that use industry-standard, high-level design languages.

The Designer Series Development System meets all of these requirements. Its graphical Design Flow Manager clearly depicts and guides you through the Actel FPGA design process. You don't have to be an expert to begin designing Actel FPGAs with Designer Series. Design Flow Manager queries you to supply any missing information needed to proceed from one step to the next. It also blends seamlessly into your design environment, and it works closely with EDA software from companies such as Aldec, Cadence, Data I/O, Intergraph, Mentor Graphics, MINC, Synopsys, Synplicity, and Viewlogic. Design Flow Manager supports both schematic and high-level design methodologies, or a combination of the two.



Actel's software development focus is on supporting your design methodology, even if it includes a mixture of tools from multiple EDA vendors. Actel development systems fully support your choice of tools.

The industry trend is toward high-level design using Verilog HDL or VHDL. The Actel FPGA architecture, with its fine-grain logic modules and its abundant routing resources is ideal for implementing high-level designs. Actel provides high-quality, high-level design language support for synthesis products from Cadence, Exemplar, IST, Mentor Graphics, Synopsys, and Synplicity.

You don't need an in-depth understanding of Actel's FPGA architecture to achieve excellent results. The architectural expertise required to use Actel FPGAs effectively and efficiently is built into the Designer Series Development System. Tools are included to squeeze maximum performance from your VHDL descriptions or automatically generate optimized data-path elements. These tools transparently take advantage of any special features offered in an Actel FPGA family.

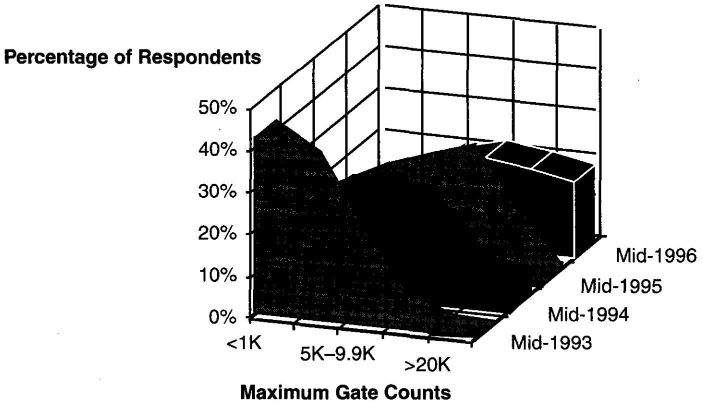
The Move Toward Synthesis

The tools you choose must be able to handle the complexity of your design. Each year, you squeeze more logic into a smaller space. The graph in Figure 1 shows that design complexity is increasing rapidly. In 1996, 80 percent of FPGA design starts are greater than 5,000 gates. As gate counts increase, it becomes more difficult to create and debug designs.

One way to deal with increasing design complexity is to adopt a high-level design methodology using Verilog HDL or VHDL. Describing your design at a higher level of abstraction allows you to more easily manage the design and debug of complex devices, to describe your design independently of target technology, and to provide an easy way to reuse all or part of your design in the future. Many synthesis tools are available to transform your high-level description into a physical implementation. The trend toward synthesis-based designs is charted in Figure 2. In 1996, fully 80 percent of all designers are using synthesis to capture some portion of their designs.

FPGA design can present a significant challenge for synthesis tools. The quality of synthesis results depends directly on FPGA architecture. An architecture that combines simple logic building blocks (logic modules) with abundant routing resources allows synthesis tools to efficiently implement an FPGA design. An architecture with complex logic modules or with insufficient routing resources prevents synthesis from generating efficient results. Inefficient synthesis reduces system performance and may require a larger FPGA.

Actel's FPGAs have both simple logic modules and plentiful routing resources, making them highly synthesis friendly. Actel's fine-grain architecture provides synthesis tools with a simple building block. The simplicity of the building block results in very little functionality being wasted when the synthesis tool generates a gate-level result. Actel FPGAs also provide plentiful routing resources, guaranteeing predictable place-and-route results.



Source: Data from Collett International, August 93 and August 94

Figure 1 • *Increase in Complexity of FPGA Design Starts*

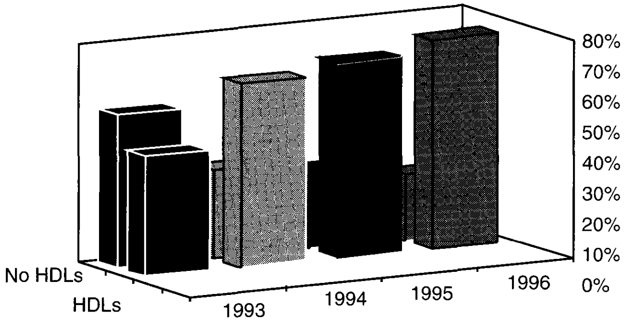


Figure 2 • *Adoption of High-Level Design Methodology*

Actel-Supported Design Methodologies

The Designer Series Development System is flexible enough to accommodate a wide range of design styles. It doesn't matter whether you're currently designing systems with PALs and TTL logic or with ASICs. Designer Series includes tools that make it easy to incorporate FPGAs into your system.

PLD Design Methods

If you design products from PLDs and TTL parts, Designer Series provides the tools to import your PLD descriptions and to generate macros that are functionally equivalent to your TTL logic. These tools allow you to capitalize on existing design practices while moving up to higher-capacity devices that integrate more logic. Designer Series makes it easy to combine state machines (in PALs) with data-path logic (in TTL components) and to implement them in a single FPGA. You have the flexibility to define part of your design by using Boolean equations and the rest with schematics. Designer Series combines both descriptions and implements your design in an FPGA, and it allows you to easily improve product performance and decrease costs by implementing your designs in an FPGA.

ASIC Design Methods

If you design products around ASICs, Designer Series is an excellent fit. Designer Series supports industry standards, such as EDIF, LPM, SDF, Verilog, VHDL, and VITAL. Describe your Actel FPGA by using any of a wide selection of EDA tools, and be assured Designer Series will turn it into an FPGA. (See "EDA Vendor Support" on page 2-17 for more information.)

You can enter your whole design by using schematic capture, or you can describe it completely using Verilog HDL or VHDL. You can even describe your design as a combination of schematics and a high-level design language. No matter how you capture it, there are connections available to link your design environment with Designer Series.

Support for your environment doesn't stop at the creation of an FPGA. In addition to verifying your design's performance by using the built-in timing analyzer, you can backannotate the timing values from the FPGA implementation and analyze them in a wide variety of simulators. You can even use the optional Actionprobe diagnostics tool in conjunction with an Activator series programmer to observe signals inside your programmed FPGA—displaying them at system speed—on your oscilloscope or logic analyzer. Actel development systems provide you with a wealth of tools that allow you to quickly create a working FPGA.

CPLD and FPGA Design Methods

Even if you currently design products around FPGAs or CPLDs, you'll find that Designer Series presents significant advantages in predictability, place-and-route efficiency, and design iteration flexibility. Use the EDA tools of your choice, as described in the previous section, and with Designer Series and Actel FPGAs, manually routing those last few nets becomes a chore of the past. Designer Series' place-and-route algorithms will automatically complete a 95 percent utilized FPGA—with fixed pins. The place-and-route algorithms have an incremental place-and-route mode that allows you to make small changes (like a typical bug fix) and quickly see the results of your change. Only the portions of your design that have changed are affected, so your design's timing changes little during debug.

The documents that follow in the rest of Section 2 cover the Designer Series Development System capabilities and EDA integration in greater detail. Read on for more information.

Designer Series Development System

The Designer Series Development System is a significant, feature-rich upgrade of Actel's FPGA development software. It has been enhanced for extremely intuitive operation and greater control for experienced designers. This document will describe Designer Series' new features in detail.

Features

- The Graphical Design Flow Manager visually describes the FPGA place-and-route process.
- The Design State Manager requests needed information and synchronizes the place-and-route database with the original design.
- The place-and-route algorithms automatically complete designs that use up to 95 percent of available modules—even with fixed pins.
- The DirectTime Layout option automates the achievement of design performance requirements.

Designer Series Overview

Figure 1 shows how Designer Series components interrelate and link to EDA tools. For more details on the EDA tools supported by Designer Series see "EDA Vendor Support" on page 2-17.

Designer Series can import a range of netlist formats directly. Supported formats include EDIF, Verilog, and Actel's ADL. These formats allow Designer Series to connect with most EDA tools.

The ACTgen Macro Builder tool, the ACTmap VHDL Synthesis tool, and compile components provide interfaces between Designer Series and EDA tools. The compile function simply

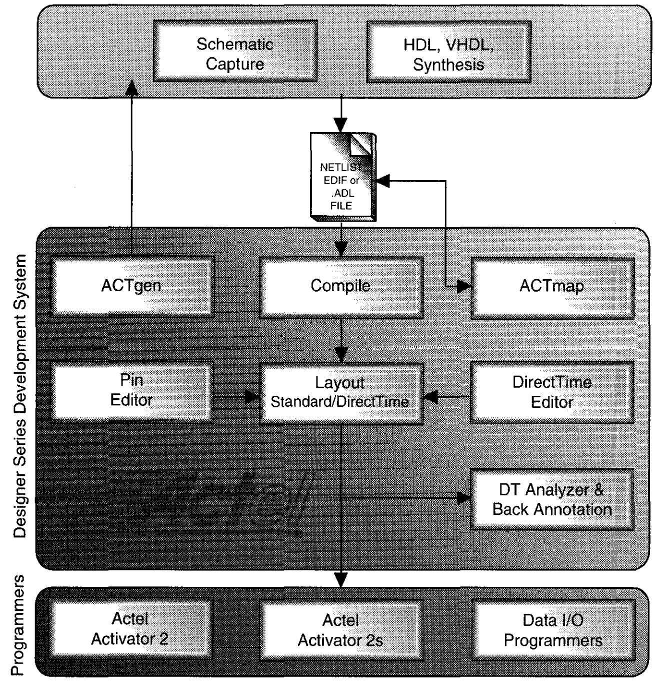


Figure 1 • *Components of Designer Series*

converts source logic into an equivalent compatible with the Actel FPGA architecture. Compile also checks the design integrity against Actel's design rules and automatically combines multiple source logic elements together to use FPGA resources efficiently. After compile finishes, the design is ready to be efficiently implemented in an Actel FPGA.

ACTgen Macro Builder and ACTmap VHDL Synthesis provide capabilities that simplify the task of obtaining maximum performance and optimum resource usage from an Actel FPGA. ACTgen Macro Builder is a rule-based generator of macro logic elements that supports trade-offs between efficient use of FPGA resources and high-performance results. ACTgen creates macro descriptions that can be easily included in a schematic or synthesis design description. ACTgen Macro Builder effortlessly and automatically generates customized macro functions that are optimized for a specific Actel architecture.

ACTmap VHDL Synthesis is a complete VHDL synthesis tool that can convert and incorporate PAL designs into an Actel FPGA, retarget designs from one Actel family to another, and optimize gate-level descriptions. ACTmap VHDL Synthesis is a streamlined synthesis tool that supports the synthesizable portion of the full IEEE-1076 VHDL standard. With no confusing special-purpose switches to specify, its simple controls provide excellent results.

When all design blocks have been imported into Designer Series by using these tools, the next step is to place and route the project for an FPGA. The standard place-and-route algorithm is included in all Designer Series Development System products. This time-proven algorithm automatically places and routes densely packed FPGAs—even when a design uses more than 95 percent of the logic module resources. Placement and routing is automatic even when all pin assignments are fixed. Signal-to-pin assignment can be accomplished using the PinEdit utility. PinEdit displays an outline of an FPGA package and allows an unassigned signal to be dragged and dropped onto a free pin. Assigning pins and placing and routing a design are straightforward.

DirectTime Layout is an optional feature of the standard place-and-route algorithm. DirectTime Layout provides direct control over the performance of an Actel FPGA and allows timing requirements to be specified prior to place and route. The DirectTime Layout algorithm takes these timing specifications into account while it makes trade-offs in logic module placement and signal routing. DirectTime Layout typically provides design performance equivalent to a faster speed grade.

Once place and route is complete, it is time to analyze the results. The built-in timing analyzer or an external timing simulator can be used to determine whether the timing requirements for a design have been met.

After verifying design timing, the next step is to program sample devices. The Activator 2 four-at-a-time programmer or the Activator 2S single-unit programmer can be used. Data I/O Unisite programmers can also be used.

The programmed devices are then assembled onto a prototype board for physical verification to assure the design functions as part of a complete system. Actionprobe diagnostics are an option that simplifies the debugging process. Actionprobe plugs into an Activator programmer and displays an FPGA's internal signals on an oscilloscope or logic analyzer. Actionprobe is a powerful tool for finding errors inside a programmed FPGA.

The Designer Series Development System provides a comprehensive suite of FPGA design tools. From interfacing to EDA tools to debugging a programmed FPGA, Designer Series provides easy-to-use FPGA design tools. Only Actel's Designer Series pulls together all the needed tools to provide an easy path to silicon.

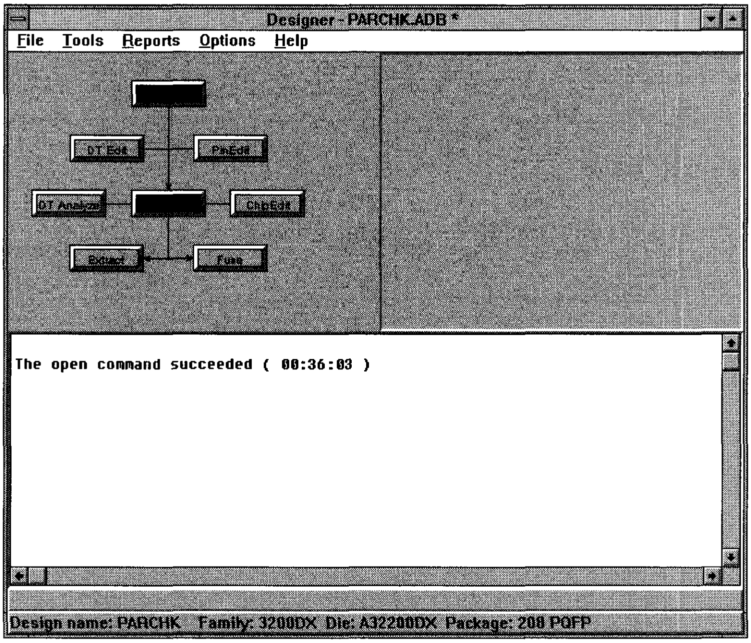
The rest of this document discusses the individual Designer Series components in greater detail.

The Designer Series Development System

Designer Series greatly simplifies Actel FPGA design. Its components ease design capture and streamline FPGA implementation. Designer Series is combined with ACTmap VHDL Synthesis and ACTgen Macro Builder to provide all the capabilities you need to implement an FPGA.

Designer Series offers a greatly improved user interface. The user interface is based on Design Flow Manager, a visual guide to the entire FPGA design process. Design Flow Manager (see Figure 2) shows each of the steps taken during FPGA implementation. With the Design Flow Manager's graphical guidance, Designer Series is extremely intuitive.

The Design Flow Manager also features an underlying Design State Manager to keep track of the information required to begin each step and to keep you informed of the design's current status. If information is missing at the beginning of a step, Designer Series will query for it. Once a step is finished, the corresponding box changes to a green color. Any operation that invalidates a completed step causes the Design Flow Manager to reset that step's color back to gray, indicating that changes have been made to require that step (and possibly subsequent steps) to be repeated. The Design State Manager can be instructed to watch the design's source files for changes. In this mode, any design source change results in a warning message indicating that design source may be out of synchronization with the placed and routed version.



```
The open command succeeded ( 00:36:03 )
```

Design name: PARCHK Family: 3200DX Die: A32200DX Package: 208 PQFP

Figure 2 • *The Design Flow Manager*

Design Flow Manager makes Designer Series easy to learn and use. The graphical Design Flow Manager is complemented by a comprehensive online help system, which provides context-sensitive descriptions of Designer Series' features. Online help is fully indexed to put complete, step-by-step instructions at your fingertips.

Place and Route

Designer Series provides place and route for Actel FPGAs. Starting with the release, Designer Series supports two ways to place and route a design. Standard layout is included with every Designer Series product. It uses Actel's proven layout algorithms automatically to place and route designs that use up to 95 percent of FPGA logic resources—even with all pins assigned. With Standard layout, you influence design performance by assigning criticality weights to a design's signals.

Designer Series can also place and route a design through the DirectTime Layout option. DirectTime Layout works from a timing specification on a design. This can be as simple as the target frequency or as detailed as a set of delay specifications for each of a design's critical nets. More detailed information on DirectTime Layout follows in "DirectTime Layout: Specification-Driven Design."

Both Standard and DirectTime Layout offer an incremental mode of operation. Incremental mode is useful when changes are made to a design that has previously been placed and routed. Changes are usually made to correct a detected design problem. Typically, these changes are made to a localized portion of the design. Incremental mode performs place and route only for the portion of a design that has changed. The remainder of the design is left untouched. As a result, incremental mode minimizes the impact of changes on design timing and reduces the time it takes to place and route a design that has not been substantially changed.

DirectTime Layout: Specification-Driven Design

The DirectTime Layout option (Figure 3) improves upon the place-and-route algorithms in the basic Designer Series Development System. DirectTime Layout works from a timing specification, which can be entered by using the DirectTime Editor. (See Figure 4.) The timing specification is either a target delay for the overall design or a list of path delay constraints for the critical paths in a design. The DirectTime Layout algorithm works from the timing specification and interactively makes trade-offs during the place-and-route phase. The DirectTime Layout algorithm takes advantage of the plentiful routing resources and the fine-grain logic modules to achieve the requested design timing.

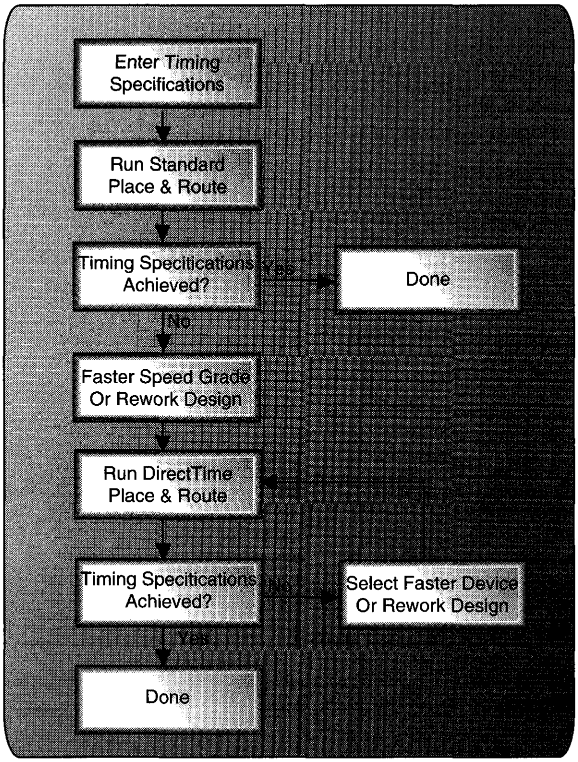
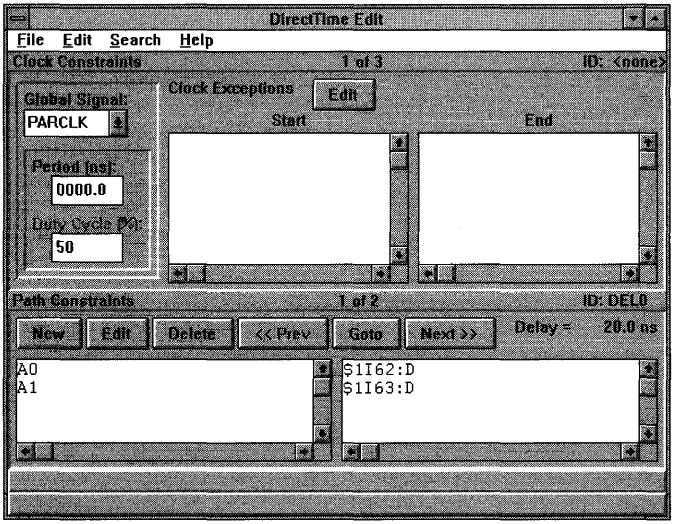

Enter Timing Specifications
Run Standard Place & Route
Timing Specifications Achieved?
Yes
Done
No
Faster Speed Grade Or Rework Design
Run DirectTime Place & Route
Timing Specifications Achieved?
No
Select Faster Device Or Rework Design
Yes
Done

Figure 3 • *Flow Diagram for DirectTime*



Clock Constraints 1 of 3 ID: <none>

Global Signal: PARCLK

Period [ns]: 0000.0

Duty Cycle [%]: 50

Path Constraints 1 of 2 ID: DEL0

Delay = 20.0 ns

```
A0          $1I62:D
A1          $1I63:D
```

Figure 4 • *DirectTime Editor*

DirectTime Layout performs a preplacement timing analysis of the design. Timing requirements that cannot physically be met with the selected FPGA are flagged at this point. During the place-and-route phase, the timing of each critical net is dynamically analyzed. Routing options are examined and discarded until each timing specification is met. DirectTime Layout's sophisticated algorithm simplifies the task of achieving a design's targeted performance.

DirectTime Layout fully supports high-level design practices. Timing requirements can be specified within a synthesized netlist and transferred into DirectTime through the Delay Constraint File (DCF) format. This allows timing requirements to be defined as the design description is entered.

Timing Analysis

The DirectTime Analyzer (see Figure 5) is a versatile timing analysis tool that provides crucial feedback on the timing results of Standard or DirectTime place and route. It gives easy access to information about a design's timing. DirectTime Analyzer provides a familiar spreadsheet view of the timing for selected signals, and it automatically sorts them, with longest delays shown first.

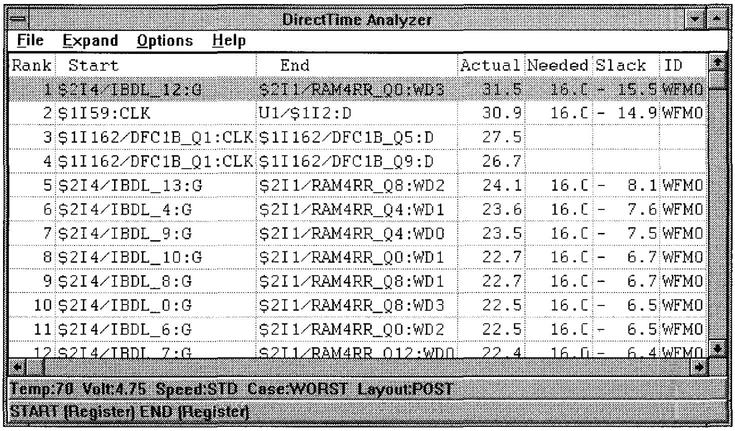
DirectTime Analyzer also provides a Filters dialog box (Figure 6), which makes it easy to select the signals to be displayed and the manner in which they will appear. Four types of signals are available for display: inpad-to-register, register-to-outpad, register-to-register, and inpad-to-outpad. It is easy to review the timing for related signal types, because the total delay for each selected path is initially displayed, no matter how many levels of logic contribute to that path. Expanding information on a signal also is easy. Double-click a spreadsheet line to show a table of delays contributed by each logic level between the path's start and end points.

DirectTime Analyzer even provides a graphical path delay expansion (see Figure 7), which diagrams the logic elements in a selected path and displays the relative delay contributions of each element and of interconnect. The display is similar to a project PERT chart, whereby the width of each box represents the delay of the corresponding logic element and the length of each line represents the interconnect delay. This graphic view makes it easy to determine the location of any timing bottlenecks.

DirectTime Analyzer can also display the timing results of a DirectTime Layout. In this scenario, the Analyzer tabulates both requested and achieved timing. If DirectTime Layout fails to achieve the requested timing, it's easy to see where difficulties occurred. Often, a faster speed grade will solve the problem. If not, the violating path can be expanded to determine whether there is a timing bottleneck that can be fixed with a logic change.

Simulation Interfaces

Postroute timing simulation is an important part of the design verification process. The results of implementing an Actel FPGA design can be thoroughly explored. Designer Series will backannotate postroute delay values to simulation products from Cadence, Mentor Graphics, Model Technologies, SimuCad, Synario, Veribest, Viewlogic, and others using the Standard Delay Format (SDF). Timing analysis can be accomplished with Verilog and VHDL simulators using Actel's Verilog and VITAL libraries.

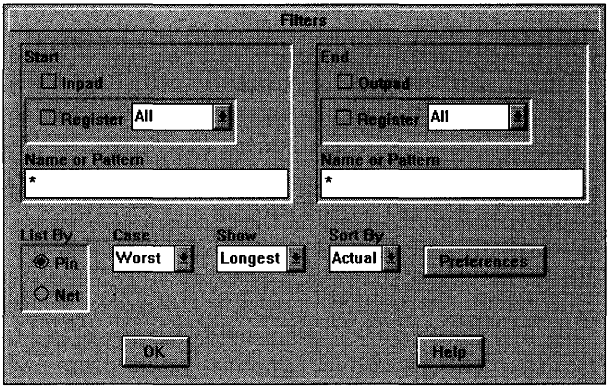


Rank	Start	End	Actual	Needed	Slack	ID
1	$2I4/IBDL_12:G	$2I1/RAM4RR_Q0:WD3	31.5	16.C	- 15.5	WFM0
2	$1I59:CLK	U1/$1I2:D	30.9	16.C	- 14.9	WFM0
3	$1I162/DFC1B_Q1:CLK	$1I162/DFC1B_Q5:D	27.5			
4	$1I162/DFC1B_Q1:CLK	$1I162/DFC1B_Q9:D	26.7			
5	$2I4/IBDL_13:G	$2I1/RAM4RR_Q8:WD2	24.1	16.C	- 8.1	WFM0
6	$2I4/IBDL_4:G	$2I1/RAM4RR_Q4:WD1	23.6	16.C	- 7.6	WFM0
7	$2I4/IBDL_9:G	$2I1/RAM4RR_Q4:WD0	23.5	16.C	- 7.5	WFM0
8	$2I4/IBDL_10:G	$2I1/RAM4RR_Q0:WD1	22.7	16.C	- 6.7	WFM0
9	$2I4/IBDL_8:G	$2I1/RAM4RR_Q8:WD1	22.7	16.C	- 6.7	WFM0
10	$2I4/IBDL_0:G	$2I1/RAM4RR_Q8:WD3	22.5	16.C	- 6.5	WFM0
11	$2I4/IBDL_6:G	$2I1/RAM4RR_Q0:WD2	22.5	16.C	- 6.5	WFM0
12	$2I4/IBDL_7:G	$2I1/RAM4RR_Q12:WD0	22.4	16.0	- 6.4	WFM0

Temp:70 Volt:4.75 Speed:STD Case:WORST Layout:POST

START (Register) END (Register)

Figure 5* • *DirectTime Analyzer's Timing Spreadsheet

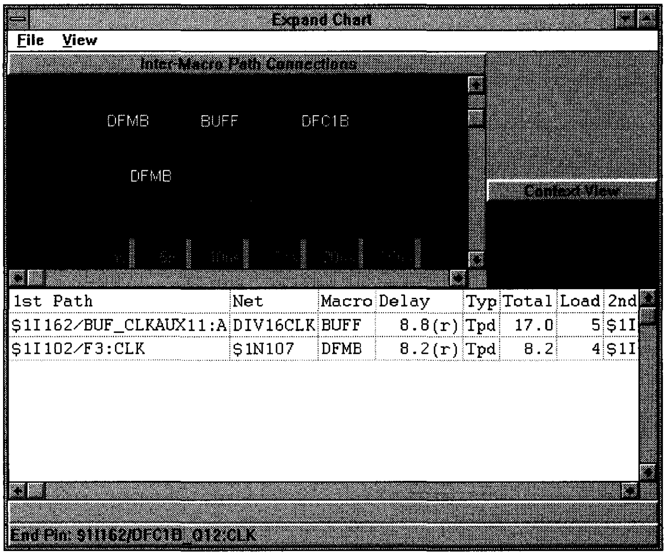


Start
□ Inpad
□ Register: All
Name or Pattern: *

End
□ Outpad
□ Register: All
Name or Pattern: *

List By: ● Pin ○ Net
Case: Worst
Show: Longest
Sort By: Actual

Figure 6 • *Filters to Make Signal Selection Simple*



1st Path	Net	Macro	Delay	Typ	Total	Load	2nd
$1I162/BUF_CLKAUX11:A	DIV16CLK	BUFF	8.8(r)	Tpd	17.0	5	$1I
$1I102/F3:CLK	$1N107	DFMB	8.2(r)	Tpd	8.2	4	$1I

End Pin: $1I162/DFC1B_Q12:CLK

Figure 7 • *DirectTime Analyzer's Graphical View*

Programming

The ultimate product of the FPGA design process is a programmed FPGA. Actel FPGAs can be programmed by using Data I/O Unisite or 3900 series programmers. In addition, Actel offers two programmers—the single-device Activator 2S (see Figure 8), and the four-device-at-one-time Activator 2 (see Figure 9). Both Activator programmers accept a variety of programming adapters to handle all FPGA device and package type combinations.

The Activators include SCSI connections that allow interfaces to Hewlett-Packard, Sun, and PC workstations. The Activator Programming Software (APS) runs on all three platforms. The Activator's programming algorithms are software configured, which allows APS to be quickly modified to support newly introduced FPGAs. The flexibility of the Activator 2 and Activator 2S provides the basis for additional capabilities: the debugger and the Actionprobe diagnostics tool.

Once the Activator has programmed a device, the FPGA check sum can be compared with the check sum calculated by the Designer Series software. This verifies that the FPGA has been programmed correctly.

The Debugger and Actionprobe Diagnostics

The Activator programmers have additional capabilities that allow them to act as functional testers and to provide in-circuit observation of the FPGAs internal signals. The Actionprobe diagnostics (which include the debugger) are an option that combines software and hardware to provide physical verification that a programmed FPGA is functioning correctly. The debugger can apply stimulus to a programmed FPGA and display the FPGA's response. Stimulus is applied in single-step mode, applying an input vector to the FPGA and then reporting its response.

The Actionprobe diagnostic hardware takes advantage of the antifuse programming circuitry to make internal signals visible and connects the Activator to the FPGA. The Actionprobe diagnostic tool is extremely useful. It enhances a designer's ability to determine a problem's cause.

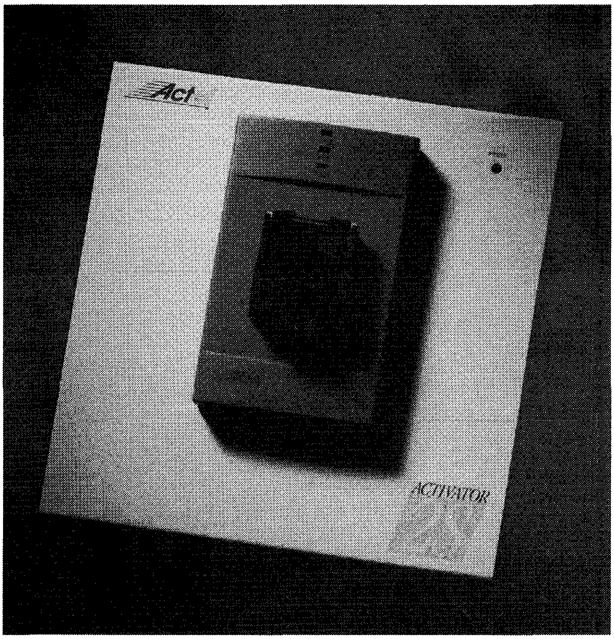


Figure 8 • Actel's Activator 2S

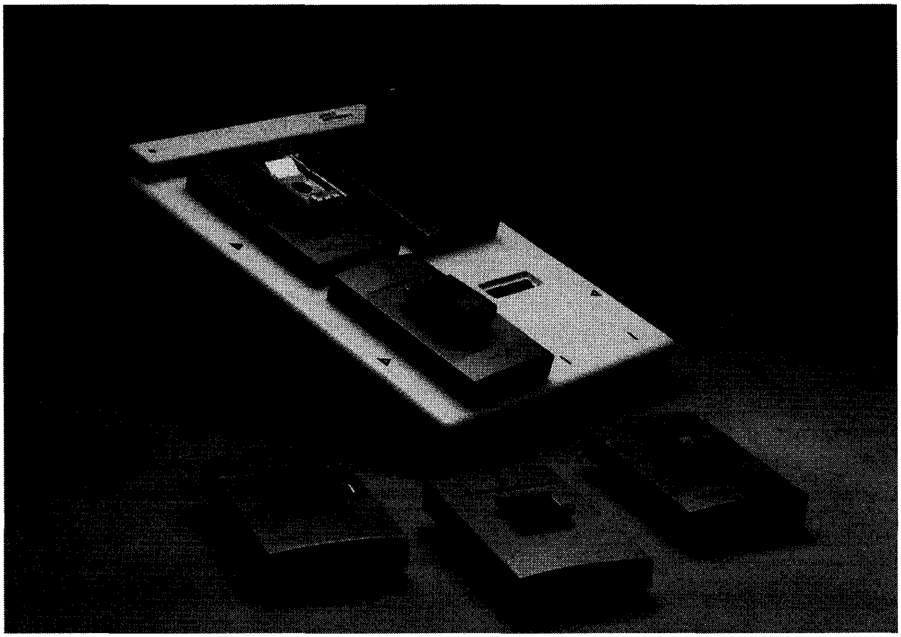


Figure 9 • *Actel's Activator 2*

ACTmap VHDL Synthesis

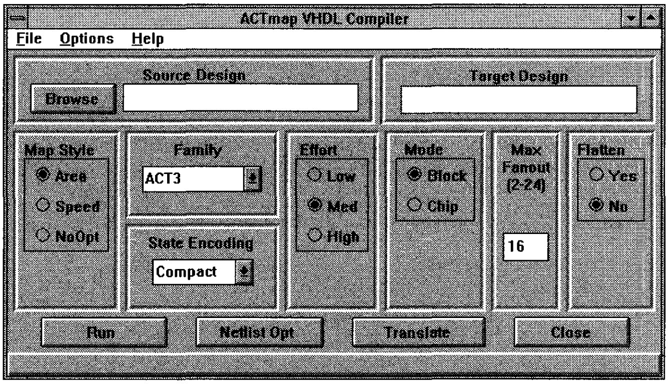
The features of ACTmap VHDL Synthesis include the following:

- Provides architectural expertise for efficient synthesis and logic optimization
- Accepts the synthesizable behavioral subset of VHDL
- Passes global clock constraints from VHDL to Designer Series
- Infers ACTgen macros
- Retargets existing designs from one Actel family to another
- Imports PAL designs and state-machine descriptions

ACTmap VHDL Synthesis provides a simple user interface for synthesis, logic optimization, retargeting among Actel FPGAs families, and importing of existing PAL designs. (See Figure 10.) ACTmap VHDL can synthesize a VHDL design description into an Actel FPGA. In addition, it can synthesize individual design blocks and interconnect them into a top-level design, or it can synthesize a complete FPGA from a high-level design description.

ACTmap VHDL Synthesis also provides highly effective optimization. It can optimize a design for either maximum operating speed or minimum area. It represents a design by using binary decision diagrams, which map efficiently onto Actel's multiplexer-based logic modules. ACTmap VHDL Synthesis can map a design onto a fixed library of logic functions, delivering high-quality synthesis results quickly, or it can map Boolean equations directly into the logic module for more efficient resource use. The choice between library or Boolean matching is user selectable.

ACTmap VHDL Synthesis can retarget existing FPGA designs from one FPGA family to another. An existing ACT 1 design can be pushed to higher performance—easily—by using ACTmap VHDL Synthesis to retarget the design into an ACT 3 device. ACTmap VHDL Synthesis can also help consolidate several small FPGAs into a single larger device. For example, four A1020 devices could be retargeted and combined into a single 1280XL FPGA. Wherever possible, ACTmap VHDL Synthesis takes advantage of family-specific features during the retargeting operation.



Map Style: ● Area ○ Speed ○ NoOpt

Family: ACT3

State Encoding: Compact

Effort: ○ Low ● Med ○ High

Mode: ● Block ○ Chip

Max Fanout (2-24): 16

Flatten: ○ Yes ● No

Figure 10 • *ACTmap VHDL Synthesis provides a simple user interface to synthesis*

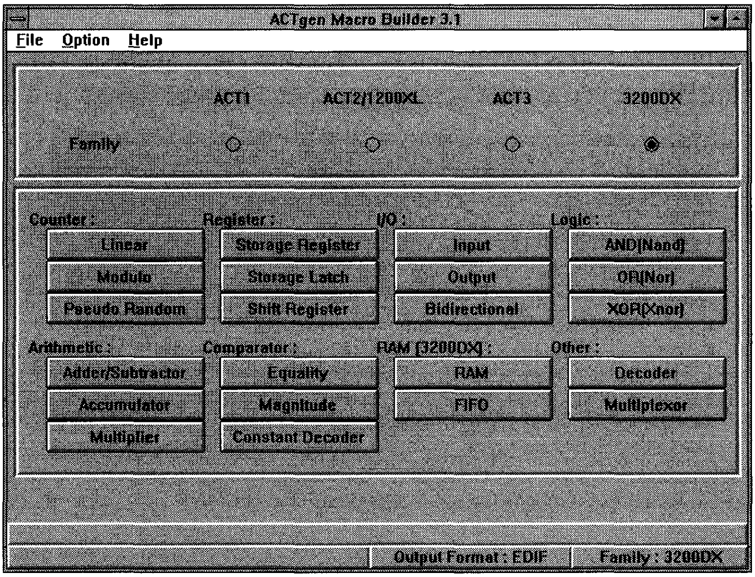
ACTgen Macro Builder

The features of ACTgen Macro Builder include the following:

- Automatically generates high-performance custom data-path macros from selected parameters
- Supports trade-offs between high performance and efficient use of resources
- Guarantees correct macro functions, because macros are built from thoroughly tested rules
- Top-level user interface provides access to all capabilities

ACTgen Macro Builder automatically generates data-path macros from a parametric specification. The intuitive user interface makes it easy to select a macro type (see Figure 11), and then to specify a variation within that type. Once the variation is selected bit width and the set of control signals present on a macro can be defined. (See Figure 12.) It is also easy to get the right balance between efficient resource use and high performance by selecting among the different variations available for each macro type. In Figure 12, the Compact and Ripple variations use the smallest number of logic resources to implement the function; the Balanced and Fast Enable counters are high performance with a somewhat higher logic resource use; and the Pre-Scaled and Register Look-Ahead counters provide the highest performance by using more resources.

ACTgen Macro Builder can generate its output in EDIF, Verilog, and VHDL, making it easy to interface with today's popular design tools. The EDIF output can be read into most schematics, automatically generating a symbol for the generated macro. The Verilog and VHDL formats allow ACTgen-created macros to be quickly instantiated into high-level design description in preparation for synthesis. ACTgen Macro Builder makes it easy to get high performance synthesis results for data-path elements, like counter, that typically yield lower performance when synthesized.

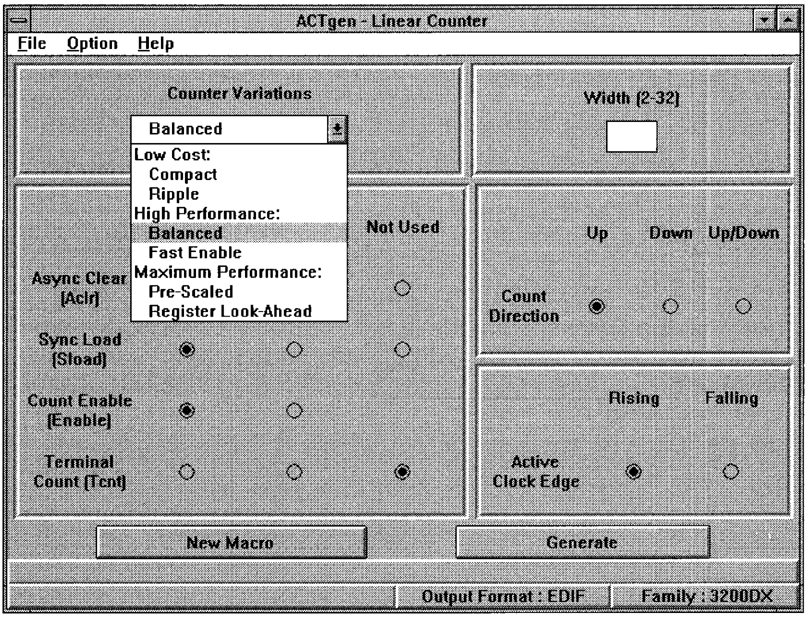


Family ○ ACT1 ○ ACT2/1200XL ○ ACT3 ● 3200DX

Output Format : EDIF

Family : 3200DX

Figure 11 • Available Macro Types



ACTgen - Linear Counter

Counter Variations

Balanced

Low Cost:
- Compact
- Ripple

High Performance:
- Balanced
- Fast Enable

Maximum Performance:
- Pre-Scaled
- Register Look-Ahead

Width [2-32]

			Not Used
Async Clear [Aclr]			○
Sync Load [Sload]	●	○	○
Count Enable [Enable]	●	○	
Terminal Count [Tcnt]	○	○	●

	Up	Down	Up/Down
Count Direction	●	○	○

	Rising	Falling
Active Clock Edge	●	○

Output Format : EDIF

Family : 3200DX

Figure 12 • Selecting Variation, Control Signals, and Bit Width

EDA Vendor Support

This document describes the EDA environments that Actel supports. This document also covers each of the Actel-supported vendors. Within each vendor's tool set the features will be described that are supported by Actel. Also listed are the software and computer requirements for integrating Designer Series with the EDA front-end tools.

Actel's Alliance Partners

Actel's Alliance program was established to assist EDA vendors in providing support for Actel FPGAs. The Alliance program provides early technical information on new Actel releases to all partners so they can offer timely support. Table 1 lists the complete set of EDA vendors that support the design of Actel FPGAs.

Table 1 *• Alliance Program EDA Vendors*

Company	Contact	Address
Acugen	J. W. Brooks (603) 881-8821	427-3 Amherst St., Suite 391, Nashua, NH 03063
Aldec	Stanley Hyduke (805) 499-6867	3525 Old Conejo Road, Suite 111 Newbury Park, CA 91320
Cadence	Itzhak Shapira (408) 944-7734	2655 Seely Road, Bldg. 6, San Jose, CA 95134
Compass	John Goldsworthy (408) 383-4720 ext. 52931	1865 Lundy Ave., San Jose, CA 95131
Escalade	Mark Miller (408) 481-1300	2475 Augustine Dr., 2nd Floor, Santa Clara, CA 95054
Exemplar Logic	Mary Murphy (510) 337-3785	815 Atlantic Ave., Suite 105, Alameda, CA 94501-2274
Intergraph	Will Wong (415) 691-6447	381 East Evelyn Avenue, Mountain View, CA 94041
Isdata	Ralph Remme 49 721 751087	Daimlerstr. 51, Karlsruhe, D-76185, Germany
IST	Gabriele Saucier 33 76 574687	Europole, 4 Place Robert Schuman 38024 Grenoble Cedex 1, France
Logic Modeling (Synopsys)	Marnie McCollow (503) 531-2412	19500 N.W. Gibbs Drive, Beaverton, OR 97006
Logical Devices	David Motarjemi (305) 974-0967	1201 N.W. 65th Place, Ft. Lauderdale, FL 33309
Mentor Graphics	Sam Picken (503) 685-1298	8005 S.W. Boeckman Road, Wilsonville, OR 97070-7777
Minc	Wayne Merrill (719) 590-1155	6755 Earl Drive, Colorado Springs, CO 80918
OrCAD	Troy Scott (503) 671-9500	9300 S.W. Nimbus, Beaverton, OR 97005
Quad Design	Hector Lai (805) 988-8250	1385 Del Norte Road, Camarillo, CA 93010
Simucad	John Williamson (415) 487-9700	32970 Alvarado-Niles Rd., Suite 744, Union City, CA 94587
Synario Design Automation	Dave Kohlmeiyer (206) 867-6802	10525 Willows Road N.E., Redmond, WA 98073-9746
Synopsys	Lynn Fiance (415) 694-4289	700 East Middlefield Road, Mountain View, CA 94043
Synplicity	Alisa Yaffa (415) 961-4962	465 Fairchild Dr., Suite 115, Mountain View, CA 94043
Teradyne	Bill Loring (617) 422-2769	179 Lincoln St., M/S L50, Boston, MA 02111
Veda Design Automation (formerly Genrad)	Rastgow Shale (408) 496-4518	2041 Mission College Blvd., Suite 259 Santa Clara, CA 95054
Viewlogic	Dave Orecchio (508) 480-0881	293 Boston Post Road, Marlboro, MA 01752
Zuken	Dwight Dagenais (408) 562-0177	3945 Freedom Circle, Suite 1100, Santa Clara, CA 95054

Designer Series for Cadence Design Systems

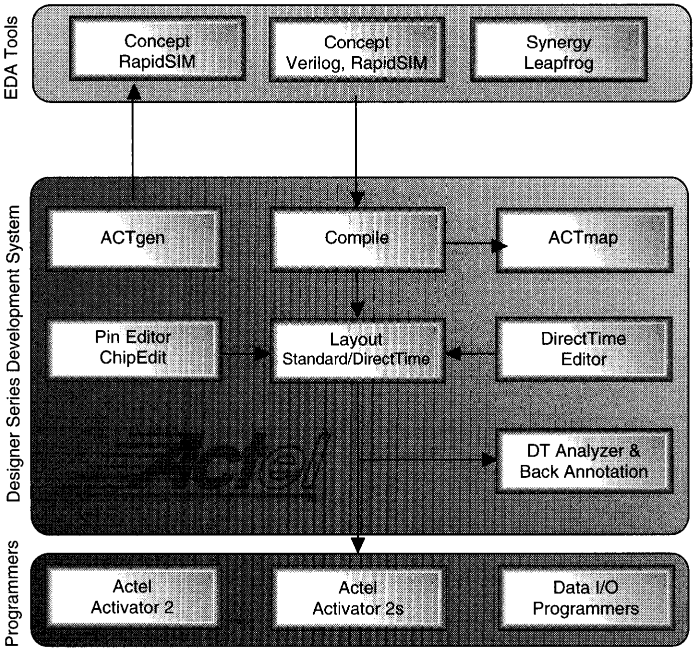


Figure 1 • The Cadence EDA Environment

The Designer Series Development System for the Cadence Design Systems environment (see Figure 1) allows FPGAs with from 1,000 to 30,000 gates to be designed with Cadence's Concept (Logic Workbench) or Composer schematic capture tools. Designer Series provides libraries that support the design of ACT 1, the Integrator Series, and the Accelerator Series families of FPGAs. In addition, Designer Series provides ACTmap VHDL Synthesis (a complete VHDL synthesis and logic optimization tool) and ACTgen Macro Builder, which creates complex logic functions to match user-specified parameters.

For Cadence, Designer Series supports the following combinations of schematic capture and simulation:

- Composer and Verilog XL
- Concept and RapidSIM
- Concept and Verilog

These design flows allow schematics to be created using Composer or Concept. Designer Series products include netlisters that convert Composer or Concept schematics into a netlist format that Designer Series can accept. The netlist is then simulated with unit delays to verify the design functionally before running place and route. After place and route, actual module and net delays are backannotated to the netlist for timing simulation. Actel's DirectTime Analyzer is used to perform static timing analysis on a design—identifying critical path delays and performance deltas relative to design specifications. Once the implementation has met all timing requirements, the design can be programmed into an FPGA.

Actel's optional DirectTime Layout feature can be added to the basic Designer Series Cadence interface. DirectTime Layout supports specification-driven design. With DirectTime Layout, the system clock frequency or delay constraints for individual signals can be entered. After Actel's standard place-and-route algorithm is executed, the results are shown in the DirectTime Analyzer. DirectTime Analyzer displays a comparison between required versus actual delays. DirectTime Layout can typically improve design performance by an amount equivalent to the next fastest speed grade. It makes performance-runtime trade-offs during placement and routing to achieve the requested timing.

If the results of standard layout are substantially slower than the required timing, more aggressive measures are needed. The designer has the option of moving to a faster-speed-grade FPGA or even to a faster family. DirectTime Analyzer makes these decisions easy by clearly showing whether the target performance is reached and showing the distance between target and achieved when target is missed.

There are many high-level design solutions that support Actel in the Cadence environment. Cadence's Synergy and PIC Designer products as well as Synopsys's Design and FPGA Compilers all work within the Cadence environment. Synthesized blocks in any of these synthesis tools can be combined with schematics to support mixed-level design definition. Cadence provides Actel libraries for the PIC Designer and Synergy synthesis products. Actel's Synopsys synthesis libraries can be added to the Designer Series Development System to support Synopsys synthesis in the Cadence environment. Actel also makes available Verilog and VITAL VHDL libraries so that designs captured in these languages can be simulated directly with Verilog XL or Leapfrog.

The Designer Series Development System for Cadence provides a tight link between Cadence's suite of design capture and analysis tools and Actel's FPGA implementation software. The integration of Designer Series with the Cadence tools provides a design environment that delivers high-performance, high-capacity FPGA solutions quickly. Combining the Actel architecture with powerful place-and-route software provides short turnaround time for ECNs. Designer Series makes it easy to include custom logic in any product.

Sun SPARC or HP 700

Hardware Requirements

- 64 MB RAM
- 125 MB DISK (executables), 5 MB DISK (per design)
- CD-ROM drive

Software Requirements

- Sun OS 4.1.3 or later, Solaris 5.3 or later, OR HP-UX 9.03 or later
- Version 9404 or later

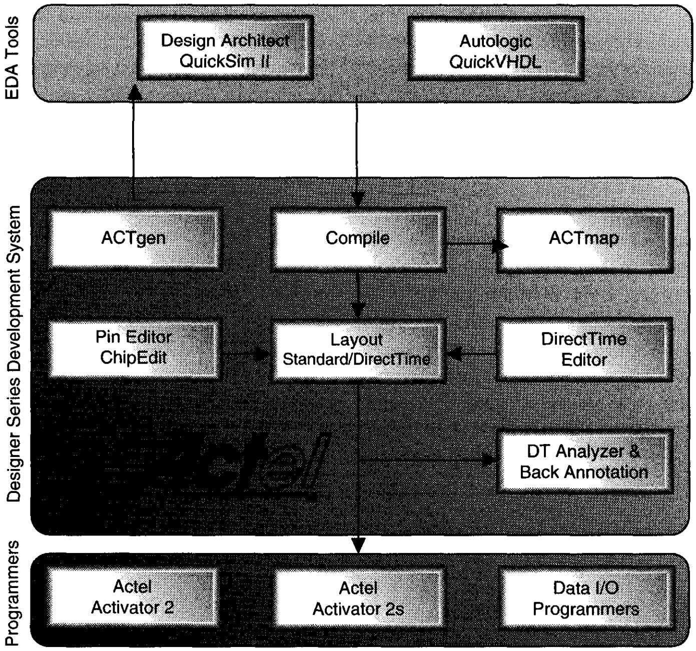


Figure 2* • *The Mentor Graphics EDA Environment

The Designer Series Development System for the Mentor Graphics environment (see Figure 2) allows FPGAs with from 1,000 to 30,000 gates to be designed with Mentor Graphics' Design Architect. Designer Series provides libraries that support the design of ACT 1, the Integrator Series, and the Accelerator Series families of FPGAs. In addition, Designer Series provides ACTmap VHDL Synthesis (a complete VHDL synthesis and logic optimization tool) and ACTgen Macro Builder, which creates complex logic functions to match user-specified parameters.

The design flow for Mentor Graphics allows schematics to be entered using Design Architect. Designer Series products include netlisters that convert Design Architect schematics into EDIF, which Designer Series can accept. The netlist is then simulated with unit delays to verify the design functionally before running place and route. After place and route, actual module and net delays can be backannotated to the netlist for timing simulation. Actel's DirectTime Analyzer can be used to perform static timing analysis on a design—identifying critical path delays and performance deltas relative to design specifications. Once the implementation has met all timing requirements, the design can be programmed into an FPGA.

Actel's optional DirectTime Layout feature can be added to the basic Designer Series Mentor Graphics interface. DirectTime Layout supports specification-driven design. With DirectTime Layout, the system clock frequency and delay constraints for individual signals can be entered. After Actel's standard place-and-route algorithm is executed, the results are shown in DirectTime Analyzer. DirectTime Analyzer displays a comparison between required versus actual delays. DirectTime Layout can typically improve design performance by an amount equivalent to the next fastest speed grade. It makes performance-runtime trade-offs during placement and routing to achieve the requested timing.

If the results of standard layout are substantially slower than the required timing, more aggressive measures are needed. The designer has the option of moving to a faster-speed-grade FPGA or even to a faster family. DirectTime Analyzer makes these decisions easy by clearly showing whether the target performance is reached and showing the distance between target and achieved when target is missed.

There are many high-level design solutions that support Actel in the Mentor Graphics environment. Mentor Graphics' Autologic and Autologic II products as well as Synopsys's Design and FPGA Compilers all work within the Mentor

Graphics environment. Synthesized blocks in any of these synthesis tools can be combined with schematics to support mixed-level design definition. Mentor Graphics provides the Actel libraries for Autologic synthesis products. Actel's Synopsys synthesis libraries can be added to the Designer Series Development System to support Synopsys synthesis in the Mentor Graphics environment. Actel also makes available Verilog and VITAL VHDL libraries so that designs captured in these languages can be simulated directly with QuickSim or QuickVHDL.

The Designer Series Development System for Mentor Graphics provides a tight link between the Mentor Graphics suite of design capture and analysis tools and Actel's FPGA implementation software. The integration of Designer Series with the Mentor Graphics tools provides a design environment that delivers high-performance, high-capacity FPGA solutions quickly. Combining the Actel architecture with powerful place-and-route software provides short turnaround time for ECNs. Designer Series makes it easy to include custom logic in any product.

Sun SPARC or HP 700

Hardware Requirements

- 64 MB RAM
- 125 MB DISK (executables), 5 MB DISK (per design)
- CD-ROM drive

Software Requirements

- Sun OS 4.1.3 or later, Solaris 5.3 or later, OR HP-UX 9.03 or later
- Version 8.2_5 or later (A.1 recommended)

Designer Series for Viewlogic

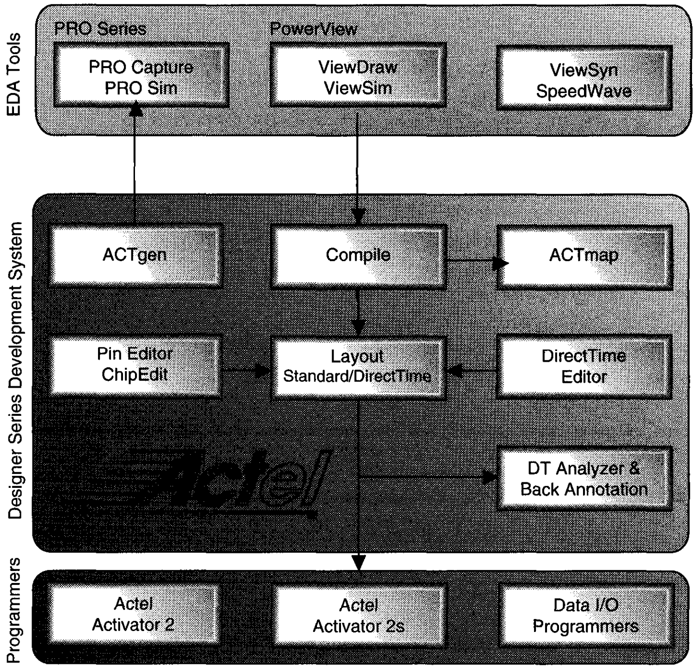


Figure 3 • *The Viewlogic EDA Environment*

The Designer Series Development System for the Viewlogic environment (see Figure 3) allows FPGAs with from 1,000 to 30,000 gates to be designed with Viewlogic's Workview Office or PowerView tools. Designer Series provides libraries that support the design of ACT 1, the Integrator Series, and the Accelerator Series families of FPGAs. In addition, Designer Series provides ACTmap VHDL Synthesis (a complete VHDL synthesis and logic optimization tool) and ACTgen Macro Builder, which creates complex logic functions to match user-specified parameters.

For Viewlogic, Designer Series supports the following workstation-product combinations:

- 486 and Pentium PCs: Workview Office
- HP 700 and Sun workstations: PowerView

These design flows allow schematics to be created using PowerView on the workstations or Workview Office on the PC. The Designer Series products include netlisters that convert PowerView or Workview Office schematics into an EDIF netlist that Designer Series can accept. The netlist is then simulated with unit delays to verify the design functionally before running place and route. After place and route, actual module and net delays are backannotated to the netlist for timing simulation. Actel's DirectTime Analyzer is used to perform static timing analysis on a design—identifying critical path delays and performance deltas relative to design specifications. Once the implementation has met all timing requirements, the design can be programmed into an FPGA.

Actel's optional DirectTime Layout feature can be added to the basic Designer Series Viewlogic interface. DirectTime Layout supports specification-driven design. With DirectTime Layout, the system clock frequency and delay constraints for individual signals can be entered. After Actel's standard place-and-route algorithm is executed, the results are shown in DirectTime Analyzer. DirectTime Analyzer displays a comparison between required versus actual delays. DirectTime Layout can typically improve design performance by an amount equivalent to the next fastest speed grade. It makes performance-runtime trade-offs during placement and routing to achieve the requested timing.

If the results of standard layout are substantially slower than the required timing, more aggressive measures are needed. The designer has the option of moving to a faster-speed-grade FPGA or even to a faster family. DirectTime Analyzer makes these decisions easy by clearly showing whether the target performance is reached and showing the distance between target and achieved when target is missed.

There are many high-level design solutions that support Actel in the Viewlogic environment. Viewlogic's ViewSynthesis as well as Synopsys's Design and FPGA Compilers all work within the Viewlogic environment. Synthesized blocks in any of these synthesis tools can be combined with schematics to support mixed-level design definition. Viewlogic provides Actel libraries for the ViewSynthesis product. Actel's Synopsys synthesis libraries can be added to the Designer Series Development System to support Synopsys synthesis in the Viewlogic environments. Actel also makes available Verilog and VITAL VHDL libraries so that designs captured in these languages can be simulated directly with Chronologic or Speedwave.

The Designer Series Development System for Viewlogic provides a tight link between Viewlogic's suite of design capture and analysis tools and Actel's FPGA implementation software. The integration of Designer Series with the Viewlogic tools provides a design environment that delivers high-performance, high-capacity FPGA solutions quickly. Combining the Actel architecture with powerful place-and-route software provides short turnaround time for ECNs. Designer Series makes it easy to include custom logic in any product.

486 or Pentium PC

Hardware Requirements

- VGA, EGA graphics card
- 32 MB RAM
- 60 MB virtual disk
- 70 MB DISK (executables), 5 MB DISK (per design)
- CD-ROM drive

Software Requirements

- Windows 3.1 or later, Windows NT 3.5.1 or later, Windows 95
- PROSeries 6.1 or later
- Workview Office 7.1 or later

Sun SPARC or HP 700

Hardware Requirements

- 64 MB RAM
- 125 MB DISK (executables), 5 MB DISK (per design)
- CD-ROM drive

Software Requirements

- Sun OS 4.1.3 or later, Solaris 5.3 or later, OR HP-UX 9.03 or later
- PowerView 5.3 or later

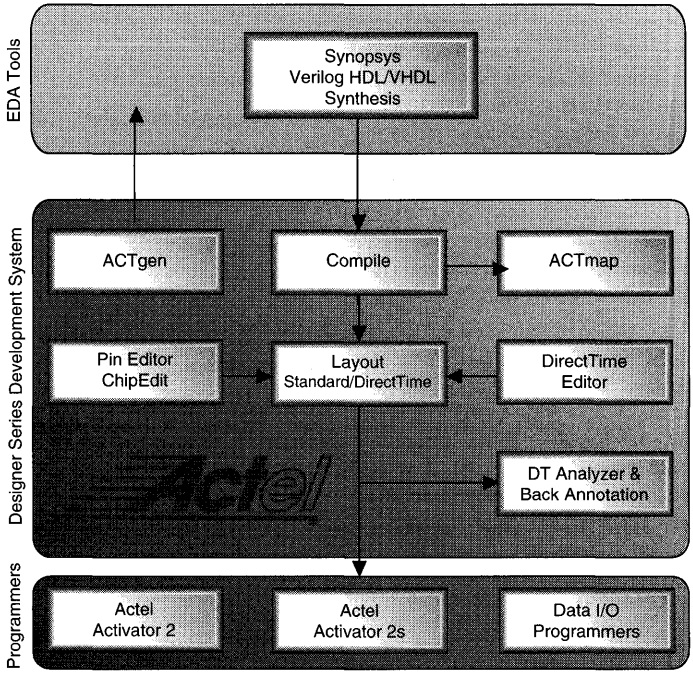


Figure 4 • The Synopsys EDA Environment

The Synopsys libraries support synthesis of Actel FPGAs from Verilog HDL or VHDL. (See Figure 4.) Support for synthesis, DesignWare, and VHDL System Simulator (VSS) is included. The Synopsys libraries provide a complete top-down design solution. Actel FPGA designs can be compiled with either the Design Compiler or the FPGA Compiler, with the FPGA Compiler yielding more efficient, high-performance FPGA implementations.

The combination of Actel's fine-grain, routing-resource-rich architecture and Synopsys's powerful synthesis program delivers high-performance, high-capacity FPGAs. The Synopsys libraries allow designs to be efficiently mapped to the Actel architecture, taking complete advantage of all possible logic module functions. Design implementation efficiency is enhanced by the DesignWare library, which contains Actel-optimized adders, counters, multiplexers, registers, and so on. These logic elements all can be instantiated into a high-level design description. Adders, comparators, and subtracters can be inferred from arithmetic operators that are used in the behavioral description. Synopsys has included Actel-specific optimization, such as sequential mapping, which takes full advantage of the multiplexer-based, flip-flop-oriented FPGA architecture. The combination of Synopsys's optimizations for Actel FPGAs and the optimized DesignWare logic elements generates designs that are typically within 5 percent of what can be achieved by laborious handcrafting.

Simulation of the design with its surrounding test bench can be accomplished by using VSS and Actel's VHDL libraries for Synopsys or by using any Verilog simulator and the separately available Verilog libraries. Both the Verilog and VHDL simulation libraries support preroute functional simulation to verify the design's functionality and accurate postroute timing simulation with backannotated (using SDF) actual delays. Timing simulation can be performed with best-case, worst-case, or typical delays to account for variations in voltage, process, and temperature. This accurate timing simulation can identify areas in which timing error is causing a design to malfunction.

The Synopsys libraries support many of the FPGA-specific architectural feature, such as ACT 3's complex I/O cells, 1200XL's wide-decode feature, and 3200DX's quad-clock and embedded RAM features. Synopsys's support of architectural specialties allows designers to maximize FPGA performance and usefulness by synthesizing a design that includes high-speed FIFOs or dual-port RAMs in a 32200DX device, or by targeting the I/O flip-flops of the ACT 3 family to locate a high-speed counter right on a data bus.

The DesignWare libraries provide a strong advantage in the design of Actel FPGAs. Actel provides a synthetic library in DesignWare, which includes commonly used functions such as accumulators, adders, comparators, counters, decoders, multiplexers, and registers up to 32 bits wide. Each logic function can be created with a designer-selected combination of control signals. By taking advantage of the synthetic library, the VHDL or Verilog code can be fine-tuned for optimal Actel implementation. These libraries are based upon Actel's ACTgen Macro Builder parameterized macro generator, which allows designers to achieve performance comparable to schematic-drawn macros. DesignWare elements can be instantiated into the code or inferred by the compilers, resulting in an extremely efficient design process.

Sun SPARC or HP 700

Hardware Requirements

- 64 MB RAM
- 125 MB DISK (executables), 5 MB DISK (per design)
- CD-ROM drive

Software Requirements

- Sun OS 4.1.3 or later, Solaris 5.3 or later, OR HP-UX 9.03 or later
- Synopsys 3.2b or later

High-Level Design Resources

Vendor Libraries

Actel provides support for synthesis and simulation products offered by leading EDA vendors. Synthesis technology libraries are available for Synospys FPGA Compiler, including library support for Synospys DesignWare and Synopsys VSS VHDL simulator.

Synthesis technology libraries are available from all leading third-party synthesis vendors, including Exemplar Galileo, Mentor Graphics Autologic, Synplicity, Cadence Synergy, MINC/IST ASYL, and Viewlogic ViewSyn. Contact your sales representative, or check the Actel World Wide Web Homepage for the latest availability.

CAE Alliance Program

Actel maintains cooperative marketing and development relationships with industry-leading EDA vendors. The goal is to give users their tools of choice while ensuring quality of results from third-party vendors.

Standard Libraries

Actel provides support for third-party EDA products though standard simulation libraries based on the VITAL 3.0 and OVI 2.0 language standards and on SDF 2.0 timing delay files. These libraries are available on all Actel support platforms at nominal costs. The SDF format is included in Designer Series.

ACTmap VHDL Synthesis

Actel bundles the ACTmap VHDL Synthesis tool with every Designer Series system. This product provides language support for VHDL (IEEE1076, 1993), PALASM 2.0, Actel's proprietary netlist, ADL, and EDIF 200.

This product is described in more detail in "Designer Series Development System" on page 2-5.

Methodology

Actel FPGAs are supported by various design methodologies ranging from schematic-capture/gate-level simulation to full language-based design in a system-simulation environment. The right methodology for you often depends on the board-level design methodology adopted by your team. FPGAs offer the flexibility to customize logic and thereby minimize the cost of design errors, but a proper methodology can save numerous hours of debugging during the design cycle. Errors caught early are the cheapest to correct, so a functional verification is a critical first step, regardless of the design capture approach.

Benefits of High-Level Design

A key benefit of high-level design is the ability to specify and verify functionality before implementation. Models for standard ICs, such as processors and memory, are also widely available so that board or system-level functional verification can be accomplished. Functionally correct code blocks can also serve as building blocks for future devices, a process termed *design reuse*. This is a powerful way to reduce design cycle time.

FPGA Implementation

After functional verification, the critical step for FPGA implementation is synthesis and optimization, followed by timing verification. A proper methodology is required. Actel has developed synthesis guidelines that lead to better results in less time, when followed correctly. These guidelines are available, through our World Wide Web site on the Internet, and in the product documentation shipped with the Designer Series software.

World Wide Web Support

A section of Actel's Web page is dedicated to high-level design methodology and issues. It is accessible from the Homepage or by directly accessing http://www.actel.com/hld.

Information provided includes FAQ (frequently asked questions), the latest vendor support with product version and release dates, methodology guidelines and application notes, and customer benchmarks and references. In addition, a message folder is available for sharing issues and questions with other engineers on the Internet.

Training

Actel offers synthesis training for ACTmap VHDL Synthesis. In addition, several third-party vendors include Actel-specific material in their training courses. In special situations, on-site synthesis training can be arranged.

Consulting Services

In some cases, direct support from Actel's methodology consulting team or from Actel-certified third-party design consultants can be arranged.

Macro Libraries

Section 3: Macro Libraries

Integrator Series Hard Macro Library – Graphical Symbols

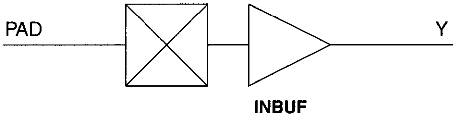
The following illustrations show all the available hard macros. The Actel *Macro Library Guide* contains module count, combinability, and pin loading information of each hard macro. It also includes a complete truth table for each macro.

Most Integrator Series hard macros (1200XL and 3200DX) are implemented by a single logic module. The following Integrator Series hard macros require more than one logic module to implement:

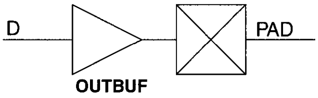
- Two-module hard macro gates: AND4D, AOI4, AX1A, MXC1, MXT, NAND4, NOR4, OAI3, OR4D
- Two-module adders: FA1A, FA1B, FA2A, HA1, HA1A, HA1B, HA1C
- Two-module latches: DL2A, DL2B, DL2C, DL2D, DLE2A, DLE3A, DLM2A
- Two-module flip-flops: DFC1, DFC1A, DFC1E, DFC1G, DFM3, DFM3E, DFP1, DFP1A, DFP1B, DFP1C, DFP1D, DFP1F, DFPC, DFPCA, JKF2C, JKF2D, JKF3A, JKF3B, JKF3C, JKF3D, JKF4B, JKFPC

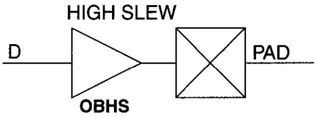
Two-module hard macro gates have a "2" displayed on some input pins. This indicates that the input to output path has two levels of logic delay for these input pins only. Also, the full adders have a "2" on the "S" output pins. This indicates that there are two levels of logic delay from the input pins to the "S" output pin. Refer to the Integrator Series Timing Characteristics for detailed timing information.

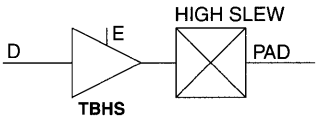
Input Buffers

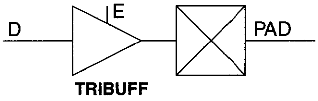


Output Buffers



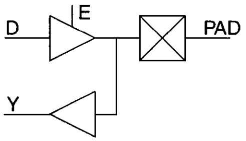




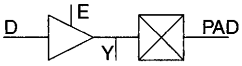


Bidirectional Buffers

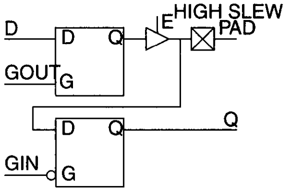
BBHS



BIBUF

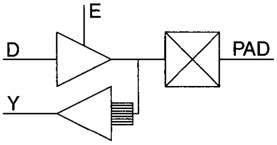


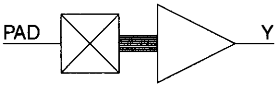
BBDLHS



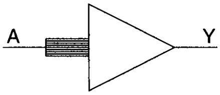
Clock (Dedicated Network) Buffers

CLKBIBUF



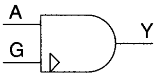


CLKBUF

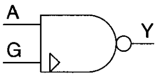


CLKINT

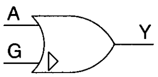
CLKBUF Interface Macros



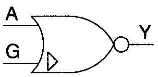
GAND2



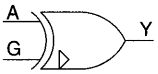
GNAND2



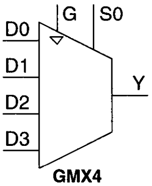
GOR2



GNOR2

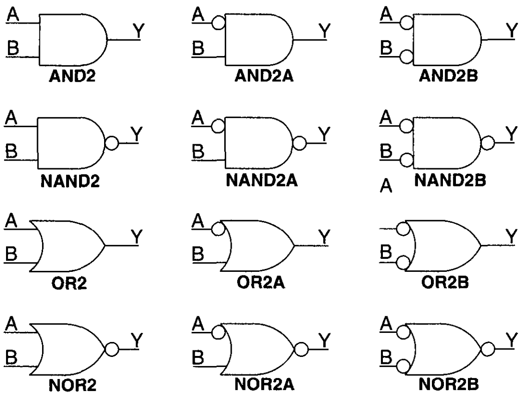


GXOR2

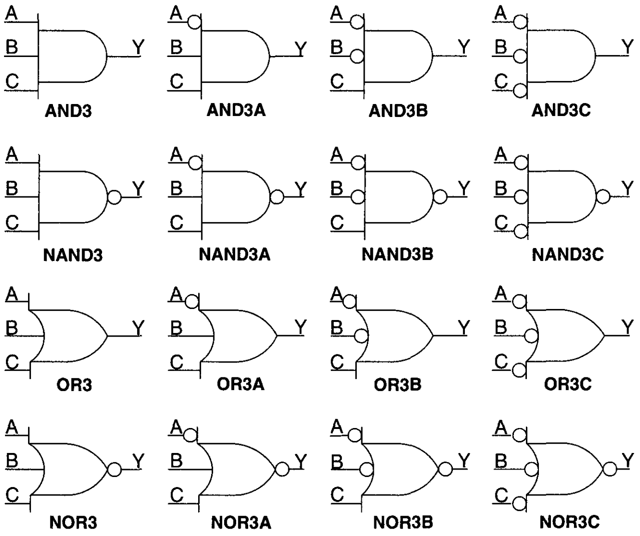


GMX4

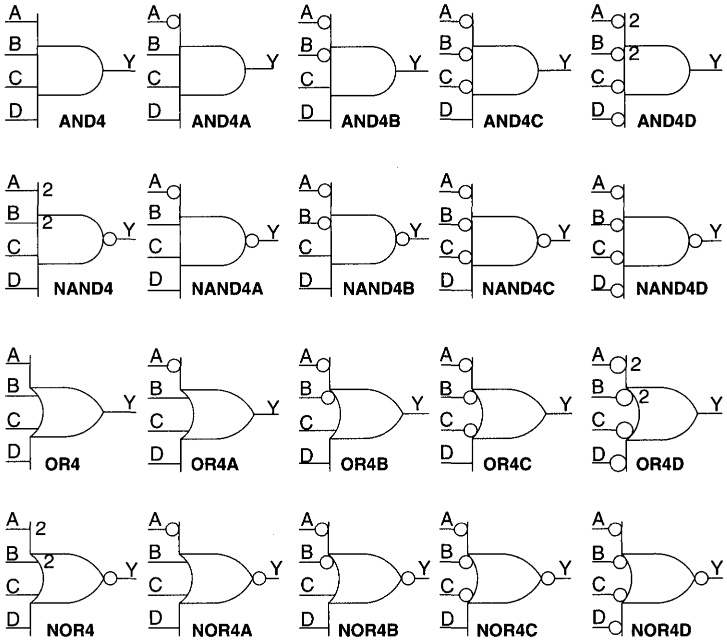
2-Input Gates



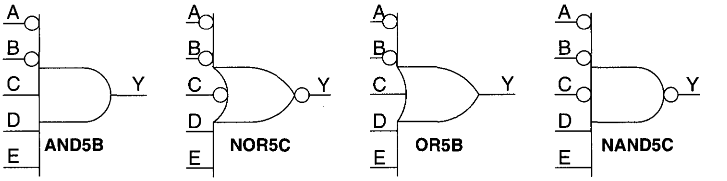
3-Input Gates



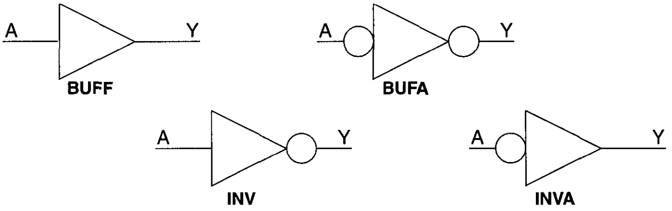
4-Input Gates



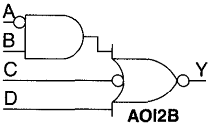
5-Input Gates



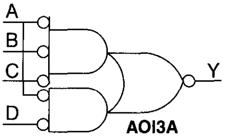
Buffers



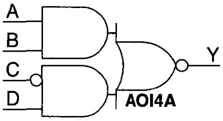
AND-OR Gates (continued)



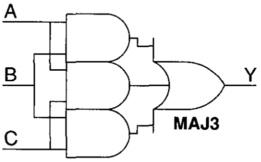
AOI2B



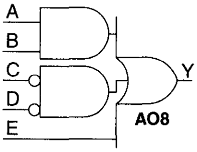
AOI3A



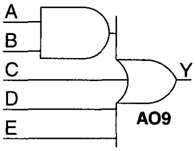
AOI4A



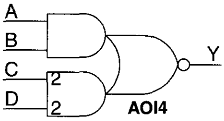
MAJ3



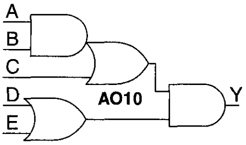
AO8



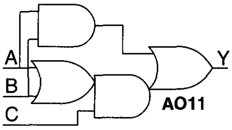
AO9



AOI4

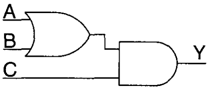


AO10

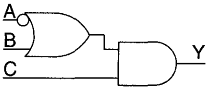


AO11

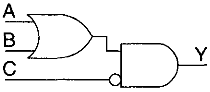
OR-AND Gates



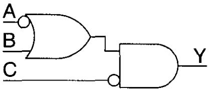
OA1



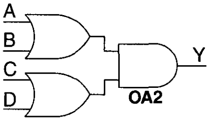
OA1A



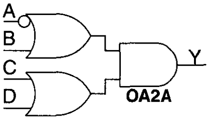
OA1B



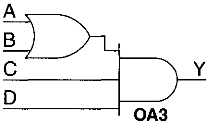
OA1C



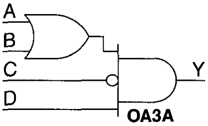
OA2



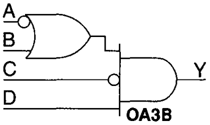
OA2A



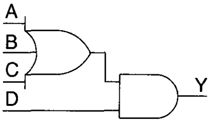
OA3



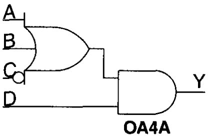
OA3A



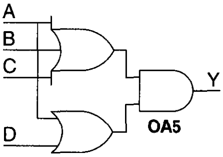
OA3B



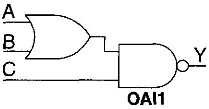
OA4



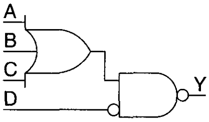
OA4A



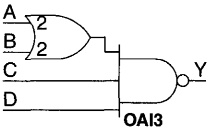
OA5



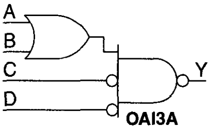
OAI1



OAI2A

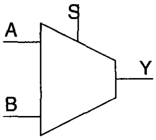


OAI3

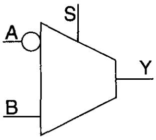


OAI3A

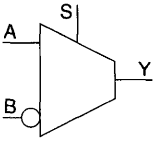
Multiplexors



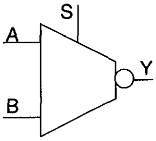
MX2



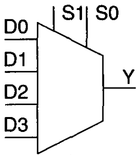
MX2A



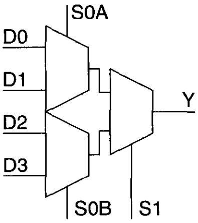
MX2B



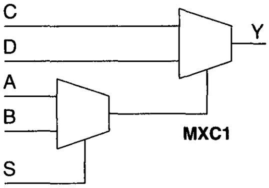
MX2C



MX4

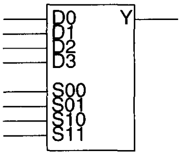


MXT

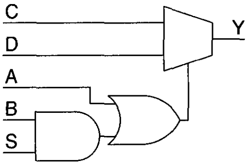


MXC1

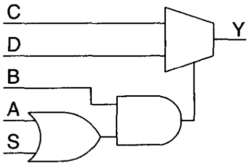
Combinatorial



CM8

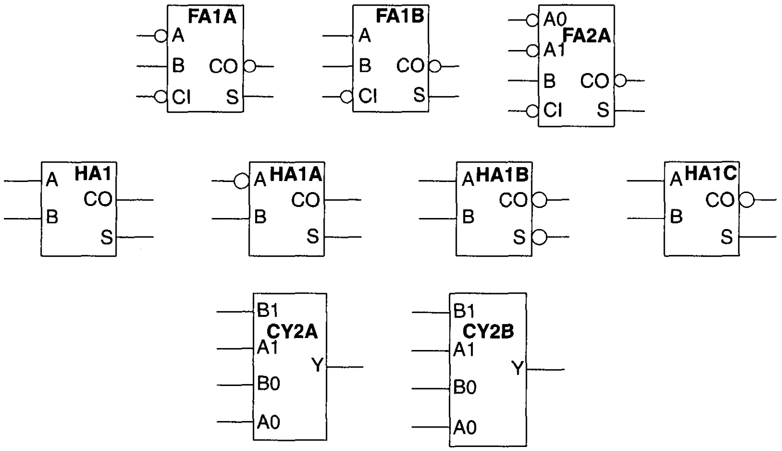


CS1

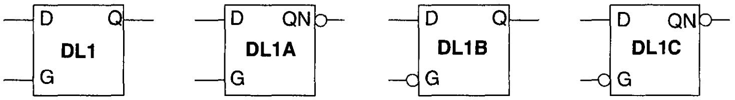


CS2

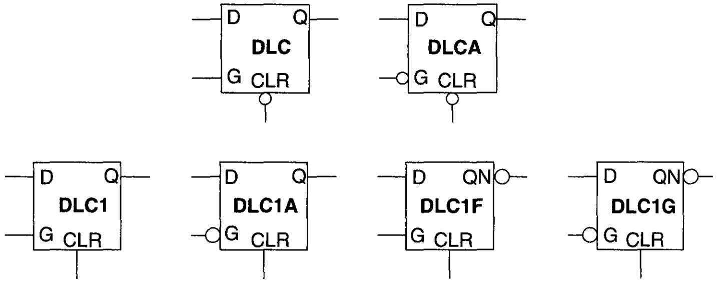
Adders



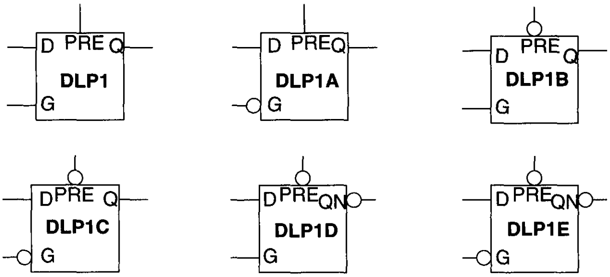
D-Latches



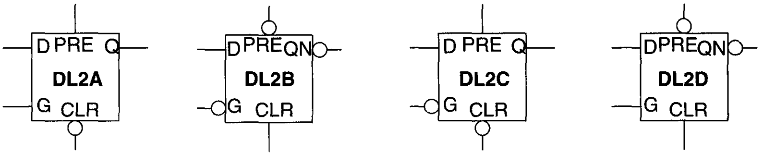
D-Latches with Clear



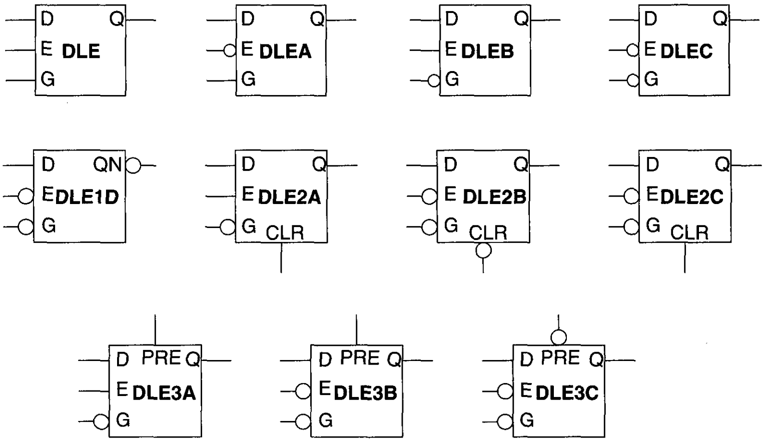
D-Latches with Preset



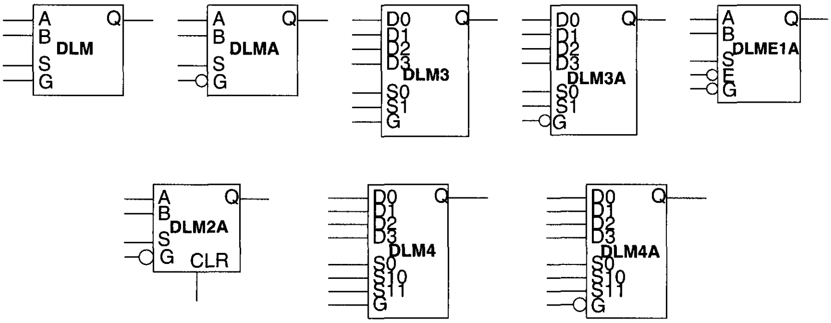
D-Latches with Preset and Clear



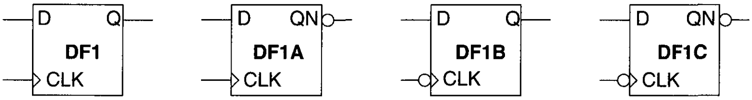
D-Latches with Enable



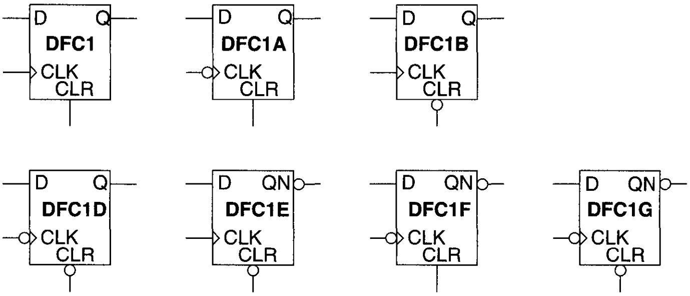
Mux Latches



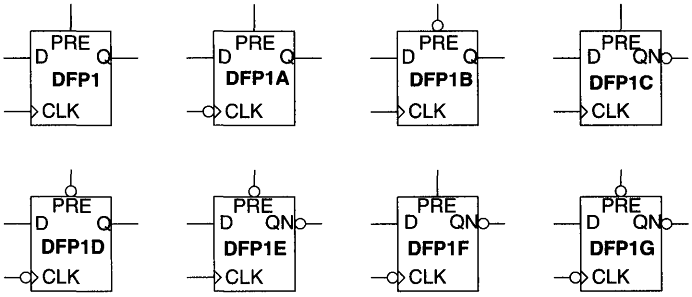
D-Type Flip-Flops



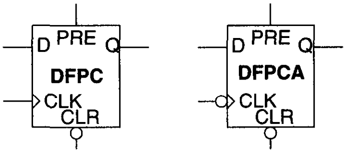
D-Type Flip-Flops with Clear



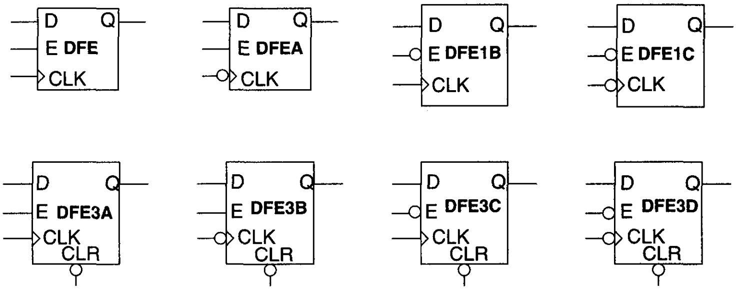
D-Type Flip-Flops with Preset



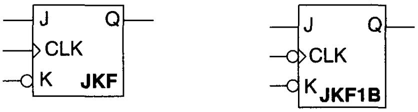
D-Type Flip-Flops with Preset and Clear



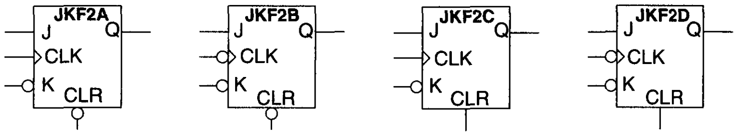
D-Type Flip-Flops with Enable



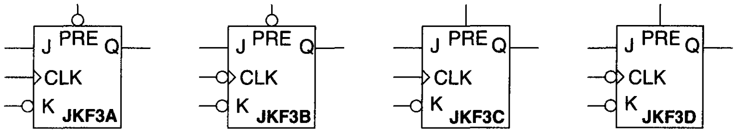
JK Flip-Flops



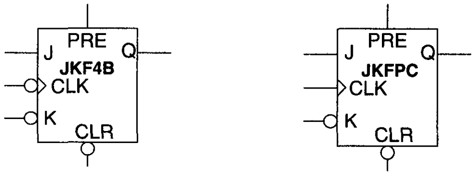
JK Flip-Flops with Clear



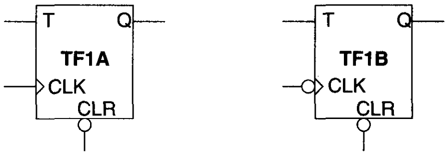
JK Flip-Flops with Preset



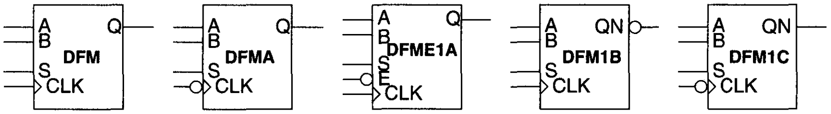
JK Flip-Flops with Preset and Clear



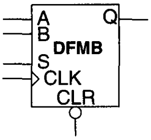
Toggle Flip-Flops

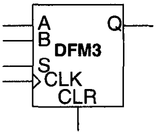


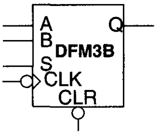
Mux Flip-Flops

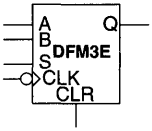


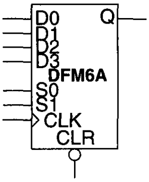
Mux Flip-Flops with Clear

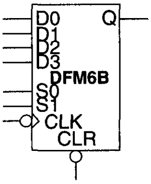


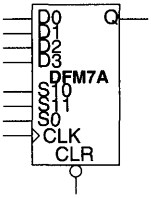


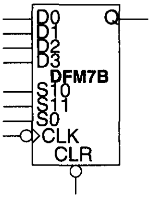




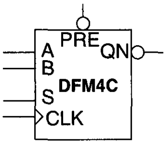


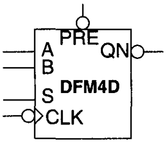




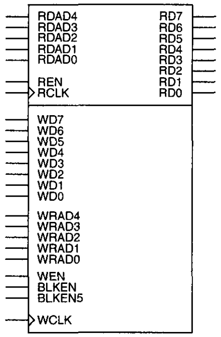


Mux Flip-Flops with Preset

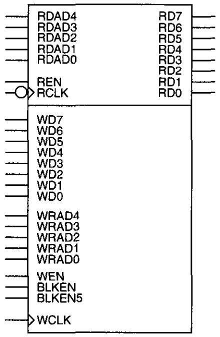




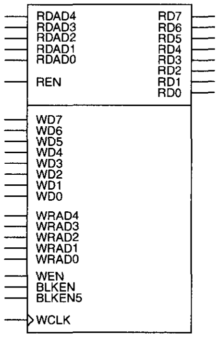
Special Macros for the 3200DX Family only



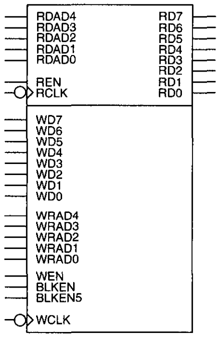
RAM8RR



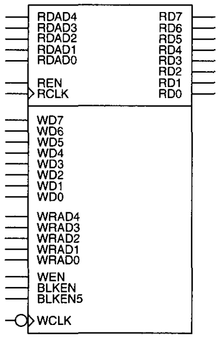
RAM8RF



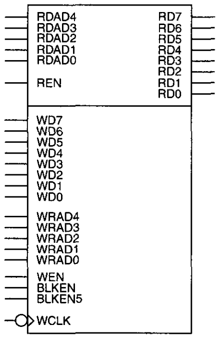
RAM8RA



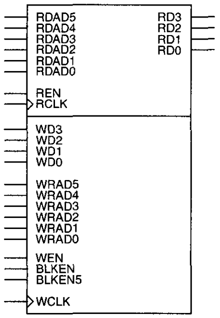
RAM8FF



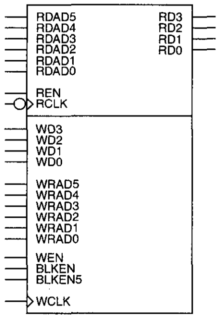
RAM8FR



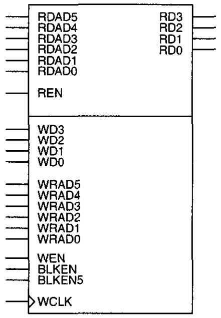
RAM8FA



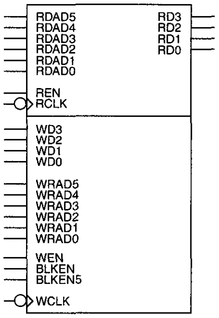
RAM4RR



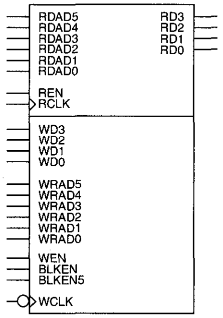
RAM4RF



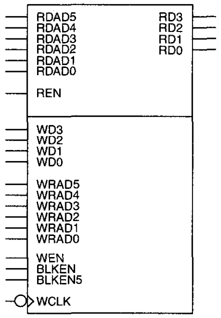
RAM4RA



RAM4FF

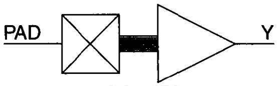


RAM4FR

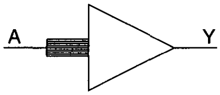


RAM4FA

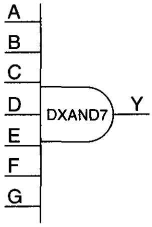
Special Macros for the 3200DX Family only (continued)



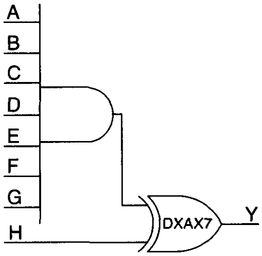
QCLKBUF



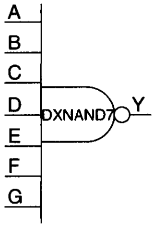
QCLKINT



DXAND7



DXAX7



DXNAND7

Integrator Series Macro Library – Tables of Hard, Soft, TTL, and ACTgen Macros

Hard Macros—Combinatorial

			Modules	
Function	Macro	Description	S	C
Combinatorial Logic Module	CM8	Combinational Module (Full 1200XL and 3200DX Logic Module)		1
Sequential Logic Module	DFM7A	4-input D-Type Flip-Flop with Multiplexed Data, active low Clear, and active high clock	1	
	DFM7B	4-input D-Type Flip-Flop with Multiplexed Data, active low Clear, and clock	1	
Adder	FA1A	1-bit adder, carry in and carry out active low, A-input active low		2
	FA1B	1-bit adder, carry in and carry out active low		2
	FA2A	2-bit adder, carry in and carry out active low, A0 and A1 inputs active low		2
	HA1	Half-Adder		2
	HA1A	Half-Adder with active low A-input		2
	HA1B	Half-Adder with active low carry out and sum		2
	HA1C	Half-Adder with active low carry out		2
AND	AND2	2-input AND		1
	AND2A	2-input AND with active low A-input		1
	AND2B	2-input AND with active low inputs		1
	AND3	3-input AND		1
	AND3A	3-input AND with active low A-input		1
	AND3B	3-input AND with active low A- and B-inputs		1
	AND3C	3-input AND with active low inputs		1
	AND4	4-input AND		1
	AND4A	4-input AND with active low A-input		1
	AND4B	4-input AND with active low A- and B-inputs		1
	AND4C	4-input AND with active low A-, B-, and C-inputs		1
	AND4D	4-input AND with active low inputs		2
	AND5B	5-input AND with active low A- and B-inputs		1
AND-OR	AO1	3-input AND-OR		1
	AO10	5-input AND-OR-AND		1
	AO11	3-input AND-OR		1
	AO1A	3-input AND-OR with active low A-input		1
	AO1B	3-input AND-OR with active low C-input		1
	AO1C	3-input AND-OR with active low A- and C-inputs		1
	AO1D	3-input AND-OR with active low A- and B-inputs		1
	AO1E	3-input AND-OR with active low inputs		1

Hard Macros—Combinatorial (continued)

			Modules	
Function	Macro	Description	S	C
AND-OR	AO2	4-input AND-OR		1
	AO2A	4-input AND-OR with active low A-input		1
	AO2B	4-input AND-OR with active low A- and B-inputs		1
	AO2C	4-input AND-OR with active low A- and C-inputs		1
	AO2D	4-input AND-OR with active low A-, B-, and C-inputs		1
	AO2E	4-input AND-OR with active low inputs		1
	AO3	4-input AND-OR		1
	AO3A	4-input AND-OR		1
	AO3B	4-input AND-OR		1
	AO3C	4-input AND-OR		1
	AO4A	4-input AND-OR		1
	AO5A	4-input AND-OR		1
	AO6	2-wide 4-input AND-OR		1
	AO6A	2-wide 4-input AND-OR with active low D-input		1
	AO7	5-input AND-OR		1
	AO8	5-input AND-OR with active low C- and D-inputs		1
	AO9	5-input AND-OR		1
	AOI1	3-input AND-OR-INVERT		1
	AOI1A	3-input AND-OR-INVERT with active low A-input		1
	AOI1B	3-input AND-OR-INVERT with active low C-input		1
	AOI1C	3-input AND-OR-INVERT with active low A- and B-inputs		1
	AOI1D	3-input AND-OR-INVERT with active low inputs		1
	AOI2A	4-input AND-OR-INVERT with active low A-input		1
	AOI2B	4-input AND-OR-INVERT with active low A- and C-inputs		1
	AOI3A	4-input AND-OR-INVERT with active low inputs		1
	AOI4	2-wide 4-input AND-OR-INVERT		2
	AOI4A	2-wide 4-input AND-OR-INVERT with active low C-input		1
AND-XOR	AX1	3-input AND-XOR with active low A-input		1
	AX1A	3-input AND-XOR-INVERT with active low A-input		2
	AX1B	3-input AND-XOR with active low A- and B-inputs		1
	AX1C	3-input AND-XOR		1
Buffer	BUFF	Buffer with active high input and output		1
	BUFA	Buffer with active low input and output		1
Clock Net	CLKINT	Clock Net Interface	0	0
	GAND2	2-input AND Clock Net		1
	GMX4	4-to-1 Multiplexor Clock Net		1
	GNAND2	2-input NAND Clock Net		1
	GNOR2	2-input NOR Clock Net		1
	GOR2	2-input OR Clock Net		1
	GXOR2	2-input Exclusive OR Clock Net		1

Hard Macros—Combinatorial (continued)

			Modules	
Function	Macro	Description	S	C
Inverter	INV	Inverter with active low output		1
	INVA	Inverter with active low input		1
Majority	MAJ3	3-input complex AND-OR		1
MUX	MX2	2-to-1 Multiplexor		1
	MX2A	2-to-1 Multiplexor with active low A-input		1
	MX2B	2-to-1 Multiplexor with active low B-input		1
MUX	MX2C	2-to-1 Multiplexor with active low output		1
	MX4	4-to-1 Multiplexor		1
	MXC1	Boolean		2
	MXT	Boolean		2
NAND	NAND2	2-input NAND		1
	NAND2A	2-input NAND with active low A-input		1
	NAND2B	2-input NAND with active low inputs		1
	NAND3	3-input NAND		1
	NAND3A	3-input NAND with active low A-input		1
	NAND3B	3-input NAND with active low A- and B-inputs		1
	NAND3C	3-input NAND with active low inputs		1
	NAND4	4-input NAND		2
	NAND4A	4-input NAND with active low A-input		1
	NAND4B	4-input NAND with active low A- and B-inputs		1
	NAND4C	4-input NAND with active low A-, B-, and C-inputs		1
	NAND4D	4-input NAND with active low inputs		1
	NAND5C	5-input NAND with active low A-, B-, and C-inputs		1
NOR	NOR2	2-input NOR		1
	NOR2A	2-input NOR with active low A-input		1
	NOR2B	2-input NOR with active low inputs		1
	NOR3	3-input NOR		1
	NOR3A	3-input NOR with active low A-input		1
	NOR3B	3-input NOR with active low A- and B-inputs		1
	NOR3C	3-input NOR with active low inputs		1
	NOR4	4-input NOR		2
	NOR4A	4-input NOR with active low A-input		1
	NOR4B	4-input NOR with active low A- and B-inputs		1
	NOR4C	4-input NOR with active low A-, B-, and C-inputs		1
	NOR4D	4-input NOR with active low inputs		1
	NOR5C	5-input NOR with active low A-, B-, and C-inputs		1
OR	OR2	2-input OR		1
	OR2A	2-input OR with active low A-input		1
	OR2B	2-input OR with active low inputs		1
	OR3	3-input OR		1
	OR3A	3-input OR with active low A-input		1

Hard Macros—Combinatorial (continued)

			Modules	
Function	**Macro**	**Description**	**S**	**C**
OR	OR3B	3-input OR with active low A- and B-inputs		1
	OR3C	3-input OR with active low inputs		1
	OR4	4-input OR		1
	OR4A	4-input OR with active low A-input		1
	OR4B	4-input OR with active low A- and B-input		1
	OR4C	4-input OR with active low A-, B-, and C-inputs		1
	OR4D	4-input OR with active low inputs		2
	OR5B	5-input OR with active low A- and B-inputs		1
OR-AND	OA1	3-input OR-AND		1
	OA1A	3-input OR-AND with active low A-input		1
	OA1B	3-input OR-AND with active low C-input		1
	OA1C	3-input OR-AND with active low A- and C-inputs		1
	OA2	2-wide 4-input OR-AND		1
	OA2A	2 wide 4-input OR-AND with active low A-input		1
	OA3	4-input OR-AND		1
	OA3A	4-input OR-AND with active low C-input		1
	OA3B	4-input OR-AND with active low A- and C-inputs		1
	OA4	4-input OR-AND		1
	OA4A	4-input OR-AND with active low C-input		1
	OA5	4-input complex OR-AND		1
	OAI1	3-input OR-AND-INVERT		1
	OAI2A	4-input OR-AND-INVERT with active low D-input		1
	OAI3	4-input OR-AND-INVERT		1
	OAI3A	4-input OR-AND-INVERT with active low C- and D-inputs		1
XNOR	XNOR2	2-input XNOR		1
XNOR-AND	XA1A	3-input XNOR-AND		1
XNOR-OR	XO1A	3-input XNOR-OR		1
XOR	XOR2	2-input XOR		1
XOR-AND	XA1	3-input XOR-AND		1
XOR-OR	XO1	3-input XOR-OR		1

Hard Macros—Sequential

			Modules	
Function	Macro	Description	S	C
D-Type	DF1	D-Type Flip-Flop	1	
	DF1A	D-Type Flip-Flop with active low output	1	
	DF1B	D-Type Flip-Flop with active low clock	1	
	DF1C	D-Type Flip-Flop with active low clock and output	1	
	DFC1	D-Type Flip-Flop with active high Clear	1	1
	DFC1A	D-Type Flip-Flop with active high Clear and active low clock	1	1
	DFC1B	D-Type Flip-Flop with active low Clear	1	
	DFC1D	D-Type Flip-Flop with active low Clear and clock	1	
	DFE	D-Type Flip-Flop with active high Enable	1	
	DFE1B	D-Type Flip-Flop with active low Enable	1	
	DFE1C	D-Type Flip-Flop with active low Enable and clock	1	
	DFE3A	D-Type Flip-Flop with Enable and active low Clear	1	
	DFE3B	D-Type Flip-Flop with Enable and active low Clear and clock	1	
	DFE3C	D-Type Flip-Flop with active low Enable and Clear	1	
	DFE3D	D-Type Flip-Flop with active low Enable, Clear, and clock	1	
	DFEA	D-Type Flip-Flop with Enable and active low clock	1	
	DFM	2-input D-Type Flip-Flop with Multiplexed Data	1	
	DFM1B	2-input D-Type Flip-Flop with Multiplexed Data and active low output	1	
	DFM1C	2-input D-Type Flip-Flop with Multiplexed Data and active low clock and output	1	
	DFM3	2-input D-Type Flip-Flop with Multiplexed Data and Clear	1	1
	DFM3B	2-input D-Type Flip-Flop with Multiplexed Data and active low Clear and clock	1	
	DFM3E	2-input D-Type Flip-Flop with Multiplexed Data, Clear, and active low clock	1	1
	DFM4C	2-input D-Type Flip-Flop with Multiplexed Data and active low Preset and output	1	
	DFM4D	2-input D-Type Flip-Flop with Multiplexed Data and active low Preset, clock, and output	1	
	DFM6A	4-input D-Type Flip-Flop with Multiplexed Data, active low Clear, and active high clock	1	
	DFM6B	4-input D-Type Flip-Flop with Multiplexed Data, active low Clear, and clock	1	
	DFMA	2-input D-Type Flip-Flop with Multiplexed Data and active low clock	1	
	DFMB	2-input D-Type Flip-Flop with Multiplexed Data and active low Clear	1	
	DFME1A	2-input D-Type Flip-Flop with Multiplexed Data and active low Enable	1	
	DFP1	D-Type Flip-Flop with active high Preset		2
	DFP1A	D-Type Flip-Flop with active high Preset and active low clock		2
	DFP1B	D-Type Flip-Flop with active low Preset		2
	DFP1C	D-Type Flip-Flop with active high Preset and active low output	1	1
	DFP1D	D-Type Flip-Flop with active low Preset and clock		2
	DFP1E	D-Type Flip-Flop with active low Preset and output	1	
	DFP1F	D-Type Flip-Flop with active high Preset and active low clock and output	1	1
	DFP1G	D-Type Flip-Flop with active low Preset, clock, and output	1	
	DFPC	D-Type Flip-Flop with active high Preset, active low Clear, and active high clock		2
	DFPCA	D-Type Flip-Flop with active high Preset and active low Clear and clock		2

Hard Macros—Sequential (continued)

			Modules	
Function	Macro	Description	S	C
J-K Type	JKF	JK Flip-Flop with active low K-input	1	
	JKF1B	JK Flip-Flop with active low clock and K-input	1	
	JKF2A	JK Flip-Flop with active low Clear and K-input	1	
	JKF2B	JK Flip-Flop with active low Clear, clock, and K-input	1	
	JKF2C	JK Flip-Flop with active high Clear and active low K-input	1	1
	JKF2D	JK Flip-Flop with active high Clear and active low clock and K-input	1	1
T-Type	TF1A	T-Type Flip-Flop with active low Clear	1	
	TF1B	T-Type Flip-Flop with active low Clear and clock	1	
Latch	DL1	Data Latch	1	
	DL1A	Data Latch with active low output	1	
	DL1B	Data Latch with active low clock	1	
	DL1C	Data Latch with active low clock and output	1	
	DLC	Data Latch with active low Clear	1	
	DLC1	Data Latch with active high Clear		1
	DLC1A	Data Latch with active high Clear and active low clock		1
	DLC1F	Data Latch with active high Clear and active low output		1
	DLC1G	Data Latch with active high Clear and active low clock and output		1
	DLCA	Data Latch with active low Clock and Clear	1	
	DLE	Data Latch with active high Enable	1	
	DLE1D	Data Latch with active high Enable and clock and active low input and output	1	
	DLE2B	Data Latch with active low Enable, Clear, and clock	1	
	DLE2C	Data Latch with active low Enable and clock and active high Clear		1
	DLE3B	Data Latch with active low Enable and clock and active low Preset		1
	DLE3C	Data Latch with active low Enable, Preset, and clock		1
	DLEA	Data Latch with active low Enable and active high clock	1	
	DLEB	Data Latch with active high Enable and active high clock	1	
	DLEC	Data Latch with active low Enable and clock	1	
	DLM	2-input Data Latch with Multiplexed Data	1	
	DLM3	4-input Data Latch with Multiplexed Data	1	
	DLM3A	4-input Data Latch with Multiplexed Data and active low clock	1	
	DLM4	Data Latch with Multiplexed Data	1	
	DLM4A	Data Latch with Multiplexed Data	1	
	DLMA	2-input Data Latch with Multiplexed Data and active low clock	1	
	DLME1A	2-input Data Latch with Multiplexed Data and Enable and active low clock	1	
	DLP1	Data Latch with active high Preset and clock		1
	DLP1A	Data Latch with active high Preset and active low clock		1
	DLP1B	Data Latch with active low Preset and active high clock		1
	DLP1C	Data Latch with active low Preset and clock		1
	DLP1D	Data Latch with active low Preset and output and active high clock	1	
	DLP1E	Data Latch with active low Preset, clock, and output	1	

Hard Macros (3200DX Family Only)

Function	Macro	Description
SRAM	DXRAM8RR	32x8 Rising WCLK Rising RCLK
	DXRAM8RF	32x8 Rising WCLK Falling RCLK
	DXRAM8RA	32x8 Rising WCLK Async read
	DXRAM8FR	32x8 Falling WCLK Rising RCLK
	DXRAM8FF	32x8 Falling WCLK Falling RCLK
	DXRAM8FA	32x8 Falling WCLK Async read
	DXRAM4RR	64x4 Rising WCLK Rising RCLK
	DXRAMRF	64x4 Rising WCLK Falling RCLK
	DXRAM4RA	64x4 Rising WCLK Async read
	DXRAM4FR	64x4 Falling WCLK Rising RCLK
	DXRAM4FF	64x4 Falling WCLK Falling RCLK
	DXRAM4FA	64x4 Falling WCLK Async read
Clock Net	QCLKBUF	Quadrant Clock Net Interface
	QCLKINT	Quadrant Clock Net Interface
Wide Decode	DXAND7	7-input AND Gate
	DXAX7	8-input AND/Exclusive-OR Gate
	DXNAND7	7-input NAND Gate

Note: *The schematic for the wide decodes are in the Macro Library Guide 1995 in (Hard-94)–(Hard-95).*

Input/Output Macros

Function	Macro	Description	I/O Modules
Buffer	IBDL	Input Buffer with Latch Clock	1
	INBUF	Input Buffer	1
	OBHS	Output Buffer, High Slew	1
	OUTBUF	Output Buffer, High Slew	1
Bidirectional	BBHS	Bidirectional Buffer, High Slew	1
	BBDLHS	Bidirectional with Input Latch and Output Latch	1
	BIBUF	Bidirectional Buffer, High Slew (with hidden buffer at Y pin)	1
	CLKBIBUF	Bidirectional with Input Dedicated to Clock Network	1
Input	CLKBUF	Input for Dedicated Routed Clock Network	1
Output	DBDLKS	Output Buffer with Latch	1
	OBHS	Output Buffer	1
	TBHS	Tristate output, High Slew	1
	TRIBUFF	Tristate output, High Slew	1

Soft Macros

			Maximum Logic	Modules	
Function	Macro	Description	Levels	S	C
Adder	FADD10	10-bit adder	3		56
	FADD12	12-bit adder	4		9
	FADD16	16-bit adder	5		97
	FADD8	8-bit adder	4		44
	FADD9	9-bit adder with active low carry out	3		49
	VAD16C	Very fast 16-bit adder, no Carry in	3		91
	VADC16C	Very fast 16-bit adder with Carry in	3		97
Comparator	ICMP4	4-bit Identity Comparator	2		5
	ICMP8	8-bit Identity Comparator	3		9
	MCMPC2	2-bit Magnitude Comparator with Enable	3		9
	MCMPC4	4-bit Magnitude Comparator with Enable	4		18
	MCMPC8	8-bit Magnitude Comparator with Enable	6		36
Counter	CNT4A	4-bit binary counter with load and clear	4	4	8
	CNT4B	4-bit binary counter with load, clear, carry-in, carry-out	4	4	7
	FCTD16C	Fast 16-bit Down Counter, parallel loadable	2	19	33
	FCTD8A	Fast 8-bit Down Counter, parallel loadable	1	10	18
	FCTD8B	Fast 8-bit Down Counter, parallel loadable	1	9	13
	FCTU16C	Fast 16-bit Up Counter, parallel loadable	2	19	31
	FCTU8A	Fast 8-bit Up Counter, parallel loadable	1	10	17
	FCTU8B	Fast 8-bit Up Counter, parallel loadable	1	9	12
	UDCNT4A	4-bit up/down counter with load, carry-in, and carry-out	5	4	13
	VCTD16C	Very fast 16-bit down counter, delay after load, registered control inputs	1	34	41
	VCTD2CP	2-bit down counter, prescaler, delay after load, used to build VCTD counters	1	5	2
	VCTD2CU	2-bit down counter, upper bits, delay after load, used to build VCTD counters	1	2	3
	VCTD4CL	4-bit down counter, lower bits, delay after load, used to build VCTD counters	1	4	7
	VCTD4CM	4-bit down counter, middle bits, delay after load, used to build VCTD counters	1	4	8
Decoder	DEC2X4	2-to-4 decoder	1		4
	DEC2X4A	2-to-4 decoder with active low outputs	1		4
	DEC3X8	3-to-8 decoder	1		8
	DEC3X8A	3-to-8 decoder with active low outputs	1		8
	DEC4X16A	4-to-16 decoder with active low outputs	2		20
	DECE2X4	2-to-4 decoder with enable	1		4
	DECE2X4A	2-to-4 decoder with enable and active low outputs	1		4
	DECE3X8	3-to-8 decoder with enable	2		11
	DECE3X8A	3-to-8 decoder with enable and active low outputs	2		11
Latch	DLC8A	Octal latch with clear active low 8-bit Data Latch with active low Clear	1	8	

Soft Macros (continued)

Function	Macro	Description	Maximum Logic Levels	Modules	
				S	C
	DLE8	Octal latch with enable 8-bit Data Latch with active high Enable	1	8	
	DLM8	Octal latch with multiplexed data 8-bit Data Latch with Multiplexed Data	1	8	
MUX	MX16	16-to-1 Multiplexor	2		5
	MX8	8-to-1 Multiplexor with active high output	2		3
	MX8A	8-to-1 Multiplexor with active low output	2		3
Multiplier	SMULT8	8-bit by 8-bit Multiplier			242
Shift Register	SREG4A	4-bit shift register with clear active low	1	4	
	SREG8A	8-bit shift register with clear active low	1	8	

Soft Macros—TTL Equivalent

Function	Macro	Description	Maximum Logic Levels	Modules	
				S	C
	TA00	2-input NAND	1		1
	TA02	2-input NOR	1		1
	TA04	Inverter	1		1
	TA07	Buffer	1		1
	TA08	2-input AND	1		1
	TA10	3-input NAND	1		1
	TA11	3-input AND	1		1
	TA138	3-to-8 decoder with enable and active low outputs	2		12
	TA139	2-to-4 decoder with active low enable and outputs	1		4
	TA150	16-to-1 multiplexor with active low enable	3		6
	TA151	8-to-1 multiplexor with enable and both active low and active high output	3		5
	TA153	4-to-1 multiplexor with active low enable	2		2
	TA154	4-to-16 decoder with active low outputs and select lines	2		22
	TA157	2-to-1 multiplexor with active low enable	1		1
	TA160	4-bit decade counter with active low clear and load	4	4	8
	TA161	4-bit binary counter with active low clear and load	3	4	6
	TA164	8-bit serial in, parallel out shift register, active low clear	1	8	
	TA169	4-bit Up/Down Counter	6	4	14
	TA174	hex D-type flip-flop with active low clear	1	6	
	TA175	quadruple D-type flip-flop with active low clear	1	4	
	TA181	ALU	5		36
	TA190	4-bit up/down decode counter with up/down mode	7	4	31
	TA191	4-bit up/down binary counter with up/down mode	7	4	30
	TA194	4-bit bidirectional universal shift register	1	4	4
	TA195	4-bit parallel-access shift register	1	4	1
	TA20	4-input NAND	1		2
	TA21	4-input AND	1		1

Soft Macros—TTL Equivalent (continued)

Function	Macro	Description	Maximum Logic Levels	Modules S	Modules C
	TA269	8-bit up/down binary counter	8	8	28
	TA27	3-input NOR	1		1
	TA273	octal register with clear	1	8	
	TA280	9-bit odd/even parity generator and checker	4		9
	TA32	2-input OR	1		1
	TA377	octal register with active low enable	1	8	
	TA40	4-input NAND	1		2
	TA42	4 to 10 decoder	1		10
	TA51	AND-OR-Invert	1		2
	TA54	4-wide 2-input AND-OR-Invert	2		5
	TA55	2-wide 4-input AND-OR-Invert	2		3
	TA688	8-bit identity comparator	3		9
	TA86	2-input exclusive OR	1		1

ACTgen Macro Timing Characteristics
(Worst-Case Commercial Conditions, Prelayout Timing Delays)

Sequential Macros*

Macro Type**	# Bits	Area (Seq/Total)	Levels	STD (MHz)	–1 (MHz)	–2 (MHz)	–3 (MHz)
Standard Counter	4	4/9	2	74	86.6	98.4	110
(load and count enable)	8	8/20	3	47	55	62.5	70
	16	16/46	3	45	52.7	60	67
	32	32/93	3	39	45.6	52	58
Pre-Scaled Counter	16	19/46	1	119	139	158	177
(load or count enable)	32	39/105	1	119	139	158	177
Shift Register	4	4/4	1	159	186	211.5	237
(SISO and SIPO)	8	8/10	1	159	186	211.5	237
	16	16/21	1	159	186	211.5	237
	32	32/42	1	159	186	211.5	237
Shift Register	4	4/4	1	159	186	211.5	237
(PISO and PIPO)	8	8/12	1	159	186	211.5	237
	16	16/24	1	159	186	211.5	237
	32	32/48	1	159	186	211.5	237
Accumulator	4	4/15	3	54	63.2	72	80.5
	8	8/35	4	39	45.6	52	58
	16	16/92	4	33	38.6	44	49.2
	32	32/191	5	25.5	30	34	38

** Delays based on DX3265 die*

***Macro type is based on default ACTgen parameters*

ACTgen Macro Timing Characteristics
(Worst-Case Commercial Conditions, Prelayout Timing Delays)

Combinatorial Macros*

Macro Type**	# Bits	Area (Comb)	Levels	STD (ns)	–1 (ns)	–2 (ns)	–3 (ns)
Adder	4	11	2	12	10.2	9	8
	8	27	3	19.5	16.6	14.6	13
	16	73	3	24	20.2	18	16
	32	155	4	33	28	25	22
Subtracter	4	15	3	18	15.4	13.6	12
	8	34	4	25.3	21.5	19	17
	16	89	4	29	24.6	21.7	19.4
	32	187	5	38.2	32.5	28.7	25.6
Magnitude Comparator	4	7	3	16	13.6	12	10.7
	8	14	4	21.8	18.5	16.4	14.6
	16	30	4	23.2	19.7	17.4	15.5
	32	61	5	29	24.7	21.8	19.4
Equality Comparator	4	5	2	10.2	8.7	7.7	6.8
	8	9	3	16	13.6	12	10.7
	16	19	3	16	13.6	12	10.7
	32	39	4	20.4	17.3	15.3	13.7
Decoder	4:16	24	2	14.4	12.3	10.8	9.7
	5:32	44	2	17.2	14.6	12.9	11.5
AND/OR	8	3	2	10.2	8.7	7.65	6.8
	16	5	2	10.2	8.7	7.65	6.8
	32	12	3	15.3	13	11.5	10.3
Multiplexer	8:1	3	2	11	9	8	7.3
	16:1	5	2	12	10.5	9	8.2
	32:1	15	3	23	20	17.4	15.5
Parity	8	7	3	15.3	13	11.5	10.3
	16	15	3	16.7	14.2	12.5	11.2
	32	31	4	21.8	18.5	16.4	14.6

** Delays based on DX3265 die*

***Macro type is based on default ACTgen parameters*

Accelerator Series Hard Macro Library – Graphical Symbols

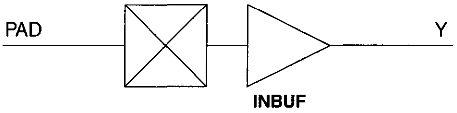
The following illustrations show all the available hard macros. The Actel *Macro Library Guide* contains module count, combinability, and pin loading information of each hard macro. It also includes a complete truth table for each macro.

Most of the Accelerator Series (ACT 3) hard macros are implemented by a single logic module. The following Accelerator Series hard macros require more than one logic module to implement:

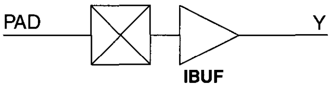
- Two-module hard macro gates: AND4D, AOI4, AX1A, MXC1, MXT, NAND4, NOR4, OAI3, OR4D
- Two-module adders: FA1A, FA1B, FA2A, HA1, HA1A, HA1B, HA1C
- Two-module latches: DL2A, DL2B, DL2C, DL2D, DLE2A, DLE3A, DLM2A
- Two-module flip-flops: DFC1, DFC1A, DFC1E, DFC1G, DFM3, DFM3E, DFP1, DFP1A, DFP1B, DFP1C, DFP1D, DFP1F, DFPC, DFPCA, JKF2C, JKF2D, JKF3A, JKF3B, JKF3C, JKF3D, JKF4B, JKFPC

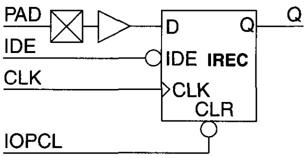
Two-module hard macro gates have a “2” displayed on some input pins. This indicates that the input to output path has two levels of logic delay for these input pins only. Also, the full adders have a “2” on the “S” output pins. This indicates that there are two levels of logic delay from the input pins to the “S” output pin. Refer to the Accelerator Series Timing Characteristics for detailed timing information.

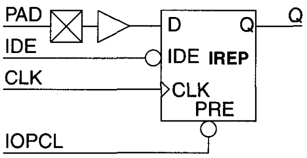
Input Buffers



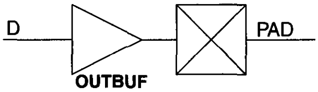
Input Buffer



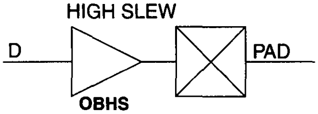




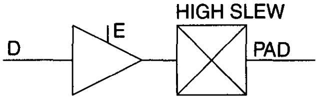
Output Buffers



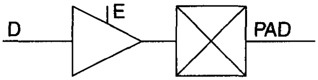
OUTBUF



OBHS



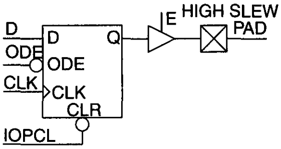
TBHS



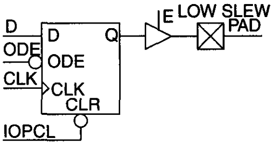
TRIBUFF

Output Buffers

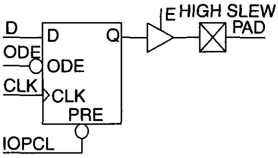
ORECTH



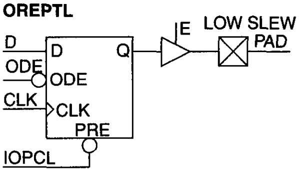
ORECTL



OREPTH

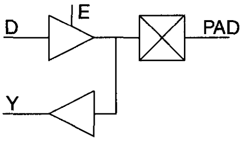


OREPTL

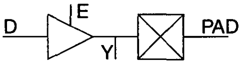


Bidirectional Buffers

BBHS

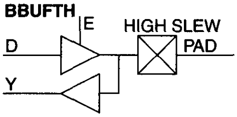


BIBUF

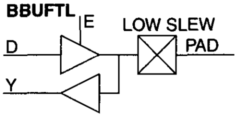


Bidirectional Buffers

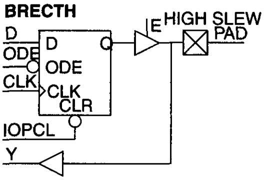
BBUFTH



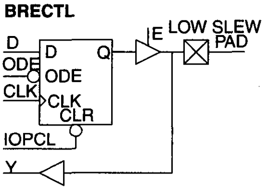
BBUFTL



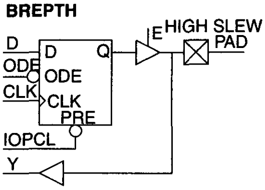
BRECTH



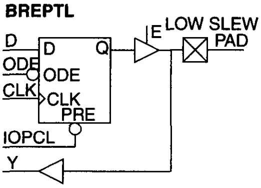
BRECTL



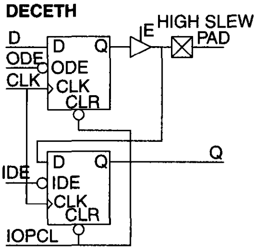
BREPTH



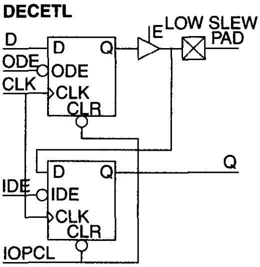
BREPTL



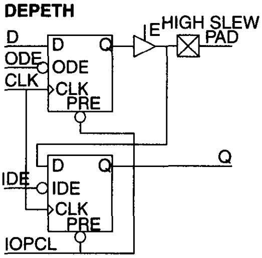
DECETH



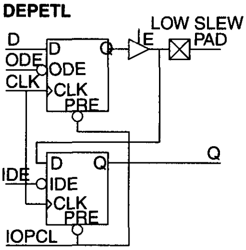
DECETL



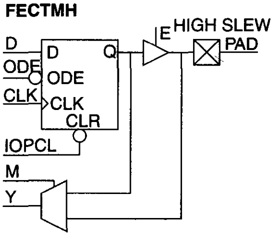
DEPETH



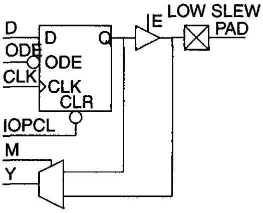
DEPETL



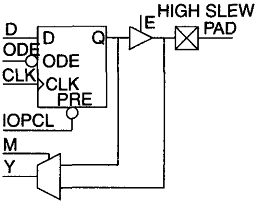
FECTMH



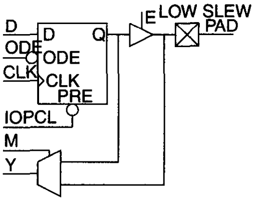
FECTML



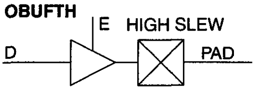
FEPTMH



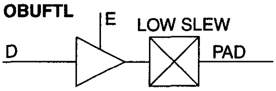
FEPTML



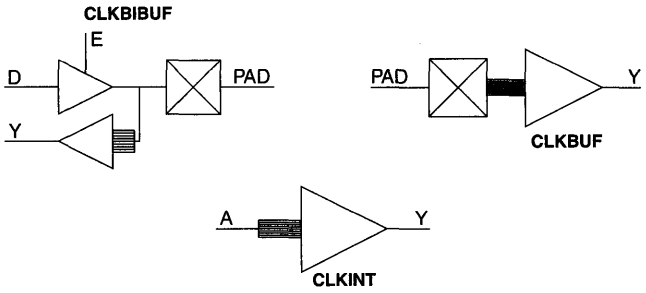
OBUFTH



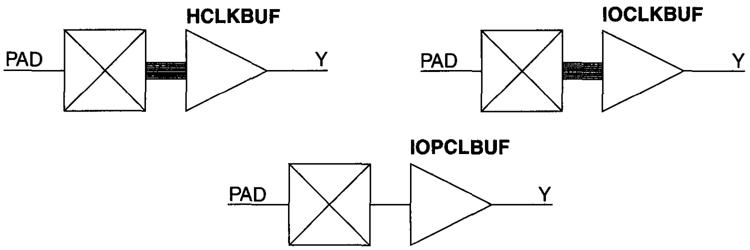
OBUFTL



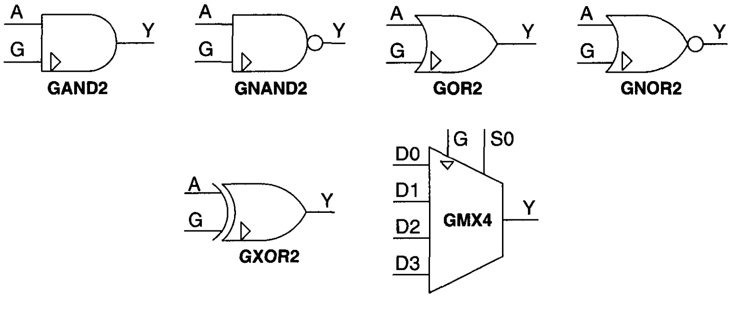
Routed Clock Buffers



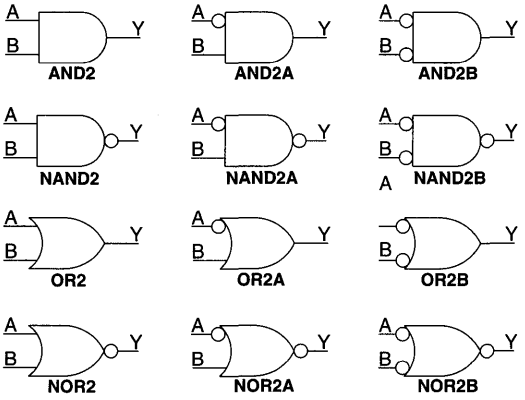
Clock (Dedicated Network) Buffers



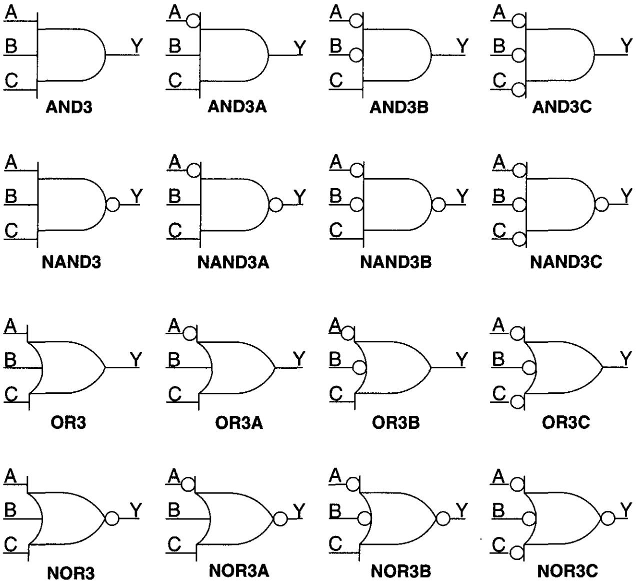
CLKBUF Interface Macros



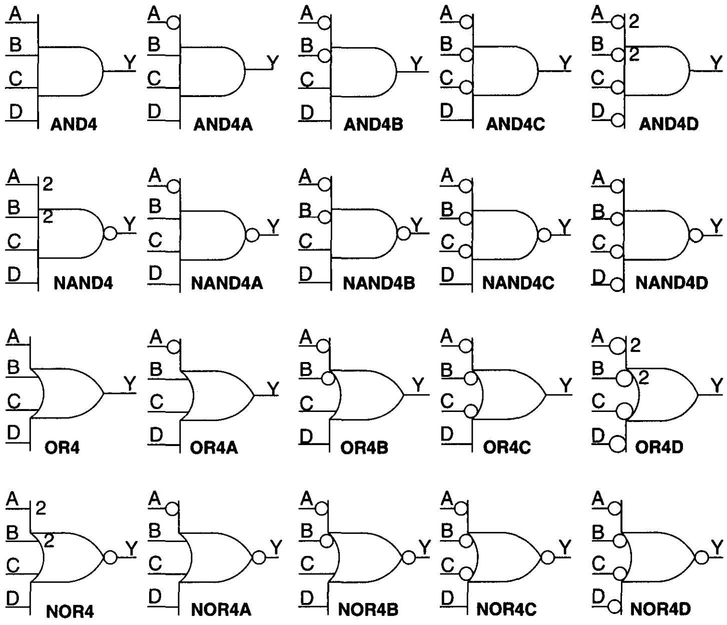
2-Input Gates



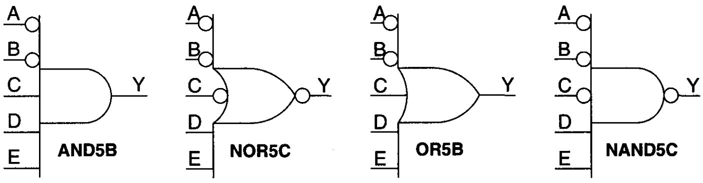
3-Input Gates



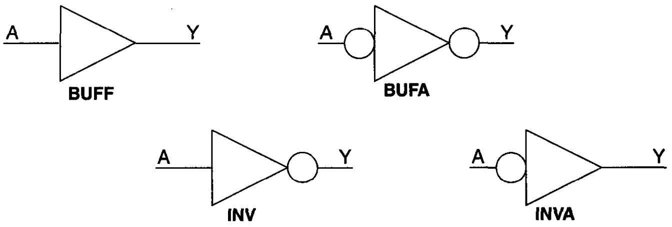
4-Input Gates



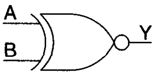
5-Input Gates



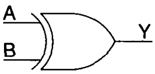
Buffers



XOR Gates

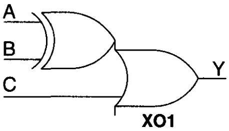


XNOR2

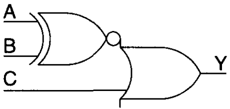


XOR2

XOR-OR Gates

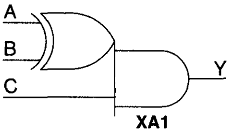


XO1

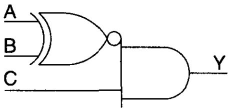


XO1A

XOR-AND Gates

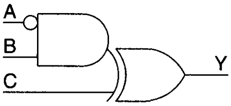


XA1

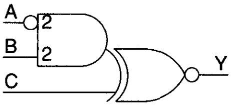


XA1A

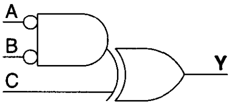
AND-XOR Gates



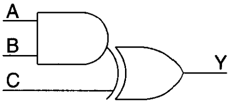
AX1



AX1A

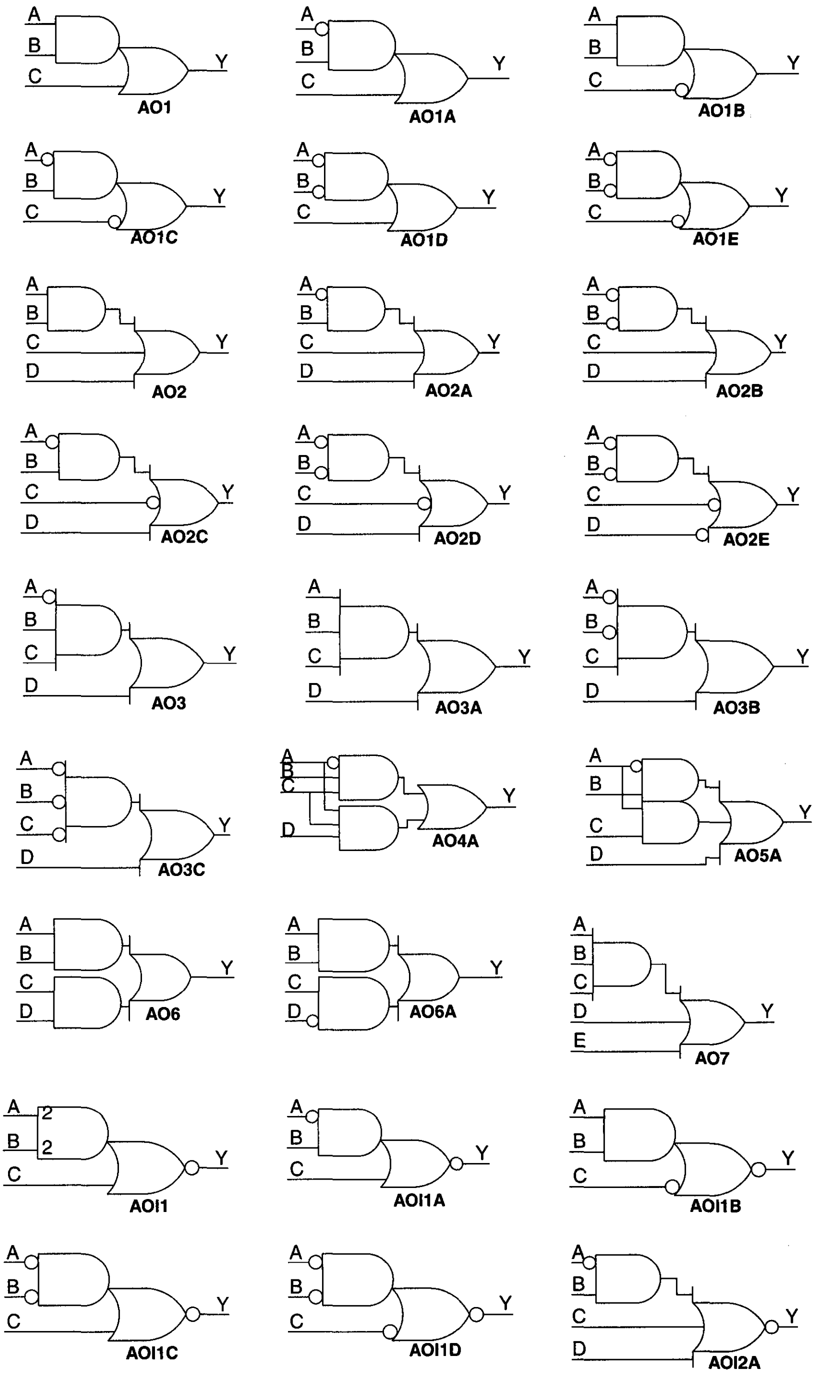


AX1B

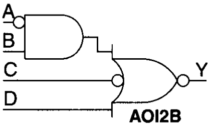


AX1C

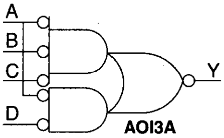
AND-OR Gates


A
B
C
Y
AO1
AO1A
AO1B
AO1C
AO1D
AO1E
D
AO2
AO2A
AO2B
AO2C
AO2D
AO2E
AO3
AO3A
AO3B
AO3C
AO4A
AO5A
AO6
AO6A
E
AO7
2
2
AOI1
AOI1A
AOI1B
AOI1C
AOI1D
AOI2A

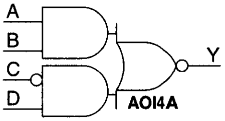
AND-OR Gates (continued)



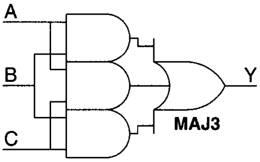
AOI2B



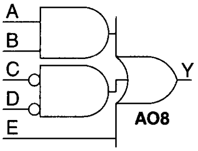
AOI3A



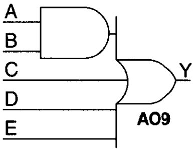
AOI4A



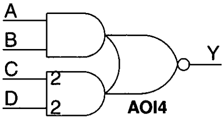
MAJ3



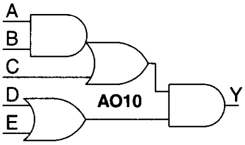
AO8



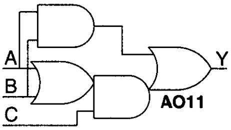
AO9



AOI4

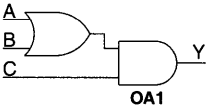


AO10

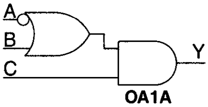


AO11

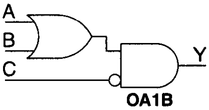
OR-AND Gates



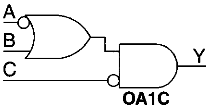
OA1



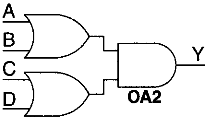
OA1A



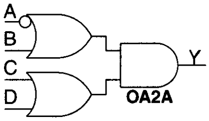
OA1B



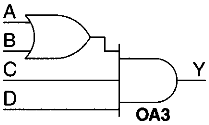
OA1C



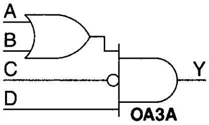
OA2



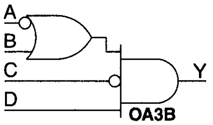
OA2A



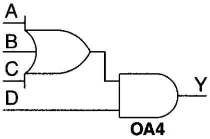
OA3



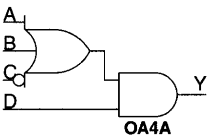
OA3A



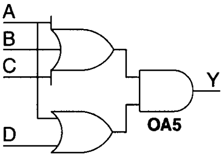
OA3B



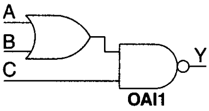
OA4



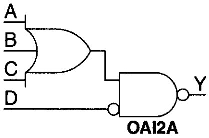
OA4A



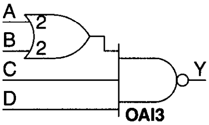
OA5



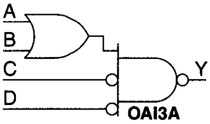
OAI1



OAI2A

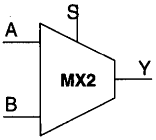


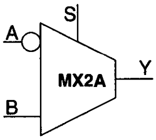
OAI3

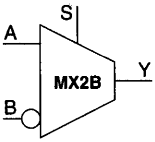


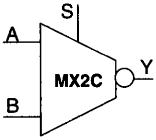
OAI3A

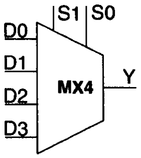
Multiplexors

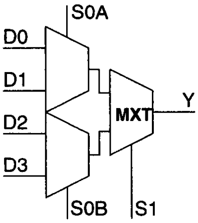


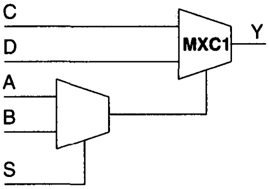




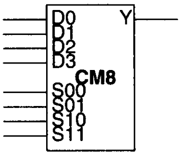


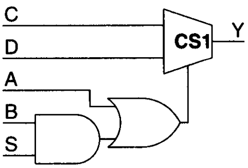


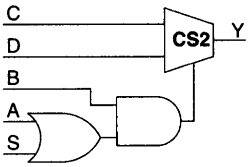




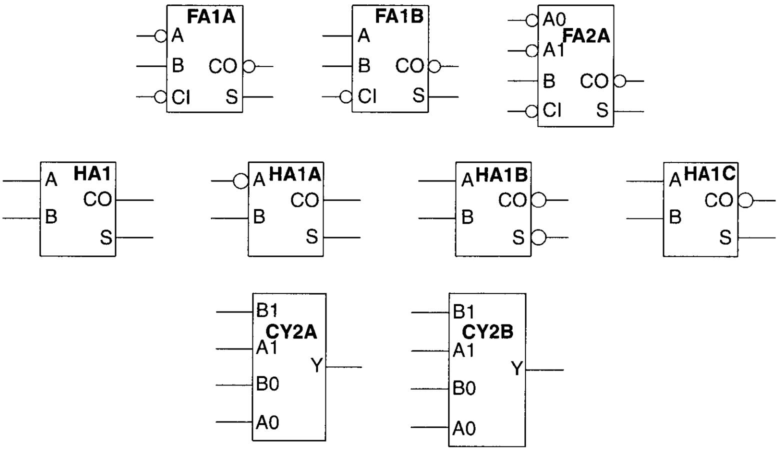
Combinatorial



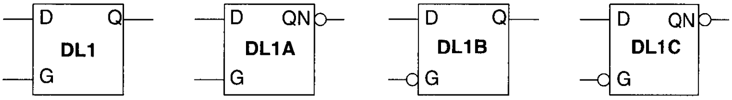




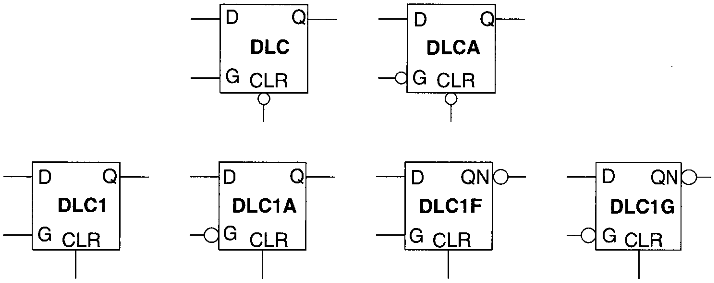
Adders



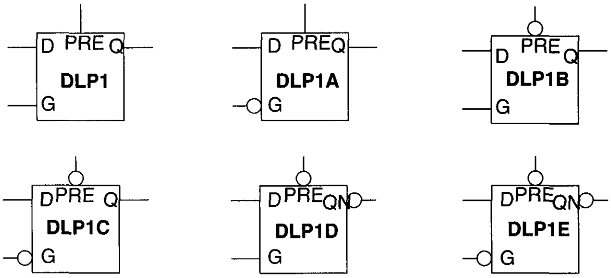
D-Latches



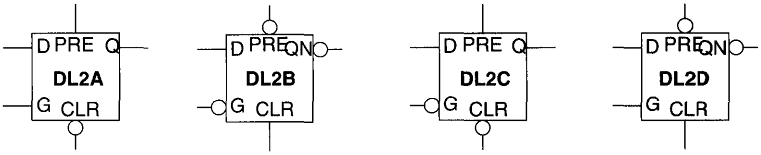
D-Latches with Clear



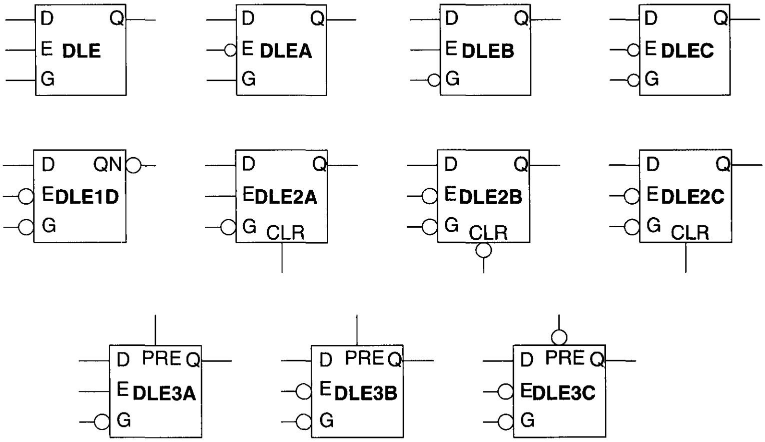
D-Latches with Preset



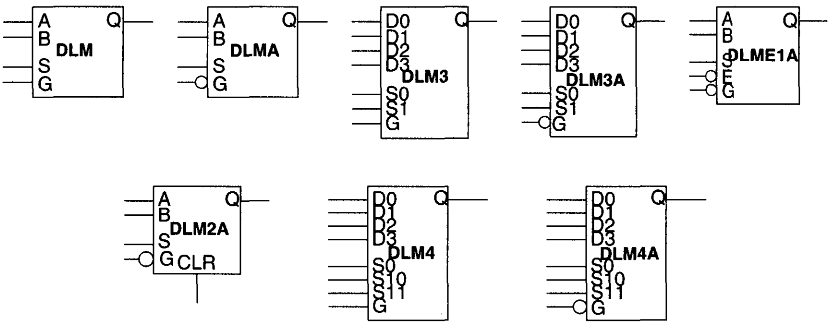
D-Latches with Preset and Clear



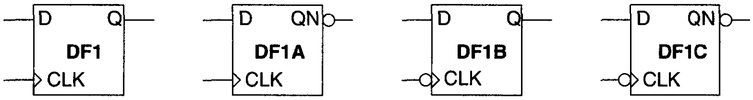
D-Latches with Enable



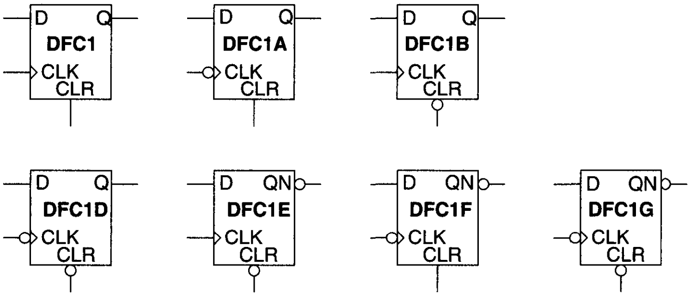
Mux Latches



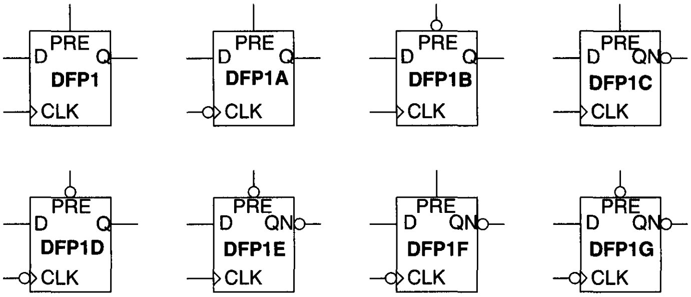
D-Type Flip-Flops



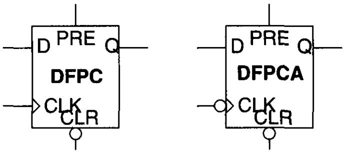
D-Type Flip-Flops with Clear



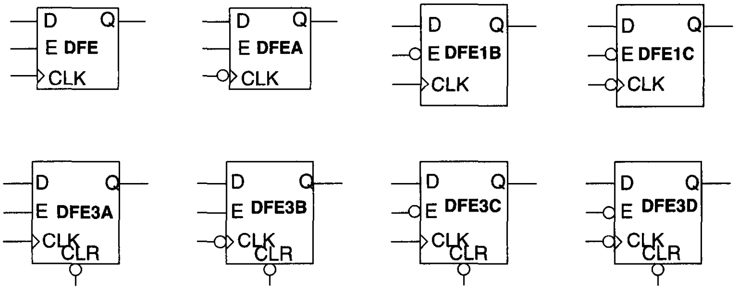
D-Type Flip-Flops with Preset



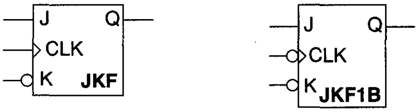
D-Type Flip-Flops with Preset and Clear



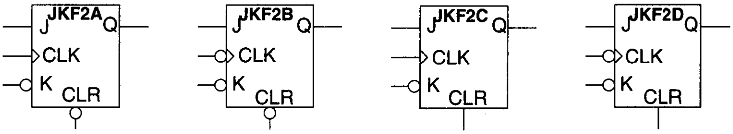
D-Type Flip-Flops with Enable



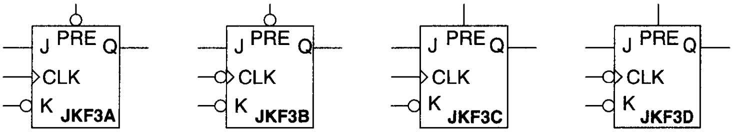
JK Flip-Flops



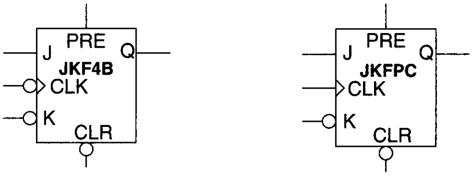
JK Flip-Flops with Clear



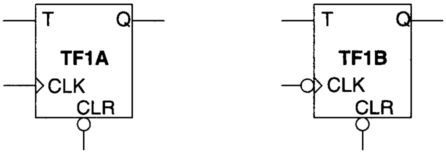
JK Flip-Flops with Preset



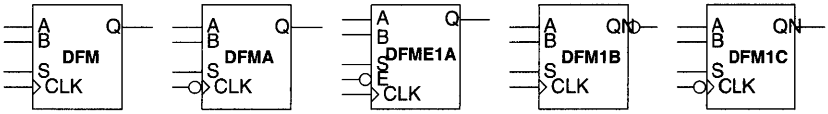
JK Flip-Flops with Preset and Clear



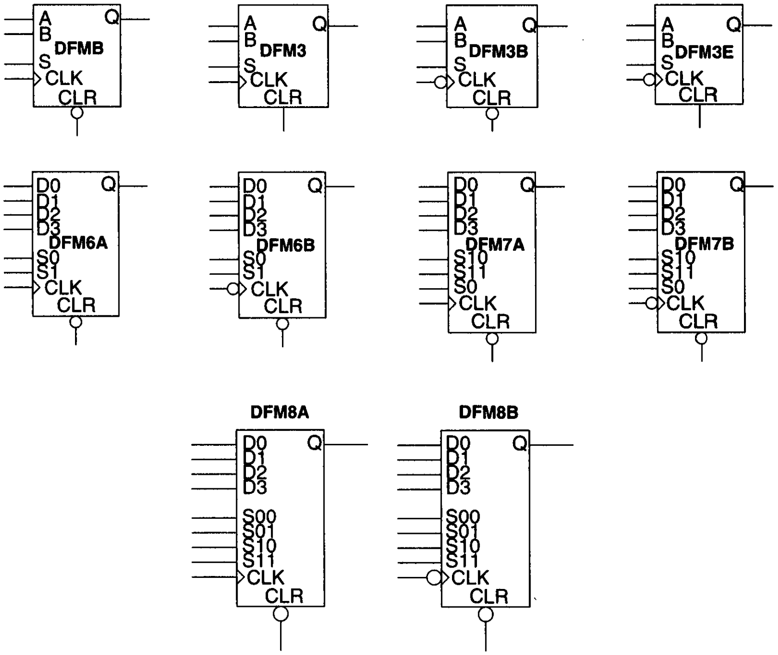
Toggle Flip-Flops



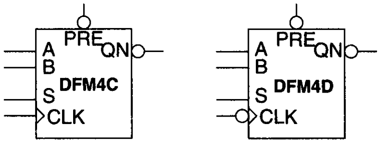
Mux Flip-Flops



Mux Flip-Flops with Clear



Mux Flip-Flops with Preset



Accelerator Series Macro Library – Tables of Hard, Soft, TTL, and ACTgen Macros

Hard Macros—Combinatorial

			Modules	
Function	**Macro**	**Description**	**S**	**C**
Combinatorial Logic Module	CM8	Combinational Module (Full ACT 3 Logic Module)		1
Sequential Logic Module	DFM8A	4-bit D-Type Flip-Flop with Multiplexed Data, active low Clear, and active high clock	1	
	DFM8B	4-bit D-Type Flip-Flop with Multiplexed Data, active low Clear, and active low clock	1	
Adder	FA1A	1-bit adder, carry in and carry out active low, A-input active low		2
	FA1B	1-bit adder, carry in and carry out active low		2
	FA2A	2-bit adder, carry in and carry out active low, A0 and A1 inputs active low		2
	HA1	Half-Adder		2
	HA1A	Half-Adder with active low A-input		2
	HA1B	Half-Adder with active low carry out and sum		2
	HA1C	Half-Adder with active low carry out		2
AND	AND2	2-input AND		1
	AND2A	2-input AND with active low A-input		1
	AND2B	2-input AND with active low inputs		1
	AND3	3-input AND		1
	AND3A	3-input AND with active low A-input		1
	AND3B	3-input AND with active low A- and B-inputs		1
	AND3C	3-input AND with active low inputs		1
	AND4	4-input AND		1
	AND4A	4-input AND with active low A-input		1
	AND4B	4-input AND with active low A- and B-inputs		1
	AND4C	4-input AND with active low A-, B-, and C-inputs		1
	AND4D	4-input AND with active low inputs		2
	AND5B	5-input AND with active low A- and B-inputs		1
AND-OR	AO1	3-input AND-OR		1
	AO10	5-input AND-OR-AND		1
	AO11	3-input AND-OR		1
	AO1A	3-input AND-OR with active low A-input		1
	AO1B	3-input AND-OR with active low C-input		1
	AO1C	3-input AND-OR with active low A- and C-inputs		1
	AO1D	3-input AND-OR with active low A- and B-inputs		1
	AO1E	3-input AND-OR with active low inputs		1

Hard Macros—Combinatorial (continued)

			Modules	
Function	Macro	Description	S	C
AND-OR	AO2	4-input AND-OR		1
	AO2A	4-input AND-OR with active low A-input		1
	AO2B	4-input AND-OR with active low A- and B-inputs		1
	AO2C	4-input AND-OR with active low A- and C-inputs		1
	AO2D	4-input AND-OR with active low A-, B-, and C-inputs		1
	AO2E	4-input AND-OR with active low inputs		1
	AO3	4-input AND-OR		1
	AO3A	4-input AND-OR		1
	AO3B	4-input AND-OR		1
	AO3C	4-input AND-OR		1
	AO4A	4-input AND-OR		1
	AO5A	4-input AND-OR		1
	AO6	2-wide 4-input AND-OR		1
	AO6A	2-wide 4-input AND-OR with active low D-input		1
	AO7	5-input AND-OR		1
	AO8	5-input AND-OR with active low C- and D-inputs		1
	AO9	5-input AND-OR		1
	AOI1	3-input AND-OR-INVERT		1
	AOI1A	3-input AND-OR-INVERT with active low A-input		1
	AOI1B	3-input AND-OR-INVERT with active low C-input		1
	AOI1C	3-input AND-OR-INVERT with active low A- and B-inputs		1
	AOI1D	3-input AND-OR-INVERT with active low inputs		1
	AOI2A	4-input AND-OR-INVERT with active low A-input		1
	AOI2B	4-input AND-OR-INVERT with active low A- and C-inputs		1
	AOI3A	4-input AND-OR-INVERT with active low inputs		1
	AOI4	2-wide 4-input AND-OR-INVERT		2
	AOI4A	2-wide 4-input AND-OR-INVERT with active low C-input		1
AND-XOR	AX1	3-input AND-XOR with active low A-input		1
	AX1A	3-input AND-XOR-INVERT with active low A-input		2
	AX1B	3-input AND-XOR with active low A- and B-inputs		1
	AX1C	3-input AND-XOR		1
Buffer	BUFF	Buffer, with active high input and output		1
	BUFA	Buffer, with active low input and output		1
Clock Net	CLKINT	Clock Net Interface	0	0
	GAND2	2-input AND Clock Net		1
	GMX4	4-to-1 Multiplexor Clock Net		1
	GNAND2	2-input NAND Clock Net		1
	GNOR2	2-input NOR Clock Net		1
	GOR2	2-input OR Clock Net		1
	GXOR2	2-input Exclusive OR Clock Net		1

Hard Macros—Combinatorial (continued)

			Modules	
Function	Macro	Description	S	C
Inverter	INV	Inverter with active low output		1
	INVA	Inverter with active low input		1
Majority	MAJ3	3-input complex AND-OR		1
MUX	MX2	2-to-1 Multiplexor		1
	MX2A	2-to-1 Multiplexor with active low A-input		1
	MX2B	2-to-1 Multiplexor with active low B-input		1
MUX	MX2C	2-to-1 Multiplexor with active low output		1
	MX4	4-to-1 Multiplexor		1
	MXC1	Boolean		2
	MXT	Boolean		2
NAND	NAND2	2-input NAND		1
	NAND2A	2-input NAND with active low A-input		1
	NAND2B	2-input NAND with active low inputs		1
	NAND3	3-input NAND		1
	NAND3A	3-input NAND with active low A-input		1
	NAND3B	3-input NAND with active low A- and B-inputs		1
	NAND3C	3-input NAND with active low inputs		1
	NAND4	4-input NAND		2
	NAND4A	4-input NAND with active low A-input		1
	NAND4B	4-input NAND with active low A- and B-inputs		1
	NAND4C	4-input NAND with active low A-, B-, and C-inputs		1
	NAND4D	4-input NAND with active low inputs		1
	NAND5C	5-input NAND with active low A-, B-, and C-inputs		1
NOR	NOR2	2-input NOR		1
NOR	NOR2A	2-input NOR with active low A-input		1
	NOR2B	2-input NOR with active low inputs		1
	NOR3	3-input NOR		1
	NOR3A	3-input NOR with active low A-input		1
	NOR3B	3-input NOR with active low A- and B-inputs		1
	NOR3C	3-input NOR with active low inputs		1
	NOR4	4-input NOR		2
	NOR4A	4-input NOR with active low A-input		1
	NOR4B	4-input NOR with active low A- and B-inputs		1
	NOR4C	4-input NOR with active low A-, B-, and C-inputs		1
	NOR4D	4-input NOR with active low inputs		1
	NOR5C	5-input NOR with active low A-, B-, and C-inputs		1
OR	OR2	2-input OR		1
	OR2A	2-input OR with active low A-input		1
	OR2B	2-input OR with active low inputs		1
	OR3	3-input OR		1
	OR3A	3-input OR with active low A-input		1

Hard Macros—Combinatorial (continued)

			Modules	
Function	**Macro**	**Description**	**S**	**C**
OR	OR3B	3-input OR with active low A- and B-inputs		1
	OR3C	3-input OR with active low inputs		1
	OR4	4-input OR		1
	OR4A	4-input OR with active low A-input		1
	OR4B	4-input OR with active low A- and B-input		1
	OR4C	4-input OR with active low A-, B-, and C-inputs		1
	OR4D	4-input OR with active low inputs		2
	OR5B	5-input OR with active low A- and B-inputs		1
OR-AND	OA1	3-input OR-AND		1
	OA1A	3-input OR-AND with active low A-input		1
	OA1B	3-input OR-AND with active low C-input		1
	OA1C	3-input OR-AND with active low A- and C-inputs		1
	OA2	2-wide 4-input OR-AND		1
	OA2A	2 wide 4-input OR-AND with active low A-input		1
	OA3	4-input OR-AND		1
	OA3A	4-input OR-AND with active low C-input		1
	OA3B	4-input OR-AND with active low A- and C-inputs		1
	OA4	4-input OR-AND		1
	OA4A	4-input OR-AND with active low C-input		1
	OA5	4-input complex OR-AND		1
	OAI1	3-input OR-AND-INVERT		1
	OAI2A	4-input OR-AND-INVERT with active low D-input		1
	OAI3	4-input OR-AND-INVERT		1
	OAI3A	4-input OR-AND-INVERT with active low C- and D-inputs		1
XNOR	XNOR2	2-input XNOR		1
XNOR-AND	XA1A	3-input XNOR-AND		1
XNOR-OR	XO1A	3-input XNOR-OR		1
XOR	XOR2	2-input XOR		1
XOR-AND	XA1	3-input XOR-AND		1
XOR-OR	XO1	3-input XOR-OR		1

Hard Macros—Sequential

			Modules	
Function	Macro	Description	S	C
D-Type	DF1	D-Type Flip-Flop	1	
	DF1A	D-Type Flip-Flop with active low output	1	
	DF1B	D-Type Flip-Flop with active low clock	1	
	DF1C	D-Type Flip-Flop with active low clock and output	1	
	DFC1	D-Type Flip-Flop with active high Clear	1	1
	DFC1A	D-Type Flip-Flop with active high Clear and active low clock	1	1
	DFC1B	D-Type Flip-Flop with active low Clear	1	
	DFC1D	D-Type Flip-Flop with active low Clear and clock	1	
	DFE	D-Type Flip-Flop with active high Enable	1	
	DFE1B	D-Type Flip-Flop with active low Enable	1	
	DFE1C	D-Type Flip-Flop with active low Enable and clock	1	
	DFE3A	D-Type Flip-Flop with Enable and active low Clear	1	
	DFE3B	D-Type Flip-Flop with Enable and active low Clear and clock	1	
	DFE3C	D-Type Flip-Flop with active low Enable and Clear	1	
	DFE3D	D-Type Flip-Flop with active low Enable, Clear, and clock	1	
	DFEA	D-Type Flip-Flop with Enable and active low clock	1	
	DFM	2-bit D-Type Flip-Flop with Multiplexed Data	1	
	DFM1B	2-bit D-Type Flip-Flop with Multiplexed Data and active low output	1	
	DFM1C	2-bit D-Type Flip-Flop with Multiplexed Data and active low clock and output	1	
	DFM3	2-bit D-Type Flip-Flop with Multiplexed Data and Clear	1	1
	DFM3B	2-bit D-Type Flip-Flop with Multiplexed Data and active low Clear and clock	1	
	DFM3E	2-bit D-Type Flip-Flop with Multiplexed Data, Clear, and active low clock	1	1
	DFM4C	2-bit D-Type Flip-Flop with Multiplexed Data and active low Preset and output	1	
	DFM4D	2-bit D-Type Flip-Flop with Multiplexed Data and active low Preset, clock, and output	1	
	DFM6A	4-bit D-Type Flip-Flop with Multiplexed Data, active low Clear, and active high Clock	1	
	DFM6B	4-bit D-Type Flip-Flop with Multiplexed Data, active low Clear, and clock	1	
	DFM7A	4-bit D-Type Flip-Flop with Multiplexed Data, active low Clear, and active high clock	1	
	DFM7B	4-bit D-Type Flip-Flop with Multiplexed Data, active low Clear and clock	1	
	DFMA	2-bit D-Type Flip-Flop with Multiplexed Data and active low clock	1	
	DFMB	2-bit D-Type Flip-Flop with Multiplexed Data and active low Clear	1	
	DFME1A	2-bit D-Type Flip-Flop with Multiplexed Data and active low Enable	1	
	DFP1	D-Type Flip-Flop with active high Preset		2
	DFP1A	D-Type Flip-Flop with active high Preset and active low clock		2
	DFP1B	D-Type Flip-Flop with active low Preset		2
	DFP1C	D-Type Flip-Flop with active high Preset and active low output	1	1
	DFP1D	D-Type Flip-Flop with active low Preset and clock		2
	DFP1E	D-Type Flip-Flop with active low Preset and output	1	

Hard Macros—Sequential (continued)

			Modules	
Function	**Macro**	**Description**	**S**	**C**
D-Type	DFP1F	D-Type Flip-Flop with active high Preset and active low clock and output	1	1
	DFP1G	D-Type Flip-Flop with active low Preset, clock, and output	1	
	DFPC	D-Type Flip-Flop with active high Preset, active low Clear, and active high clock		2
	DFPCA	D-Type Flip-Flop with active high Preset, and active low Clear and clock		2
J-K Type	JKF	JK Flip-Flop with active low K-input	1	
	JKF1B	JK Flip-Flop with active low clock and K-input	1	
	JKF2A	JK Flip-Flop with active low Clear and K-input	1	
	JKF2B	JK Flip-Flop with active low Clear, clock, and K-input	1	
	JKF2C	JK Flip-Flop with active high Clear and active low K-input	1	1
	JKF2D	JK Flip-Flop with active high Clear and active low clock and K-input	1	1
T-Type	TF1A	T-Type Flip-Flop with active low Clear	1	
	TF1B	T-Type Flip-Flop with active low Clear and clock	1	
Latch	DL1	Data Latch	1	
	DL1A	Data Latch with active low output	1	
	DL1B	Data Latch with active low clock	1	
	DL1C	Data Latch with active low clock and output	1	
	DLC	Data Latch with active low Clear	1	
	DLC1	Data Latch with active high Clear		1
	DLC1A	Data Latch with active high Clear and active low clock		1
	DLC1F	Data Latch with active high Clear and active low output		1
	DLC1G	Data Latch with active high Clear and active low clock and output		1
	DLCA	Data Latch with active low Clock and Clear	1	
	DLE	Data Latch with active high Enable	1	
	DLE1D	Data Latch with active high Enable and clock and active low input and output	1	
	DLE2B	Data Latch with active low Enable, Clear, and clock	1	
	DLE2C	Data Latch with active low Enable and clock and active high Clear		1
	DLE3B	Data Latch with active low Enable and clock and active low Preset		1
	DLE3C	Data Latch with active low Enable, Preset, and clock		1
	DLEA	Data Latch with active low Enable and active high clock	1	
	DLEB	Data Latch with active high Enable and active high clock	1	
	DLEC	Data Latch with active low Enable and clock	1	
	DLM	2-bit Data Latch with Multiplexed Data	1	
	DLM3	4-bit Data Latch with Multiplexed Data	1	
	DLM3A	4-bit Data Latch with Multiplexed Data and active low clock	1	
	DLM4	Data Latch with Multiplexed Data	1	

Hard Macros—Sequential (continued)

			Modules	
Function	Macro	Description	S	C
Latch	DLM4A	Data Latch with Multiplexed Data	1	
	DLMA	2-bit Data Latch with Multiplexed Data, and active low clock	1	
	DLME1A	2-bit Data Latch with Multiplexed Data and Enable and active low clock	1	
	DLP1	Data Latch with active high Preset and clock		1
	DLP1A	Data Latch with active high Preset and active low clock		1
	DLP1B	Data Latch with active low Preset and active high clock		1
	DLP1C	Data Latch with active low Preset and clock		1
	DLP1D	Data Latch with active low Preset and output and active high clock	1	
	DLP1E	Data Latch with active low Preset, clock, and output	1	

Input/Output Macros

Function	Macro	Description	I/O Modules
Buffer	BBHS	Bidirectional Buffer, High Slew	1
	BBUFTH	Bidirectional Buffer, Tristate Enable, High Slew	1
	BBUFTL	Bidirectional Buffer, Tristate Enable, Low Slew	1
	BIBUF	Bidirectional Buffer, High Slew (with hidden buffer at Y pin)	1
	HCLKBUF	Dedicated High-Speed S-Module Clock Buffer	1
	IBUF	Input Buffer	1
	INBUF	Input Buffer	1
	IOCLKBUF	Dedicated I/O Module Clock Buffer	1
	IOPCLBUF	Dedicated I/O Module IOPCL Buffer	1
	OBHS	Output buffer, High Slew	1
	OBUFTH	Output Buffer, Tristate Enable, High Slew	1
	OBUFTL	Output Buffer, Tristate Enable, Low Slew	1
	OUTBUF	Output Buffer, High Slew	1
Bidirectional	BRECTH	Bidirectional, Output Register with Clear, Data Enable, Tristate Enable, High Slew	1
	BRECTL	Bidirectional, Output Register with Clear, Data Enable, Tristate Enable, Low Slew	1
	BREPTH	Bidirectional, Output Register with Preset, Data Enable, Tristate Enable, High Slew	1
	BREPTL	Bidirectional, Output Register with Preset, Data Enable, Tristate Enable, Low Slew	1
	CLKBIBUF	Bidirectional with Input Dedicated to Clock Network	1
	DECETH	Bidirectional, Double Registered with Clear, Data Enable, Tristate Enable, High Slew	1
	DECETL	Bidirectional, Double Registered with Clear, Data Enable, Tristate Enable, Low Slew	1
	DEPETH	Bidirectional, Double Registered with Preset, Data Enable, Tristate Enable, High Slew	1
	DEPETL	Bidirectional, Double Registered with Preset, Data Enable, Tristate Enable, Low Slew	1

Input/Output Macros (continued)

Function	Macro	Description	I/O Modules
Input	CLKBUF	Input for Dedicated Routed Clock Network	1
	IREC	Input Register with Clear	1
	IREP	Input Register with Preset	1
Output	FECTMH	Output Register with Muxed Feedback, Clear, Data Enable, Tristate Enable, High Slew	1
	FECTML	Output Register with Muxed Feedback, Clear, Data Enable, Tristate Enable, Low Slew	1
	FEPTMH	Output Register with Muxed Feedback, Preset, Data Enable, Tristate Enable, High Slew	1
	FEPTML	Output Register with Muxed Feedback, Preset, Data Enable, Tristate Enable, Low Slew	1
	ORECTH	Output Register with Clear, Data Enable, Tristate Enable, High Slew	1
	ORECTL	Output Register with Clear, Data Enable, Tristate Enable, Low Slew	1
	OREPTH	Output Register with Preset, Data Enable, Tristate Enable, High Slew	1
	OREPTL	Output Register with Preset, Data Enable, Tristate Enable, Low Slew	1
	TBHS	Tristate output, High Slew	1
	TRIBUFF	Tristate output, High Slew	1

Soft Macros

Function	Macro	Description	Maximum Logic Levels	Modules	
				S	C
Adder	FADD10	10-bit adder	3		56
	FADD12	12-bit adder	4		9
	FADD16	16-bit adder	5		97
	FADD8	8-bit adder	4		44
	FADD9	9-bit adder with active low carry out	3		49
	VAD16C	Very fast 16-bit adder, no Carry in	3		97
	VADC16C	Very fast 16-bit adder with Carry in	3		97
Comparator	ICMP4	4-bit Identity Comparator	2		5
	ICMP8	8-bit Identity Comparator	3		9
	MCMPC2	2-bit Magnitude Comparator with Enable	3		9
	MCMPC4	4-bit Magnitude Comparator with Enable	4		18
	MCMPC8	8-bit Magnitude Comparator with Enable	6		36
Counter	CNT4A	4-bit binary counter with load and clear	4	4	8
	CNT4B	4-bit binary counter with load, clear, carry-in, carry-out	4	4	7
	FCTD16C	Fast 16-bit Down Counter, parallel loadable	2	19	33
	FCTD8A	Fast 8-bit Down Counter, parallel loadable	1	10	18
	FCTD8B	Fast 8-bit Down Counter, parallel loadable	1	9	13
	FCTU16C	Fast 16-bit Up Counter, parallel loadable	2	19	31
	FCTU8A	Fast 8-bit Up Counter, parallel loadable	1	10	17
	FCTU8B	Fast 8-bit Up Counter, parallel loadable	1	9	12
	UDCNT4A	4-bit up/down counter with load, carry-in, and carry-out	5	4	13

Soft Macros (continued)

Function	Macro	Description	Maximum Logic Levels	Modules	
				S	C
Counter	VCTD16C	Very fast 16-bit down counter, delay after load, registered control inputs	1	34	41
	VCTD2CP	2-bit down counter, prescaler, delay after load, use to build VCTD counters	1	5	2
	VCTD2CU	2-bit down counter, upper bits, delay after load, use to build VCTD counters	1	2	3
	VCTD4CL	4-bit down counter, lower bits, delay after load, use to build VCTD counters	1	4	7
	VCTD4CM	4-bit down counter, middle bits, delay after load, use to build VCTD counters	1	4	8
Decoder	DEC2X4	2-to-4 decoder	1		4
	DEC2X4A	2-to-4 decoder with active low outputs	1		4
	DEC3X8	3-to-8 decoder	1		8
	DEC3X8A	3-to-8 decoder with active low outputs	1		8
	DEC4X16A	4-to-16 decoder with active low outputs	2		20
	DECE2X4	2-to-4 decoder with enable	1		4
	DECE2X4A	2-to-4 decoder with enable and active low outputs	1		4
	DECE3X8	3-to-8 decoder with enable	2		11
	DECE3X8A	3-to-8 decoder with enable and active low outputs	2		11
Latch	DLC8A	octal latch with clear active low 8-bit Data Latch with active low Clear	1	8	
	DLE8	octal latch with enable 8-bit Data Latch with active high Enable	1	8	
	DLM8	octal latch with multiplexed data 8-bit Data Latch with Multiplexed Data	1	8	
MUX	MX16	16-to-1 Multiplexor	2		5
	MX8	8-to-1 Multiplexor with active high output	2		3
	MX8A	8-to-1 Multiplexor with active low output	2		3
Multiplier	SMULT8	8-bit by 8-bit Multiplier			242
Shift Register	SREG4A	4-bit shift register with clear active low	1	4	
	SREG8A	8-bit shift register with clear active low	1	8	

Soft Macros—TTL Equivalent

Function	Macro	Description	Maximum Logic Levels	Modules	
				S	C
	TA00	2-input NAND	1		1
	TA02	2-input NOR	1		1
	TA04	Inverter	1		1
	TA07	Buffer	1		1
	TA08	2-input AND	1		1
	TA10	3-input NAND	1		1
	TA11	3-input AND	1		1
	TA138	3-to-8 decoder with enable and active low outputs	2		12
	TA139	2-to-4 decoder with active low enable and outputs	1		4
	TA150	16-to-1 multiplexor with active low enable	3		6
	TA151	8-to-1 multiplexor with enable and both active low and active high output	3		5
	TA153	4-to-1 multiplexor with active low enable	2		2
	TA154	4-to-16 decoder with active low outputs and select lines	2		22
	TA157	2-to-1 multiplexor with active low enable	1		1
	TA160	4-bit decade counter with active low clear and load	4	4	8
	TA161	4-bit binary counter with active low clear and load	3	4	6
	TA164	8-bit serial in, parallel out shift register, active low clear	1	8	
	TA169	4-bit Up/Down Counter	6	4	14
	TA174	hex D-type flip-flop with active low clear	1	6	
	TA175	quadruple D-type flip-flop with active low clear	1	4	
	TA181	ALU	5		36
	TA190	4-bit up/down decode counter with up/down mode	7	4	31
	TA191	4-bit up/down binary counter with up/down mode	7	4	30
	TA194	4-bit bidirectional universal shift register	1	4	4
	TA195	4-bit parallel-access shift register	1	4	1
	TA20	4-input NAND	1		2
	TA21	4-input AND	1		1
	TA269	8-bit up/down binary counter	8	8	28
	TA27	3-input NOR	1		1
	TA273	octal register with clear	1	8	
	TA280	9-bit odd/even parity generator and checker	4		9
	TA32	2-input OR	1		1
	TA377	octal register with active low enable	1	8	
	TA40	4-input NAND	1		2
	TA42	4 to 10 decoder	1		10
	TA51	AND-OR-Invert	1		2
	TA54	4-wide 2-input AND-OR-Invert	2		5
	TA55	2-wide 4-input AND-OR-Invert	2		3
	TA688	8-bit identity comparator	3		9
	TA86	2-input exclusive OR	1		1

ACTgen Macro Timing Characteristics
(Worst-Case Commercial Conditions, Prelayout Timing Delays)

Sequential Macros*

Macro Type**	# Bits	Area (Seq/Total)	Levels	STD (MHz)	–1 (MHz)	–2 (MHz)	–3 (MHz)
Standard Counter (load and count enable)	4	4/9	2	91	106.5	121	135.6
	8	8/20	2	59	69	78.5	88
	16	16/46	3	57	66.7	75.8	85
	32	32/97	3	52	61	69	77.5
Pre-Scaled Counter (load or count enable)	16	19/46	1	149	174	198	222
	32	39/105	1	149	174	198	222
Shift Register (SISO and SIPO)	4	4/4	1	182	213	242	271
	8	8/10	1	182	213	242	271
	16	16/21	1	182	213	242	271
	32	32/42	1	182	213	242	271
Shift Register (PISO and PIPO)	4	4/4	1	182	213	242	271
	8	8/12	1	182	213	242	271
	16	16/24	1	182	213	242	271
	32	32/48	1	182	213	242	271
Accumulator	4	4/15	3	65	76	86.5	97
	8	8/35	4	48	56	64	71.5
	16	16/92	4	43	50	57	64
	32	32/191	5	34	40	45	50.7

**Delays based on 1425A die*

***Macro type is based on default ACTgen parameters*

ACTgen Macro Timing Characteristics

(Worst-Case Commercial Conditions, Prelayout Timing Delays)

Combinatorial Macros*

MacroType**	# Bits	Area (Comb)	Levels	STD (ns)	–1 (ns)	–2 (ns)	–3 (ns)
Adder	4	11	2	9.8ns	8.3	7.4	6.6
	8	27	3	15.3ns	13	11.5	10.3
	16	73	3	17.7ns	15	13	12
	32	155	4	24.3ns	20.7	18	16.3
Subtracter	4	15	3	14.5	12	11	10
	8	34	4	20	17	15	13.4
	16	89	4	22	18.7	16.5	14.8
	32	187	5	28	23.8	21	18.8
Magnitude Comparator	4	7	3	13.3	11	10	9
	8	14	4	18	15.3	13.5	12
	16	30	4	19	16	14	12.6
	32	61	5	23.5	20	17.6	15.8
Equality Comparator	4	5	2	8.6	7.3	6.5	5.8
	8	9	3	13.3	11.3	10	9
	16	19	3	13.3	11.3	10	9
	32	39	4	17.2	14.6	13	11.5
Decoder	4:16	24	2	11	9.4	8.3	7.4
	5:32	44	2	12.5	10.6	9.4	8.4
AND/OR	8	3	2	8.6	7.3	6.5	5.8
	16	5	2	8.6	7.3	6.5	5.8
	32	12	3	13	11	9.7	8.6
Multiplexer	8:1	3	2	9	7.6	6.7	6
	16:1	5	2	10	8.3	7.3	6.6
	32:1	15	3	19	16	14	12.6
Parity	8	7	3	13	11	9.6	8.6
	16	15	3	13.7	11.6	10.3	9.2
	32	31	4	18	15.3	13.5	12

**Delays based on 1425A die*

***Macro type is based on default ACTgen parameters*

ACT 1 Hard Macro Library Overview – Graphical Symbols

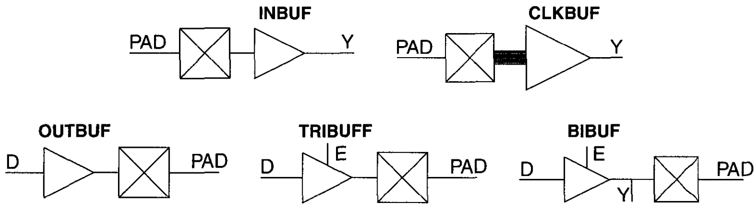
The following illustrations show all the available hard macros. The Actel *Macro Library Guide* contains module count, combinability, and pin loading information of each hard macro. It also includes a complete truth table for each macro.

Most ACT 1 hard macros are implemented by a single logic module. The following ACT 1 hard macros require two logic modules to implement:

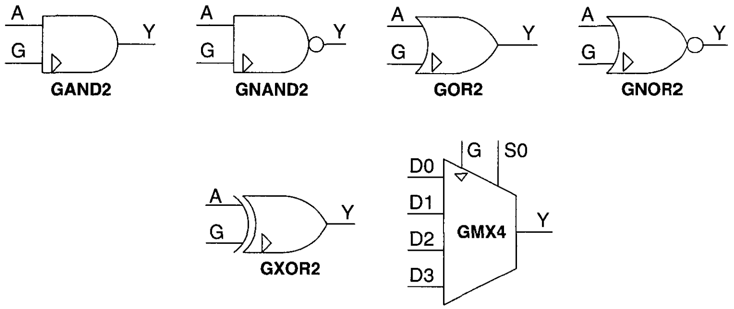
- All flip-flops
- All adders
- Two-module hard macro gates: AND4, AND4A, AND4D, AOI1, AOI4, NAND3, NAND4, NAND4A, NAND4B, NOR4, NOR4C, NOR4D, OAI3, OR3C, OR4B, OR4C, OR4D

Two-module hard macro gates have a "2" displayed on some input pins. This indicates that the input to output path has two levels of logic delay for these input pins only. Also, the full adders have a "2" on the "S" output pins. This indicates that there are two levels of logic delay from the input pins to the "S" output pin. Refer to the ACT 1 Timing Characteristics for detailed timing information of all ACT 1 macros.

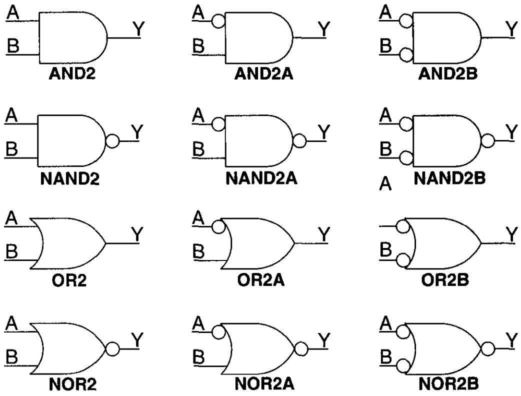
Input and Output Buffers



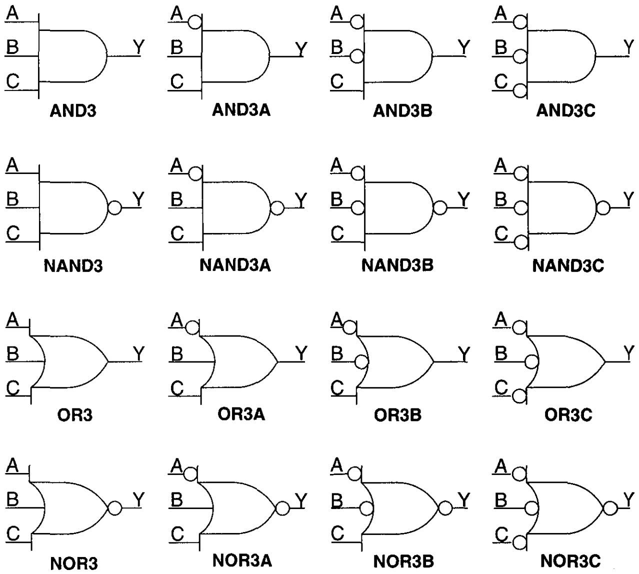
CLKBUF Interface Macros



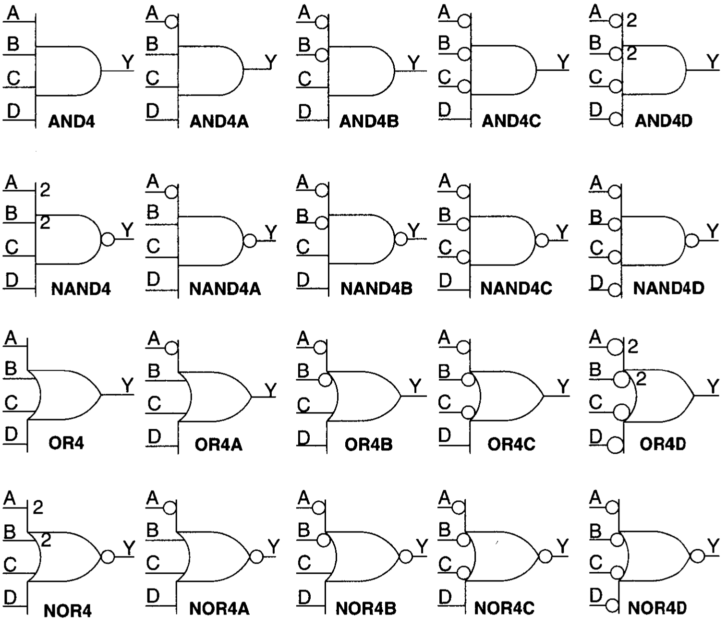
2-Input Gates



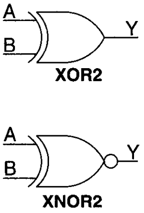
3-Input Gates



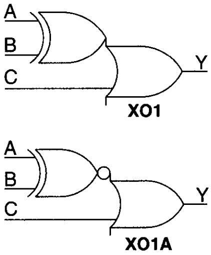
4-Input Gates



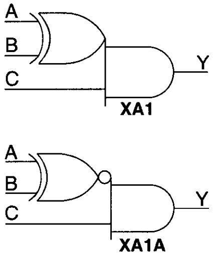
XOR Gates



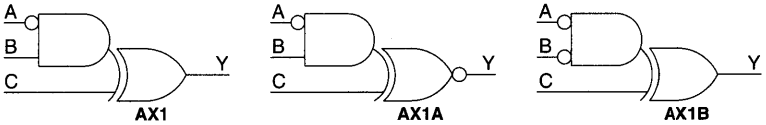
XOR-OR Gates



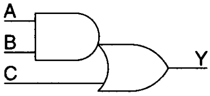
XOR-AND Gates



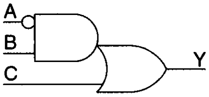
AND-XOR Gates



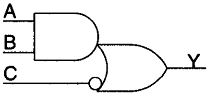
AND-OR Gates



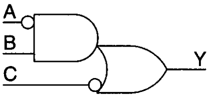
AO1



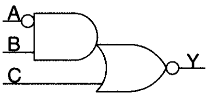
AO1A



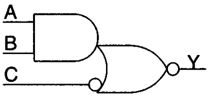
AO1B



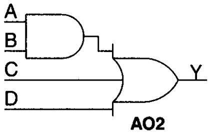
AO1C



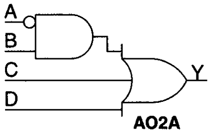
AOI1A



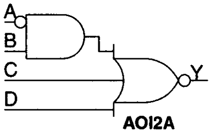
AOI1B



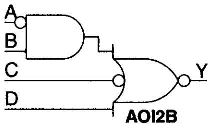
AO2



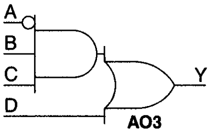
AO2A



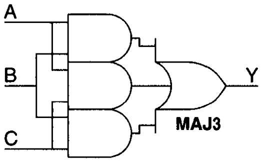
AOI2A



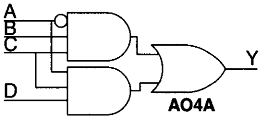
AOI2B



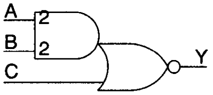
AO3



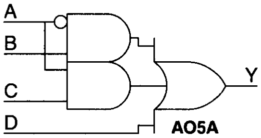
MAJ3



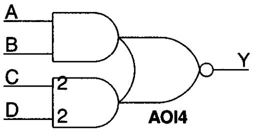
AO4A



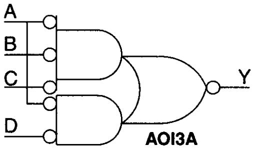
AOI1



AO5A

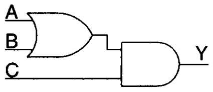


AOI4

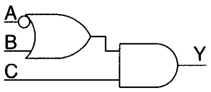


AOI3A

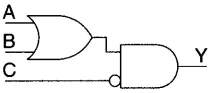
OR-AND Gates



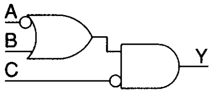
OA1



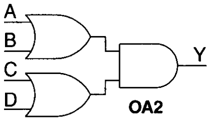
OA1A



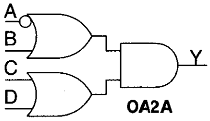
OA1B



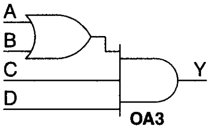
OA1C



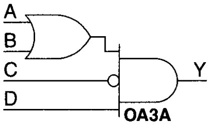
OA2



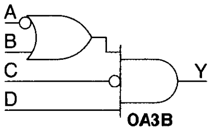
OA2A



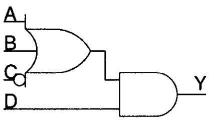
OA3



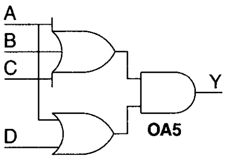
OA3A



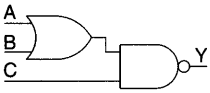
OA3B



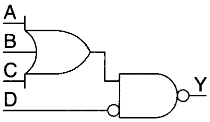
OA4A



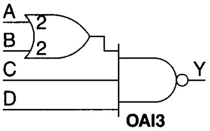
OA5



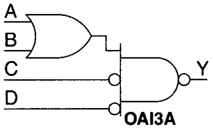
OAI1



OAI2A

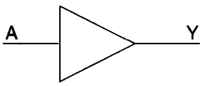


OAI3

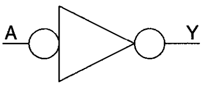


OAI3A

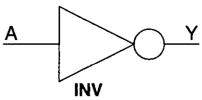
Buffers



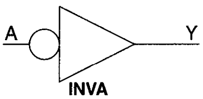
BUFF



BUFA

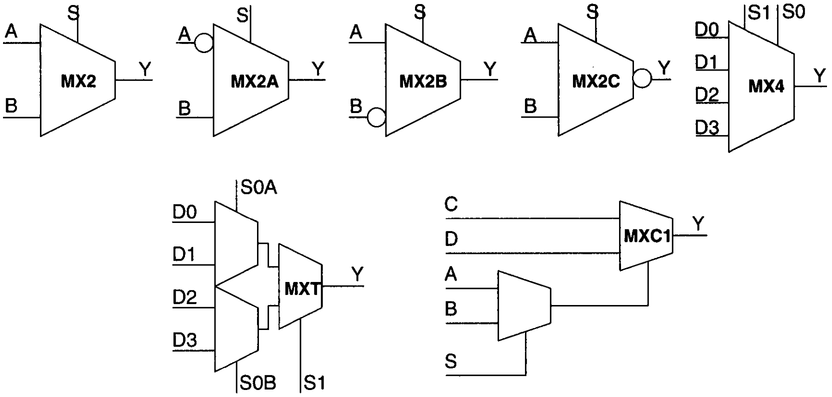


INV

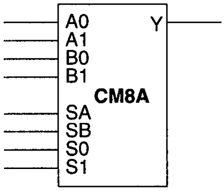


INVA

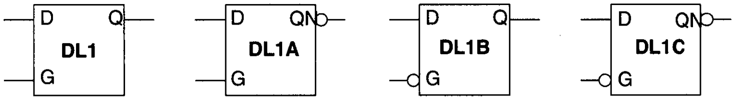
Multiplexors



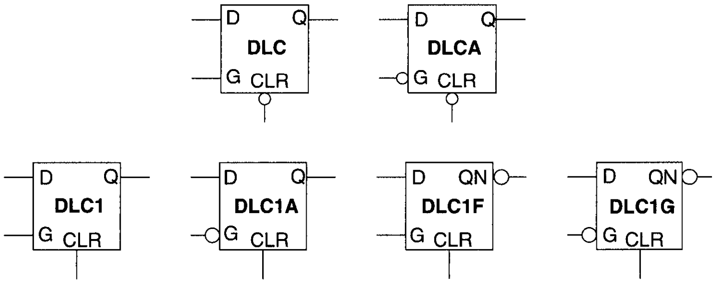
Combinatorial



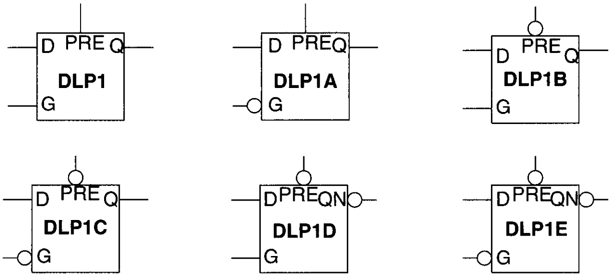
D-Latches



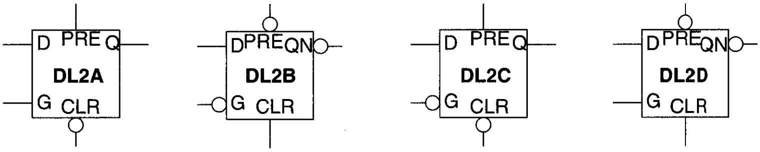
D-Latches with Clear



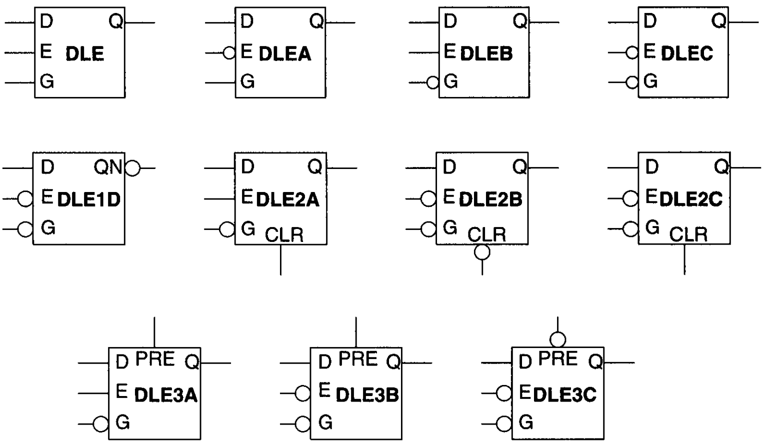
D-Latches wilth Preset



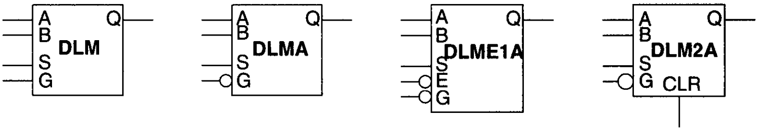
D-Latches with Preset and Clear



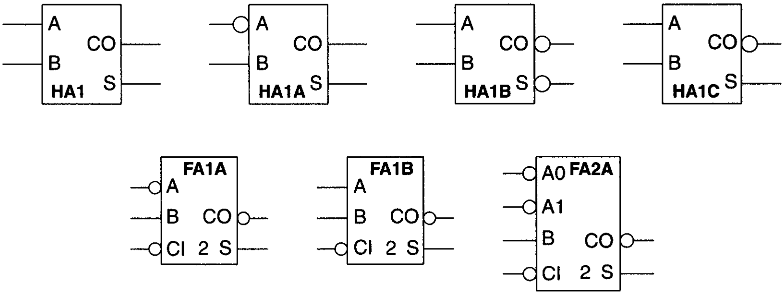
D-Latches with Enable



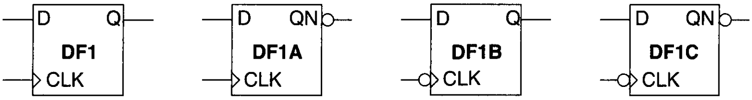
Mux Latches



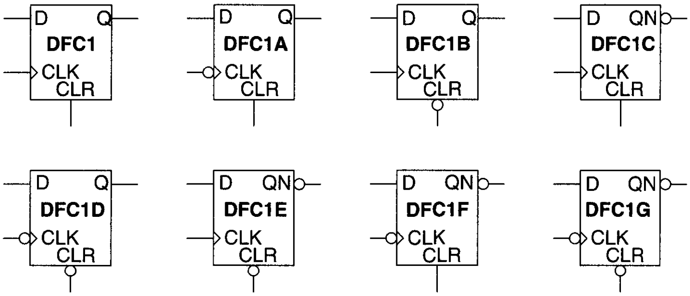
Adders



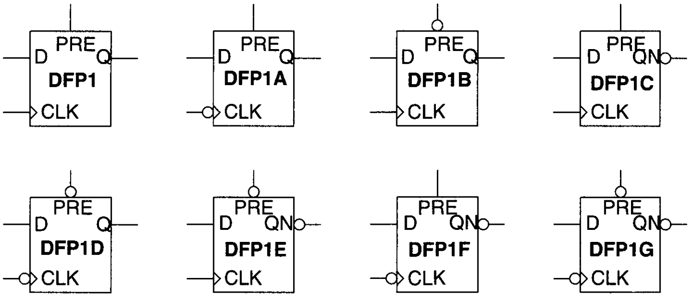
D-Type Flip-Flops



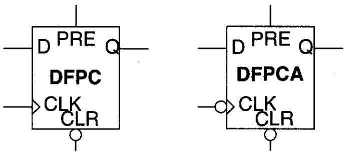
D-Type Flip-Flops with Clear



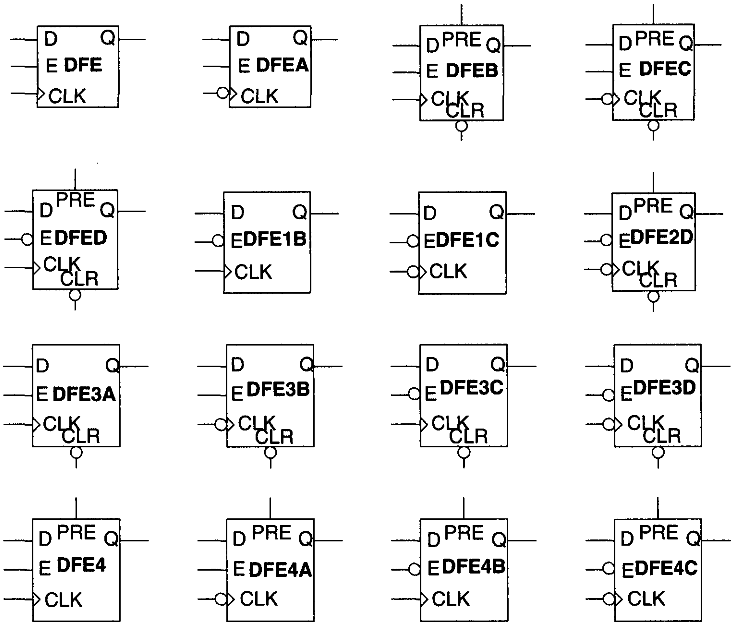
D-Type Flip-Flops with Preset



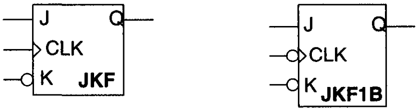
D-Type Flip-Flops with Preset and Clear



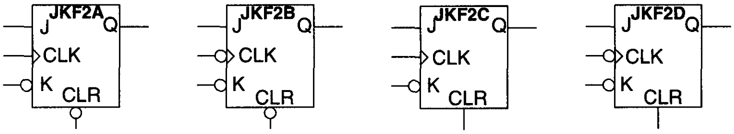
D-Type Flip-Flops with Enable



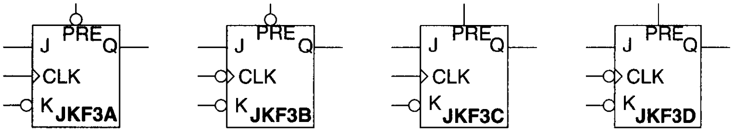
JK Flip-Flops



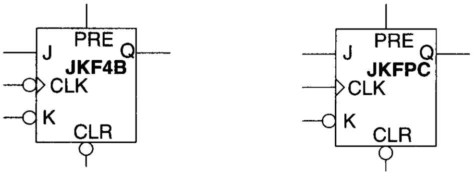
JK Flip-Flops with Clear



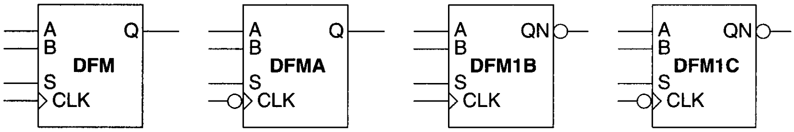
JK Flip-Flops with Preset



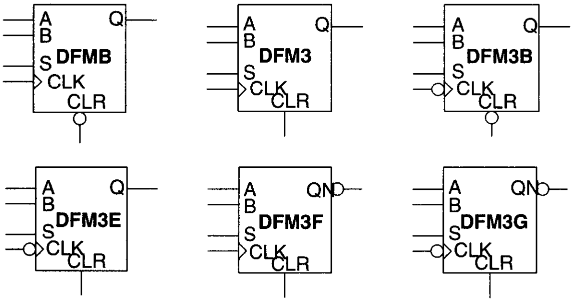
JK Flip-Flops with Preset and Clear



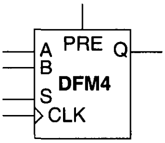
Mux Flip-Flops

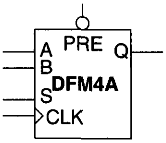


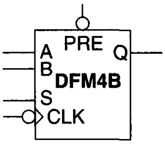
Mux Flip-Flops with Clear

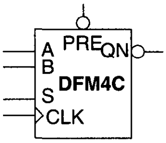


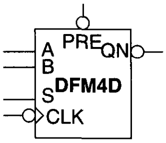
Mux Flip-Flops with Preset

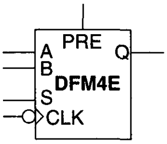




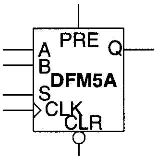


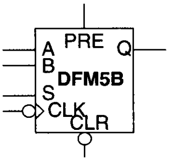






Mux Flip-Flops with Preset and Clear





ACT 1 Macro Library – Tables of Hard, Soft, and TTL Macros

Hard Macros—Combinatorial

Function	Macro	Description	Modules C
Adder	FA1A	1-bit adder, carry in and carry out active low, A-input active low	2
	FA1B	1-bit adder, carry in and carry out active low	2
	FA2A	2-bit adder, carry in and carry out active low, A0 and A1 inputs active low	2
	HA1	Half-Adder	2
	HA1A	Half-Adder with active low A-input	2
	HA1B	Half-Adder with active low carry out and sum	2
	HA1C	Half-Adder with active low carry out	2
AND	AND2	2-input AND	1
	AND2A	2-input AND with active low A-input	1
	AND2B	2-input AND with active low inputs	1
	AND3	3-input AND	1
	AND3A	3-input AND with active low A-input	1
	AND3B	3-input AND with active low A- and B-inputs	1
	AND3C	3-input AND with active low inputs	1
	AND4	4-input AND	2
	AND4A	4-input AND with active low A-input	2
	AND4B	4-input AND with active low A- and B-inputs	1
	AND4C	4-input AND with active low A-, B-, and C-inputs	1
	AND4D	4-input AND with active low inputs	2
AND-OR	AO1	3-input AND-OR	1
	AO1A	3-input AND-OR with active low A-input	1
	AO1B	3-input AND-OR with active low C-input	1
	AO1C	3-input AND-OR with active low A- and C-inputs	1
	AO2	4-input AND-OR	1
	AO2A	4-input AND-OR with active low A-input	1
	AO3	4-input AND-OR	1
	AO4A	4-input AND-OR	1
	AO5A	4-input AND-OR	1
	AOI1	3-input AND-OR-INVERT	2
	AOI1A	3-input AND-OR-INVERT with active low A-input	1
	AOI1B	3-input AND-OR-INVERT with active low C-input	1
	AOI2A	4-input AND-OR-INVERT with active low A-input	1
	AOI2B	4-input AND-OR-INVERT with active low A- and C-inputs	1

3

ACT 1

Hard Macros—Combinatorial (continued)

Function	Macro	Description	Modules C
AND-OR	AOI3A	4-input AND-OR-INVERT with active low inputs	1
	AOI4	2-wide 4-input AND-OR-INVERT	2
AND-XOR	AX1	3-input AND-XOR with active low A-input	1
	AX1A	3-input AND-XOR-INVERT with active low A-input	1
	AX1B	3-input AND-XOR with active low A- and B-inputs	1
Buffer	BUF	Buffer with active high input and output	1
	BUFA	Buffer with active low input and output	1
Clock Net	GAND2	2-input AND Clock Net	1
	GMX4	4-to-1 Multiplexor Clock Net	1
	GNAND2	2-input NAND Clock Net	1
	GNOR2	2-input NOR Clock Net	1
	GOR2	2-input OR Clock Net	1
	GXOR2	2-input Exclusive OR Clock Net	1
Combinatorial	CM8A	Combinational Module	1
Inverter	INV	Inverter with active low output	1
	INVA	Inverter with active low input	1
Majority	MAJ3	3-input complex AND-OR	1
MUX	MX2	2-to-1 Multiplexor	1
	MX2A	2-to-1 Multiplexor with active low A-input	1
	MX2B	2-to-1 Multiplexor with active low B-input	1
MUX	MX2C	2-to-1 Multiplexor with active low output	1
	MX4	4-to-1 Multiplexor	1
	MXC1	Boolean	
	MXT	Boolean	
NAND	NAND2	2-input NAND	1
	NAND2A	2-input NAND with active low A-input	1
	NAND2B	2-input NAND with active low inputs	1
	NAND3	3-input NAND	
	NAND3A	3-input NAND with active low A-input	1
	NAND3B	3-input NAND with active low A- and B-inputs	1
	NAND3C	3-input NAND with active low inputs	1
	NAND4	4-input NAND	2
	NAND4A	4-input NAND with active low A-input	
	NAND4B	4-input NAND with active low A- and B-inputs	
	NAND4C	4-input NAND with active low A-, B-, and C-inputs	1
	NAND4D	4-input NAND with active low inputs	1
NOR	NOR2	2-input NOR	1
	NOR2A	2-input NOR with active low A-input	1
	NOR2B	2-input NOR with active low inputs	1
	NOR3	3-input NOR	1
	NOR3A	3-input NOR with active low A-input	1

Hard Macros—Combinatorial (continued)

Function	Macro	Description	Modules C
NOR	NOR3B	3-input NOR with active low A- and B-inputs	1
	NOR3C	3-input NOR with active low inputs	1
	NOR4	4-input NOR	2
	NOR4A	4-input NOR with active low A-input	1
	NOR4B	4-input NOR with active low A- and B-inputs	1
	NOR4C	4-input NOR with active low A-, B-, and C-inputs	2
	NOR4D	4-input NOR with active low inputs	2
OR	OR2	2-input OR	1
	OR2A	2-input OR with active low A-input	1
	OR2B	2-input OR with active low inputs	1
	OR3	3-input OR	1
	OR3A	3-input OR with active low A-input	1
	OR3B	3-input OR with active low A- and B-inputs	1
	OR3C	3-input OR with active low inputs	2
	OR4	4-input OR	1
	OR4A	4-input OR with active low A-input	1
	OR4B	4-input OR with active low A- and B-input	2
	OR4C	4-input OR with active low A-, B-, and C-inputs	2
	OR4D	4-input OR with active low inputs	2
OR-AND	OA1	3-input OR-AND	1
	OA1A	3-input OR-AND with active low A-input	1
	OA1B	3-input OR-AND with active low C-input	1
	OA1C	3-input OR-AND with active low A- and C-inputs	1
	OA2	2-wide 4-input OR-AND	1
	OA2A	2 wide 4-input OR-AND with active low A-input	1
	OA3	4-input OR-AND	1
	OA3A	4-input OR-AND with active low C-input	1
	OA3B	4-input OR-AND with active low A- and C-inputs	1
	OA4A	4-input OR-AND with active low C-input	1
	OA5	4-input complex OR-AND	1
	OAI1	3-input OR-AND-INVERT	1
	OAI2A	4-input OR-AND-INVERT with active low D-input	1
	OAI3	4-input OR-AND-INVERT	2
	OAI3A	4-input OR-AND-INVERT with active low C- and D-inputs	1
XNOR	XNOR2	2-input XNOR	1
XNOR-AND	XA1A	3-input XNOR-AND	1
XNOR-OR	XO1A	3-input XNOR-OR	1
XOR	XOR2	2-input XOR	1
XOR-AND	XA1	3-input XOR-AND	1
XOR-OR	XO1	3-input XOR-OR	1

Hard Macros—Sequential

Function	Macro	Description	Modules C
D-Type	DF1	D-Type Flip-Flop	2
	DF1A	D-Type Flip-Flop with active low output	2
	DF1B	D-Type Flip-Flop with active low clock	2
	DF1C	D-Type Flip-Flop with active low clock and output	2
	DFC1	D-Type Flip-Flop with active high Clear	2
	DFC1A	D-Type Flip-Flop with active high Clear and active low clock	2
	DFC1B	D-Type Flip-Flop with active low Clear	2
	DFC1C	D-Type Flip-Flop with Clear, Sequential	2
	DFC1D	D-Type Flip-Flop with active low Clear and clock	2
	DFC1E	D-Type Flip-Flop with Clear, Sequential	2
	DFC1F	D-Type Flip-Flop with Clear, Sequential	2
	DCF1G	D-Type Flip-Flop with Clear, Sequential	2
	DFE	D-Type Flip-Flop with active high Enable	2
	DFE1B	D-Type Flip-Flop with active low Enable	2
	DFE1C	D-Type Flip-Flop with active low Enable and clock	2
	DFE2D	D-Type Flip-Flop with Enable, Sequential	2
	DFE3A	D-Type Flip-Flop with Enable and active low Clear	2
	DFE3B	D-Type Flip-Flop with Enable and active low Clear and clock	2
	DFE3C	D-Type Flip-Flop with active low Enable and Clear	2
	DFE3D	D-Type Flip-Flop with active low Enable, Clear, and clock	2
	DFE4	D-Type Flip-Flop with Enable, Sequential	2
	DFE4A	D-Type Flip-Flop with Enable, Sequential	2
	DFE4B	D-Type Flip-Flop with Enable, Sequential	2
	DFE4C	D-Type Flip-Flop with Enable, Sequential	2
	DFEA	D-Type Flip-Flop with Enable and active low clock	2
	DFEB	D-Type Flip-Flop with Enable, Sequential	2
	DFEC	D-Type Flip-Flop with Enable, Sequential	2
	DFED	D-Type Flip-Flop with Enable, Sequential	2
	DFM	2-input D-Type Flip-Flop with Multiplexed Data	2
	DFM1B	2-input D-Type Flip-Flop with Multiplexed Data and active low output	2
	DFM1C	2-input D-Type Flip-Flop with Multiplexed Data and active low clock and output	2
	DFM3	2-input D-Type Flip-Flop with Multiplexed Data and Clear	2
	DFM3B	2-input D-Type Flip-Flop with Multiplexed Data and active low Clear and clock	2
	DFM3E	2-input D-Type Flip-Flop with Multiplexed Data, Clear, and active low clock	2
	DFM3F	D-Type Flip-Flop with Multiplexed Data, Enable and Sequential	2
	DFM3G	D-Type Flip-Flop with Multiplexed Data, Enable and Sequential	2
	DFM4	D-Type Flip-Flop with Multiplexed Data, Enable and Sequential	2
	DFM4A	D-Type Flip-Flop with Multiplexed Data, Enable and Sequential	2
	DFM4B	D-Type Flip-Flop with Multiplexed Data, Enable and Sequential	2

Hard Macros—Sequential (continued)

			Modules
Function	Macro	Description	C
D-Type	DFM4C	2-input D-Type Flip-Flop with Multiplexed Data and active low Preset and output	2
	DFM4D	2-input D-Type Flip-Flop with Multiplexed Data and active low Preset, clock, and output	2
	DFM4E	D-Type Flip-Flop with Multiplexed Data, Enable and Sequential	2
	DFM5A	D-Type Flip-Flop with Multiplexed Data, Enable and Sequential	2
	DFM5B	D-Type Flip-Flop with Multiplexed Data, Enable and Sequential	2
	DFMA	2-input D-Type Flip-Flop with Multiplexed Data and active low Clock	2
	DFMB	2-input D-Type Flip-Flop with Multiplexed Data and active low Clear	2
	DFME1A	2-input D-Type Flip-Flop with Multiplexed Data and active low Enable	2
	DFP1	D-Type Flip-Flop with active high Preset	2
	DFP1A	D-Type Flip-Flop with active high Preset and active low clock	2
	DFP1B	D-Type Flip-Flop with active low Preset	2
	DFP1C	D-Type Flip-Flop with active high Preset and active low output	2
	DFP1D	D-Type Flip-Flop with active low Preset and clock	2
	DFP1E	D-Type Flip-Flop with active low Preset and output	2
	DFP1F	D-Type Flip-Flop with active high Preset and active low clock and output	2
	DFP1G	D-Type Flip-Flop with active low Preset, clock, and output	2
	DFPC	D-Type Flip-Flop with active high Preset, active low Clear, and active high clock	2
	DFPCA	D-Type Flip-Flop with active high Preset and active low Clear and clock	2
J-K Type	JKF	JK Flip-Flop with active low K-input	2
	JKF1B	JK Flip-Flop with active low clock and K-input	2
	JKFPC	JK Flip-Flop, Sequential	2
	JKF2A	JK Flip-Flop with active low Clear and K-input	2
	JKF2B	JK Flip-Flop with active low Clear, clock, and K-input	2
	JKF2C	JK Flip-Flop with active high Clear and active low K-input	2
	JKF2D	JK Flip-Flop with active high Clear and active low clock and K-input	2
	JKF3A	JK Flip-Flop, Sequential	2
	JKF3B	JK Flip-Flop, Sequential	2
	JKF3C	JK Flip-Flop, Sequential	2
	JKF3D	JK Flip-Flop, Sequential	2
	JKF4B	JK Flip-Flop, Sequential	2
Latch	DL1	Data Latch	1
	DL1A	Data Latch with active low output	1
	DL1B	Data Latch with active low clock	1
	DL1C	Data Latch with active low clock and output	1
	DL2A	Sequential, Data Latch	1
	DL2B	Sequential, Data Latch	1
	DL2C	Sequential, Data Latch	1
	DL2D	Sequential, Data Latch	1
	DLC	Data Latch with active low Clear	1

Hard Macros—Sequential (continued)

Function	Macro	Description	Modules C
Latch	DLC1	Data Latch with active high Clear	1
	DLC1A	Data Latch with active high Clear and active low clock	1
	DLC1F	Data Latch with active high Clear and active low output	1
	DLC1G	Data Latch with active high Clear and active low clock and output	1
	DLCA	Data Latch with active low Clock and Clear	1
	DLE	Data Latch with active high Enable	1
	DLE1D	Data Latch with active high Enable and clock and active low input and output	1
	DLE2A	Sequential, Data Latch with Enable	1
	DLE2B	Data Latch with active low Enable, Clear, and clock	1
	DLE2C	Data Latch with active low Enable and clock and active high clear	1
	DLE3A	Sequential, Data Latch with Enable	1
	DLE3B	Data Latch with active low Enable and clock and active low Preset	1
	DLE3C	Data Latch with active low Enable Preset and clock	1
	DLEA	Data Latch with active low Enable and active high clock	1
	DLEB	Data Latch with active high Enable and active high clock	1
	DLEC	Data Latch with active low Enable and clock	1
	DLM	2-input Data Latch with Multiplexed Data	1
	DLM2A	Sequential, Data Latch with Multiplexed Data	1
	DLMA	2-input Data Latch with Multiplexed Data and active low clock	1
	DLME1A	2-input Data Latch with Multiplexed Data and Enable and active low clock	1
	DLP1	Data Latch with active high Preset and clock	1
	DLP1A	Data Latch with active high Preset and active low clock	1
	DLP1B	Data Latch with active low Preset and active high clock	1
	DLP1C	Data Latch with active low Preset and clock	1
	DLP1D	Data Latch with active low Preset and output and active high clock	1
	DLP1E	Data Latch with active low Preset, clock, and output	1

Input/Output Macros

Function	Macro	Description	I/O Modules
Buffer	INBUF	Input Buffer	1
	OUTBUF	Output buffer, High Slew	1
Bidirectional	BIBUF	Bidirectional Buffer, High Slew (with hidden buffer at Y pin)	1
Input	CLKBUF	Input for Dedicated Routed Clock Network	1
Output	TRIBUFF	Tristate output, High Slew	1

Soft Macros

Function	Macro	Description	Maximum Logic Levels	Modules	
				S	C
Adder	FADD16	16-bit adder	5		79
	FADD8	8-bit adder	4		37
	FADD24	24-bit adder	6		120
	FADD32	32-bit adder	7		160
Comparator	ICMP4	4-bit Identity Comparator	2		5
	ICMP8	8-bit Identity Comparator	3		9
	MCMP16	16-bit Magnitude Comparator	5		93
	MCMPC2	2-bit Magnitude Comparator with Enable	3		9
	MCMPC4	4-bit Magnitude Comparator with Enable	4		18
	MCMPC8	8-bit Magnitude Comparator with Enable	6		36
Counter	CNT4A	4-bit binary counter with load and clear	4		18
	CNT4B	4-bit binary counter with load, clear, carry-in, carry-out	4		15
	UDCNT4A	4-bit up/down counter with load, carry-in, and carry-out	6		24
Decoder	DEC2X4	2-to-4 decoder	1		4
	DEC2X4A	2-to-4 decoder with active low outputs	1		4
	DEC3X8	3-to-8 decoder	1		8
	DEC3X8A	3-to-8 decoder with active low outputs	1		9
	DEC4X16A	4-to-16 decoder with active low outputs	2		20
	DECE2X4	2-to-4 decoder with enable	1		4
	DECE2X4A	2-to-4 decoder with enable and active low outputs	1		5
	DECE3X8	3-to-8 decoder with enable	2		11
	DECE3X8A	3-to-8 decoder with enable and active low outputs	2		11
Latch	DLC8A	octal latch with clear active low 8-bit Data Latch with active low Clear	1		8
	DLE8	octal latch with enable 8-bit Data Latch with active high Enable	1		8
	DLM8	octal latch with multiplexed data 8-bit Data Latch with Multiplexed Data	1		8
MUX	MX16	16-to-1 Multiplexor	2		5
	MX8	8-to-1 Multiplexor with active high output	2		3
	MX8A	8-to-1 Multiplexor with active low output	2		3
Multiplier	SMULT8	8-bit by 8-bit Multiplier			241
Registers	REGE8A	8-bit register, with enable, preset, and active low clear	1		20
	REGE8B	8-bit register, with enable, preset, and active low clear and negative edge clock	1		20
Shift Register	SREG4A	4-bit shift register with clear active low	1		8
	SREG8A	8-bit shift register with clear active low	1		18

Soft Macros—TTL Equivalent

Function	Macro	Description	Maximum Logic Levels	Modules	
				S	C
	TA138	3-to-8 decoder with enable and active low outputs	2		12
	TA139	2-to-4 decoder with active low enable and outputs	1		4
	TA151	8-to-1 multiplexor with enable and both active low and active high output	3		5
	TA153	4-to-1 multiplexor with active low enable	2		2
	TA157	2-to-1 multiplexor with active low enable	1		1
	TA161	4-bit binary counter with active low clear and load	3		22
	TA164	8-bit serial in, parallel out shift register, active low clear	1		18
	TA169	4-bit Up/Down Counter	6		25
	TA181	ALU	5		36
	TA194	4-bit bidirectional universal shift register	1		14
	TA195	4-bit parallel-access shift register	1		11
	TA269	8-bit up/down binary counter	8		50
	TA273	octal register with clear	1		18
	TA280	9-bit odd/even parity generator and checker	4		9
	TA377	octal register with active low enable	1		16

ACT 2 Macro Library

The ACT 2 Macro Library is equivalent to the 1200XL Macro Library. Please refer to the "Integrator Series Hard Macro Library – Graphical Symbols," starting on page 3-1, and the "Integrator Series Macro Library – Tables of Hard, Soft, TTL, and ACTgen Macros," starting on page 3-17, for all library information for this family of devices.

Designing with Actel Devices

Section 4: Designing with Actel Devices

Introduction to Actel FPGA Architecture

This overview of the different Actel device families covers the major architectural features in sufficient detail to ensure the reader is familiar enough with Actel devices to get the most out of the application notes in the rest of this section. Details on the functionality of each device are available in the individual device data sheets.

FPGA Architecture—User Requirements

Digital system design is becoming more difficult. Systems require ever-increasing complexity and performance, but time-to-market pressures continue to limit development times. System cost is also an important constraint, so a solution must meet stringent financial targets. These competing requirements demand a digital logic design solution optimized to meet all of the capacity, performance, cost, and time-to-market requirements. An optimized architecture is needed to balance all of these competing demands. The Actel architecture meets all of these requirements by providing the correct balance of capacity, performance, cost, and ease of use through its innovative combination of an optimized logic module, abundant interconnect resources, efficient silicon usage, and powerful software design tools.

Actel Device Architecture

The underlying architecture of an Actel FPGA is very similar to that of a conventional gate array. The core of the device consists of simple logic modules used to implement the required logic gates and storage elements. These logic modules are interconnected with an abundance of segmented routing tracks. Unlike gate arrays, the segment lengths are predefined and can be connected with low-impedance switching elements to create the precise routing length required of the interconnect signal. Surrounding the logic core is the interface to the I/O pads of the devices. This interface consists of I/O modules that translate and interconnect the logic signals from the core of the device to the FPGA output pads. A block diagram of a generic Actel FPGA is given in Figure 1.

The major elements of the Actel FPGA architecture are thus the I/O modules, interconnect resources, clocking resources, and logic modules. Each Actel FPGA family has a slightly different mix of these resources, optimized for different cost, performance, and density requirements. Table 1 shows the capabilities of each Actel FPGA family. Each capability is defined in the sections following Table 1.

Logic Module Descriptions

The optimal logic module should provide the user with the correct mix of performance, efficiency, and ease of design required to implement the application. If the logic module provides performance without efficiency, the cost and capacity requirements of the design might not be achieved. Similarly, if cost and capacity targets are achieved at the expense of performance and ease of use, the device may not be usable. The optimal logic module must balance these trade-offs carefully to ensure that the many conflicting goals of the designer are achievable.

Simple Logic Module

The first Actel logic module was the Simple Logic Module used in the ACT 1 family. It is shown in Figure 2. It is a multiplexer-based logic module. Logic functions are implemented by interconnecting signals from the routing tracks to the data inputs and select lines of the multiplexers. Inputs can also be tied to a logical 1 or 0 if required, since these signals are always available in the routing channel.

A surprising number of useful logic functions can be implemented with this module. Clearly, multiplexing is very efficient, but random logic and sequential logic functions are also efficient. These options provide the designer with an excellent mix of logic capabilities, required for applications demanding a variety of logic functions. Figure 3 shows an example logic function implemented with the Actel Simple Logic Module. Notice that latches can be implemented in a single logic module per bit and that registers require two logic modules per bit. The ACT 1 logic module is thus extremely flexible in covering a wide range of combinatorial and sequential logic mixes.

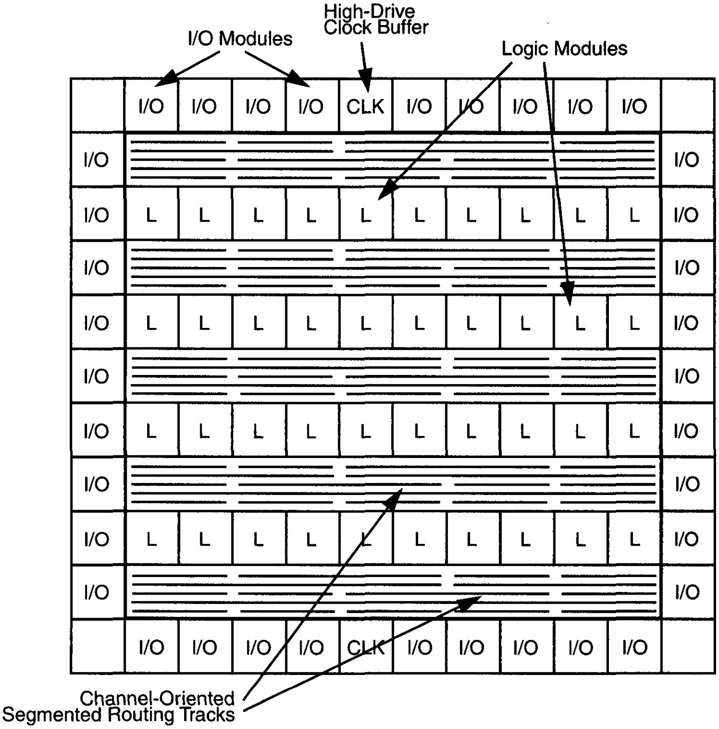


Figure 1 • *Basic Actel FPGA Architecture*

Table 1 • *Quick Summary of Key Architectural Features of Actel FPGA Families*

Capability	**ACT 1**	**ACT 2/1200XL**	**3200DX**	**ACT 3**
Core Module	Simple Logic Module	Combinatorial and Sequential Modules	Combinatorial and Sequential Modules Wide Decode Modules Embedded Dual-Port SRAM	Combinatorial and Enhanced Sequential Modules
Interconnect	Channeled	Channeled	Channeled	Channeled
Clocking Resources	Routed Clock (1)	Routed Clocks (2)	Routed Clocks (2) Quad Clocks (4)	Routed Clocks (2) Dedicated Array Clock Dedicated I/O Clock
I/O Module	Simple I/O Module	Latched I/O Module	Latched I/O Module	Registered I/O Module

Combinatorial Logic Module

Some improvements were made to the Simple Logic Module when the second generation ACT 2 family was developed. The Simple Logic Module was replaced with two different logic modules, one for implementing combinatorial logic, (the Combinatorial Logic Module) and one for implementing storage elements (the Sequential Logic Module). The Combinatorial Logic Module, shown in the diagram in Figure 4, is similar to the Simple Logic Module, but an additional logic gate was placed on the first-level multiplexer. The added gate improves the implementation of some combinatorial functions. (Some five-input gates are now available.) Also, the first-level multiplexer lines in the Simple Logic Module were combined in the Combinatorial Logic Module. In the Simple Logic Module, the separate multiplexer select lines were used to implement latches and registers efficiently. This was not required in the Combinatorial Logic Module because of the addition the Sequential Logic Module. Figure 5 shows an example of a logic function implemented with the Combinatorial Logic Module.

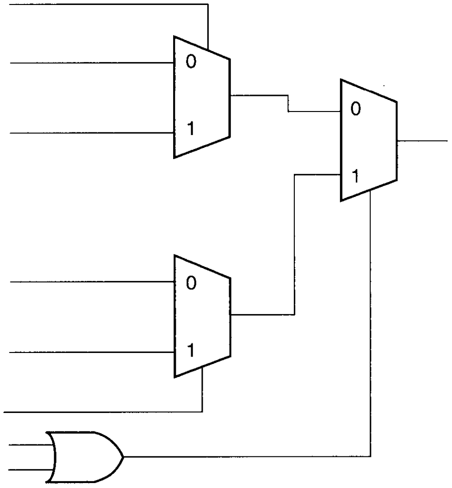


Figure 2 • *Simple Logic Module*

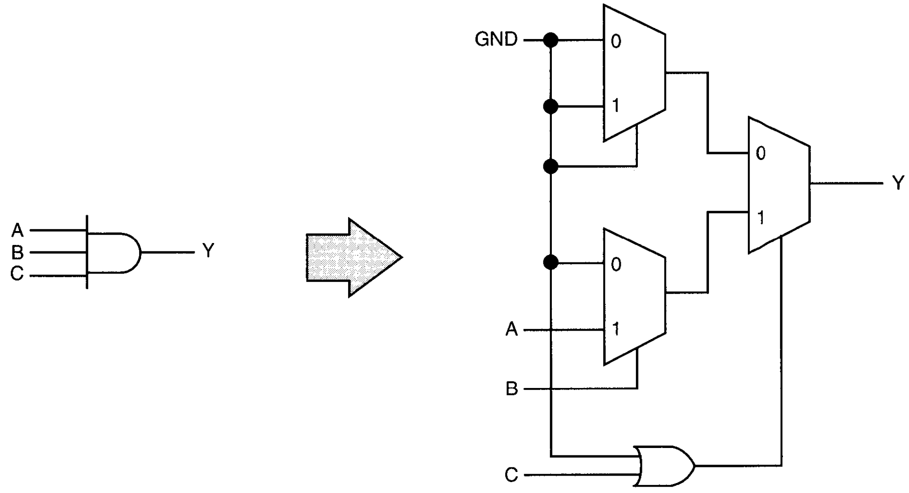


Figure 3 • *Example of a Logic Function Implemented with the ACT 1 Simple Logic Module*

Sequential Logic Module

The Sequential Logic Module, shown in the diagram in Figure 6, has a combinatorial logic front end with a dedicated storage element on the output of the logic module. The storage element can be either a register or a latch. (It can also be bypassed so the logic module can be used as a Combinatorial Logic Module.) The clock input can be selected to be either active high or active low. One of the logic gates is missing on the combinatorial logic section, making it slightly different from the Combinatorial Logic Module. The exclusion of this one logic gate allows the reset signal, which is shared with the combinatorial logic section, to be made available to the storage element without increasing the number of total module inputs required. If the storage element is bypassed, the reset signal is used to implement the required combinatorial module input. In the Integrator Series, sequential and combinatorial modules are interleaved, resulting in a 50-50 mix of logic modules. This has been determined to be an optimal mix for a wide variety of designs and results in excellent utilization.

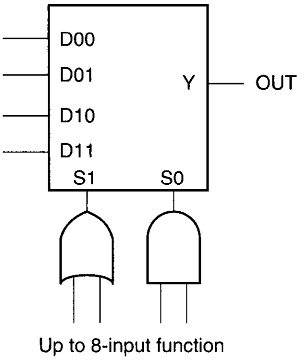


Figure 4 • Combinatorial Logic Module

Wide Decode Logic Module

Every member of the 3200DX family has a number of special logic modules optimized for implementing wide-input combinatorial logic functions directly driving device output pads. The Wide Decode Logic Module consists of a seven-input AND gate with selectable inversion on the output. The output of this module bypasses the normal routing network and connects directly to a particular output buffer. This feature minimizes the delay from the module output to the device pad and is ideal for implementing wide-decode functions typically implemented in small PLD devices. The Wide Decode Logic Module output is also available to the core logic modules, so it can be used in conjunction with other logic functions internal to the device. For more details on the Wide Decode Logic Module, see the "Integrator Series FPGAs" data sheet and the "3200DX Wide Decode Modules" application note in this data book.

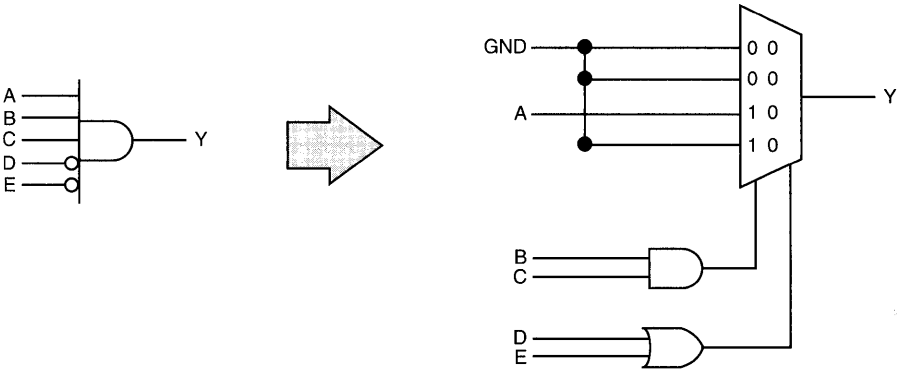


Figure 5 • Example of a Logic Function Implemented with the Combinatorial Logic Module

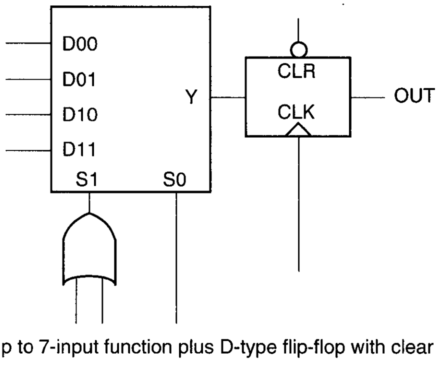


Figure 6 • Sequential Logic Module

Embedded Dual-Port SRAM

Some members of the 3200DX family include dedicated blocks of high-speed dual-port SRAM. The SRAM blocks are arranged in 256-bit blocks, which can be configured as 32 x 8 or 64 x 4. SRAM blocks can be cascaded to form wider or deeper memory blocks. The SRAM is dual ported, with separate read and write addresses, a separate data input port (for writes), and a separate data output port (for reads). Reads and writes are controlled by separate clocked read and write enables, easing timing requirements for using the SRAM. The dual-port structure is ideal for implementing FIFO, burst buffers, and internal register storage for status, control, or constant data. The 3200DX devices have from 8 SRAM blocks (on the A32100DX) to 16 SRAM blocks (on the A32400DX). For more detail on the structure and functionality of the dual-port SRAM blocks, see the

"Integrator Series FPGAs" data sheet and the "3200DX Dual-Port Random Access Memory" application note in this data book.

Enhanced Sequential Logic Module

The Enhanced Sequential Logic Module used in the ACT 3 family is a refinement of the Sequential Logic Module and is shown in the diagram in Figure 7. The reset input on the register in the sequential section is not shared with the combinatorial logic function, so the full combinatorial logic is available in the Combinatorial Logic Module to be used in front of the register. This makes all single module combinatorial logic functions usable in front of the storage element, simplifying design via schematics or synthesis inputs, and it can result in speed improvements for wide-input functions.

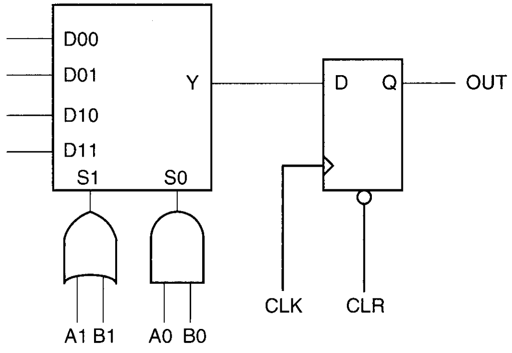


Figure 7 • S-Module Diagram

Channeled Interconnect

All Actel devices use a channeled interconnect architecture to make connections between internal logic modules and device I/O pins. This architecture is similar to that of a channeled gate array, in that horizontal tracks span the length of the array with a variety of predefined segmentation lengths. This makes a huge amount of routing resources available and ensures that signals usually have the length of available track that they require. In addition, tracks can be joined together to construct longer tracks, when required, by programming an interconnect fuse. Logic module outputs span four channels (two above and two below) and can be connected to any track. This means that most signals require only two fuses to connect any logic module output to any logic module input. There are enough routing resources available in Actel devices so that place and route is an automatic task. No hand routing is required. For more details on the interconnect resources available in Actel devices, refer to the device family data sheets.

Clocking Resources

Actel devices have a wide range of clocking flexibility. Every sequential element's clock input can be connected to regular interconnects within the channel, as well as to optimized clocking resources. Regular interconnects offer the most flexibility, allowing for thousands of potential separate clocks. Each Actel device also has dedicated clocking resources on-chip to improve clock performance and to simplify the design of sequential signals. Clocking resources can also be used, in most cases, as high-drive global signals like reset, output enable, or select signals. Each FPGA family is slightly different in the way it implements clocking functions. For more details on each type of clocking resource, refer to the associated device data sheets and application notes.

Routed Clocks

All Actel FPGA families have one or two special buffers that provide high-drive, low-skew signals and that can be used to drive any signal requiring these characteristics. These routed clocks are distributed to every routing channel and are available to every logic module. This allows a routed clock signal to be used by both Sequential and Combinatorial Logic Modules, offering maximum flexibility with slightly lower performance than dedicated clocks.

Dedicated Array Clock

The ACT 3 family has an additional clocking resource consisting of a high-speed dedicated clock buffer optimized for driving the sequential modules in the core array. This clock buffer can be driven from an external pin or from an internal signal. The dedicated array clock is optimized for driving sequential modules and can't drive storage elements built from combinatorial modules.

Dedicated I/O Clock

The ACT 3 family has another clocking resource consisting of a high-speed dedicated clock buffer optimized for driving the sequential modules in the I/O modules. This clock buffer also can be driven from an external pin or from an internal signal. The dedicated I/O clock is optimized for driving I/O modules and can't drive storage elements in the array. If all storage elements need to be driven from a common clock, the array clock and I/O clock can be connected together externally.

Quad Clocks

The 3200DX family has an additional clocking resource consisting of four special high-drive buffers called quad clocks. Each buffer provides a high-drive signal that spans about one-quarter of the device (a quadrant). These buffers can be used for fast local clocks (perhaps for prescaled shifters or counters), for wide-width mux selects, or for I/O enables. Note that since these are quadrant oriented, only a single quadrant clock can be used per quadrant. Quad clocks can be connected together internally to span up to one-half of the device. Additionally, the quad clocks can be sourced from internal signals as well as external pins. Thus they can be used as internally driven high-fanout nets.

I/O Module Descriptions

Each Actel FPGA family has a slightly different I/O module. The Simple I/O Module, found in the ACT 1 family, is optimized for low cost, and the Latched I/O Module, found in the Integrator Series, offers a balance of speed and cost (value). The Registered I/O Module in ACT 3 is optimized for high speed in synchronous applications. More details on each I/O module can be found in the associated device data sheets and application notes.

Simple I/O Module

The Simple I/O Module (Figure 8) used in the ACT 1 family was the first I/O module developed by Actel and is a simple I/O buffer with interconnections to the logic array. All input, output, and three-state control signals are available to the array. Outputs are TTL and CMOS compatible and sink or source 10 mA at TTL levels.

Latched I/O Module

The Latched I/O Module, shown in the diagram in Figure 9, is used in the Integrator Series and is slightly more complicated than the Simple I/O Module. The Latched I/O Module contains input and output latches that can be used as such or combined with internal latches to construct input or output registers. Outputs are TTL and CMOS compatible and sink or source 10 mA at TTL levels.

Registered I/O Module

The Registered I/O Module, used in the ACT 3 family devices, is optimized for speed and functionality in synchronous system designs. It contains complete registers at both the input and output paths, as shown in the diagram in Figure 10. Data can be stored in the output register (under control of the ODE, output data enable, signal), or it can bypass the register if the OTB control bit is tied low. Both the output register and the input register can be cleared or preset via the global IOPCL signal, and both are clocked via the IOCLK global signal. Notice that the output of the output register can be selected as an input to the array (on the Y signal). This allows state machines, for example, to be built right into the I/O module for fast clock-to-output requirements. The input register can be used to register the data from the I/O pad and can be enabled via the IEN signal. (IDE is used in conjunction with IEN but is usually tied low if the input register is bypassed or unused.)

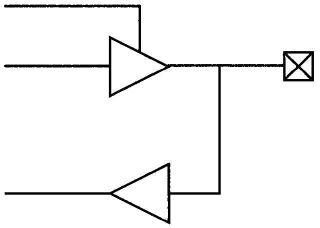


Figure 8 • Simple I/O Module

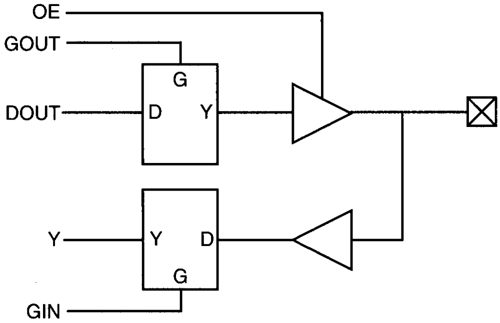


Figure 9 • Latched I/O Module

Summary

This introduction has given the reader a quick start to the architecture of Actel FPGA families. The reader can now continue exploring Actel devices and applications by turning to a variety of applications and data sheet resources.

- If you are interested in finding out more about a particular device, consult the appropriate data sheet.
- If you are interested in seeing the application of Actel devices to your particular design problem, or one similar to it, consult the table of contents of the applications sections of the data book.
- If you want to see estimates of performance and capacity for real-world functions implemented in Actel devices, turn to the applications sections to find information on estimating performance and capacity of Actel FPGAs.

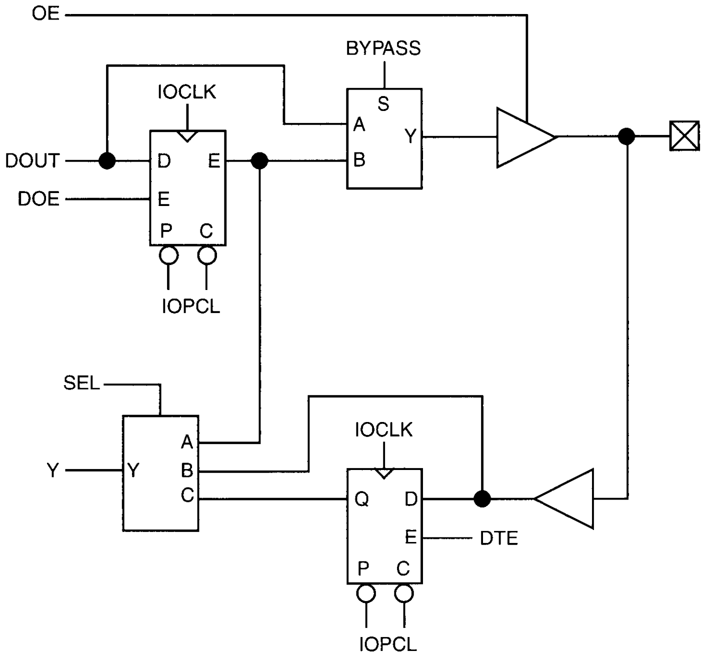


Figure 10 • *Registered I/O Module*

Designing with FPGAs Compared with SSI/MSI Devices

Field programmable gate arrays (FPGAs) are powerful devices for implementing complex digital systems. FPGAs are best used with an understanding of the key differences between FPGAs and previous logic technologies. This document focuses on FPGAs compared with small scale integration (SSI) and medium scale integration (MSI) devices. Understanding these differences and using design techniques appropriate for FPGAs results in 50 to 100 percent improvement in speed and density compared to design styles that treat FPGAs and SSI/MSI equally.

Discrete Logic Replacement

An estimation of the size of an existing SSI/MSI design may be determined by using the design's parts list. The data book lists the number of Actel logic modules needed to build each of the soft macros. Multiplying the number of logic modules needed by the quantity of any part used will give the total number of logic modules required for any one function. Consider the parts list in Table 1 for a TTL design. Select an Actel equivalent macro for each item on the parts list, and note the number of modules required. (In this case, there are S-modules and C-modules, since this is a 1200XL design.) Sum the product of macro quantity and number of modules per macro for all listed macros. In this case, the design requires 250 C-modules and 134 S-Modules, which fit in a 1225XL device with a utilization of 63 percent.

Comparing Technologies

SSI/MSI building blocks are created by optimizing the number of pins on popular functions to fit in the small packages available. Logic functions are typically constructed of a few hundred popular building blocks such as counters, multiplexers, shift registers, and comparators. The typical design is optimized to reduce package count, and techniques have evolved to make the most use of a device. For example, simple state machines are constructed from counters and decoders with appropriate pins tied to logic one or zero. This technique minimizes package count, which is the primary cost factor in SSI/MSI designs. The interconnect in these designs is done on the PC board with negligible timing delays. FPGAs, on the other hand, have abundant package pins but are constrained in routing resources, so different techniques are required.

Macro Libraries

Actel provides libraries with the basic system for popular schematic capture tools. The library contains both hard macros and soft macros. Hard macros are similar to SSI components. They form the basic functional building blocks, such as gates and flip-flops. Some Actel hard macros are identical in function to TTL devices, although they have different names. For example, Actel's TA00 is a single two input NAND gate that is a substitution for the quad two input NAND SSI device 74LS00.

Table 1 • Converting Sample Designs from TTL to Actel Macros

No.	Part no.	Description	Number of Packages	Actel Macro	Per Macro		Total	
					S-module	C-module	S-module	C-module
1	74LS161	4-bit counter	3	TA161	4	10	12	30
2	74F151	8:1 multiplexer	4	TA151	0	5	0	20
3	74LS684	8-bit mag comparator	2	MCMPC8	0	36	0	72
4	74LS377	8-bit register	6	TA377	8	0	48	0
5	74F166	8-bit shift register	8	SREG8A	8	0	64	0
6	74LS74	Dual D flip-flop	9	DFPC	1	0	18	0
7	74F113	Dual JK flip-flop	4	JKF1B	1	0	8	0
8	74F04	Inverter	2	n/a	0	0	0	0
9	74F32	Quad OR gate	2	OR2	0	1	0	8
10	74F08	Quad AND gate	1	AND2	0	1	0	4

Soft macros are more complex functions built from a number of hard macros. Some soft macro examples are counters, decoders, and adders. All soft macros are easily copied, modified, and saved in user libraries. Should you need an SSI/MSI function for which there is no equivalent in the library, it is easy to build it by creating the schematic from a TTL manual, using the Actel library. The Actel FPGA Macro Library Guide contains complete information on all the Actel library components, including a specific TTL section.

If you do not require all the outputs of a soft macro, you do not have to modify the macro. The Actel combiner program (invoked during compile) will automatically eliminate any unused modules before the design is placed and routed. All macro inputs, however, must be connected to another module output, V_{CC}, or ground. For optimal module usage, it is best to copy the macro and the underlying schematic and then edit both of these to reflect the logic reduction.

Three-State Implementation

Many SSI/MSI devices have three-state outputs allowing them to be connected to a common bus. The three-state function should be implemented with a multiplexer, which is particularly effective in an Actel FPGA. Figure 1 shows a three-state SSI/MSI implementation as well as the FPGA multiplexer implementation. An 8-bit bus with four possible drivers can be implemented with only eight logic modules, or less than 3 percent of an Actel A1010 device.

State Machine Techniques

A common state machine design technique with MSI devices uses a loadable counter to implement a state machine. Load inputs are tied to a jump address. (Sometimes logic is used if more than one jump address is needed.) The counter either counts (to advance to the next state) or loads (to jump to a different state). This is efficient in MSI devices, since it requires only a couple of packages to implement a simple state machine. Although this design technique reduces MSI package count, it results in inefficient logic usage for FPGAs. The bit-per-state technique is much more efficient and easier to design when using FPGAs. (See the "Designing State Machines for FPGAs" application note in this data book.)

Another common inefficient FPGA design technique uses a single large state machine instead of multiple communicating small state machines. In MSI devices, sometimes a single microcoded state machine controls a complex data path. This works well since large registered PROMs are available to implement a design in a small number of packages. However, these designs are complicated, since each state could have several activities occurring simultaneously, and the interactions between each activity need to be checked in every state. In FPGAs, multiple communicating state machines are easier to design, since most of the communication is local, and only a few activities need to be communicated between different state machines. The distributed machines tend to have much simpler logic

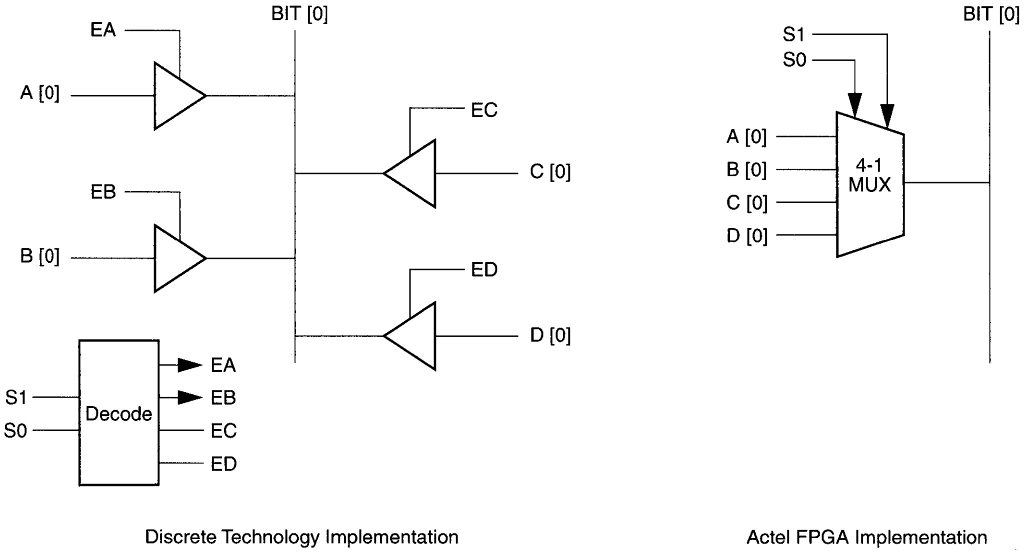


Figure 1 • Least Significant Bit of a Bus with Four Possible Drivers requirements that also fit better with the FPGAs register-rich, small-logic building-block characteristics. This approach is also better for FPGA routing, because the routing resource requirements are more distributed.

Data Path Oriented Techniques

If counters that are loaded only occasionally are required, prescaling techniques can be used to improve operating frequency. However, these techniques also result in slower load capability. Applications that need to generate long address sequences (for example, memory access) can use this load latency counter very effectively and can operate at a higher speed compared with nonlatency versions.

Nonlatency counters designed using look-ahead techniques have better performance than do MSI equivalents. These methods do not impose any additional constraints on the application, as does the load latency counter, but they take advantage of the register-rich nature of FPGAs in implementing counter functions. For example, in a 16-bit downcounter, each register should roll over after the counter reaches all zeros. Instead of detecting the all-zero case by placing combinatorial logic after the counter registers, logic can be placed in front of a register to detect the case in which the counter contains a 1 and is counting down. The register will then be active on the same cycle in which the counter contains all zeros, saving the combinatorial delay associated with the all-zero detection.

Actel provides an automated tool (ACTgen Macro Builder) to assist designers with the creation of a wide variety of counters, adders, and other datapath functions.

Random Logic Oriented Techniques

Many of the SSI-oriented techniques that designers use for random logic translate directly into FPGA devices because of the similarity of the basic building blocks. You must keep in mind that routing resources are limited inside FPGAs, whereas routing resources in SSI designs on PC boards are virtually inexhaustible. Sections of logic that use too many different clock sources and high fan-in may overly constrain routing. For example, it is usually more efficient to use a synchronous clock source with synchronous enables instead of a large number of individual clock signals to load individually selected data bits into registers because synchronous enable signals have more routing flexibility than do clock signals.

Since FPGAs are most efficient at implementing logic at the input of registers, a good rule of thumb in implementing random logic is to use logic at the input of registers instead of at the outputs wherever possible. For example, multiplexing a signal prior to a register is more efficient than multiplexing the signal after the register.

Conclusion

SSI/MSI designers can easily use FPGA technology to reap the benefits of lower cost, smaller board size, and lower power. However, using different techniques that are better suited for FPGAs will allow between 50 and 100 percent improvement in performance and capacity.

Designing with FPGAs Compared with PLD Devices

Field programmable gate arrays (FPGAs) are powerful devices for implementing complex digital systems. FPGAs are best used with an understanding of the key differences between FPGAs and previous logic technologies. This document focuses on FPGAs compared with programmable logic devices (PLDs). Understanding these differences and using design techniques appropriate for FPGAs results in 50 to 100 percent improvement in speed and density compared with design styles that treat FPGAs and PLDs equally.

Comparing Technologies

PLDs are array-oriented devices that typically have an AND-OR structure with wide-input AND gates feeding a narrower OR gate. A register is typically available at the output of each OR. This architecture is termed *logic rich* because there are typically many more logic gates than registers available. The ratio of gates to registers can be as high as 5 to 1. Because of the large delay through the wider logic module, PLDs pay a significant speed penalty when multiple levels of logic are required. Speeds tend to be more predictable in PLDs because of the larger *speed quanta*.

FPGAs on the other hand, are register rich, with a logic-to-register ratio closer to 2 to 1. (This ratio is equivalent to the traditional gate array usage and tends to be related to high density designs' need for more registers than are needed by the traditional "glue logic" oriented low-density applications.) FPGA logic structures are optimized for functions narrower than those of PLDs. FPGAs have a smaller speed quanta than do PLDs, so logic functions can be incremented in complexity while incrementing the delay only a little each time. In addition, signals that need to be fast can be sourced near the bottom of the logic tree, minimizing the number of logic levels required, and slow signals can be sourced at the top of the logic tree, where more logic levels are required.

Estimating PLD Logic Replacement

Estimating the number of logic modules needed to replace an existing PLD is straightforward. Figures 1 shows an example of a typical PLD file converted to Actel logic modules. Typical PLD functions are address decoding and state machine control. Figure 1 is a wide-register AND function decoding 25 inputs and using only 6 logic modules. Since the first AND5B gate is combined with the DFC1B flip-flop in a sequential module, the 25-bit decode is done in only one logic level. A second level could be added to provide 32-bit decode with 8 modules or 64-bit decode with 16 modules.

Figure 2 is a familiar three-product term AND/OR array using only four logic modules to implement a typical state machine equation. Figure 3 shows the product and sum terms expanded further. In the PLD equation examples, note how easily the number of either product or sum terms can be expanded. Small incremental changes in both delay and size are added each time the array is enlarged. This is in sharp contrast to the large step-function increase in delay and size when a second PLD array must be used to implement a particular equation.

PLD equation conversion shows the flexibility of the Actel logic module, implementing a variety of different combinatorial and sequential functions in few modules without wasting dedicated resources. With ACT 2, 1200XL, 3200DX, and ACT 3 devices, the cost of using a sequential function is equal to the cost of using a combinatorial function. In contrast, PLDs are rich in AND/OR gating but lacking in registers. Encoding states in a PLD requires using the smallest number of flip-flops possible, but with Actel you are free to use any combination of flip-flops and logic.

State Machine Techniques

The traditional PLD design techniques for implementing state machines are geared toward the logic-rich and register-lean architecture of the standard PLD. A small number of state registers are used (usually the theoretical minimum), since registers are scarce. This approach requires a larger amount of combinatorial logic to decode the state, but PLDs usually are able to provide enough combinatorial logic to do this effectively. Using this technique for FPGAs would not be an efficient use of FPGA strengths—numerous registers and fast narrow logic gates. A bit-per-state approach, whereby each state uses a separate register instead of encoding states in multiple registers, results in faster and more efficient state machines in FPGAs. (Refer to the "Designing State Machines for FPGAs" application note in this data book.) In many cases, speed improves by 50 to 100 percent compared with the PLD-oriented methodology of an encoded state machine.

In PLD-oriented designs, logic is typically used to develop outputs from state machines. Usually this requires an additional level of logic after the state register and adds

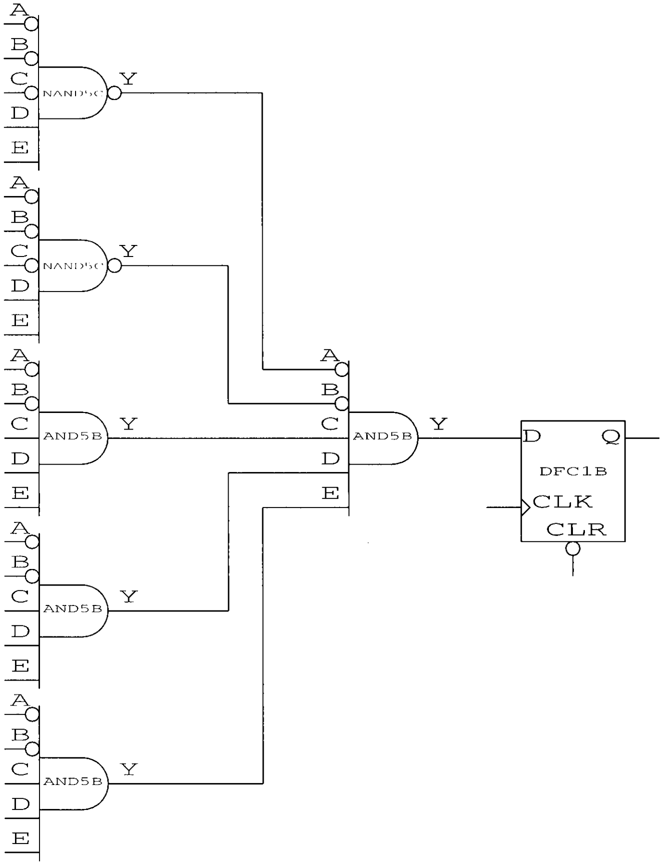


Figure 1* • *PLD Implementation with Wide Fan-In

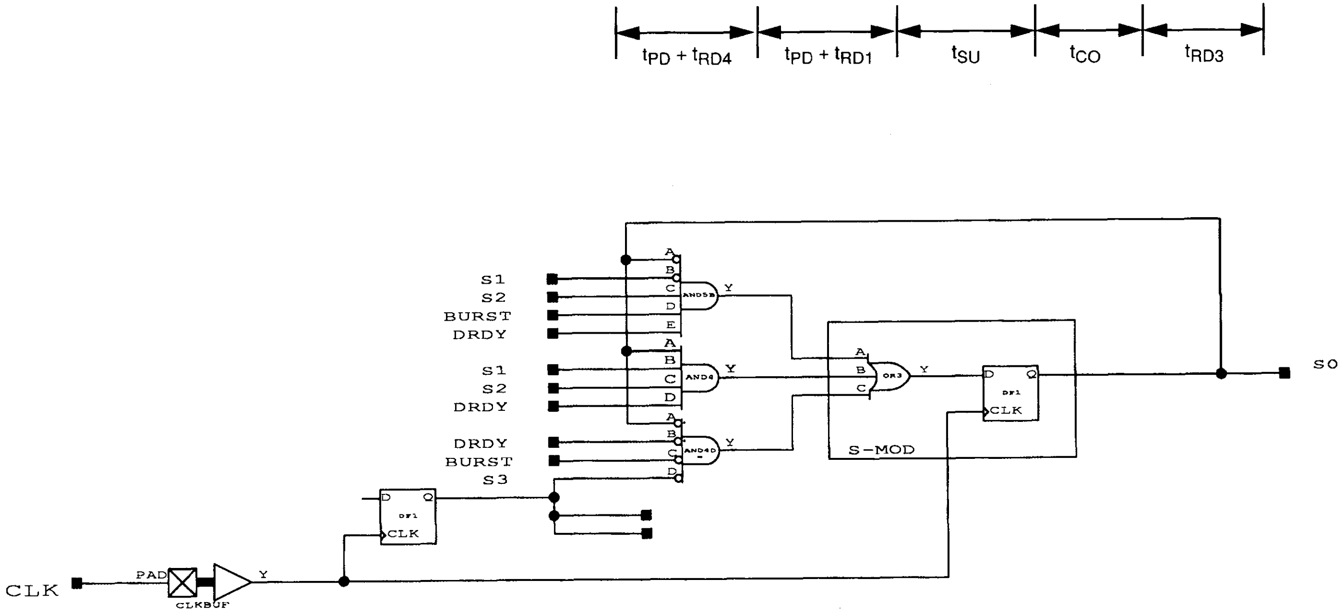
delay. In FPGAs, this level of logic can be eliminated in many cases by combining the logic in front of the state bits in which an output is active. For example, if the chip enable (CE) output from a state machine needs to be active in states 3 and 5, the logic feeding state bits 3 and 5 can be ORed together and registered to create the CE output without incurring a logic delay after the register. Since the logic in front of state bits is simple, usually no additional delay or logic resources are required in front of the new register.

Another popular state machine design technique for PLDs uses counters to generate a sequence of wait states. For example, a state machine may need to wait for 16 cycles until a data transfer can begin. A 4-bit counter can be used to generate the required state sequence. This is fairly efficient in PLD architectures because of the logic-rich and register-lean characteristics of the count function. It is not as good a fit for FPGAs, however. In FPGAs, registers are rich, and a shift register is more efficient and faster than a counter. A normal shift register will require one register per wait state. If very large delays are required, a feedback shift register can be used to implement only one state fewer than a counter, but it requires much less logic and is significantly faster.

Actel provides an automated tool (ACTgen Macro Builder) to assist designers with the creation of counters, shift registers, and feedback shift registers. These functions can be used to augment state machine designs and simplify the design process.

Conclusion

PLD designers can use FPGA technology to reap the benefits of lower cost, smaller board size, and lower power. However, using new techniques that are better suited for FPGAs will allow between 50 and 100 percent improvement in performance and capacity.



```
S0 = S2 * /S1 * /S0 * BURST * DRDY
   + /S3 * /S0 * /BURST * /DRDY
   + S2 *  S1 *  S0 * DRDY
```

Figure 2 • *Implementing State Machine*

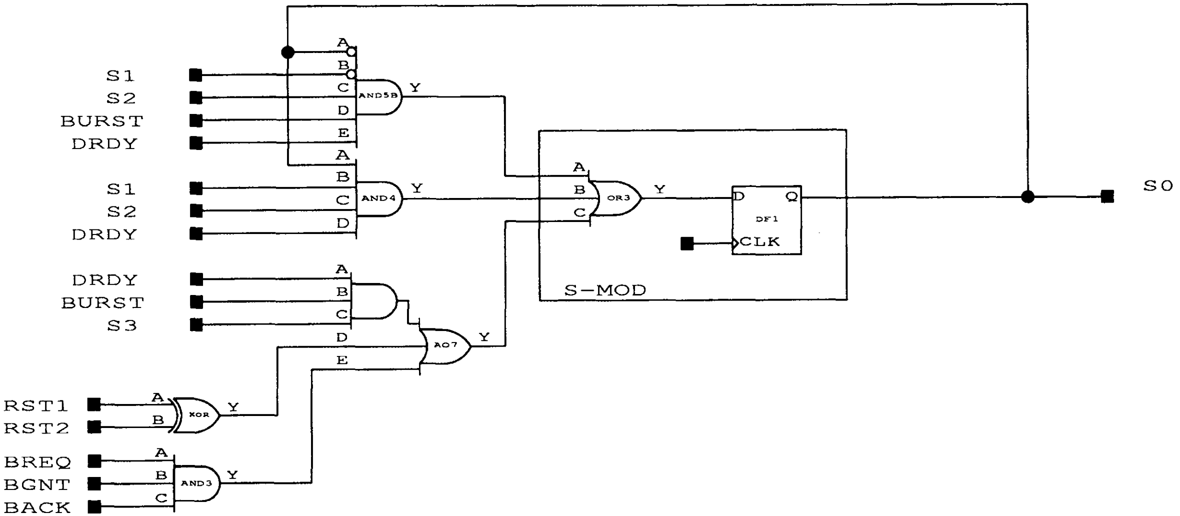


Figure 3 • Expanded State Machine

FPGA Design for ASIC-Experienced Designers

Actel FPGAs allow designers familiar with ASIC and HLD flow to make an easy transition to FPGA design. The Actel flow is similar to the typical HLD flow, but optimum results are achieved only when the designer keeps in mind a few key concepts: the Actel architecture, coding style, and methodology. This application note introduces the ASIC-knowledgeable designer to designing with Actel FPGAs and describes in detail the key concepts required to obtain good synthesis results.

FPGA Design with High-Level Design Tools

Actel FPGAs closely follow the expected ASIC design flow. HLD design source can be simulated with a behavioral simulator and a test bench. Once the design simulates correctly, a behavioral compiler can generate netlists that can be used for postsynthesis verification. If the verification is successful, the netlist can be compiled in the back-end place-and-route tools from Actel Designer Series 3.0. Post-place-and-route timing information can be backannotated for final timing verification. In addition, Designer Series 3.0 allows timing constraint entry for place and route, which can help ensure that timing targets are achieved by the place-and-route tools.

This flow should be similar to the flow that experienced ASIC designers have used. Actel FPGAs can support this flow because of the good correlation between pre-place-and-route capacity and timing estimates and post-place-and-route results. This is possible because of the abundant routing resources available on Actel FPGAs. These resources ensure that place-and-route issues don't impact capacity and performance estimates.

The ASIC designer can ensure good synthesis results by keeping a few key concepts in mind when using holds for Actel FPGAs. Actel architecture, HLD coding style, and methodology are the most important concepts and they are covered in detail in the rest of this application note.

Actel FPGA Architecture

Actel FPGA devices are based on an architecture that is inherently friendly to high-level synthesis tools. The Actel logic module is optimal in implementing logic synthesized from high-level tools. The logic module is small enough to be efficient at implementing common functions produced by synthesis tools (larger logic modules waste silicon area when only simple functions are required), but it is powerful enough to fit synthesis-produced, speed-critical functions into a single logic module. (For example, a five-input NAND gate and a 4-to-1 multiplexor each fit into a single Actel combinatorial logic module.)

Actel devices also provide an abundance of routing resources by using an antifuse interconnect element. The routing resource available on every Actel device ensures that logic placed into Actel logic modules can be efficiently interconnected automatically. No designer intervention is required. FPGA architectures that use other interconnect technologies (such as SRAM or EPROM) must limit the amount of interconnect available on the device; otherwise, device sizes would grow beyond manufacturing limits. (For example, an 8,000-gate SRAM device typically has 100,000 switching elements; the same-capacity Actel device has 700,000 elements, with a significantly smaller die size.) Thus, the optimal module size and abundant routing resources of Actel devices make them uncommonly friendly to high-level design methodologies.

The Actel 1200XL device architecture is shown in detail in Figure 1. The device is similar in structure to a channeled gate array, with rows of logic modules separated by rows of routing channels. Outputs from the logic modules enter the routing channel and can be connected to any of the routing tracks in the routing channel. Routing tracks are segmented, offering the module output a variety of possible connections. Long connections are available for signals that must span a major portion of the device, and short tracks are available for signals that need to span only a short distance. Module outputs span multiple routing channels, giving each output access to all tracks in the two channels above and below. This large interconnect flexibility makes Actel devices very routable and ensures that timing and capacity estimates made by high-level tools can be achieved by the back-end place-and-route tools.

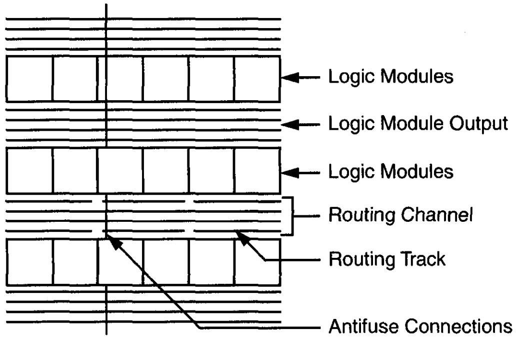


Figure 1 • Actel 1200XL Device Architecture

The logic modules used in the Actel 1200XL devices are shown in Figure 2. The combinatorial logic module is a 4-to-1 multiplexer with the addition of an AND gate on one select line and an OR gate on the other select line. This logic module can be used to construct a variety (over 760) of logic functions by connecting the appropriate logic signal and module inputs. Synthesis software automatically creates the required logic functions from these modules, so logic designers need not worry about the details of the logic implementation. (However, it will be shown that a little understanding of the multiplexer structure underlying the Actel architecture can be useful to the high-level designer.) The sequential module in the 1200XL family is similar to the combinatorial module, with the addition of a D-type register at the module output. One of the multiplexer select inputs loses an input to make the active low clear function on the register available. The register can also be turned into a level-sensitive latch, if preferred.

High-Level Design Techniques

It is inherently easy to synthesize logic for Actel devices, but the designer who understands some basics of the underlying architecture can improve the efficiency of the resulting logic. The two main categories under which these techniques fall are coding style and methodology. Coding style is an approach selected from the wealth of possible implementations available in most high-level tools to describe the required behavior of a logic function. Methodology covers the sequence and the various steps (flow) a designer selects to take a finished high-level design description and translate it into silicon.

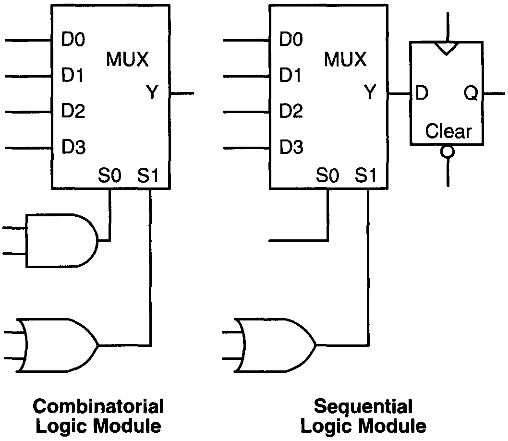


Figure 2 • Actel 1200XL Logic Modules

Coding Style

The coding style a designer uses can either help or hinder the synthesis tools in translating the desired logic function into the targeted FPGA. Only a small amount of knowledge about the underlying device architecture is needed to select between alternative approaches to defining a desired logic function. A simple but important example of this can be seen when implementing select functions in high-level languages. Two common alternative approaches to these functions (or two different coding styles) would be to use IF/THEN/ELSE statements or to use CASE statements. Each approach implements the same logic function, but the implementation in Actel FPGAs can be quite different. Figure 3 shows a 4-to-1 multiplexer described in a CASE statement in VHDL. The CASE statement in the middle of the code implements a logic function with one of the logic inputs (c, d, e, or f) selected by the 2-bit vector s4 and provided on the output m2y. The resulting implementation in an Actel logic module is also shown in Figure 3. Only a single logic module is required, since the synthesis software mapped the 4-to-1 select function into a 4-to-1 multiplexer.

Figure 4 shows an alternative implementation of the select function using IF/THEN/ELSE statements. Again, the 2-bit vector s4 is used to select one of the four inputs (c, d, e, or f). The function is identical to the function implemented previously by using the CASE statement. The resulting implementation in Actel, also shown in Figure 4, is much different, however. The IF/THEN/ELSE logic results in a multiple module implementation and increases both silicon area and delay.

A VHDL function with multiplexer coding style would be the following:

```
library ieee;
use ieee.std_logic_1164.ALL;

ENTITY mux IS
 PORT (c, d, e, f : IN std_logic;
        s4 : IN std_logic_vector(1 downto 1);
        m2y : OUT std_logic );
END mux;

ARCHITECTURE my_mux_behave OF mux IS

BEGIN

 CASE s4 IS
        WHEN "00" => m2y <= c;
        WHEN "01" => m2y <= d;
        WHEN "10" => m2y <= e;
        WHEN others => m2y <= f;
 END CASE

END PROCESS mux1

END my_mux_behave;
```

Synthesis would synthesize this VHDL as follows:

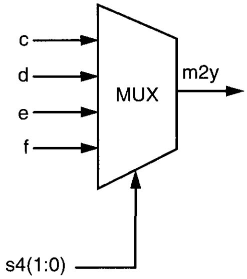


Figure 3 • *CASE Statement Example*

Thus, a designer who understands that Actel devices are optimized for multiplexer-based logic will use CASE statements when possible. This will result in the most efficient silicon implementation when targeting Actel devices, but it doesn't require any device-specific code.

Another architectural aspect of Actel devices that the knowledgeable designer should keep in mind is the structure of the register used in the synchronous logic module. The register is optimized for a D-type with an active low clear. In addition, since the register is preceded by the combinatorial logic module, many more complex register functions can be synthesized by the high-level design software. For example, Figure 5 shows the code for a clearable, D-type register with an Enable. The Enable function can be implemented by the synthesis software in a single logic module by using the multiplexer in front of the register in the synchronous logic module. Alternatively, when the designer uses random logic

The same VHDL function described previously using the CASE statement, can be written as follows:

```
library ieee;
use ieee.std_logic_1164.ALL;

ENTITY my_if_then IS
 PORT (c, d, e, f : IN std_logic;
        s4 : IN std_logic_vector(1 downto 1);
        m2y : OUT std_logic );
END my_if_then;

ARCHITECTURE my_mux_behave OF my_if_then IS

BEGIN

myif1 : PROCESS (s4, c, d, e, f)

BEGIN

if s4 = "00" then
    m2y <= c;
    elsif s4="01" then
      m2y <= d;
        elsif s4="10" then
          m2y <= e;
            else
              m2y <= f;
  endif

END PROCESS myif1;

END my_if_behave;
```

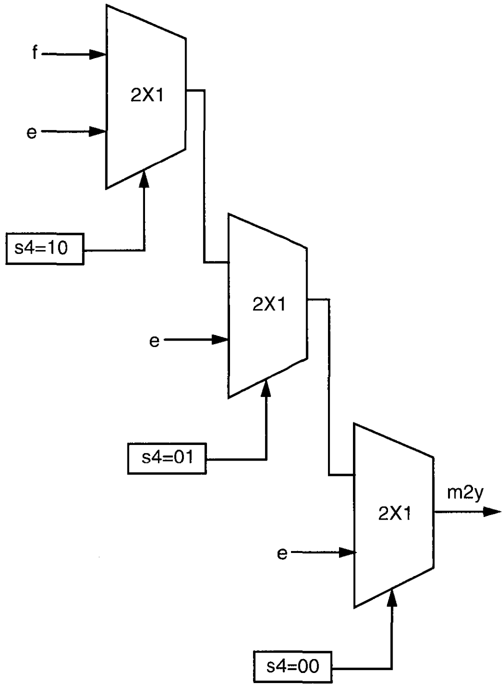


Figure 4 • *If-Then-Else Example*

in front of the register, the synthesis software will specify combinatorial logic, which can be implemented in the logic module in front of the register. Using the simple register description (simple D-type with active low clear) and CASE statements for data-path selection will ensure that functions such as a 4-to-1 multiplexer driving a D-type register efficiently map into a single Actel module.

```
IF (clear = '0') THEN
ELSIF (clk'event and clk='1') THEN
    IF (en= '1') THEN
    Q <= d;
END IF;
END IF;
```

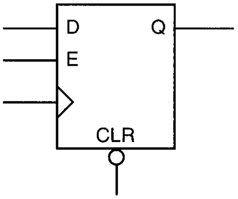
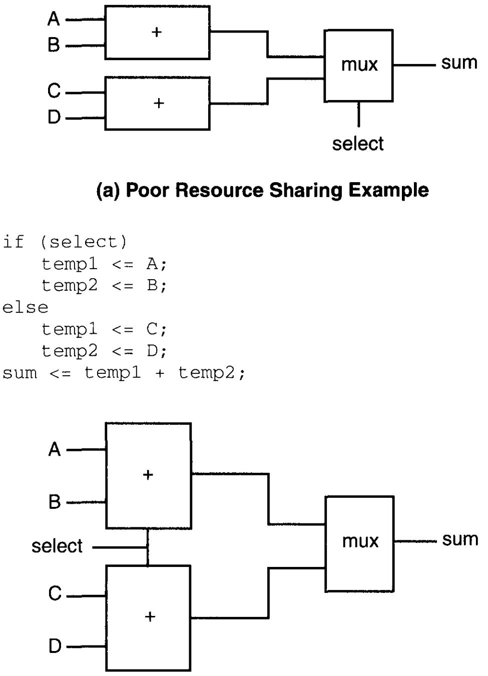


Figure 5 • Clearable D-type Register with an Enable

High-level designs often use complex data-path elements. The designer who understands the relative size and performance of these blocks can optimize the result by using proper coding style. Take for example a data path that requires the addition of two pairs of numbers. A sum needs to be generated for A_n and B_n, as well as for C_n and D_n. If the sums are not required simultaneously, the designer might decide to select the outputs after the sums are generated. The code for this approach is given in Figure 6a. First the additions are defined, and then the outputs are selected by using the IF/THEN/ELSE statement. The resulting implementation uses two adders and one multiplexer. If the designer realizes that an adder is much more expensive than a multiplexer (in terms of delay and module count), the definition can be reordered to do the selection first and then the addition. The code for this approach is shown in Figure 6b. The resulting implementation requires two multiplexers, but a single adder. This will result in considerable savings in both modules and delay. Notice that resource sharing approaches are architecturally independent. These savings would apply to almost any target technology.

As shown earlier, coding style can have a major impact on the resulting silicon implementation. The designer who understands some key aspects of the underlying technology or the resource costs for key functions can improve the results generated from high-level design tools. With Actel devices, a designer needs only to use CASE statements, simple active low clearable registers, and resource sharing to improve synthesis results. The optimal logic module and abundant routing resources will ensure that the resulting logic will be efficiently mapped into silicon.

```
if (select)
   sum <= A + B;
else
   sum <= C + D;
```



(a) Poor Resource Sharing Example

```
if (select)
   temp1 <= A;
   temp2 <= B;
else
   temp1 <= C;
   temp2 <= D;
sum <= temp1 + temp2;
```

(b) Correct Resource Sharing Example

Figure 6 • Resource Sharing Example

Methodology

The methodology a designer selects for a design can also influence the efficiency achieved with synthesis tools. For example, FPGA devices that do a minimum amount of postprocessing on the output of synthesis tools to fit the logic into the device can provide accurate capacity and performance estimates to the high-level tools. This reduces the number of iterations required to go from a high-level description to a silicon implementation of the design. Some architectures must do postprocessing on the high-level synthesis tool output to fit the logic into large logic modules, while keeping the number of inputs to the module low so that routing resources are not overtaxed. Actel devices require very little postprocessing because of the optimal module size and abundant routing resources.

Another methodology issue related to postprocessing is backannotation of delays. If there is a good correlation between the high-level logic and the postrouting fit, accurate delay estimates can be provided to the synthesis tool for timing analysis and simulation. If postprocessing has split logic to improve routability, the correlation between logic delays and the high-level logic representation will be broken, reducing the accuracy of timing simulation to the point of uselessness. In these cases, timing must be handled differently and may break the desired development flow. Again, Actel devices have a high correlation between high-level timing estimates and post-place-and-route delays, making backannotation for simulation easy and accurate.

Actel place-and-route tools are timing driven (Designer Series 3.0), and the timing constraints can be passed from the high-level tools. This allows top-level timing requirements to be easily used by the designer to guide the low-level place-and-route software, without the need to learn new tools or introduce another source of data—thus minimizing the possibility of errors. This "forward annotation" of delays is a powerful feature of the Actel design flow. Pin assignment can also be forward annotated, making it possible to specify completely the function, pinout, and timing for the target device, all at the highest level of the design.

Library support is also an important aspect of the development methodology. Actel provides synthesis vendors with special libraries composed of predesigned macros for common logic functions. In addition, key high-level functions (such as adders, counters, and FIFOs) are predesigned by Actel and can be "called" by the synthesis tools on an as-needed basis. These functions are implemented by Actel experts to be the most efficient designs possible and are generated based on parameters specified by the high-level tool (bit widths, up versus down counters, etc.).

A new approach to synthesis just now reaching the market does not use the library approach for random logic Boolean equations; instead it implements logic directly in the Actel logic module. This "libraryless" approach improves efficiency since the Actel module can create so many logic functions. Because it is unwieldy to create a library containing all possible elements, the required logic function is generated instead by using the Actel logic module directly. This improves efficiency when the function required is not a common function and would be overlooked by a library-based approach. For example, the logic function shown in Figure 7 would normally map into two logic modules, one a two-input XOR and the other a two-input OR. Synthesis software that maps directly into the Actel logic module (CM8) would find the solution where the entire function fits into a single Actel logic module. The resulting implementation, shown in Figure 8, requires a single logic module and only a single level of delay.

Conclusion

Actel devices are inherently efficient for high-level designs because of their optimized logic modules and abundant routing resources. Experienced designers have found ways to squeeze even more efficiency out of these devices by understanding the influences of coding style and methodology on the resulting silicon implementation. Simple things such as using CASE statements and active low clearable registers can get the last drop of efficiency out of a design. The ability of high-level tools to accurately estimate capacity and performance when mapping to Actel devices also increases the efficiency of using these tools. Using Actel devices and high-level design tools, a designer has the capability to move up to higher-complexity designs.

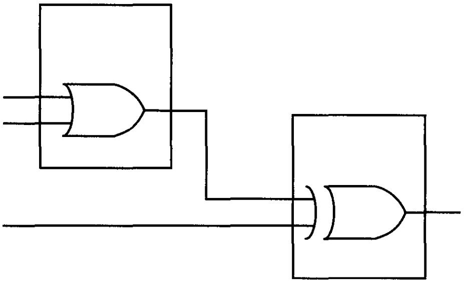


Figure 7 • Two Module Implementation

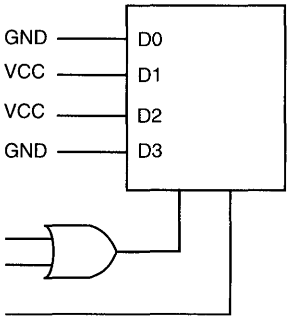


Figure 8 • Single Module CM8 Implementation

Comparing FPGA Solutions

The key to comparing FPGA solutions is to highlight all issues that can significantly impact your end product and compare how each FPGA vendor addresses these concerns. The next task is to determine the relative value of each issue and then make an informed decision regarding your programmable solution.

Issues

Issues that can impact your products include the following:

- Schedule
- Product requirements (performance, area, cost, etc.)
- Compatibility with existing tool flows

Each of these issues is key to being able to deliver competitive products in a timely manner.

Schedule (Time to Market)

If time to market is a major concern for product success, then the FPGA should fit into this requirement as well. Some of the following are keys to ensuring that the FPGA is not the bottleneck in the design process:

- Performance and capacity must be predictable. When a design is partitioned (block diagram), the various functions and performance requirements are defined for each block. Ensuring that each block can achieve these goals prior to implementation is one key for controlling a schedule. Failure to meet a goal can cause severe schedule slips, forcing design modifications and potential vendor changes.
- The vendor must support fully automatic layout (place and route) that achieves performance and capacity expectations.
- Circuit boards must be built prior to FPGA completion to minimize schedule time. (This requires manual definition of I/O locations for the FPGA.) In many FPGA architectures, manual I/O assignment can significantly impact the performance and capacity of a device; therefore, you should verify with your FPGA vendor that this strategy will have little or no impact on your capacity and performance.
- Implementation and timing verification can represent a major portion of the design process. The FPGA vendor must have the tools and macros to move quickly through this process.

Note: *Actel has always delivered automatic layout tools that provide predictable performance and capacity by leveraging the strengths of its device architecture. The flexibility of the architecture also makes it possible to assign pins manually prior to device completion with minimal impact on performance or capacity and to prevent the printed circuit board from being the schedule-limiting factor. Actel also provides an array of implementation assistants including libraries (synthesis and schematic), ACTgen Macro Builder, and ACTmap VHDL Synthesis to speed users through the function-generation period of development. Actel also provides a variety of prelayout and postlayout tools to verify performance and to debug potential timing problems quickly.*

Product Requirements

Product requirements are the basic definition of an end product. The definition may include a variety of requirements such as cost, performance, area, function, I/Os, power consumption, etc. Meeting each of these goals can be critical for the success of your products. The following breaks down some of these attributes and provide information you should have to make an informed decision regarding your FPGA solution.

Performance

If performance is the primary goal, the following approach can be used to evaluate potential solutions.

1. Determine the performance of each major function in the design. It's a good idea to track capacity information as well, since high capacity can significantly impact the performance of many FPGA architectures.
2. Perform a critical path analysis to determine whether required performance is met. If performance is not met, then speed-graded versions or optional design implementations should be considered. Remember to consider both internal and external (I/O) performance.
3. Determine from the FPGA supplier whether the functions can be effectively mapped into the silicon at your performance and capacity expectations.

Note: *Again, Actel's goal is to provide complete cost/performance information to assist in making informed decisions. Capacity information for all functions is provided in terms of C-modules and S-modules, making it easy to track capacity. Performance numbers are provided in terms of both logic module delay and average routing delay so that there are no surprises. In fact, the defined delays represent average achievable performance from the automatic layout tools, providing performance margin for most designs, even at full device capacity. See individual data sheets for performance information on simple functions and the "Macro Libraries" section of the data book for information on more complex functions.*

Area

Another important factor driving the move to ASICs and FPGAs has been the ever-decreasing size requirements for products. This is especially true in the portable applications market. If minimizing board area is an important concern, then consider the following:

- Reduce package count when possible.
- Use smaller outline packaging when possible, such as TQFP and VQFP packaging.
- Use the smallest package that still meets minimum I/O count requirements.

Note: *Actel's antifuse-based devices provide single chip solutions with no requirement for external boot (including a ROM) circuitry. Actel also supports a wide variety of TQFP and VQFP packaging to meet small outline needs. In addition, the small die size of Actel FPGAs allows us to package high-capacity devices in small packages for applications that are gate limited instead of I/O limited.*

Cost

Cost can be one of the most difficult attributes to quantify when comparing multiple FPGA vendors. The main reason is that no effective standards exist in the industry for defining device capacity or performance. Each vendor has a unique architecture that is better for some functions than for others, and each vendor defines performance of devices uniquely. Performance and cost are closely coupled. Faster speed grade versions exist, but with higher price tags. So, if cost is a primary concern, then the following steps should be followed:

1. For each major function, estimate the number of resources required and the performance of each function for each vendor of interest.
2. Perform a critical path analysis to determine whether required performance is met. If performance is not met, then speed-graded versions, optional design implementations, or relaxation of performance requirements should be considered. Remember to consider both internal and external (I/O) performance in the analysis.
3. Determine the smallest device that accommodates the major functions, and ensure that it fits cost goals. (You may want to build in some overhead for unexpected changes and additions.)
4. Determine from the FPGA supplier whether the functions can be effectively mapped into the silicon at your capacity and performance expectations.

Note: *One of Actel's goals is to provide complete cost/performance information to assist in making informed decisions. Capacity information for all functions is provided in terms of C-modules and S-modules, making it easy to determine the required device size. In addition, all AC performance numbers are provided in terms of both logic module delay and average routing delay so that there are no surprises. See individual data sheets for simple functions and the "Macro Libraries" section of the data book for information on more complex functions. The architecture supports up to 100 percent usability of all resources in the device through automatic layout by leveraging architectural strengths.*

Tool Compatibility

Companies develop specific design flows based on specific CAE vendors to solve problems that are critical to their success. In order to be an effective hardware solution, FPGA development should also fit into this overall design flow strategy.

Note: *Actel FPGA development is based on interaction with third-party CAE tools. Actel's goal is to provide value-added functions over and above CAE tools while supporting the basic needs that designers face in developing FPGAs. See the "Software Development Systems" section of the data book for additional details.*

Application Note

Estimating Performance and Capacity of Actel Devices

It is important to know, prior to getting too far into the design process, if a particular FPGA can implement the required application at the capacity and performance level of a target device. If the application does not fit, features need to be sacrificed, or a higher-capacity device (if it exists) substituted for the lower-capacity device. If the speed of the target device isn't sufficient, a higher-speed device or significant changes to the design might be needed to achieve the required performance. With Actel FPGAs, designers can estimate application performance and capacity requirements prior to the design, with minimal effort, and they can increase their confidence that the desired application will fit in the target device. This application note presents a technique, using a small library of common logic functions, to make performance and capacity estimates for Actel devices.

Performance Estimation Techniques

It is possible to make some fairly accurate estimates of the performance and capacity of Actel devices, because of the abundance of high-speed routing resources available in the devices. Estimates are less dependent on placement and routing limitations, unlike other FPGA architectures. For Actel devices, estimates for applications can be based on other estimates for common building blocks, and the combined estimates will be reasonably accurate. This technique may not work for other, less routing-intensive architectures.

A small library of common logic functions is shown in Table 1. The estimated performance of each function is given for each of Actel's FPGA families. For example, the performance of a 16-bit accumulator in the A1200XL family is around 33 MHz (for the standard, –0, speed grade). Most of these blocks are generated by Actel's ACTgen Macro Builder tool (an automated logic generator) and provide good estimates for post-place-and-route performance. The two state machines are representatives of a typical simple state machine and a more complicated state machine implementation. The small state machine is an 8-state state machine with 10 simple transition terms and 8 simple outputs. The large state machine is a 16-state state machine with 26 complex transitions and 8 outputs. More or less complex machines can be extrapolated from these two data points for more accurate performance estimates.

Table 1 *• Logic Function Performance Estimates*

Function	Family			
	ACT 1	1200XL	3200DX	ACT 3
2-input XOR*	9.3 ns	7 ns	7 ns	5.5 ns
4-input AND/OR gate*	9.3 ns	7 ns	7 ns	5.5 ns
16-bit Accumulator	20 MHz	33 MHz	33 MHz	43 MHz
16-bit Shift Register	80 MHz	159 MHz	159 MHz	182 MHz
16-bit Counter	30 MHz	45 MHz	45 MHz	57 MHz
16-bit Prescaled Counter	80 MHz	119 MHz	119 MHz	149 MHz
16-bit Register	80 MHz	159 MHz	159 MHz	182 MHz
32x8 FIFO (fast)	NA	NA	60 MHz	NA
32x8 FIFO (compact)	NA	NA	40 MHz	NA
Small State Machine	20 MHz	60 MHz	60 MHz	80 MHz
Large State Machine	12 MHz	40 MHz	40 MHz	66 MHz

** Single module delays with fanout of 4*

A Scaling Method for Speed Grades

Performance estimates for speed grade versions can be made by multiplying the standard, –0, speed grade performance by the associated scaling factor. These scaling factors are given in Table 2. For example, a –3 version of the 16-bit accumulator in the A1200XL family would be estimated at 33 MHz * 1.54, or 50 MHz. Alternatively, the designer could estimate the overall application performance based on the standard speed grade and then scale the application performance based on the speed grade scaling factors. This approach is probably preferred if the application has several key functions and is moderately to very complicated.

An Application Example

Performance estimates for an application may require a combination of logic functions. If the designer can identify the key functions in the target application, perhaps as a "back of the envelope" block diagram, estimates for each block can be made and combined as the overall estimate. Figure 1 shows a block diagram for an example application. It is a graphics controller for an LCD display. The display is bit mapped, with the image data stored in RAM. The controller interleaves simple operations on the bits (SET, RESET, XOR, XOR with mask, SHIFT, etc.) with reading the data from memory and sending it to the display.

Table 2 • Scaling Factors for FPGA Speed Grades: Frequency and Delay

	Delay Scaling				Frequency Scaling			
	Std	–1	–2	–3	Std	–1	–2	–3
A1010B/A1020B	1.00	0.85	0.75	0.65	1.00	1.18	1.33	1.54
1200XL	1.00	0.85	0.75	0.65	1.00	1.18	1.33	1.54
3200DX	1.00	0.85	0.75	0.65	1.00	1.18	1.33	1.54
A1400A	1.00	0.85	0.75	0.65	1.00	1.18	1.33	1.54

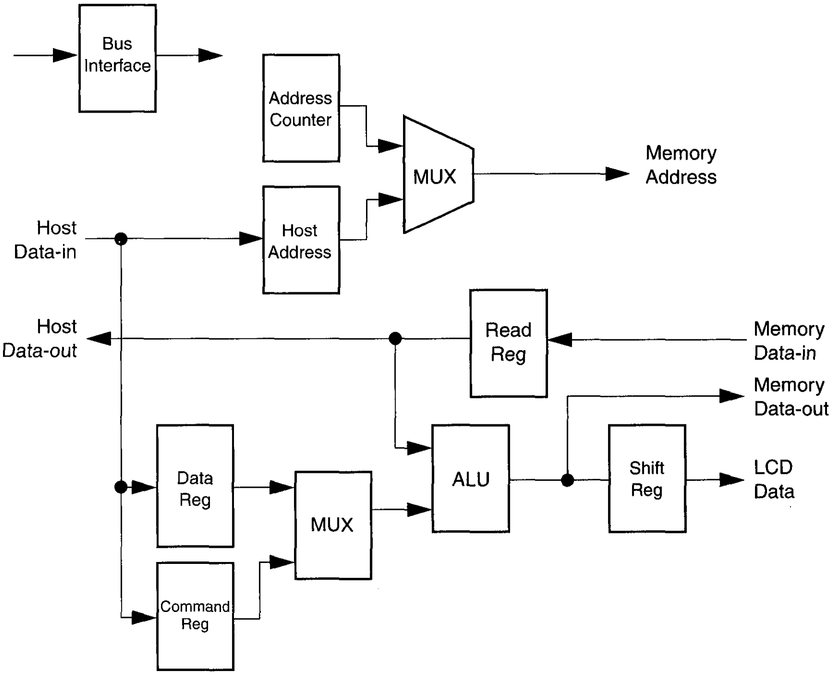


Figure 1 • LCD Controller Block Diagram

The address counter is used to address the memory during display operations. A multiplexer is used to select between display addresses and the host address register. The desired address is loaded into the memory address register. The addressed value is read from memory and loaded into the serial shift register, which provides a serial bit stream to the display. Bit operations are executed in the ALU, with data sources from the memory register, the host data register, and the bit mask register. Notice that all registers can be read by the host processor over the host data bus (via the multiplexer) to simplify coding and debugging. A small state machine controls the host-controller interface and provides the required timing for the display.

The key function blocks are given in Table 3. Associated with each block is the closest equivalent function from Table 1. The performance of that function in a particular device family can be estimated directly from Table 1. For example, the ALU is most similar to the 16-bit accumulator in Table 1, so its performance estimate is 35 MHz in ACT 3 with the standard speed grade. Other family's estimates are also listed in Table 3.

Estimates can also be made by using the scaling factors in Table 2. This allows comparisons between high-speed devices and standard speed devices. For example, a –3 speed grade 3200DX device would have around a 50 MHz ALU estimated performance (1.54 for –3 speed grade x 33 MHz standard speed grade performance estimate).

Note that more extensive performance estimation tables are available in the "Macro Libraries" section in this data book. In addition, the ACTgen Macro Builder Test Drive can be used to evaluate the capacity and performance of particular functions not tabulated. The ACTgen Test Drive is available from Actel free of charge. Finally, for the most up-to-date tables, including future family estimates, visit Actel's World Wide Web Homepage at www.actel.com.

Capacity Estimation Techniques

Capacity estimates can be made by using a technique similar to that used for estimating performance. Table 4 gives estimates for the module count for a number of common logic building blocks in each Actel family of devices. Module counts for larger building blocks can be easily estimated. Capacity for each block can be estimated by finding the closest corresponding function in Table 4. The total capacity estimate can be found by simply summing the estimates for all blocks. In some cases, the capacity estimates depend on whether the functions need to be high-speed. Actel devices allow higher speed implementations for some logic functions by using extra logic modules. Make sure you select the module count based on the expected performance requirements. For example, a pre-16-bit counter in ACT 3 requires more modules than does the regular 16-bit counter. Table 4 gives the total capacity estimates for the example application.

Table 3 • *Associated Block Estimated Performance for Standard Speed Devices*

Block	Estimated Block	ACT 1	1200XL	3200DX	ACT 3
ALU	16-bit Accumulator	20 MHz	33 MHz	33 MHz	43 MHz
Address Counter	16-bit Counter	30 MHz	45 MHz	45 MHz	57 MHz
Shift Register	Shift Register	80 MHz	159 MHz	159 MHz	182 MHz
Bus State Machine	Small State Machine	20 MHz	60 MHz	60 MHz	80 MHz

Table 4 • *Capacity Estimates for the Example Application*

Application Block	Estimated Block	ACT 1	1200XL	3200DX	ACT 3
Address Counter	Counter	64	46	46	46
ALU	Accumulator	100	92	92	92
Shift Register	Shift Register	40	21	21	21
Bus State Machine	Small State Machine	32	18	18	16
Data Registers (4)	Register	64	64	64	64
Multiplexers (4)	Multiplexer	64	80	80	80
Total		364	321	321	319

Details for the Expert

These techniques are estimates only and leave out some of the details associated with the architecture of Actel devices. These details are usually second-order effects, but even better estimates can be made if these factors are kept in mind. Try them after you are familiar with the simple estimation techniques.

- Clock frequency limit performance in some very simple blocks, like shift registers. Higher-capacity devices in a family have slightly slower maximum clock frequencies. If a simple logic function must operate at a very high speed, consult the appropriate device data sheet after selection to determine if the maximum performance is affected.
- Logic modules can sometimes be combined in Actel FPGAs when logic functions are connected. For example, a simple register and a 4-to-1 multiplexer can be combined into a single module. If you have many multiplexers and registers in a data path, count only one block for capacity purposes. The other block comes free!
- When estimating capacity for wider data paths, add a few extra percentage points to account for buffering on common signals. For example, a 32-bit 4-to-1 multiplexer could be estimated at about 210 percent of a 16-bit version. This is usually a second-order effect, however, and can be ignored unless it applies to a large percentage of the design.
- When logic modules are combined, propagation delays are also reduced. Thus, the performance of a 4-to-1 multiplexer in front of a simple register isn't just the sum of the two. If you take the longest delay and add one-half of the smaller, you will get a good estimate for the expected delay after combining.
- High-performance designs sometimes require logic duplication to keep fanout low. If you have a high-performance design, reserve 10 to 20 percent of the device logic modules to ensure that logic duplication can be implemented.
- Actel devices are composed of two different types of logic modules. Sequential logic modules include a dedicated flip-flop; combinatorial modules don't. If you have a register-rich design, you may need to estimate the total number of registers and check your target device to ensure that enough registers are present. Since Actel devices are composed of about a 50-50 mix of sequential logic modules and combinatorial logic modules, it is unusual to run out of registers, but it can happen.

Conclusion

Actel devices have good performance and capacity predictability because of the large amount of routing resources available on the device. This allows the performance and estimation techniques presented here to be used to determine the application capacity and performance fit prior to the detailed design of the application. A small amount of time put into doing this estimation can result in a much better choice for the target family.

3200DX Dual-Port Random Access Memory (RAM)

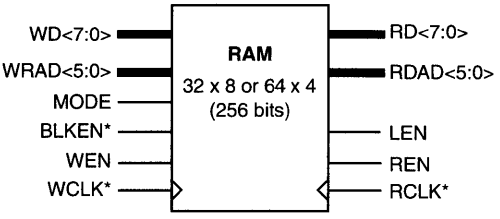
With the dramatic changes in the processing power of microprocessors and microcontrollers, peripheral components and systems have become the bottleneck of system performance. An efficient peripheral system must be capable of transferring data in high-speed bursts. Thus, the CPU can optimize the system performance by accessing the system bus with minimal interruption. A common method of increasing data throughput to an I/O interface while minimizing CPU wait states is to use first-in-first-out (FIFO) queues to buffer transferred data between the host bus and the peripheral systems. Some common applications using FIFO buffers are SCSI and IDE interfaces, bus-width conversion applications (e.g., 8-bit to 32-bit conversions), and asynchronous transfer mode (ATM) network interface cards. Typically, FIFOs are either implemented in discrete logic devices or designed using standard dual-port RAM devices with the RAM control logic implemented in PALs or PLDs.

The 3200DX family provides for a system's memory needs by supplying blocks of synchronous dual-port SRAM (Figure 1). Other FPGAs offer memory, but at a high cost of using their SRAM programming elements. Using the SRAM programming elements depletes the already constrained routing resources.

The 3200DX family offers dual-port SRAM capable of supporting a system speed of 100 MHz. The SRAM comes in blocks with a selectable 32 x 8-bit or 64 x 4-bit configuration. Designers can also connect the SRAM blocks to build wider (x16, x32, etc.) or deeper (128, 256, etc.) banks of memory. If the design's needs are smaller, however, the remaining SRAM is not necessarily wasted. It can also serve as internal logic registers. The block structure of the 3200DX SRAM makes it particularly suitable for data-path designs, in which packets of data must be handled as a unit.

The SRAM of the 3200DX family offers dual independent read and write ports. The dual-port structure makes the memory suitable for FIFO applications such as telecommunication framers. Because the two ports have independent clocks, the memory can also serve in video and graphics applications, in which data arrives in bursts and is read out at a steady rate. Furthermore, the SRAM blocks will be fully supported by Actel's macro builder, ACTgen. Using ACTgen, the user can generate parameterized dual-port memory and FIFO blocks simply by clicking menu selections.

Figure 1 shows the 3200DX dual-port SRAM block interface. Each RAM block contains 256 bits. These 256 bits can be configured as 32 bytes (32 x 8), or as 64 nibbles (64 x 4). The following is a brief functional overview of the 3200DX SRAM blocks.



**signals with polarity selection*

Figure 1* • *3200DX Dual-Port SRAM Block Interface

Write Operations

The write port is always synchronous. Data is written into the RAM on the rising edge of WCLK (Write Clock) whenever BLKEN (Block Enable) and WEN (Write Enable) are both active. WCLK and BLKEN have polarity selection. Write addresses (WRAD[5:0]), write data (WD[7:0]), BLKEN, and WEN are synchronized to the appropriate edge of WCLK. The MODE input is used to choose between byte and nibble modes. In byte mode, WRAD5 is not used, as 5 bits can address the 32 bytes of each block. In nibble mode, WRAD5 is used and the data inputs are connected in pairs (WD0 and WD4 to LSB, WD1 and WD5 to next higher-order bit, etc.).

The SRAM blocks can be cascaded together to create deeper blocks of memory using the ACTgen Macro Builder.

Read Operations

There are two modes of operation at the READ port. In synchronous mode, read addresses (RDAD[5:0]) are synchronized to the read clock (RCLK), and the outputs change in response to a rising or falling edge of this clock. In asynchronous mode, all internal latches are forced transparent and read data outputs (RD[7:0]) change in response to a change of RDAD (Read Address). LEN is used to choose synchronous or asynchronous mode. Read operations occur on RCLK or RDAD and change whenever REN is HIGH. When REN is LOW, the current state of the output is held.

Table 1 summarizes the 3200DX family features, including the dual-port SRAM in each device.

Table 1 • 3200DX Family

	3265DX	32100DX	32140DX	32200DX	32300DX
SRAM bits	0	2,048	0	2,560	3,072
Global Clocks	2	2	2	2	2
Quadrant Clocks	0	4	0	4	4
I/O_{max}	126	152	176	202	250
Wide Decode Cells	20	20	24	24	28
JTAG	No	Yes	Yes	Yes	Yes

3200DX Wide Decode Modules

When simple programmable logic devices (PLDs) first became available, designers quickly embraced the new technology. Programmable logic allowed designers to develop and test logic designs quickly and immediately correct design errors. Soon they began to demand larger, more flexible devices so that they could incorporate much of their system into programmable logic. Field Programmable Gate Arrays (FPGAs) extended the programmable logic concept to become more adaptable and encompass many thousands of gates. The increased size, however, did not solve all the system designers' problems. The richness of the FPGA as a design element came at the cost of speed for simpler functions such as address decoding.

Today's digital system designers have a tough job. Their circuits must perform complex and diverse tasks at ever-increasing clock speeds. Further, system designs often need small amounts of fast random-access memory (RAM), which conventional FPGAs don't offer. To date, whatever is available on programmable logic is a simple by-product of the architecture and has limited applicability to system needs. The result has been that system designers have had to add fast PLDs or small SRAMs to their FPGA-based designs in order to meet speed-critical decoding and memory needs. Actel's 3200DX family addresses FPGA limitations that designers have had to work around. In addition to providing high capacities (up to 40,000 gates), this family eliminates the need for additional ICs, because it includes system logic elements such as fast decode and dual-port SRAM blocks.

Address Decoding

Introduction of faster processor-based systems dictates high-speed interface requirements between the modules within these systems. One of the most generic interface issues in microprocessor-based systems is the address decoding task. Figure 1 shows a typical system consisting of a microprocessor and peripheral components. Typically, the N-bit-wide processor address line, along with some control signals, must be decoded. The address is decoded by an address decoder, which generates a chip-select signal for each device in the system. The performance of the address decoder is critical, since the present address on the address line is valid for only a short duration (depending on the system requirements and timing specifications). With the 3200DX devices, a designer no longer needs to implement the address decoder block outside of the FPGA. (The gray box in Figure 1 shows only a few possible blocks suitable to be implemented within a 3200DX device.)

3200DX Family

Actel's 3200DX family of devices provides an array of features such as internal dual-port SRAM, Wide Decode (WD) modules, quadrant clocks, and JTAG compliance. Table 1 summarizes the 3200DX product family.

The WD modules are located at the top and bottom rows of the chip and are indicated by shaded blocks in Figure 2. The modules are accessed through one of three library hard macros. These library elements are shown in Figure 3. One main feature of the WD module is the direct connection to an I/O pin. The module can drive an I/O pin without the usual interconnect delay associated with regular logic modules. To implement the direct I/O connection, the user needs to connect an output buffer of any type (OUTBUF, TRIBUFF, etc.) to the output of the Wide Decode library element (Figure 4). In addition, the output of the WD modules can be routed within the chip for internal connections. Each WD module has a specific I/O module associated with it. Only that I/O module will accommodate its associated WD module with a direct I/O connection. Actel's software will automatically place and route the cell mapped into a WD module to its associated I/O pin.

When to Use the WD Modules

The WD modules offer two main advantages not provided by other Actel library elements:

- A direct I/O connection (not routed) with 0.5 ns delay
- Seven-input AND/NAND function implemented in one logic module

The wide AND/NAND functionality, among other advantages, makes the WD modules suitable for decoding of wide address lines up to 35 bits.

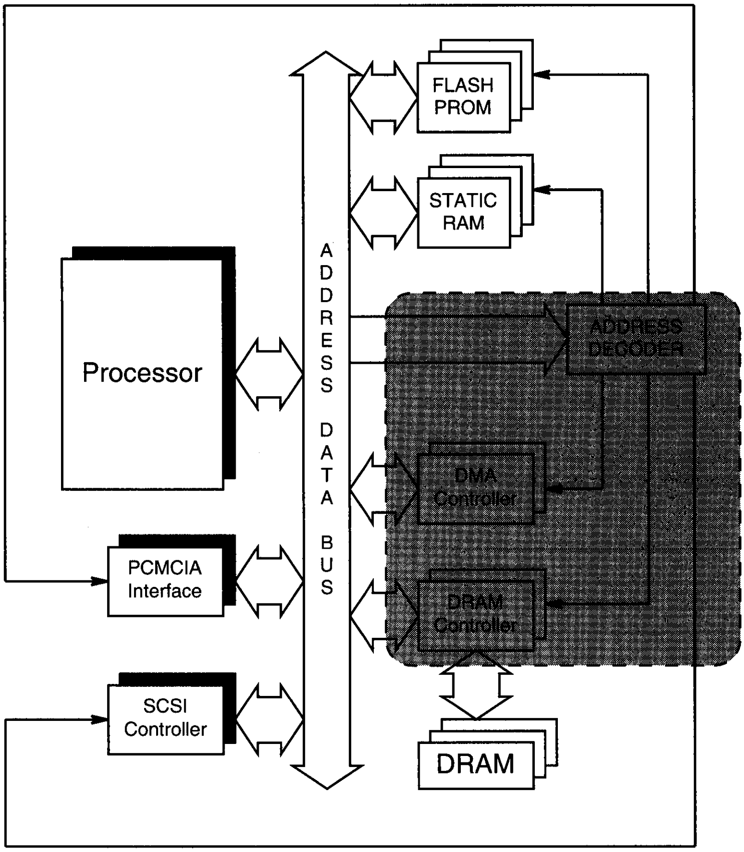


Figure 1 • *Microprocessor-Based System Block Diagram*

Table 1 • *3200DX Family*

	3265DX	32100DX	32140DX	32200DX	32300DX
Gates	6,503	9,420	14,210	20,140	30,735
C-mod	475	662	912	1,184	1,133
S-mod	510	700	954	1,230	1,888
I/O max	126	152	176	202	250
Wide Decode cells	20	20	24	24	28
JTAG	No	Yes	Yes	Yes	Yes
SRAM bits	0	2,048	0	2,560	3,072

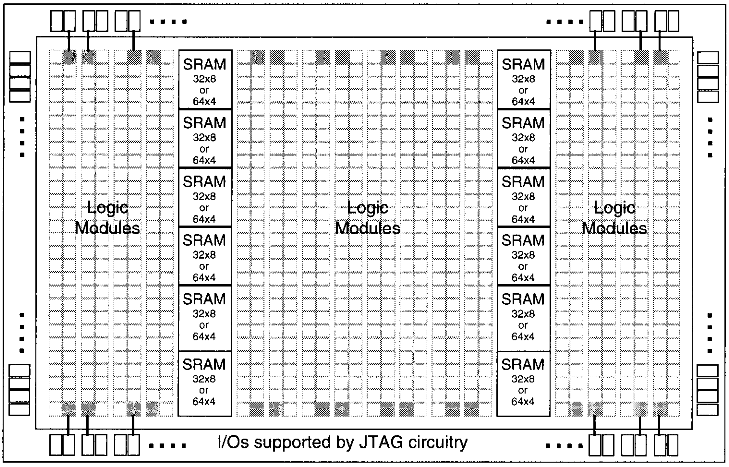


Figure 2 • *3200DX Floor Plan*

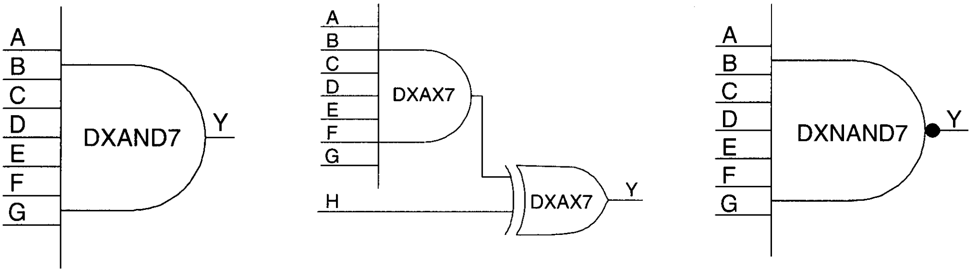


Figure 3 • *Wide Decode Library Elements*

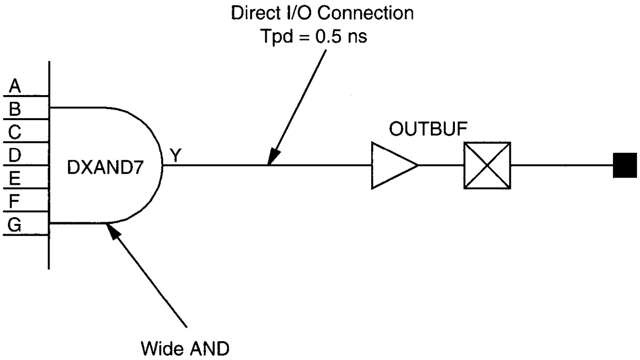


Figure 4 • *WD Module Direct I/O Connection*

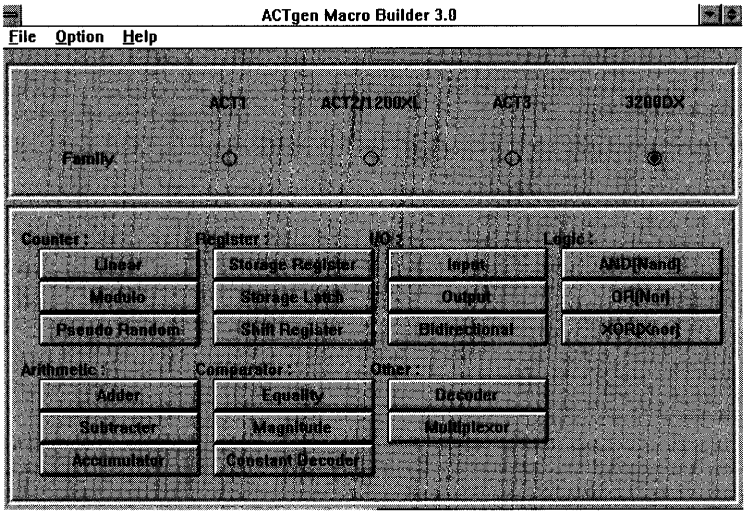
The WD module applications reach beyond just address decoders. In general, any critical path of a design in which a wider gate would save a level of logic is an appropriate place to use these modules. Also, if a critical path in your design requires a fast connection to an output pin, the WD modules can be used to accommodate the situation. Another feature that may be useful in an application is the WD module's internal Exclusive-OR. This feature can be exploited by using the DXAX7 library macro (Figure 3).

Implementing Address Decoders

The best way to take advantage of the WD modules when implementing address decoders is to use Actel's macro builder, ACTgen. Figure 5 shows the ACTgen 3.0 graphical user interface. Clicking the Constant Decoder button causes ACTgen's Constant Decoder window to appear, as shown in Figure 6.

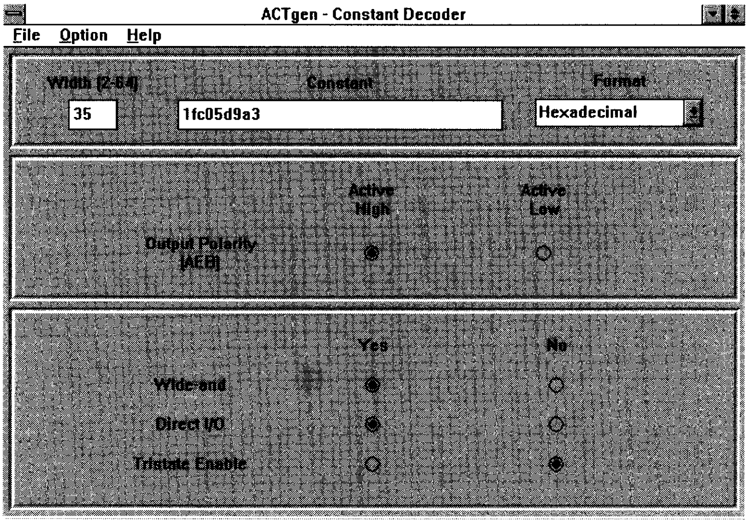
From the Constant Decoder menu, you can assign any bit width from 2 to 64. You can enter the constant to be decoded in hexadecimal, decimal, or binary. The address decoder's output sense, along with tristate options, can also be selected by simply clicking the appropriate ACTgen Constant Decoder button.

Constant decoders (or any other block) generated by ACTgen can be used in schematics or high-level language designs. Furthermore, any logic block generated by ACTgen does not need to be simulated functionally at the block level. Only the interface with other blocks needs to be simulated.



	ACT1	ACT2/1200XL	ACT3	3200DX
Family	○	○	○	●

Figure 5 • *Actel Marco Builder, ACTgen 3.0*



Width [2-64]: 35

Constant: 1fc05d9a3

Format: Hexadecimal

	Active High	Active Low
Output Polarity [AEB]	●	○

	Yes	No
Wide-and	●	○
Direct I/O	●	○
Tristate Enable	○	●

Figure 6 • ACTgen Constant Decoder

IEEE Standard 1149.1 (JTAG) in the 3200DX Family

Introduction

Due to the increasing complexity of circuit boards, testing loaded boards is becoming prohibitively expensive and more difficult to perform. Board complexity has resulted from the rapid development of surface-mount technology and from the use of multilayered boards. Finer pin spacing and the use of double-sided boards also have contributed to the increased cost and difficulty of traditional testing, which makes use of methods such as in-circuit testing by bed-of-nails and functional testing. Although functional testing can cope with complex and dense boards, it is costly because different designs require different sets of test programs.

A new method of testing reduces the cost and difficulty of board-level testing. The new standard was proposed and developed by the Joint Test Action Group (JTAG) and later adopted by IEEE as the *IEEE Standard Test Access Port and Boundary-Scan Architecture* also referred to as *IEEE Std. 1149.1* or informally as *JTAG*.

The standard provides a cost-effective method of board testing through use of the boundary-scan technique. Boundary scan provides the means to test each component's required performance, interconnections, and interaction. In addition to describing boundary scan, the standard also describes the design-for-test feature.

Overview

The Actel 3200DX family is fully compliant with the IEEE Standard 1149.1. Figure 1 shows the major parts that make up the JTAG test logic circuit. The circuit provides the required components (Test Access Port (TAP) controller and registers) to support all the mandatory boundary-scan instructions (EXTEST, SAMPLE/PRELOAD, BYPASS) and two optional instructions (HIGHZ and CLAMP). The JTAG test logic circuit also supports two private instructions, USER INSTRUCTION, and JPROBE.

JTAG Activation

The JTAG test logic circuit is activated in the Designer Series software by selecting Options/Set Die & Package. This brings up the Device Selection dialog box as shown in Figure 2. Click the Restrict JTAG Pins check box and then click the OK button. During device programming a special fuse called the *J-Fuse* is programmed, enabling the JTAG test logic circuit. The JTAG test logic circuit can also be activated when using the Actel Script language by adding the command:

```
Set ("RESTRICTJTAGPINS", "YES");
```

Test Access Port (TAP)

Each test logic function is accessed through the Test Access Port. There are four pins associated with the TAP, and they are listed in Table 1 with their corresponding descriptions. These pins are dedicated pins, which are used only with the test logic. If JTAG is not enabled, the TAP ports (TMS, TCK, and TDI) are free to be used as a regular I/O. The test logic has been designed to be in the reset state upon power-up.

Table 1 • Test Access Port

Port	Description
TCK (Test Clock Input)	Dedicated test logic clock used serially to shift test instruction, test data, and control inputs on the rising edge of the clock, and serially to shift the output data on the falling edge of the clock.
TMS (Test Mode Select)	Serial input for the test logic control bits. Data is captured on the rising edge of the test logic clock. This pin is equipped with a pull-up resistor to place the test logic in the Reset state when no input is present.
TDI (Test Data Input)	Serial input for instruction and test data. Data is captured on the rising edge of the test logic clock. This pin is equipped with a pull-up resistor.
TDO (Test Data Output)	Serial output for test instruction and data from the test logic. TDO is set to an Inactive Drive state (high impedance) when data scanning is not in progress.

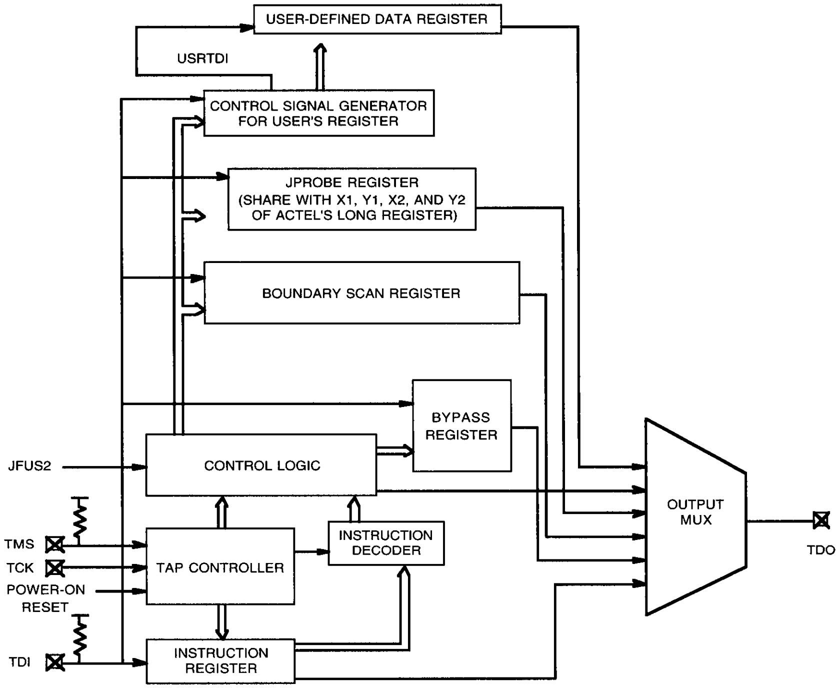
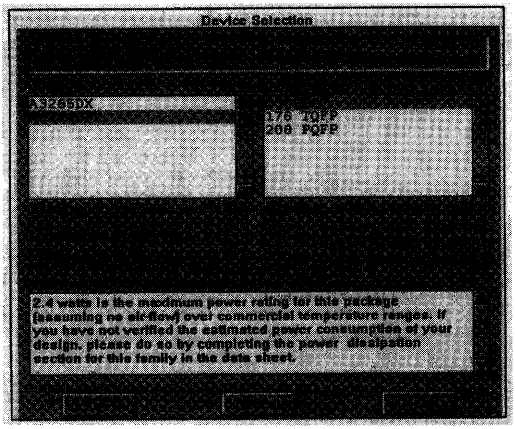


Figure 1 • *JTAG Block Diagram*



A3265DX

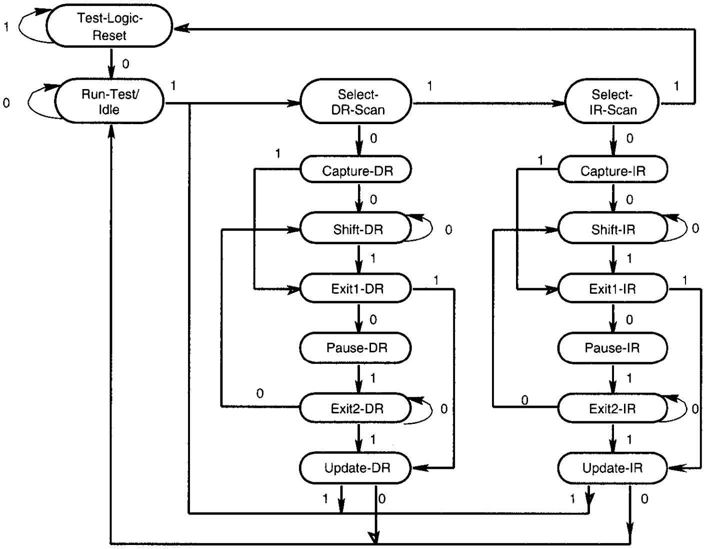
176 TQFP
208 PQFP

2.4 watts is the maximum power rating for this package (assuming no air-flow) over commercial temperature ranges. If you have not verified the estimated power consumption of your design, please do so by completing the power dissipation section for this family in the data sheet.

Figure 2 • *Device Selection Dialog Box*

TAP Controller

The TAP controller is a 16-state state machine that performs according to the state diagram shown in Figure 3. The 1's and 0's shown adjacent to the state transitions represent the TMS values that must be present at the time of a rising edge at TCK for a state transition to occur. In the states that include the letters -IR, the instruction register operates; in the states that contain the letters -DR, the test data register operates (bypass, boundary-scan, and JPROBE registers).



Note: *The value shown adjacent to the state transitions in this figure represents the signal present at TMS at the time of the rising edge at TCK.*

Figure 3 *• TAP Controller State Diagram*

The TAP controller receives two control inputs, TMS and TCK, and generates control and clock signals for the rest of the test logic architecture, as illustrated in Figure 4. The TAP controller's state changes based on the value of TMS, and at the rising edge of TCK or on power-up. Upon power-up, the TAP controller enters the Test-Logic Reset state. Since the TMS pin is equipped with a pull-up resistor, the TAP controller will remain in or return to the Test-Logic-Reset state when there is no input or when a logical 1 is on the TMS pin. To reset the controller, TMS must be high for at least five TCK cycles.

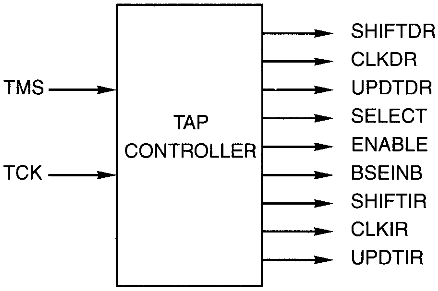


Figure 4 *• TAP Controller Block Diagram*

Instruction Register

The instruction register (IR) consists of three IR cells. Each cell has a shift-register stage and a latch stage (Figure 5). On the Capture-IR state, the shift register is loaded with bits 001, which are used for fault isolation of the board-level serial test data path. The TDI-IR-TDO path is established on the Shift-IR state. Data in the shift register is shifted toward TDO, and data in the latch remains the same. The data in the shift registers is latched out and becomes the current instruction on the falling edge of the TCK in the Update-IR state. When the TAP controller enters the Test-Logic Reset state, bits 111 are latched in IR, which corresponds to the BYPASS instruction, and the data in the shift register cell retain their previous values. Table 2 shows the summary of the operation of the instruction register.

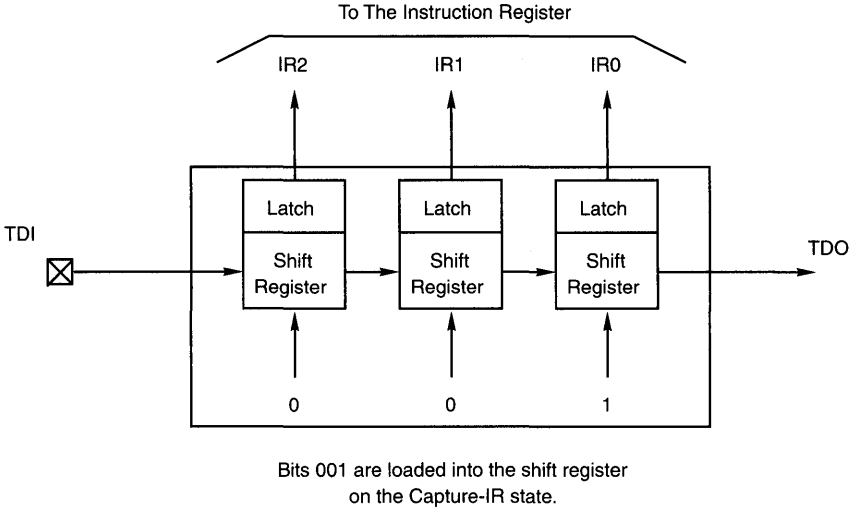

To The Instruction Register
IR2
IR1
IR0
TDI
Latch
Shift Register
Latch
Shift Register
Latch
Shift Register
TDO
0
0
1
Bits 001 are loaded into the shift register on the Capture-IR state.

Figure 5 • *Instruction Register Block Diagram*

Table 2 • *Instruction Register Operation*

Controller State	Shift-Register Stage	Latch Stage
Test-Logic-Reset	Undefined	BYPASS Instruction IR2 IR1 IRO = 1 1 1
Capture-IR	001 is loaded	Retain previous state
Shift-IR	Shift data toward TDO	Retain previous state
Exit1-IR Exit2-IR Pause-IR	Retain previous state	Retain previous state
Update-IR	Retain previous state	Latch data from the shift register
All other states	Undefined	Retain previous state

Instructions

Table 3 lists the supported instructions with their corresponding IR codes and descriptions.

Bypass Register

The bypass register is a single-bit register that provides a minimum data path between the TDI and TDO pins (Figure 6). The bypass register is selected when the BYPASS, HIGHZ, or CLAMP instruction is the current instruction in the instruction register.

On the Capture-DR controller state, 0 is loaded into the bypass register. Test data can then be shifted from the TDI to the TDO pin on the Shift-DR state. By moving into the Update-DR controller state, data movement through the bypass register is terminated. Table 4 shows the summary of the operation of the bypass register.

Table 4 • Bypass Register Operation

Controller State	Bypass Register
Test-Logic-Reset	Retain previous state
Capture-DR	0 is loaded
Shift-DR	Shift data toward TDO
Exit1-DR Exit2-DR Pause-DR	Retain previous state
Update-DR	Retain previous state
All other states	Undefined

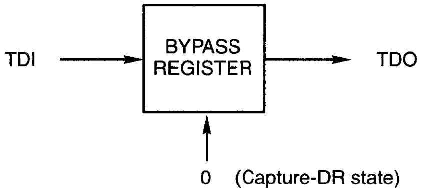


Figure 6 • Bypass Register Block Diagram

Table 3 • Supported Instructions

Instruction	IR Code = (IR2 IR1 IR0) Instruction Type	Description
EXTEST	(000) Mandatory	Permits board-level interconnect testing by applying specific test data at the output pins to the external interconnection and capturing the test result at the input pins.
SAMPLE/PRELOAD	(001) Mandatory	Used to capture data from the input pins and to apply test data into the latched parallel outputs of the boundary-scan register while the on-chip system logic is in normal operation.
JPROBE	(011) Private	Permits access to Actel's probe register for system logic signal probing.
USER INSTRUCTION	(100) Reserved	Permits access to the application-specific test data register designed by the user with Actel's logic modules.
HIGHZ	(101) Optional	Used to place the component's system logic outputs into an inactive drive state—high-impedance state.
CLAMP	(110) Optional	Permits the use of data in the boundary-scan register to be driven from component pins while the bypass register is selected as the serial path between TDI and TDO.
BYPASS	(111) Mandatory	Permits access to the single-bit shift register data path between the TDI and TDO pins to facilitate rapid movement of data through a component when the component does not require test operation.

Boundary-Scan Register

The boundary-scan register is used to observe and control the state of each system pin, including the clock pins. Each boundary-scan cell consists of serial input (SI) and serial output (SO) that are connected to each cell, as shown in Figure 7. In addition, each cell consists of a parallel input (PI) and a latched parallel output (PO) that connect to the system logic and system output. Three cells are used for each I/O: an input cell (BS2), an output cell (BS1), and an output-enable cell (BS0).

The operation of the boundary-scan register under specific boundary-scan instruction is illustrated in Tables 5 and 6.

If the EXTEST instruction is not being used in conjunction with the SAMPLE/PRELOAD instruction, the external test starts by shifting the desired test data into the boundary-scan register in the Shift-DR controller state. By moving into the Update-DR controller state, data shifting is terminated, and on the falling edge of the TCK, the data from the shift-register stage is transferred onto the parallel output of the latch stage. The external test results are loaded into the shift-register stage from the system input on the next Capture-DR controller state and are examined by shifting the data toward TDO on the next Shift-DR controller state.

During the SAMPLE/PRELOAD instruction, the Shift-DR state is used to shift out the data captured from the system input and output pins for examination during the Capture-DR state. At the same time, the Shift-DR state shifts in test data to be used by the next boundary-scan instruction other than SAMPLE/PRELOAD. The EXTEST instruction is usually initiated following the SAMPLE/PRELOAD instruction. The data preloaded during the SAMPLE/PRELOAD instruction phase becomes available at the parallel output of the boundary-scan cells when the EXTEST becomes the current instruction on the falling edge of TCK in the Update-IR state.

Similarly, the CLAMP instruction is usually initiated following the SAMPLE/PRELOAD instruction. The latched data in the boundary-scan cell becomes available to the system output pins when CLAMP becomes the current instruction and when the bypass register is selected as the data path from TDI to TDO.

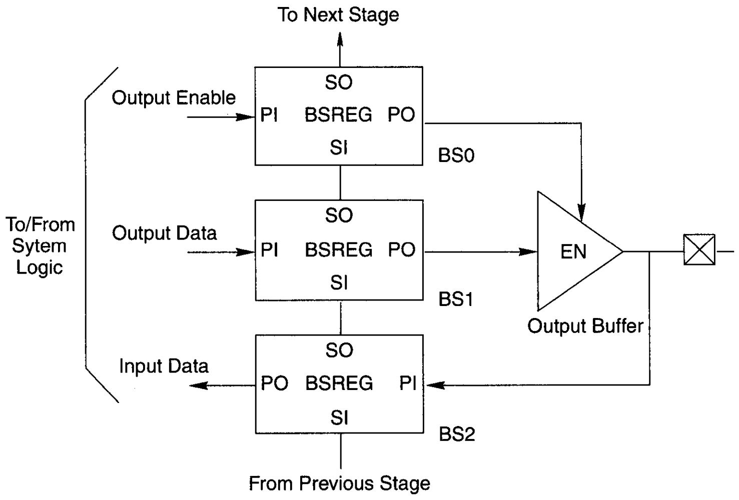


Figure 7 • Functional Schematic of Boundary-Scan Cell

Table 5 • *Summary of EXTEST Operation*

Controller State	Boundary-Scan Shift-Register Stage	Boundary-Scan Latch Stage	Parallel Output (PO)
Test-Logic-Reset	Undefined	Undefined	Parallel In = Parallel Out
Capture-DR	Data at PI is loaded	Undefined	Latched Data
Shift-DR	Shift data toward TDO	Undefined	Latched Data
Exit1-DR Exit2-DR Pause-DR	Retain previous state	Undefined	Latched Data
Update-DR	Retain previous state	Latches data from the shift register	Latched Data = Parallel Out (on the falling edge of TCK)
All other states	Retain previous state	Retain previous state	Latched Data

Table 6 • *Summary of SAMPLE/PRELOAD Operation*

Controller State	Boundary-Scan Shift-Register Stage	Boundary-Scan Latch Stage	Parallel Output (PO)
Test-Logic-Reset	Undefined	Undefined	Parallel In = Parallel Out
Capture-DR	Data on PI is loaded	Undefined	Parallel In = Parallel Out
Shift-DR	Shift data toward TDO	Undefined	Parallel In = Parallel Out
Exit1-DR Exit2-DR Pause-DR	Retain previous state	Undefined	Parallel In = Parallel Out
Update-DR	Retain previous state	Latches data from the shift register	Parallel In = Parallel Out
All other states	Retain previous state	Retain previous state	Parallel In = Parallel Out

Note: *During the SAMPLE/PRELOAD instruction, the parallel input and output of the boundary-scan cells are transparent (PI equals PO).*

JPROBE Register

The JPROBE register consists of a 2-bit shift register (labeled "JPRBREG") connected to the existing probe registers, as shown in Figure 8. The registers that are darkened are not parts of the JPROBE register. The presence of the JPROBE register and the JPROBE instruction permits the use of the internal probe circuitry to observe and analyze any signal inside an Actel chip via JTAG.

The desired probe address is shifted into the JPROBE register by first selecting the JPROBE instruction and then moving to the Shift-DR controller state. Shifting is discontinued by entering the Update-DR controller state. The probe results are loaded into the JPRBREG on the rising edge of TCK in the next Capture-DR controller state. The probe results can be examined by moving back to the Shift-DR controller state and shifting the result toward TDO. Table 7 shows the summary of the JPROBE operation. The probe results may also be observed at the probe pins (PRA and PRB), provided that these pins have been reserved for probe use.

During the JPROBE operation, DCLK and SDI inputs are disabled. However, if the current test instruction is different from JPROBE, the probe circuitry can be accessed and operated normally through the probe pins (DCLK, SDI, PRA, and PRB).

Boundary-Scan Description Lanuage (BSDL) File

Conformance to the IEEE Standard 1149.1 requires that the operation of the various JTAG components be documented. The BSDL file provides the standard format to describe the JTAG components which can be used by automatic test equipment software. The file includes the instructions that are supported, instruction bit pattern, and the boundary-scan chain order.

BSDL files for the 3200DX family is available on the Actel World Wide Web Homepage (www.actel.com) and in the Actel anonymous ftp site (ftp.actel.com).

Table 7 • Summary of the Operation of the JPROBE Register

Controller State	JPRBREG	Probe Register
Test-Logic-Reset	Undefined	Undefined
Capture-DR	Probe result loaded when valid address is in the probe register	Retain previous state
Shift-DR	Shift probe address (probe result) toward TDO	Shift probe address (probe result) toward TDO
Exit1-DR Exit2-DR Pause-DR	Retain previous state	Retain previous state
Update-DR	Retain previous state	Retain previous state
All other states	Undefined	Undefined

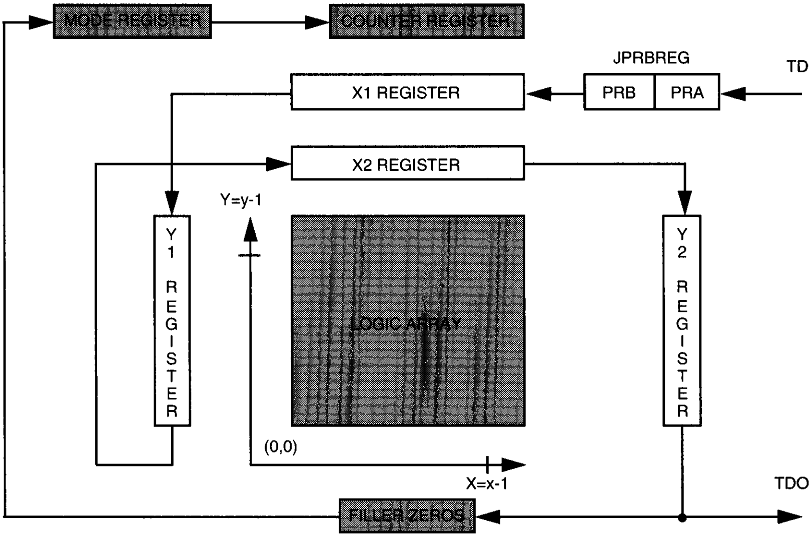


Figure 8 • Functional Schematic of the JPROBE Register

References:

1. Colin M. Maunder & Rodham E. Tulloss. *The Test Access Port and Boundary-Scan Architecture*. IEEE Computer Society Press, 10662 Los Vaqueros Circle, P.O. Box 3014, Los Alamitos, CA 90720-1264.
2. "IEEE Std 1149.1-1993, IEEE Standard Test Access Port, and Boundary-Scan Architecture." IEEE, Inc., 345 East 47th St., New York, NY 10017-2394.
3. Kenneth P. Parker. *The Boundary-Scan Handbook*. Kluwer Academic Publishers, 101 Philip Drive, Assinippi Park, Norwell, MA 02061.

3200DX Quadrant Clocks

The need for high-drive signals in large Field Programmable Gate Arrays (FPGAs) is clear. As FPGA devices continue to increase in size, the demand for high-drive signals used for clocking and resets will continue to rise. Applications with multiple clocks, enables, and resets often can use a few more high-drive lines than the FPGAs offer, and using regular routed signals for these applications can mean higher skew and lower predictability. However, the impact on the FPGA die area has always been a significant roadblock to providing a large number of high-drive signals. Actel's ACT 2 family provides two global clocks that can be used for high-drive signals. The 3200DX builds upon that base by offering four additional quadrant clocks at minimal die cost. This document details the architecture and usability of the 3200DX quadrant clocks. For more information on other 3200DX features, such as SRAM blocks, and Wide Decode modules, refer to other Actel application notes.

When to Use Quadrant Clocks

High-drive signals create a dilemma for the routing resources of FPGAs—the higher the fanout on an ordinary net, the more skew between its destination points. In addition, the higher fanout on a net can cause the net to be slower. To remedy this situation, Actel FPGAs (specifically 3200DX devices) provide global clock networks within chips. These networks are high-speed, low-skew routes dedicated to providing suitable paths for high-speed clocks. The global clock networks can be accessed internally (CLKINT macro) or externally through CLKA and CLKB pins.

As FPGA applications become larger and more complicated, more high-drive signals are implemented within the FPGAs. In addition to one or two high-speed global clocks, there might be two or three more local clocks in need of low-skew paths. Furthermore, there are often reset or enable lines with high fanouts that could use dedicated circuitry to avoid deep buffer trees. These high-drive signals are ideal for the new quadrant clock buffers. Quadrant clocks can also be accessed internally or through external pins. Table 1 summarizes the 3200DX devices and their special features, including the quadrant clocks.

How to Use the Quadrant Clocks

The quadrant clock modules are accessed through library elements similar to those of traditional clock buffers in the ACT 2 and 3200DX families. Figure 2 shows the two new library elements available to capture the quadrant clocks, as well as the existing clock buffers used to capture global clocks. The quadrant clocks can be used for signals with high fanout. If the fanout on the signal driven by the quadrant clock exceeds the quadrant limits, two or more of them can be grouped together to cover the load. The grouping process can be done automatically or manually.

Table 1 • 3200DX Family

	3265DX	32100DX	32140DX	32200DX	32300DX
Gates	6,503	9,240	14,210	20,140	30,735
Global Clocks	2	2	2	2	2
Quadrant Clocks	0	4	0	4	4
I/O MAX	126	152	176	202	250
Wide Decode Cells	20	20	24	24	28
JTAG	No	Yes	Yes	Yes	Yes
SRAM bits	0	2,048	0	2,560	3,072

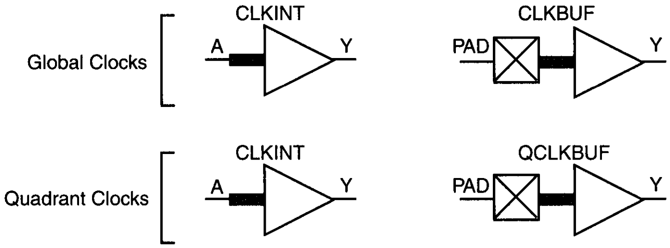


Figure 1 *• Global and Quadrant Clock Library Elements*

Fast On and Off Chip Delays with 1200XL and 3200DX I/O Latches

Introduction

Using Actel 1200XL and 3200DX devices, latched I/O buffer macros can improve clock input-to-registered-output performance. Latched I/O buffer macros also increase the speed of latching signals into the FPGA. Flip-flops created from these I/O latch macros improve performance by up to 22 percent compared with traditional approaches. The ACTgen Macro Builder tool from Actel automatically generates I/O registers using this technique. This application note compares the use of traditional approaches with the use of I/O latch macros in 1200XL and 3200DX designs for the designer interested in the details.

Master/Slave Flip-Flops

As shown in Figure 1, two level-sensitive latches can be combined to create a positive edge-sensitive flip-flop.

Where:

1. $T_{CO} = t_{CO2}$
2. $T_{SU} = t_{CKL} - t_{CO1} - t_{RD1} > t_{SU2}$
3. t_{SU2} = minimum setup time for slave

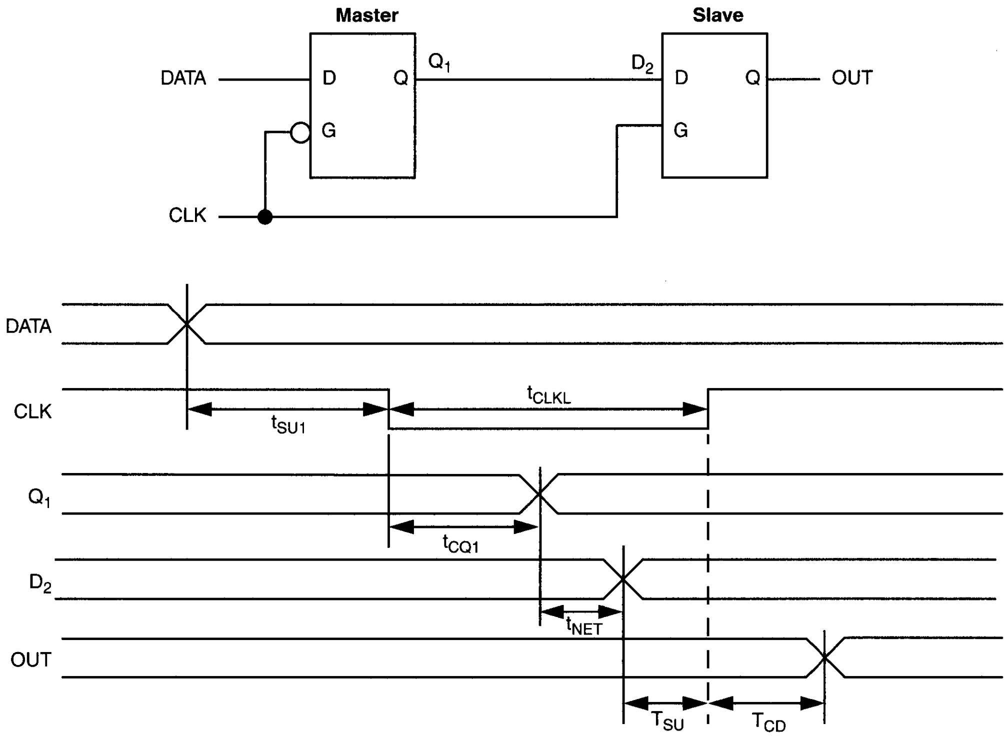
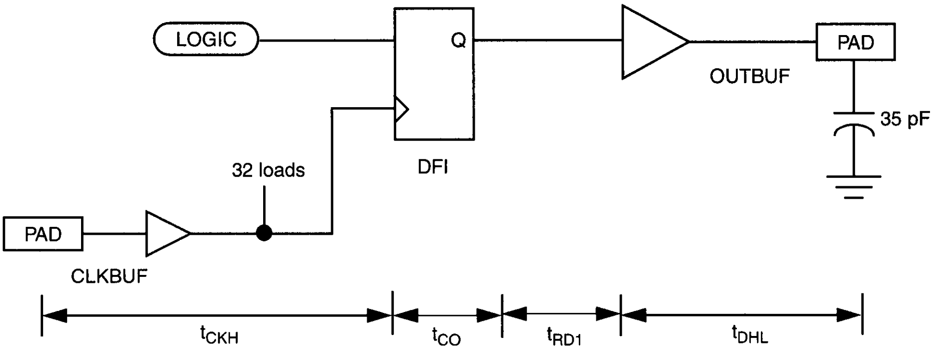


Figure 1 • Master/Slave Flip-Flop

The clock-to-output delay of the resulting flip-flop is determined by the clock-to-output delay of the slave latch. The clock period low time (t_{CKL}) must be greater than the clock-to-output delay of the master latch (t_{CO1}) plus the net delay from the master latch output (t_{RD1}) plus the setup time of the slave latch (t_{SU2}).

Constructing Registered Outputs

You can construct a registered output by combining a flip-flop macro with an output buffer as depicted in Figure 2. The clock-to-out delay for an A3265DX-2 under worst-case commercial conditions is 12.8 ns. For the A1225XL-1 under worst-case commercial conditions, clock-to-out delay is 14.3 ns.



A3265DX-2 $T_{ACO} = t_{CKH} + t_{CO} + t_{RD1} + t_{DHL}$ = 5.1 ns + 2.5 ns+ 1.3 ns + 3.9 ns = 12.8 ns

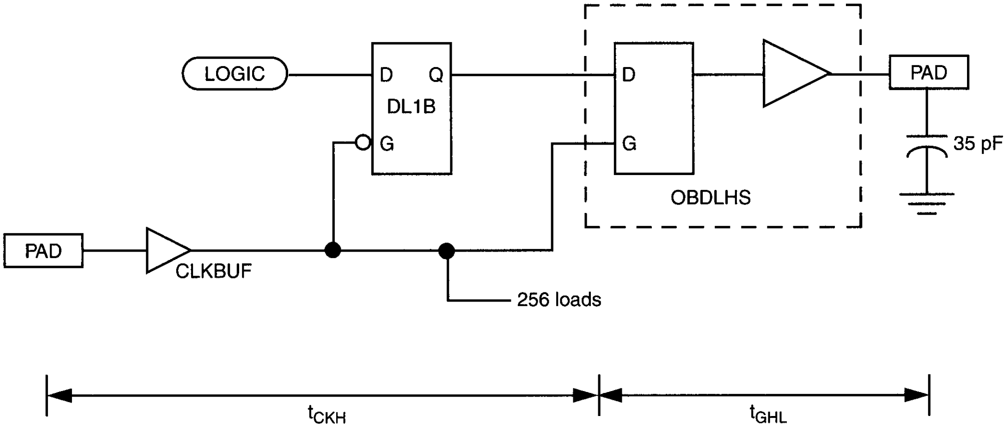
A1225XL-1 $T_{ACO} = t_{CKH} + t_{CO} + t_{RD1} + t_{DHL}$ = 5.8 ns + 3.0 ns + 0.9 ns + 4.6 ns = 14.3 ns

Figure 2 • Conventional Registered Output

I/O Latch Flip-Flops

You can also construct a registered output as a master/slave flip-flop using a logic module latch (DL1B) and a latched output buffer (OBDLHS) as shown in Figure 3.

In this case, the clock-to-out delay for an A3265DX-2 device under commercial worst-case conditions is 10.3 ns. In this case, the loading on the global clock network is assumed to be 256 loads. For similar conditions, the A1225XL-1 has a clock-to-out delay of only 11.9 ns.



A3265DX-2 $T_{LCO} = t_{CKH} + t_{GHL}$ = 5.7 ns + 4.6 ns = 10.3 ns

A1225XL-1 $T_{LCO} = t_{CKH} + t_{GHL}$ = 6.5 ns + 5.4 ns = 11.9 ns

Figure 3 • Master/Slave Registered Output

Registered inputs can also be constructed using a logic module latch (DL1) with a latched input buffer macro (IBDL) as shown in Figure 4. This is equivalent to the I/O macro, IR. Data can be latched into the FPGA most efficiently in this configuration. Because of the unique architecture of the I/O buffer latches in 1200XL and 3200DX families, the input latch has zero external setup time. This means that the data may change just before the rising edge of the clock signal. Data must be held at the input of the latch for the time t_{HEXT}.

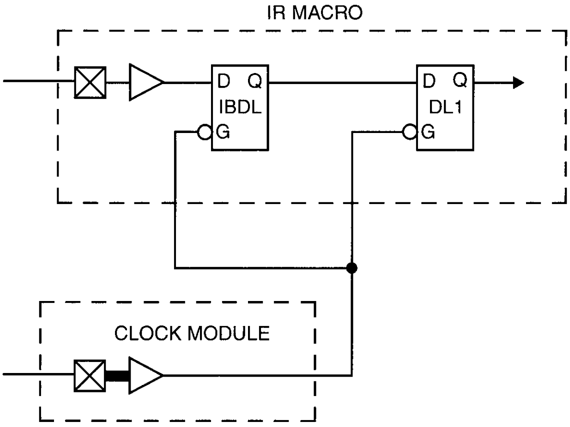
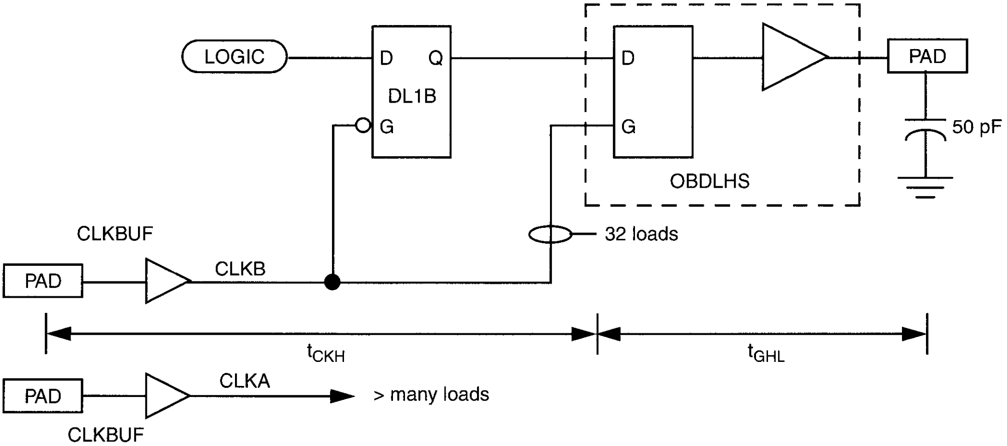


Figure 4 • Master/Slave Registered Input

Dual Clock Approach

Faster clock-to-out can be achieved by utilizing the second global clock network as an I/O clock. In this configuration, shown in Figure 5, CLKA drives the synchronous circuitry on the device and CLKB drives the I/O master/slave latches. Both clock networks are operating at the same frequency and must be connected together outside of the devices. In this case, a 9.7 ns clock-to-out delay can be achieved for the A3265DX-2 and 11.2 ns for the A1225XL-1.

Alternately, an INBUF can be used to drive the I/O latch, if fanout on the INBUF is kept less than four to achieve similar performance.



A3265DX-2 $T_{CO} = t_{CKH} + t_{GHL} = 5.1 \text{ ns} + 4.6 \text{ ns} = 9.7 \text{ ns}$

A1225XL-1 $T_{CO} = t_{CKH} + t_{GHL} = 5.8 \text{ ns} + 5.4 \text{ ns} = 11.2 \text{ ns}$

Figure 5 • Dual Clock Master/Slave Registered Output

Implementation Rules

Actel strongly suggests following these rules when constructing flip-flops with the I/O latches:

1. Use ACTgen Macro Builder to automatically create I/O registers implemented with I/O latches. You need not design them yourself. This can save a lot of time and allows you to focus on more important issues with your design.
2. Do not put combinatorial macros in the data path between master and slave latches. Added delay may prevent the flip-flop from operating properly. For inputs, combinatorial macros are allowed in the data path between the master and slave latches *only* if the logic module latch and the combinatorial latch are both combinable hard macros.
3. For outputs, do not connect the master latch output to any loads except the I/O latch D input.
4. Use a latch made from sequential logic modules for the master stage for outputs on the slave stage for inputs. Sequential module latches have better timing characteristics. They also allow combining to take place, which can improve the performance of the data being registered. The transparent-low sequential module latches are DL1B and DL1C.
5. Use net criticality to ensure that the net delay does not violate the setup requirements of the slave latch (as defined in Figure 1). Verify the timing conditions after place and route is complete. Note that an asymmetrical duty cycle on the clock signal (less than 50% low time) will provide more tolerance on the allowable net delay between latches.
6. Design to combine. The Designer Series System will automatically combine combinatorial logic into the D input of the DL1B latch if the combiner rules are met.

Using ACT 3 Family I/O Macros

The ACT 3 family from Actel has a complex I/O macro which allows users to implement high speed complex functions entirely in the I/O. Common functions like shift registers operating at 160 MHz, state machines and counters with 7.5 ns clock-to-output (pin-to-pin) and 160 MHz pipeline registers are all implementable in ACT 3 I/Os. This significantly increases capacity and I/O performance of ACT 3 devices over previous generations of FPGAs. This application note covers several examples of how designers can use these powerful I/O macros in real world applications.

ACT 3 I/O Architecture

The ACT 3 family architecture is shown in Figure 1. The logic array consists of rows of Sequential and Combinatorial logic cells with routing channels in between, much like a channeled gate array. I/O modules are located at the periphery of the array. Each I/O module is used to interconnect device signal pins to array inputs and outputs. The I/O module contains specialized circuitry to assist the designer in getting signal on and off chip. Figure 2 shows the detailed block diagram of the ACT 3 I/O module.

I/O Module

The main function of the ACT 3 I/O module is to provide a high-speed high-functionality interface between the pins of the device and the logic array. Data to/from the logic array arrives on the left side of Figure 2. U01 and U02 are the data from the array and 'Y' is the data to the array. The pad is shown on the far right side of Figure 2 and is the input/output pin of the device. Data from the array may be registered in the output register or bypassed using the OTB control signal. If registered, the data may be selectively loaded or held via the synchronous enable signal ODE and asynchronously initialized to a logic Low or High via the IOPCL input. The output drivers have both an individual slew control and individual output enable. The Preset/Clear, OTB, and Slew features are usually automatically determined when the user selects the desired I/O macro. For example, the I/O macro ORECTH shown in Figure 3A uses a Clearable output register with High Slew. Correct selections of OTB, Preset/Clear, and Slew are made automatically. Another feature of the I/O macro is the ability to feedback the contents of the output register back to the logic array. This allows embedded functions like state machines and shift registers to be easily implemented in the IO macro. Figure 3B shows the FECTML macro and the feedback multiplexor used to select between the output register and the device pad.

When data is sourced from the device pad, the input register can be used to capture incoming data. The input register has a synchronous data enable and a Preset/Clear feature similar to the output register. The IDE and IEN signals are used to control the Y multiplexer and the data enable multiplexer and provide the most commonly needed input functions. Again, these signals are usually determined automatically when the user selected the desired macro. For example, the IREC input register macro is shown in Figure 3C and has IDE, IEN, and Preset/Clear predetermined to implement the desired functions.

Actel provides an automatic software tool (ACTgen Macro Builder) to help in implementing a wide range of I/O functions in ACT 3 devices. The designer can select the width of the macro and a variety of operating modes with a few mouse clicks. A sample menu from ACTgen is shown in Figure 4.

Table 1 • ACT 3 I/O Macro Performance

Input Features	
Input Timing Register	167 MHz
Input Pipeline Register	167 MHz
Pulse Stretching Input Register	120 MHz
Input Synchronization	167 MHz
Input Address Holding Register	164 MHz
Input Control Register	167 MHz
Output Functions	
State Machine Output Register	135 MHz
Output Shift Register	167 MHz
Output Pseudo Random Counter	131 MHz
Output Datapath	131 MHz
Bi-Directional Functions	
Bi-Directional Registers	167 MHz

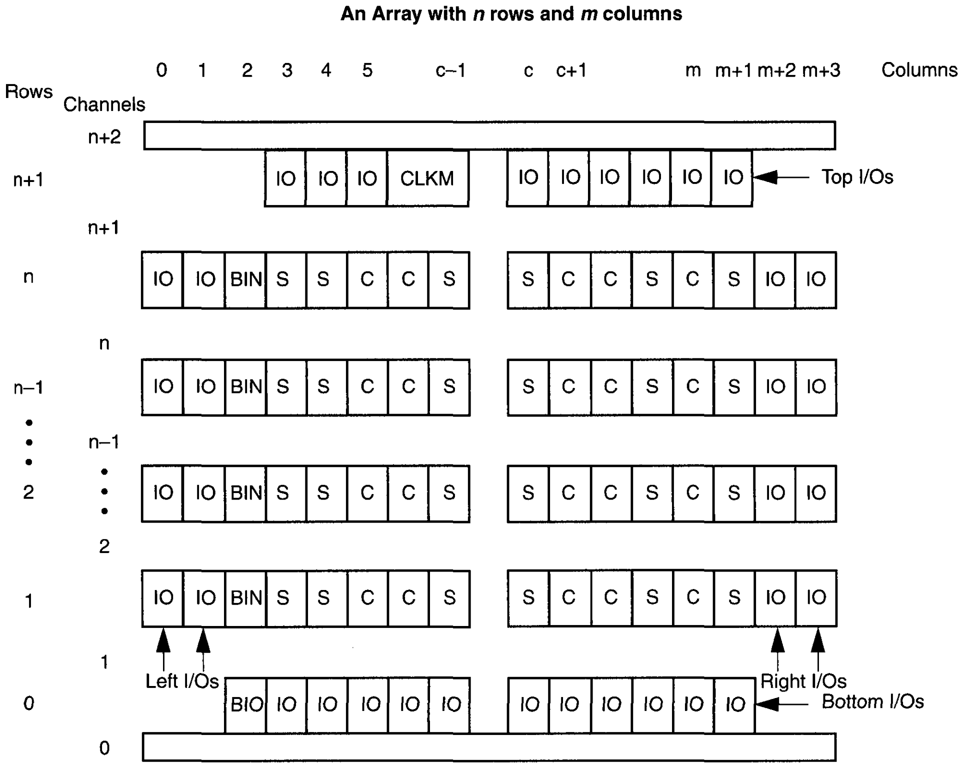

An Array with *n* rows and *m* columns
0 1 2 3 4 5 c–1 c c+1 m m+1 m+2 m+3 Columns
Rows
Channels
n+2
n+1
IO IO IO CLKM IO IO IO IO IO IO
Top I/Os
n+1
n
IO IO BIN S S C C S S C C S C S IO IO
n
n–1
IO IO BIN S S C C S S C C S C S IO IO
n–1
2
IO IO BIN S S C C S S C C S C S IO IO
2
1
IO IO BIN S S C C S S C C S C S IO IO
1
Left I/Os
Right I/Os
0
BIO IO IO IO IO IO IO IO IO IO IO IO
Bottom I/Os
0

Figure 1 • *Generalized Floor Plan of ACT 3 Device*

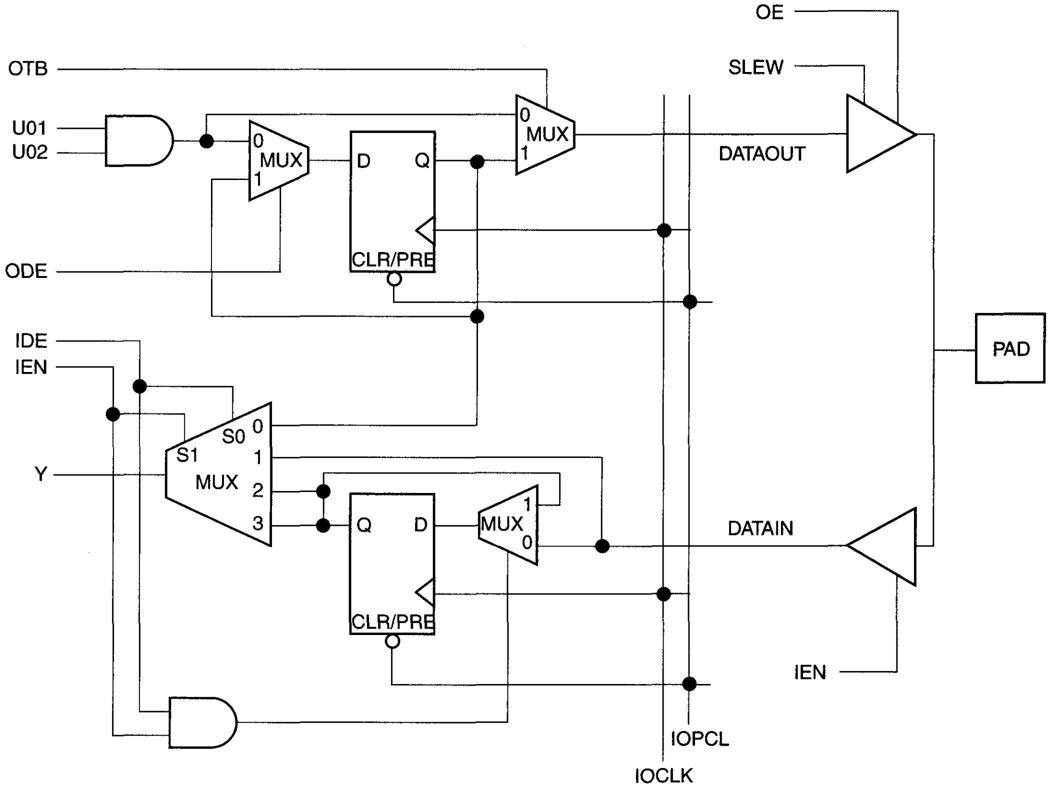
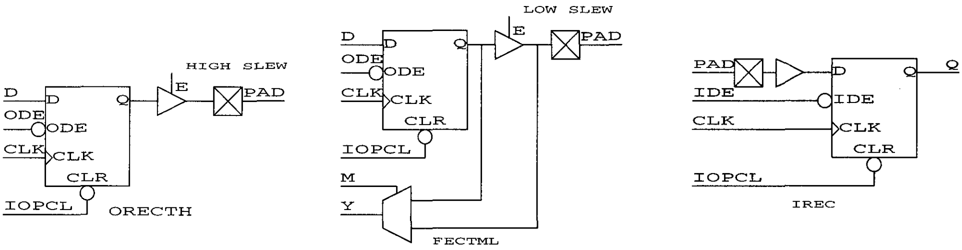


Figure 2 • *I/O Macro Block Diagram*

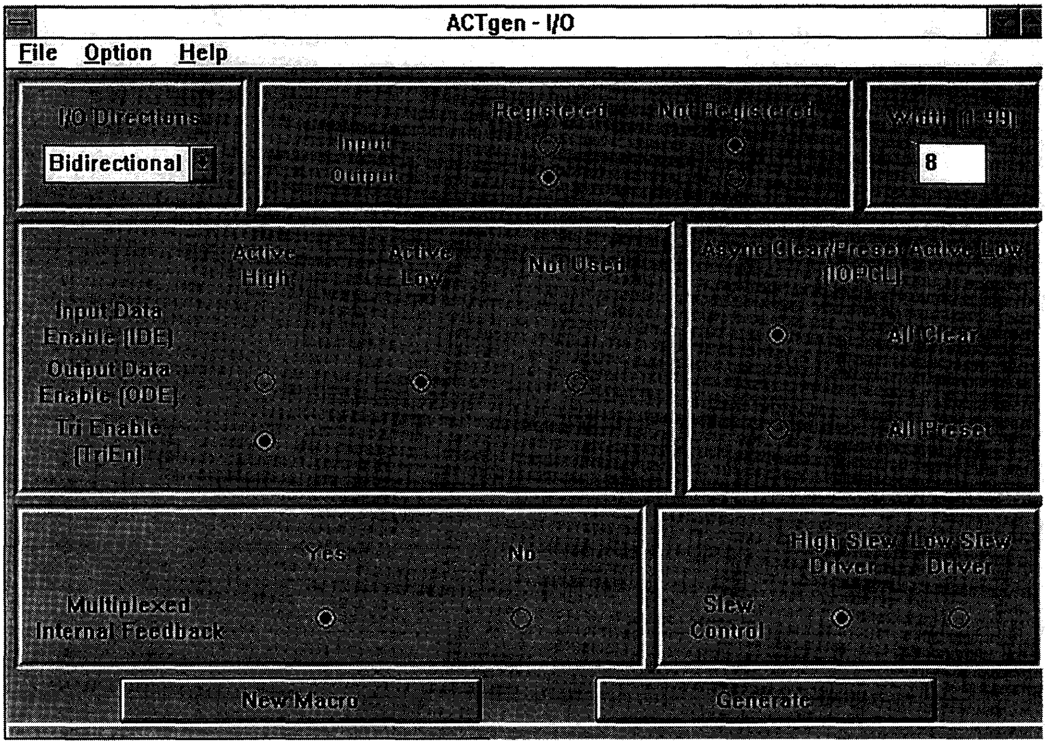


3A

3B

3C

Figure 3 • *I/O Macro Implementations*



ACTgen - I/O

I/O Directions: Bidirectional

	Registered	Not Registered
Input	○	●
Output	●	○

Width (1-99): 8

	Active High	Active Low	Not Used
Input Data Enable [IDE]			
Output Data Enable [ODE]	○	●	○
Tri Enable [TriEn]	●		

Async Clear/Preset Active Low [IOPCL]

● All Clear

○ All Preset

	Yes	No
Multiplexed Internal Feedback	●	○

	High Slew Driver	Low Slew Driver
Slew Control	●	○

Figure 4 • ACTgen Selection of ACT 3 I/O Functions

Every registered ACT 3 I/O macro (for example, DECETH, IREC, and OREPTL) includes an asynchronous preset or clear input IOPCL. This input asynchronously sets the output of I/O flip-flops to 0 or 1. The IOPCL pin of a registered I/O macro must be driven by the dedicated I/O macro, IOPCLBUF. The IOPCLBUF is externally driven in turn by the dedicated IOPCL package pin as shown in Figure 5.

Similarly, the clock (CLK) inputs of registered I/O macros are driven by a dedicated circuit. The dedicated I/O clock buffer, IOCLKBUF, is used to drive the clock pin of each I/O macro. The IOCLKBUF is driven in turn by the dedicated external package pin, IOCLK, as shown in Figure 6. See the package pin assignment diagrams in the ACT 3 data sheet for specific locations of the IOPCL and IOCLK package pins.

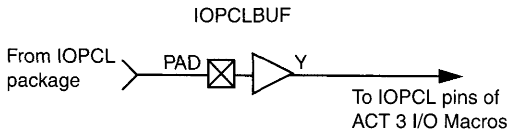


Figure 5 • *An IOPCLBUF Driven by a Dedicated IOPCL Package Pin*

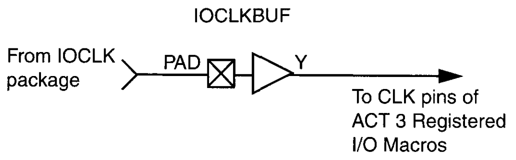


Figure 6 • *An IOCLKBUF Driven by a Dedicated IOCLK Package Pin*

If no registered I/O macros are used in an ACT 3 design, then the dedicated IOCLK and IOPCL package pins can be used as normal I/O pins. In that case, these pins can be connected to any other type of I/O buffer except IOCLKBUF, IOPCLBUF, and HCLKBUF. For example, undedicated normal I/O buffers such as INBUF and OUTBUF may be connected to these pins. If any registered I/O macros are used in an ACT 3 design, then the IOCLKBUF and IOPCLBUF macros must be included in the design, and the pin assignment must include the IOPCL and IOCLK pins. The IOPCL and IOCLK pins must be explicitly specified because the Designer Series software will not automatically create pin assignments for IOCLKBUF and IOPCLBUF.

In an ACT 3 design, all types of I/O macros can be used in a single design. For example, I/O macros without registers, I/O macros with preset, and I/O macros with clear can be used together in one design. However, IOCLKBUF must drive all the clock inputs of the registered I/O macros and the IOPCL must drive all the clear or preset inputs of the registered I/O macros. CLK and IOPCL pins of registered I/O macros cannot be connected to GND or V_{CC}. All ACT 3 I/O registered buffers share the same dedicated clock and asynchronous clear or preset network. Once the built-in dedicated IOCLK and IOPCL networks are used, all registered I/O macros will be cleared and preset at the same time by the same signal.

Implementing Three-State and Bidirectional Buses with Multiplexers in Actel FPGAs

Three-state logic is used in conventional MSI logic devices to allow buses where multiple drivers are directly connected to one or more loads. Figure 1 shows a typical bus configuration with TTL three-state bus drivers and registers. Each driving device has a control input that places all outputs in a high impedance state when asserted. (For the register, the control pin is OC; for the bus driver, there are two control pins, 1G and 2G). To prevent data collisions, only one driver can be active at a time; the other drivers must be in a high impedance state. The four NAND gates perform a logic decoding function to ensure that only one driver is active at a time. If the first bus driver is selected to be active via SELA and SELB, then the data bits W0 to W7 will drive the bus (BUS0 to BUS7). Similarly, the other data bits will be selected when the respective register is active. The loads on the bus are not shown in Figure 1.

To make effective use of routing resources for many different applications, the Actel FPGA implements internal multiple drivers on a net with multiplexers instead of three-state logic. Figure 2 depicts the Actel implementation of the three-state bus discussed above. In this case, a 4-to-1 multiplexer is used for each bit of the bus to redirect the desired signal from one of four sources (W, X, Y, or Z). In addition to replacing the three-state nature of the MSI devices, the multiplexer eliminates the need for the LS241 bus drivers (inputs W and X). The desired source is selected by the two multiplexer select lines, which eliminate the need for the decoding logic used in the MSI implementation. For greater than four sources, the MX8 8-to-1 multiplexer can be used in a similar fashion.

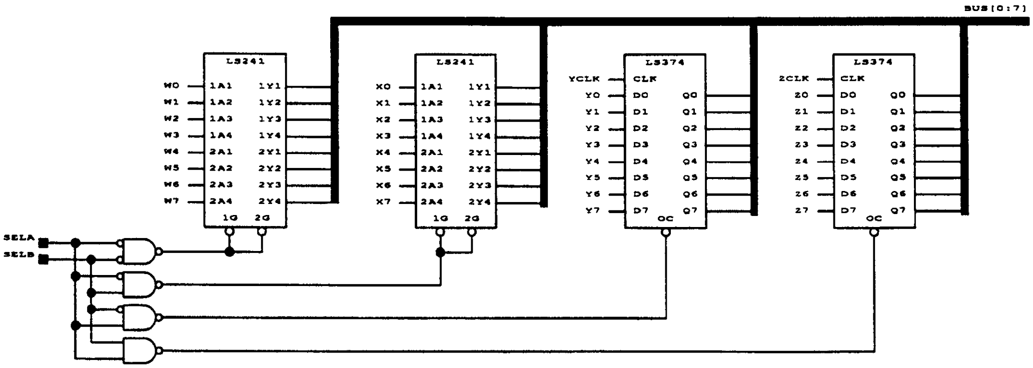


Figure 1 • *Three-State Bus Implementation with MSI Logic*

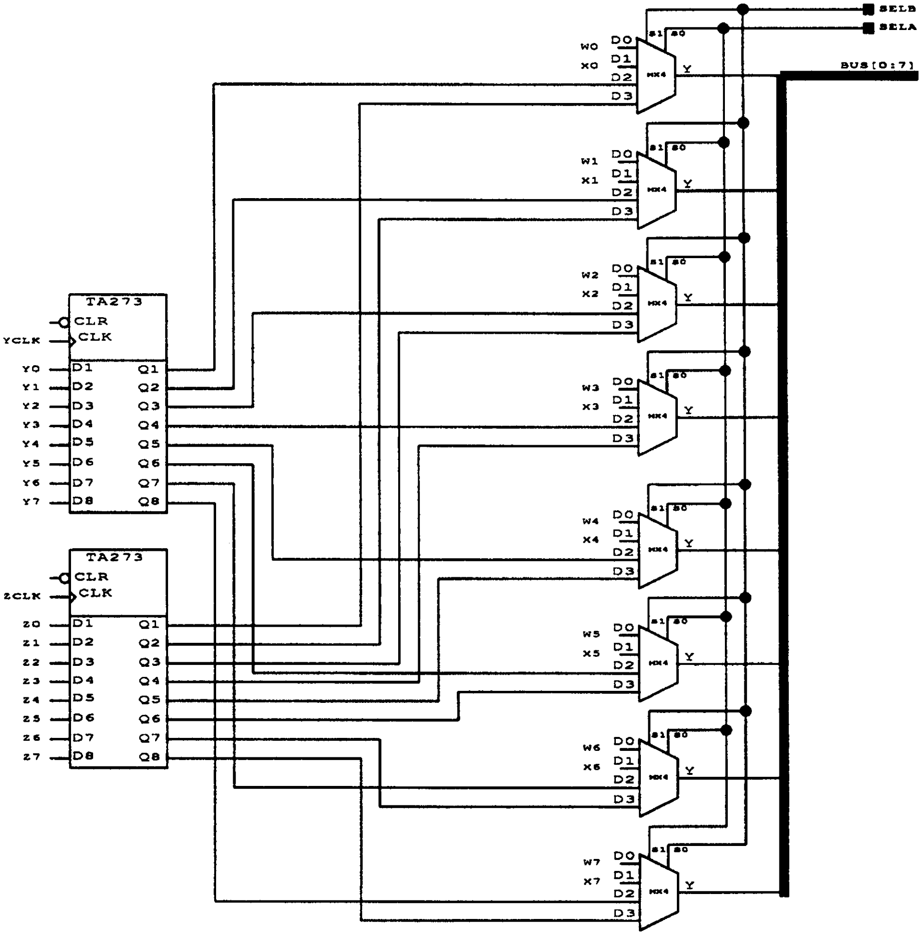


Figure 2 • *Three-State Bus with Actel FPGA Using Multiplexers*

Bidirectional signals can also be replaced readily with multiplexers in the Actel architecture. Figure 3 shows the conversion of a popular transceiver to a 2-to-1 multiplexer.

Figure 4 illustrates the conversion to the multiplexer implementation of three bidirectional transceivers driving a bus.

The multiplexer circuit assumes that A's driver also drives any inputs on the A sub-bus, B drives B's inputs, and so on. Two additional sub-buses can be made available by controlling the select lines accordingly.

If the bidirectional element is located at the FPGA pad, the BIBUF macro can be used directly. Figure 5 shows a simple circuit using the LS245 transceiver with flip-flops driving and leading the bus. Figure 6 shows the Actel implementation of the BIBUF macro and the DFM multiplexed flip-flop. Note that when enable is low, the pad drives the B flip-flop and that when enable is high, A drives both the PAD and B.

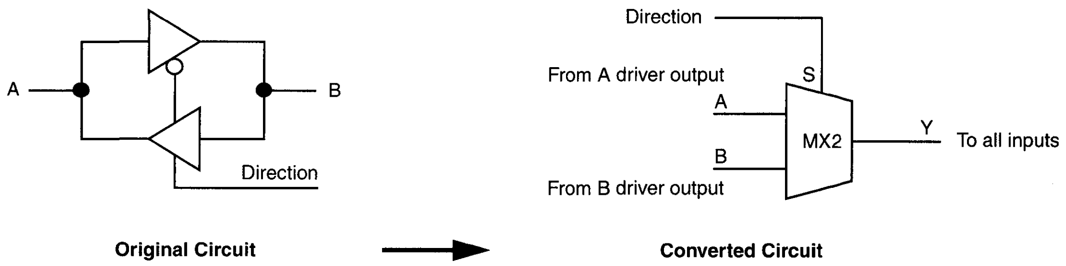


Figure 3 • *Transceiver Conversion to a 2-to-1 Multiplexer*

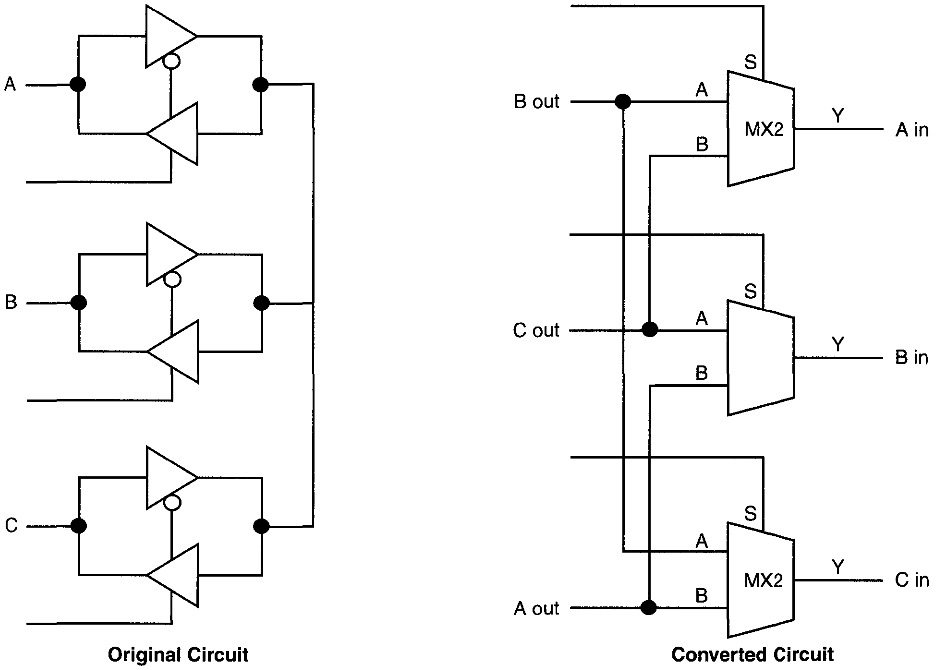


Figure 4 • *Three-Bit Transceiver Conversion*

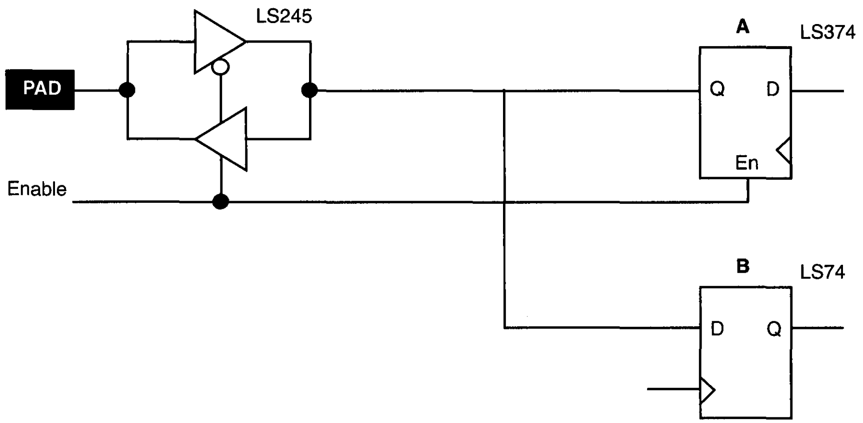


Figure 5 • Bidirectional Bus with LS245

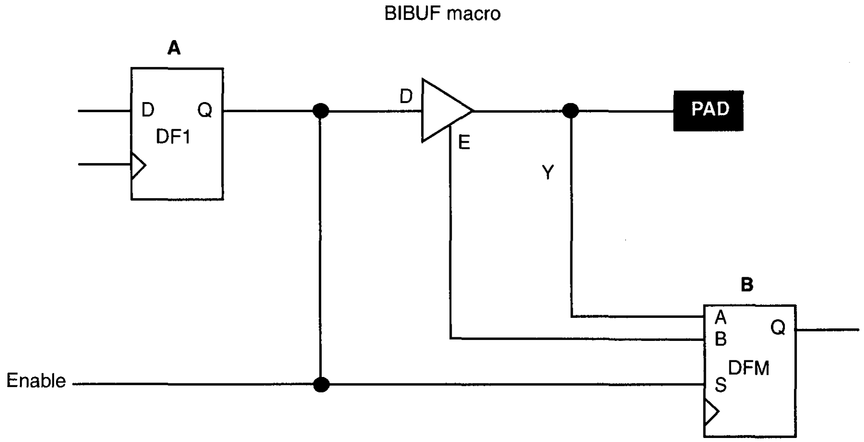


Figure 6 • Bidirectional Bus with BIBUF Macro

Actel

Application Note

Global Clock Networks

The Actel architecture provides global clock networks that allow high fanout drive for flip-flops and latches with minimal skew. Table 1 shows the available global networks and their characteristics, which are defined as follows.

- *Routed clocks* are clock networks which can be used by selecting CLKBUF/CLKBIBUF or CLKINT macros or both.
- *Dedicated clocks* are clock networks that are directly wired to sequential and I/O modules. They contain no programming elements in the path from the I/O pad driver to the input of S-modules or I/O modules; they provide sub-nanosecond skew and guaranteed performance.
- A *special* network refers to a special hard-wired input for I/O modules that can only drive preset/clear pins of I/O modules.

Note that the *Internal Drive Option* for ACT™ 2, 1200XL, 3200DX, and ACT 3 devices can only be utilized by selecting the CLKINT macro to drive an internal clock network.

ACT 1 Clock Network

A single clock distribution network is provided on ACT 1 arrays. The clock network provides unlimited fanout (the ability to drive all logic modules in the array) with minimal delay and skew time. Figure 1 illustrates the clock distribution network for an A1010 array. It is arranged similarly in the A1020 array.

The network is driven by a specific I/O pin that drives a dedicated on-chip buffer tree. Each row of logic modules (8 on the A1010 and 14 on the A1020) has a dedicated buffered clock track. The clock distribution network is selected automatically when the CLKBUF macro is used in the schematic and assigned to its dedicated package pin. The CLK I/O pin can also be used for normal I/Os by assigning INBUF, OUTBUF, TRIBUFF, or BIBUF to the CLK pin location.

Clock Balancing Scheme

Clock balancing equalizes the clock loads on each branch of the global clock network, thereby minimizing clock skew. Clock skew can cause setup and hold time problems. The clock balancing strength sets the level of clock balancing for ACT 1 designs only. It is not used for ACT 2 and ACT 3 designs because the global clock networks for those families have been designed for minimal clock skew. The results are design dependent. If more than 50 percent of the logic modules are driven by the global clock, the effect is minimal. Also, strong clock balancing may result in increased delays and reduced routability.

Table 1 *• Global Clock Attributes*

Input Pad Name	Type	Family	Number	Internal Drive Option	Macro	Note
CLK	routed	ACT 1	1	No	CLKBIBUF, CLKBUF	Can adjust skew with clock balancing
CLKA, CLKB	routed	ACT 2, 1200XL, 3200DX	2	Yes	CLKBIBUF, CLKBUF, CLKINT	Use CLKINT for Internal Drive Option
CLKA, CLKB	routed	ACT 3	2	Yes	CLKBIBUF, CLKBUF, CLKINT	Use CLKINT for Internal Drive Option
HCLK	dedicated	ACT 3	1	No	HCLKBUF	Connected to all S-modules
IOCLK	dedicated	ACT 3	1	No	IOCLKBUF	Connected to all I/O modules
IOPCL	special	ACT 3	1	No	IOPCLBUF	Connected to I/O module set and reset pins

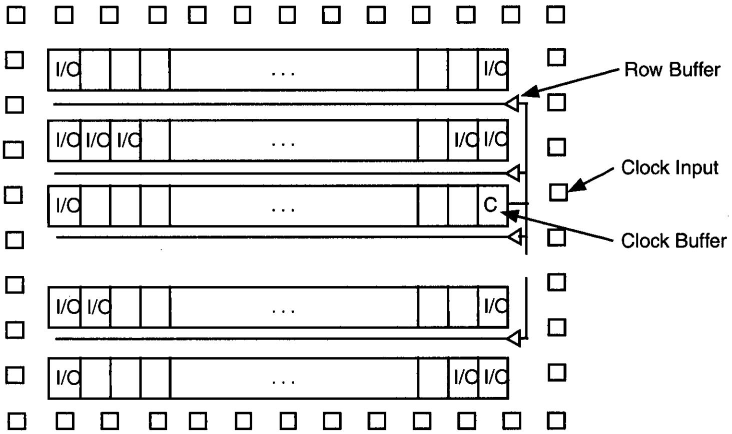


Figure 1 • *Clock Distribution Network for an A1010*

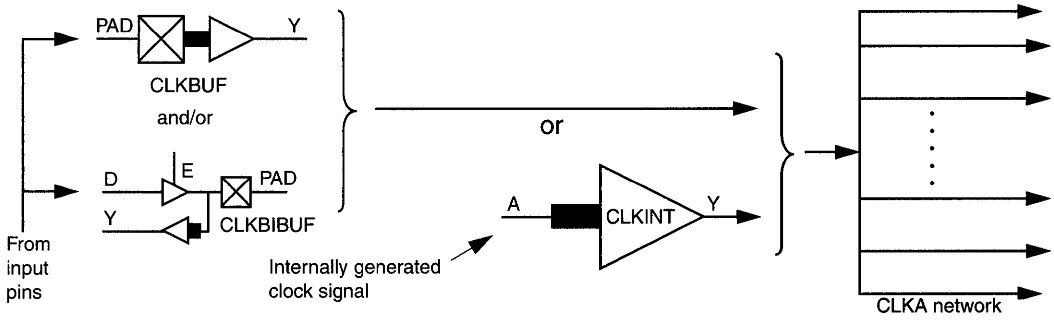


Figure 2 • *One of Two ACT 2/1200XL/3200DX Family Clock Networks (CLK0)*

ACT 2, 1200XL, and 3200DX Clock Networks

Two low-skew, high fanout clock distribution networks are provided in the ACT 2, 1200XL, and 3200DX architectures. Figure 2 illustrates the implementation of one of two clock networks using CLKBUF/CLKBIBUF or CLKINT macro or both. ACT 2, 1200XL, and 3200DX devices offer two identical clock networks. The clock modules are located in the top row of I/O modules. Clock drivers and a dedicated horizontal clock track are located in each horizontal routing channel. The clock input pads may also be used as normal I/Os, bypassing the clock networks. These networks are referred to as CLKA and CLKB. Each network has a clock module that selects the source of the clock signal and may be driven as follows:

- externally from the CLKBUF macro
- externally from the CLKBIBUF macro
- internally from the CLKINT macro

As mentioned above, the macro CLKBUF is used to connect one of the two external clock pins to a clock network, and the macro CLKINT is used to connect an internally generated clock signal to a clock network. Figure 3 illustrates the implementation of internal clock network using CLKINT macro. In the figure, the input signals are driven by regular I/O buffers, not CLKBUF or CLKBIBUF. These input signals drive a user-created Clock Conditioning Module, which generates the internal clock signal. This signal is subsequently driven by the CLKINT macro.

ACT 3 Clock Networks

The ACT 3 architecture contains four clock networks: two high performance dedicated clock networks and two general purpose routed networks. The high performance networks function at up to 250 MHz, while the general purpose routed networks function at up to 150 MHz. Figure 4 illustrates the ACT 3 family clock networks.

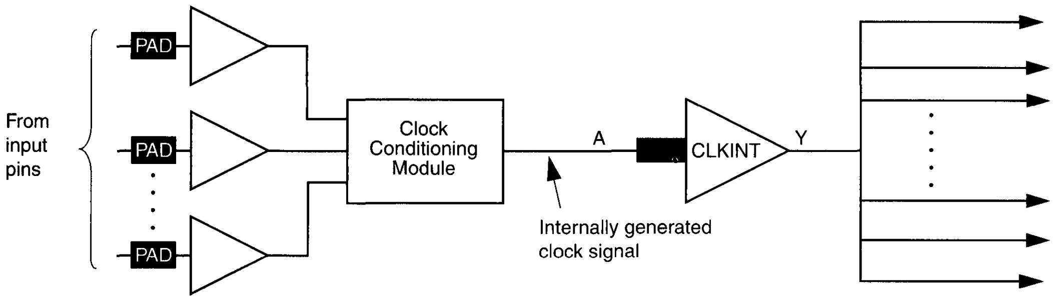
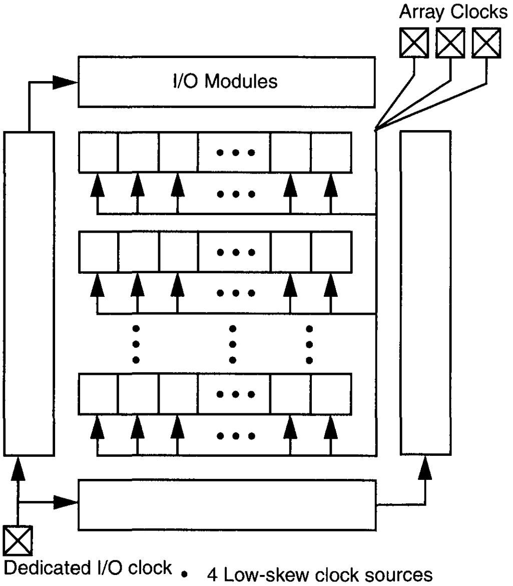


Figure 3 • *Implementation of Internal Clock Network Using CLKINT Macro*



- 4 Low-skew clock sources
  - Dedicated I/O clock
  - Dedicated array clock
  - Two routed array clocks (like ACT 2 family)
- Can tie dedicated clocks together for fully synchronous operation

Figure 4 • *ACT 3 Family Clock Networks*

Dedicated Clocks

There are two dedicated clock networks: one for the array registers known as Dedicated Array Clock (HCLK) and one for the I/O registers known as Dedicated I/O Clock (IOCLK) The clock networks are accessed by special I/Os. Figure 5 shows the macros that drive the Dedicated Array and I/O clock networks in ACT 3 devices.

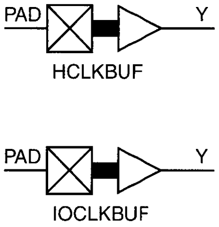


Figure 5 • *Dedicated Array and I/O Clock Buffers*

HCLK is a dedicated hard-wired clock input for sequential modules. HCLK is directly wired to each S-module and offers guaranteed clock speeds independent of the number of S-modules being driven. IOCLK is a dedicated hard-wired clock input for I/O modules. IOCLK is directly wired to each I/O module and offers guaranteed clock speeds independent of the number of I/O modules being driven. These dedicated clock networks support high performance by providing sub-nanosecond skew and guaranteed performance. Dedicated clock networks contain no programming elements in the path from the I/O pad driver to the input of S-modules or I/O modules.

Routed Clocks

The routed clock networks for ACT 3 devices have the same characteristics as the ACT 2, 1200XL, and 3200DX networks as shown in Figure 2 and Figure 3. The macros that drive routed clock networks in ACT 1, ACT 2, 1200XL, 3200DX, and ACT 3 devices are shown in Figure 6.

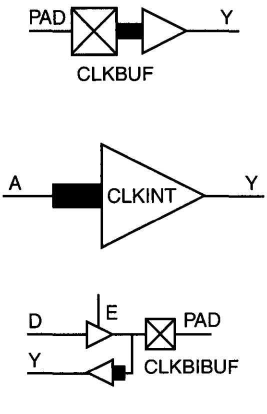


Figure 6 • Dedicated Routed Clock Buffers

CLKA and CLKB are global signals with unlimited fanout. Refer to ACT 2 family clock networks described above for more detail on ACT 3 family routed clocks.

Special Hard-wired Preset/Clear Network

IOPCL is a dedicated special hard-wired input for I/O modules. It is directly wired to the Preset and Clear inputs of all I/O registers. IOPCL functions as an I/O when no I/O preset or clear macros are used. Figure 7 shows the IOPCLBUF macro that drives the dedicated hard-wired Preset/Clear network. IOPCLBUF can only be connected to the preset/clear pins of I/O macros.

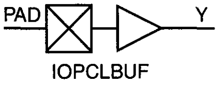


Figure 7 • Dedicated Preset/Clear Network Buffer

The routed clocks CLKA and CLKB can also be used to drive high fanout nets like resets, output enables, or data enables. This saves logic modules and results in performance increases in some cases.

Clock Connections for ACT 2, 1200XL, 3200DX, and ACT 3 Families

To minimize loading on the clock networks, only a subset of module inputs has antifuses on the clock tracks. Therefore, only a few of the C-module and S-module inputs can be connected to the clock networks. To further reduce loading on the clock network, only a subset of the horizontal routing tracks can connect to the clock inputs of the S-module. Figure 8 illustrates the connections to the clock networks.

Creating a User-Defined Clock Distribution Network

Some applications require many internal clock networks. For these type of designs or for clock networks with a small number of loads, it may be better to create a user-defined clock network. Because this clock network is not a built-in, dedicated circuit, the skews and delays cannot be guaranteed.

However, by intelligently placing the I/O pins and declaring the nets associated with the clock network as critical, the automatic placement of these macros create a network that controls clock skew and delay. The results can be verified by running a simulation or using the Actel Designer Series software. It may be necessary to place and route again to meet timing requirements. To minimize the skew between paths, try to equalize the loading in each leg of the clock distribution network.

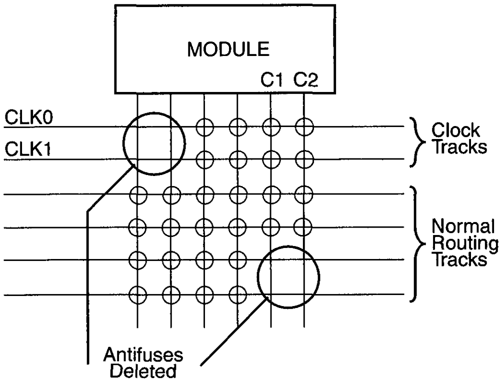


Figure 8 • Fuse Deletion on Clock Networks

Designing State Machines for FPGAs

Introduction

The traditional methodology for designing state machines has been to draw a state diagram, map the states into the minimum number of register bits, and determine the next state function for each register bit. The minimum number of register bits needed can be determined by rounding up the natural log of the number of states. This methodology results in a minimum number of registers but usually requires wide gating and complicated logic to encode the next state bit. This scheme is necessary to implement state machines using programmable logic devices (PLDs) because of the PLDs inherent lack of registers. Because FPGAs do not have this limitation, other approaches are used for efficient state machines.

This application note will discuss techniques for efficiently implementing state machines by using one-hot-encoding for Actel FPGAs.

Sample One-Hot-Encoding

One-hot-encoding is an effective approach because it takes advantage of the abundance of registers. The state diagram of a sample state machine is illustrated in Figure 1. This state machine is the control section of a four-channel DMA controller supermacro for Actel FPGAs. The state machine contains six states, seven inputs, and five outputs. Each circle represents a different state, and each arrow represents a transition between states. Inputs that cause state transitions are listed adjacent to the state transition arrows. Control outputs are labeled along with the states inside the circles. For example, in state S2, the state machine asserts the CNTD and CMREQ outputs. If the MACK input is low, the state machine remains at state S2. If the MACK input is high, the state machine goes to state S3 and asserts the CE output in the process.

State Transition Equations

The next step is determining the logic to generate the state sequence. For one-hot-encoding implementations, assign each state to a separate register and write a state transition equation for each register. The state diagram in Figure 1 shows that state S0 can be realized when state S4 is asserted and the input CONT is low, or it remains at state S0 if all four inputs—A, B, C, and D—are low. Therefore, the transition equation of state S0 can be written as

S0 := /A*/B*/C*/D*S0 + /CONT*S4

Similarly, state S1 can be achieved when state S0 is asserted and any one of the four inputs—A, B, C, or D—is high, or it remains at the same state (S1) if the input PBGNT is low. The transition equation of state 1 can be derived as

S1 := (A+B+C+D)*S0 + /PBGNT*S1

The complete state transition equations for the state machine are listed in Table 1. Note that state S3 will go to state S4 unconditionally; therefore, the transition equation for state S3 can be written as S3 := S4.

Table 1* • *State Machine Transition Equations

S0 := /A*/B*/C*/D*S0 + /CONT*S4;
S1 := (A+B+C+D)*S0 + /PBGNT*S1;
S2 := PBGNT*S1 + /MACK*S2;
S3 := /MACK*S2 + MACK*S3;
S4 := S3;
S5 := CONT*S4 + /MACK*S5;

Output Equations

Once the state transition equations are determined, the output equations can be written by encoding the states. In some cases, the output is active only in one state. Therefore, the output is simply a function of that state. For example, CE is active only in state S3, so the output equation for CE is simply CE = S3. Other output may be active in more than one state. The output equations can be written simply as functions of those states. For example, CMREQ is active in state S2 as well as state S5. Therefore, the output equation is

CMREQ = S2 + S5

Table 2 lists the output equations of the state machine.

Table 2* • *Output Equations

PBREQ = S1;
CLD = S4;
CNTLD = S2;
CMREQ = S2 + S5;
CE = S3;

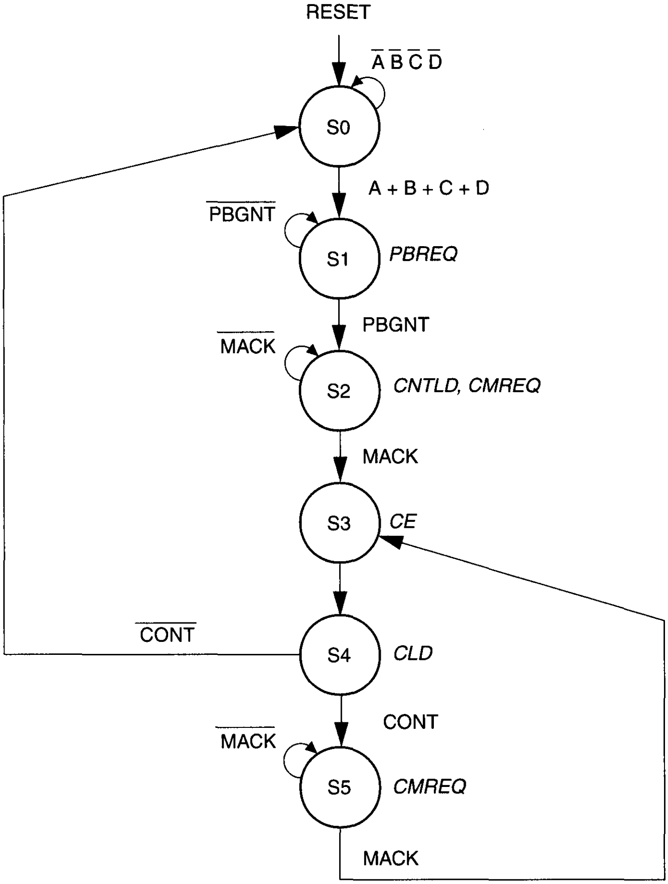


Figure 1 • *Four-Channel DMA Controller State Diagram (Control Section)*

The state machine can be captured using schematic entry, or it can be automatically mapped by using synthesis tools such as Actel ACTmap VHDL Synthesis. The completed VHDL file is shown in Figure 2. A reset signal RST has been added to initialize the circuit.

Figure 3 shows the schematic of the state machine implementation using the one-hot-encoding approach in the ACT 3 family. The circuit requires 18 logic modules after combining timing for the A1425A-2, as shown in Figure 4.

Summary

A summary of the bit-per-state methodology is as follows:

1. Draw a state diagram.
2. Assign each state to a separate register.
3. Write a state transition equation for each register.
4. Derive output equations based on active states.

Larger state machines can be implemented using this technique by distributing control to several smaller state machines and using a single master machine to coordinate activities among the state machines. This usually results in higher-performance designs. It is also easier to design and debug simpler and smaller state machines.

```
--library ieee;
--use ieee.std_logic_1164.all;
entity DMASM is
        port (A,B,C,D:  in bit;
              PBGNT, MACK, CONT :  in bit;
              RST, CLK :  in bit;
              PBREQ, CMREQ, CE, CNTLD, CLD :  out bit);
        end DMASM;
architecture BEHAVE of DMASM is
        type STATE is (S0, S1, S2, S3, S4, S5);
        signal CURRENT_STATE, NEXT_STATE: STATE;
begin
SEQ: process (RST, CLK)
begin
        if (RST = '0') then
                CURRENT_STATE <= S0;
        elsif (CLK' event and CLK = '1' ) then
                CURRENT_STATE <= NEXT_STATE;
        end if;
end process;
COMB: process (CURRENT_STATE, A, B, C, D, PBGNT, MACK, CONT)
begin
  PBREQ <= '0';
  CMREQ <= '0';
  CE    <= '0';
  CNTLD <= '0';
  CLD   <= '0';
case CURRENT_STATE is
    when S0 =>
        if (A = '1' or B = '1' or C = '1' or D = '1') then
        NEXT_STATE <= S1;
        else
        NEXT_STATE <= S0;
        end if;
    when S1 => PBREQ <= '1';
        if (PBGNT = '1') then
        NEXT_STATE <= S2;
        else
        NEXT_STATE <= S1;
        end if;
    when S2 => CNTLD <= '1'; CMREQ <= '1';
        if (MACK = '1') then
        NEXT_STATE <= S3;
        else
        NEXT_STATE <= S2;
        end if;
    when S3 => CE <= '1';
        NEXT_STATE <= S4;
    when S4 => CLD <= '1';
        if (CONT = '1') then
        NEXT_STATE <= S5;
        else
        NEXT_STATE <= S0;
        end if;
    when S5 => CMREQ <= '1';
        if (MACK = '1') then
        NEXT_STATE <= S3;
        else
        NEXT_STATE <= S5;
        end if;
    end case;
  end process;
end BEHAVE;
```

Figure 2* • *Complete State Machine VHDL file

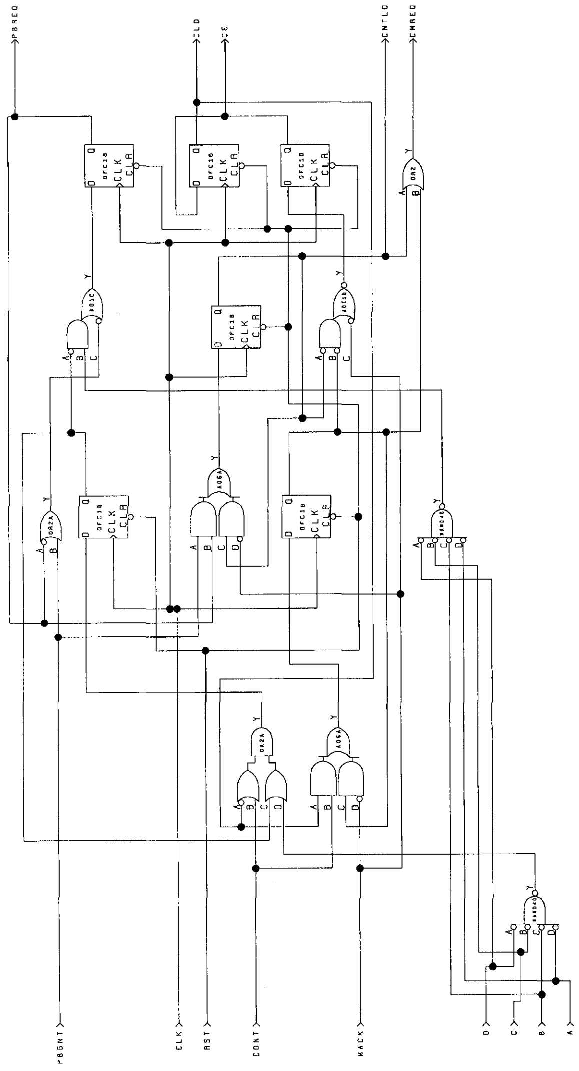


Figure 3 • *State Machine Schematic in ACT 3 Family*

Worst-Case Path (from A, B, C, D Registered Inputs)

$$= \overbrace{tCO + tRD2}^{DF1} + \overbrace{tPD + tRD1}^{NAND4D} + \overbrace{tPD + tRD1}^{AOI} + tSUD$$

$$= 2.3 + 1.4 + 2.3 + 1.0 + 2.3 + 1.0 + 0.6$$

$$= 10.9 \text{ ns}$$

Figure 4 • *State Machine Timing Using A1425A-2*

Shift Register Design

Shift registers with serially controlled data inputs and parallel outputs can be used as powerful controlled sequence generators. As a result, shift registers can be used in high-speed state machine designs.

For this application, a shift register sequentially shifts a single logic high signal while the rest of the output bits are low. This function can be implemented by simply connecting the output of the one flip-flop to the input of the next flip-flop and the output of the last flip-flop back to the input of the first flip-flop. Table 3 shows the function table of an 8-bit serial shift register. The width of the shift register is expandable by serially adding more flip-flops to the last stage. Figure 5 shows the schematic of an 8-bit shift register. Note that the shift register is designed so that it can be set to a known state with a reset signal.

State Machine Implementation

The state machine implementation is best illustrated by an example. The sample state machine has six states with three output bits. The sequence is organized such that only one output bit changes state for every clock pulse. Figure 6 shows the state diagram of the state machine.

Table 3* • *8-Bit Shift Register Function Table

RST	CLK	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
0	X	1	0	0	0	0	0	0	0
1	≠	0	1	0	0	0	0	0	0
1	≠	0	0	1	0	0	0	0	0
1	≠	0	0	0	1	0	0	0	0
1	≠	0	0	0	0	1	0	0	0
1	≠	0	0	0	0	0	1	0	0
1	≠	0	0	0	0	0	0	1	0
1	≠	0	0	0	0	0	0	0	1
1	≠	1	0	0	0	0	0	0	0

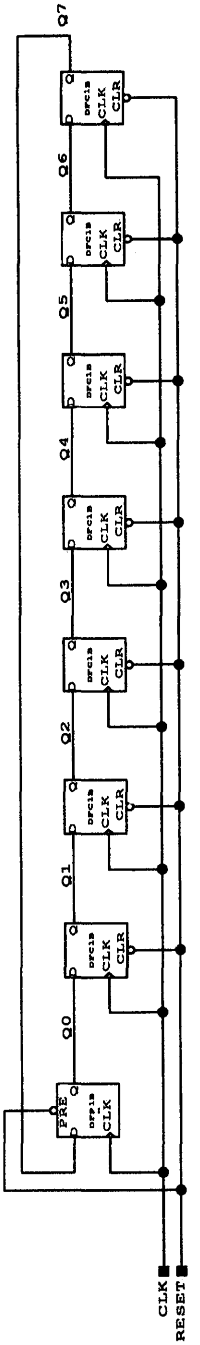


Figure 5 • Eight-Bit Shift Register Schematic

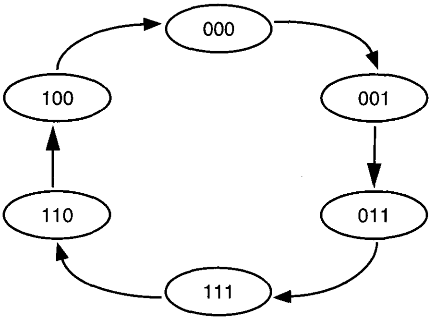


Figure 6 • *State Diagram of State Machine*

A six-state state machine requires a six-bit shift register with one register per state. Using this shift register determines the state sequence map of the state machine. The state machine creates the three output bits based on decoding the outputs of the shift register. The decoding logic is greatly minimized because there is only one output bit asserted in any state. Table 4 shows the state table of the state machine. When the outputs of the state machine are 001, the shift register outputs are 000010. In this state, Q5, Q4, Q3, Q2, and Q0 are all low and Q1 is high. Therefore, the state machine uses only Q1 for the decoding logic. The logic for the state machine outputs is based on the shift register outputs. Out0 is high when Q1 or Q2 or Q3 is high, Out1 is high when Q2 or Q3 or Q4 is high, and Out2 is high when Q3 or Q4 or Q5 is high. The complete decoding logic equations are the following:

Out0: Q1 + Q2 + Q3

Out1: Q2 + Q3 + Q4

Out2: Q3 + Q4 + Q5

Table 4 • *State Table for Example State Machine*

Shift Register Outputs						State Machine Outputs		
Q5	Q4	Q3	Q2	Q1	Q0	Out2	Out1	Out0
0	0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	1	1
0	0	1	0	0	0	1	1	1
0	1	0	0	0	0	1	1	0
1	0	0	0	0	0	1	0	0

Figure 7 shows the schematic diagram of the state machine using ACT 2 or ACT 3 macros. Note that the decoding logic requires only one level of logic to implement. Thus, using a shift register to implement state machines improves performance significantly.

Conclusion

To achieve the high performance possible with FPGAs, new register-rich techniques are needed. Two effective approaches include bit-per-state and shift register decoding. The user can manually implement these techniques or select FPGA-friendly algorithms with optimization tools such as the Actel ACTmap VHDL Synthesis tool.

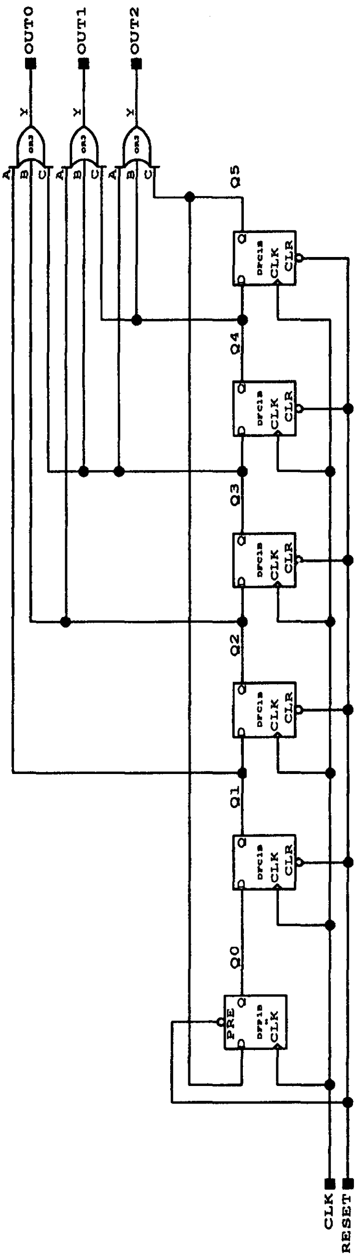


Figure 7 • *State Machine Schematic*

Implementing Multipliers with Actel FPGAs

Introduction

Hardware multiplication is a function often required for system applications such as graphics, DSP, and process control. The Actel architecture, which is multiplexer based, allows efficient implementation of multipliers with high performance. Furthermore, the Actel development tools allow the user quickly to create multipliers by using the appropriate algorithm and bit width needed for a specific application. The 1200XL family is the focus of the implementation of this application note, although other Actel families could also be used.

Multiplier Theory

The function of a binary unsigned multiplier, like its decimal counterpart, consists of a multiplicand (X), a multiplier (Y), and a product (P). The result is the product of the multiplier and the multiplicand (P = X * Y). Figure 1 shows the complete multiplication of two four-bit numbers producing an eight-bit product. As in decimal multiplication, the least significant digit of the multiplier combines with each digit of the multiplicand, forming a partial product (Y0X3, Y0X2, Y0X1, Y0X0). Three other partial products are similarly formed. To arrive at the final result, all four of the partial products are added (P7, P6...P0). Note that the most significant bit of the product (P7) is required, due to a possible carry from the other bits.

Conventional Multiplier Algorithm

The conventional approach to implementing a multiplier in digital logic is to AND individual multiplier and multiplicand bits to generate the partial products (PP1, PP2, PP3, PP4). For a four-bit multiplier, this would consist of 16 dual-input AND gates and three adders, as shown in Figure 2. The simplest method to sum the partial products is to have all three adders to be eight bits. Not all of the partial products generate eight bits, so smaller adders could be used. However, tracking which partial sums can be dropped and which need to propagate as carries to the next stage becomes complex and time-consuming, especially with larger bit widths. More important, the conventional multiplier implementation is resource intensive and does not produce optimal performance. Fortunately, another approach is possible.

L-Booth Algorithm Implementation

The L-Booth algorithm employs an alternative technique based on multiplexers, which are an ideal fit for the Actel architecture. For the four-bit implementation, the multiplier's two least significant bits are handled separately from the two most significant bits. Effectively, the multiplexer replaces the first stage of partial sum generation. Figure 3 illustrates the mathematics that explains the L-Booth algorithm. The two least significant multiplier bits (Y1, Y0) are handled separately from the two most significant bits (Y3, Y2). In both cases, the four possible combinations of the multiplier bits are covered with the multiplexer, resulting in the partial products PPA and PPB. Specifically for multiplexer A, the four combinations are zero (the trivial case), X (when Y1=0 and Y0=1), X shifted left or 2X (when Y1=1 and Y0=0), and 3X (when Y1=Y0=1). The multiplexers are eight bits deep to accommodate all eight possible inputs for the adders. To obtain the final product, the two partial sums are added with an eight-bit adder. High-speed adders are used in this implementation since shortest delay from input to output is the primary design constraint. Figure 4 shows the implementation of the L-Booth multiplier. The complete schematic for the fout-bit L-Booth multiplier is shown in Figure 5. Note the use of the five-bit adder to generate the required 3X input for the multiplexers. The name of the schematic is *LBMULT4*, indicating that it uses the L-Booth algorithm.

Multiplicand	>					**X3**	**X2**	**X1**	**X0**
Multiplier	>				x	**Y3**	**Y2**	**Y1**	**Y0**
1st partial product	>					Y0X3	Y0X2	Y0X1	Y0X0
2nd partial product	>				Y1X3	Y1X2	Y1X1	Y1X0	
3rd partial product	>			Y2X3	Y2X2	Y2X1	Y2X0		
4th partial product	>	+	Y3X3	Y3X2	Y3X1	Y3X0			
Final product	>	**P7**	**P6**	**P5**	**P4**	**P3**	**P2**	**P1**	**P0**

Figure 1 • *Four-Bit Binary Multiplication*

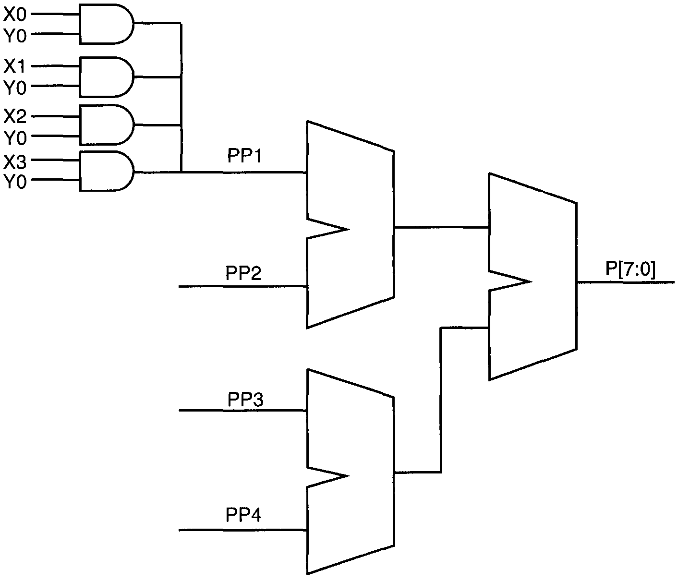


Figure 2 • *Classical Implementation of Four-Bit Multiplier*

							X3	X2	X1	X0
Only 2 LSB used	>					x			Y1	Y0
if Y1=0,Y0=0		I	0	0	0	0	0	0	0	0
if Y1=0, Y0=1		I	0	0	0	0	X3	X2	X1	X0
if Y1=1, Y0=0		I	0	0	0	X3	X2	X1	X0	0
if Y1=1, Y0=1		I	0	0	0	X3	X3+X2	X2+X1	X1+X0	0
Multiplexer A result	>		0	0	PPA5	PPA4	PPA3	PPA2	PPA1	PPA0
							X3	X2	X1	X0
Only 2 MSB used	>					x	Y3	Y2		
if Y3=0,Y2=0		I	0	0	0	0	0	0	0	0
if Y3=0, Y2=1		I	0	0	X3	X2	X1	X0	0	0
if Y3=1, Y2=0		I	0	X3	X2	X1	X0	0	0	0
if Y2=1, Y2=1		I	0	X3	X3+X2	X2+X1	X1+X0	0	0	0
Multiplexer B result	>		PPB7	PPB6	PPB5	PPB4	PPB3	PPB2	0	0
			0	0	PPA5	PPA4	PPA3	PPA2	PPA1	PPA0
		+	PPB7	PPB6	PPB5	PPB4	PPB3	PPB2	0	0
Final product	>		P7	P6	P5	P4	P3	P2	P1	P0

Figure 3 • *Four-Bit Binary Multiplication Using L-Booth Algorithm*

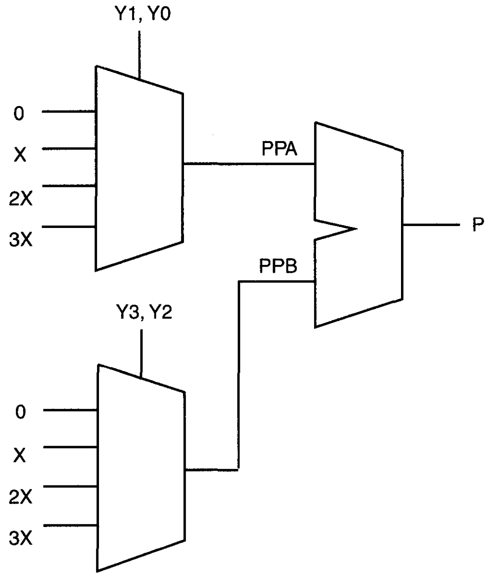


Figure 4 • *L-Booth Multiplier Implementation*

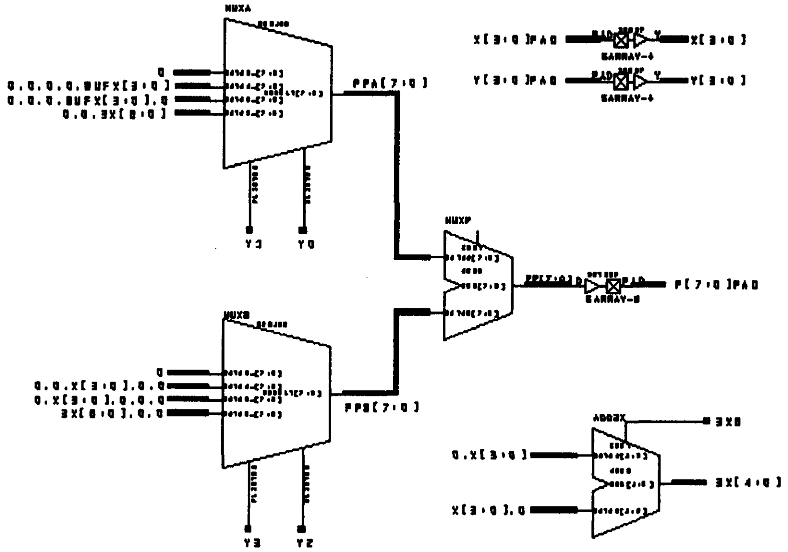


Figure 5 • Schematic Implementation of Four-Bit L-Booth Multiplier

Pipelined Multiplier

The previous multiplier implementations were entirely combinatorial. The output product is valid after all input values have propagated through the combinatorial logic. By introducing registers between the levels of logic, the stages of the multiplication can be broken up and synchronized with a clock. By doing so, the effective speed of multiple multiplications is increased, although the result is delayed by the number of register stages that are added. This delay is referred to as the *circuit latency*. Figure 6 shows the pipelined version of the four-bit multiplier, PMULT4.

Two levels of registers are used in the PMULT4 design, resulting in a latency of one cycle (i.e., the result appears one clock cycle later as shown in Figure 7). The distribution of registers is optimal since both stages contain three levels of logic. (The five-bit adder has two levels combined with one for the multiplexer, for a total of three levels. The eight-bit adder has three levels of logic internal.) This means that the multiplication can be done with three levels of combinatorial logic, one register, and data setup. Furthermore, with the 1200XL family, a combinatorial level is absorbed within the sequential module. This means that a four-bit multiplier could be done at a frequency in excess of 60 MHz. (Actual device performance is discussed in detail later in this application note.) The performance can be further increased at the cost of additional registers and circuit latency.

Design Tools

Designing multipliers with the Actel development tools is particularly easy since the basic blocks required (adders, multiplexers, and registers) can be quickly created with the ACTgen Macro Builder. Figure 8 shows the ACTgen main menu with macro category selections. As an example, an eight-bit fast adder with the name of *SAMPLE* will be created with ACTgen. The adder menu in Figure 9 shows the available options: adder variations, bus width, carry in, and carry out. The summary report is shown in Figure 10, and the generated symbol is shown in Figure 11. The multiplexers and registers can be created equally quickly with ACTgen. Using this approach, it is very easy to create multipliers of any bit width by changing the ACTgen parameters. Another modification that can be changed is the type of adder created by ACTgen. By selecting a ripple adder instead of a high-speed one, a more compact multiplier can be created. By making such a change, the four-bit multiplier would require 20 percent fewer modules.

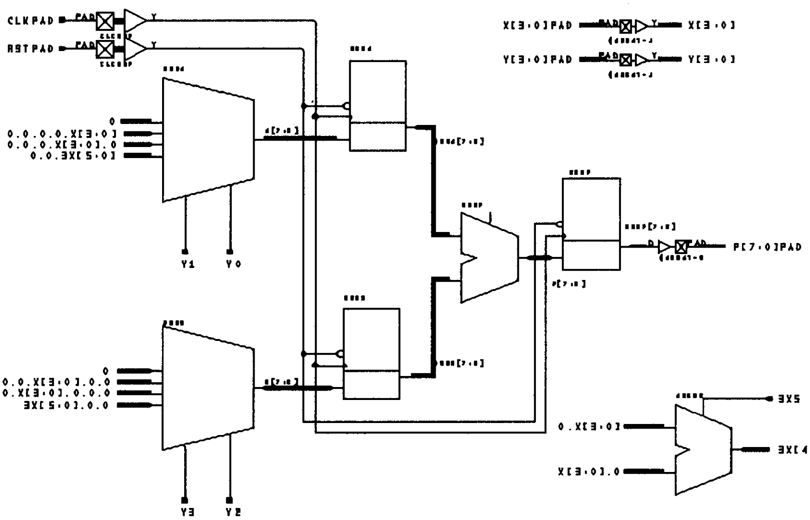


Figure 6 • Pipelined Four-Bit Multiplier

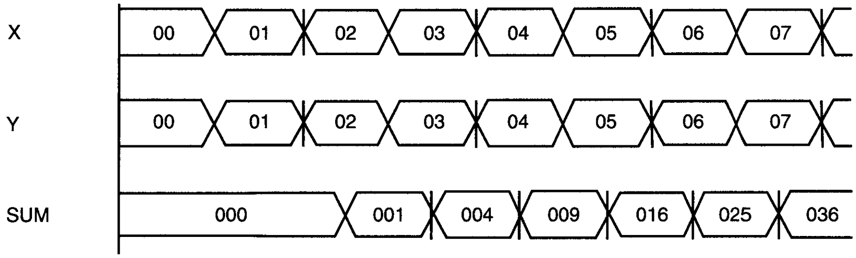
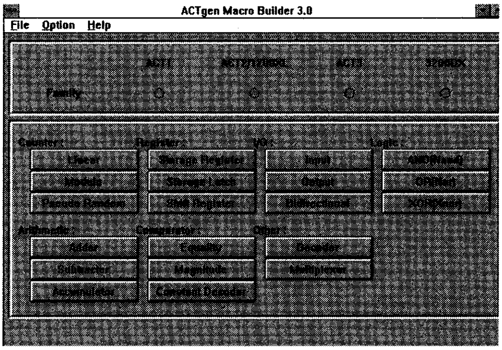
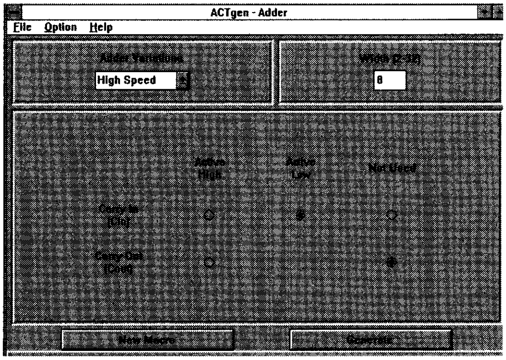


Figure 7 • Timing Waveform of Pipelined Four-Bit Multiplier with One Cycle Latency



Family: ○ ACT1 ○ ACT2/1200XL ○ ACT3 ○ 3200DX

Figure 8 • ACTgen Macro Builder Main Menu

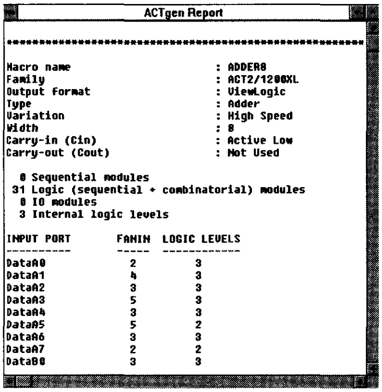


Adder Variations: High Speed

Width (2-32): 8

	Active High	Active Low	Not Used
Carry In [Cin]	○	●	○
Carry Out [Cout]	○		●

Figure 9 • ACTgen Macro Builder Adder Menu



```
ACTgen Report

********************************************************

Macro name                      : ADDER8
Family                          : ACT2/1200XL
Output Format                   : ViewLogic
Type                            : Adder
Variation                       : High Speed
Width                           : 8
Carry-in (Cin)                  : Active Low
Carry-out (Cout)                : Not Used

  0 Sequential modules
 31 Logic (sequential + combinatorial) modules
  0 IO modules
  3 Internal logic levels

INPUT PORT      FANIN  LOGIC LEVELS
----------      -----  ------------
DataA0            2         3
DataA1            4         3
DataA2            3         3
DataA3            5         3
DataA4            3         3
DataA5            5         2
DataA6            3         3
DataA7            2         2
DataB0            3         3
```

Figure 10 • ACTgen Macro Builder Summary Report

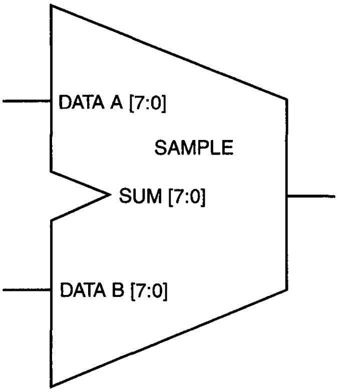


Figure 11 • ACTgen-Generated Adder Symbol

Hardware Description Language

Another approach to implementing multipliers is defining the functionality in Hardware Description Language (HDL). Generally, higher-level descriptions are fast to implement but somewhat slower than a manual or macro generator approach. A multiplier is particularly challenging since it is exclusively an arithmetic function, which is typically not well suited to pure synthesis algorithms. However, the Actel ACTmap VHDL Synthesis tool performs very well. The design was optimized for area and required only two modules more than the best manual implementation.

The VHDL source file for a four-bit multiplier is shown in Figure 12. Note how compact the required description is compared with previous schematic implementations. Furthermore, to create an eight-bit, sixteen-bit, or n-bit version would require only changing the BIT_VECTOR definitions.

```
library ASYL;
use ASYL.pkg_arith.all;

entity MULT4 is
            port(X, Y: in BIT_VECTOR(3 downto 0);
                        P: out BIT_VECTOR(7 downto 0));
end MULT4;

architecture ARCHI of
            MULT4 is
            begin       P <= X * Y;
end ARCHI;
```

Figure 12 • VHDL Source Code for Four-Bit Multiplier

Multiplier Performance

All of the multiplier approaches discussed have been implemented in the Actel 1200XL devices. Table 1 shows all of these including many of the statistics required to make an educated decision on the best approach, depending on design needs. As with most designs, there is almost always a trade-off of system performance and design resources. In the case of the multiplier implementations, this is also true. In fact, there is a monotonic relationship of increasing performance with increasing module count for every multiplier implementation. The speed is obtained by implementing the design for the 1225XL-1 device and obtaining static timer worst-case commercial conditions after place and route. The speed refers to the worst-case path from input pad to output pad for the combinatorial multipliers and the longest internal-clock-to-data path (including data setup time). The "Util" column indicates what percentage of the 1225XL is being used by the multiplier.

Tables 2 and 3 show the statistics for eight-bit and sixteen-bit multipliers, respectively. As before, performance is based on actual placed and routed designs using the Actel static timing analyzer.

Conclusion

Multipliers can be quickly and easily implemented in Actel FPGAs, providing both efficient usage and high performance. The 1200XL family is particularly well suited for the applications. There is a range of options available to the user when designing multipliers: speed/area trade-offs, latency, and design method. Armed with this application information, the Actel development tools, and the 1200XL devices, designers can effectively create multipliers to meet their individual requirements.

Table 1 • *Statistics of Four-Bit Multipliers*

Design	Description	Device	Speed	# Modules	# Levels	Latency	Util
PMULT4	4 by 4 pipelined	1225XL-1	57 MHz	67	3	1 cycle	15%
LBMULT4	4 by 4 comb	1225XL-1	24 MHz	60	6	n/a	13%
PRMULT4	4 by 4 pipelined, ripple adder	1225XL-1	21 MHz	48	10	1 cycle	11%
VHDMULT4	4 by 4 comb, VHDL source	1225XL-1	19 MHz	50	8	n/a	12%
RBMULT4	4 by 4 comb, ripple adder	1225XL-1	14 MHz	48	12	n/a	11%

Table 2 • *Statistics of Eight-Bit Multipliers*

Design	Description	Device	Speed	# Modules	# Levels	Latency	Util
PMULT8	8 by 8 pipelined	1225XL-1	44 MHz	276	3	3 cycles	62%
LBMULT8	8 by 8 comb	1225XL-1	14 MHz	232	10	n/a	52%
PRMULT8	8 by 8 pipelined	1225XL-1	10 MHz	188	17	3 cycles	42%
RBMULT8	8 by 8 comb	1225XL-1	8 MHz	164	22	n/a	37%
VHDMULT8	8 by 8 comb, VHDL source	1225XL-1	8 MHz	325	22	n/a	73%

Table 3 • *Statistics of Sixteen-Bit Multipliers*

Design	Description	Device	Speed	# Modules	# Levels	Latency	Util
PMULT16	16 by 16 pipelined	1280XL-1	28 MHz	1011	4	3 cycles	83%
LDMULT16	16 by 16 comb	1280XL-1	8 MHz	844	16	n/a	69%
PRMULT16	16 by 16 pipelined	1280XL-1	5 MHz	786	33	3 cycles	64%
RBMULT16	16 by 16 comb	1240XL-1	4 MHz	656	40	n/a	96%

Application Note

Synchronous Dividers in Actel FPGAs

Synchronous dividers are useful in numerous applications, such as prescalers, timing generators, and multi-phase clocks. Although ripple dividers use less logic, glitches and potential race conditions make them hazardous in all but the simplest applications. Synchronous dividers offer a better solution for reliable operation.

Figure 1 shows divide-by-two to divide-by-ten synchronous dividers using combinable Actel logic modules. Using the Actel ACT 2, 1200XL, 3200DX, and ACT 3 architectures, the combinatorial gates will be absorbed with the following flip-flop. The flip-flops used do not have reset pins as dividers are generally used as astable devices. The OR-AND gates are used for some of the dividers to avoid nonconvergent illegal states. To initialize the dividers for simulation, the output can be held high and clocked once for each flip-flop, then released. For example, with the ViewSim simulator, the divide-by-three synchronous divider (with output node DIV3) would be initialized with the following command file sequence:

```
h DIV3
cycle 2
r DIV3
cycle 5
```

The resulting waveforms for each of the dividers are shown in Figure 2. Note that the duty cycle is 50 percent for all even dividers and as close to this as possible for the others.

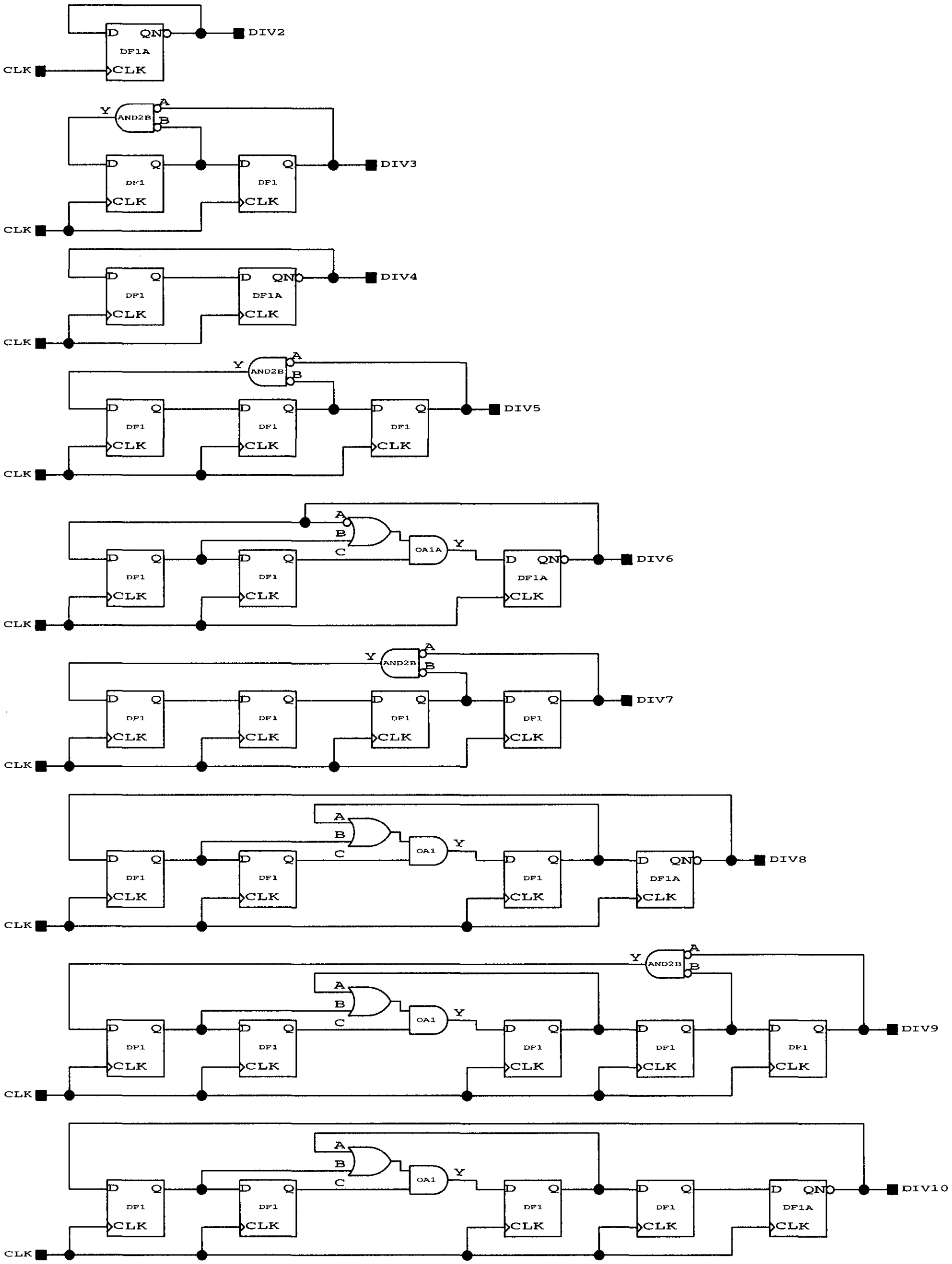


Figure 1 • Actel Implementation for Synchronous Dividers

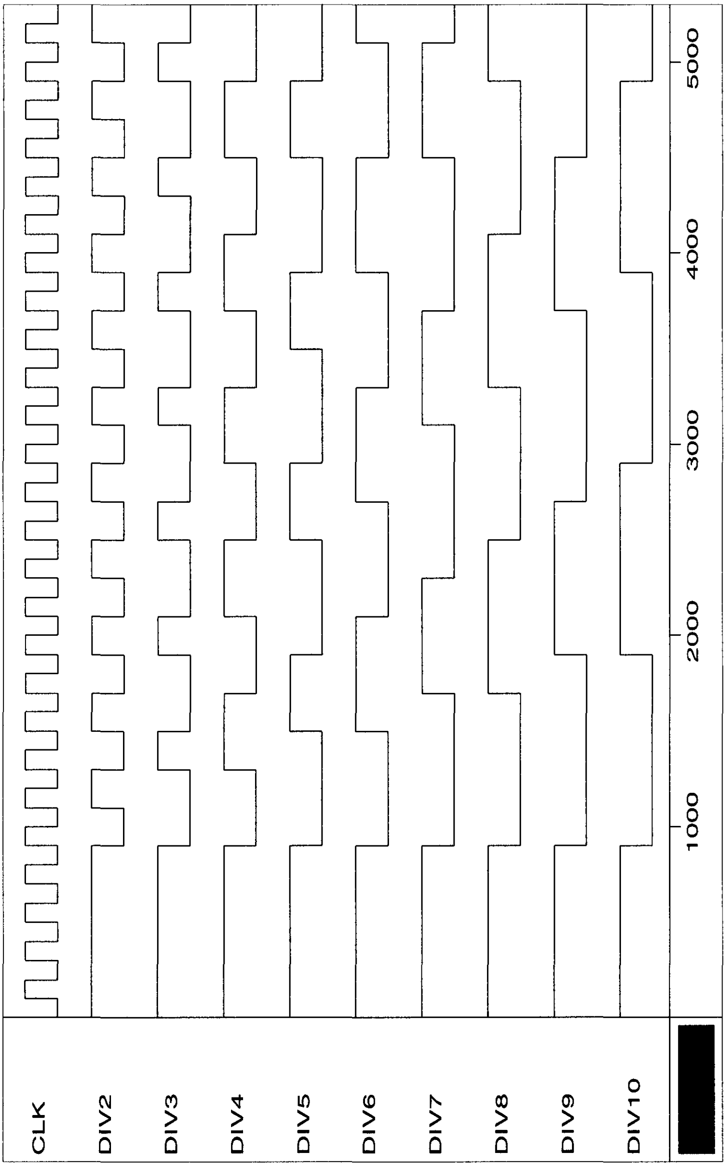


Figure 2 • *Synchronous Divider Waveforms*

Using FPGAs for Digital PLL Applications

In addition to purely digital applications, many designs use Field Programmable Gate Arrays (FPGAs) for DSP. We'll examine one such application, digital PLLs, to show various ways of implementing PLL designs using FPGAs.

Pulse Steal PLL

In telecommunications applications, it is often desirable to generate a digital signal that is locked to an incoming signal and is some multiple of its frequency. A drawing of a pulse steal PLL, which is a simple way to generate such a signal, is shown in Figure 1. Note that the design contains an ordinary oscillator but no VCO. Except for the crystal, the entire design will operate in an FPGA.

Note the frequency relationship that holds at points A and B in the figure, where

$$\text{OSC}/(\text{K}*\text{M}) = \text{Input}/\text{N} = \text{Comparison Frequency} \qquad (1)$$

The technique is based on selecting a reference oscillator frequency slightly higher than OSC. This frequency (OSC+) should be chosen so that

$$1/\text{Comparison Freq.} - (\text{K}*\text{M})/(\text{OSC+}) = .5 * (1/\text{OSC}) \qquad (2)$$

The right side of equation 2 equals one-half the period of the reference oscillator.

The reference oscillator frequency delta will cause point B (the detector flip-flop D input) to begin to precede point A (the detector flip-flop clock input) by half a period. When the edge of the D input is sufficient, the detector will clock true and begin a pulse train through the two deglitching flip-flops. The output of the second of these clears all three flip-flops and steals a pulse by disabling the divide by K output. Stealing the pulse puts point B behind A until the reference oscillator delta can move it ahead by one period, repeating the cycle. Points A and B are always within one-half a cycle of each other.

The circuit allows the frequency of the output signal to be selected simply by adjusting the values of the dividers K and M. The lock range of the loop is given by the following:

$$\text{Lock Range} = \pm\ (\text{OSC+}/\text{osc})/\text{Input} \qquad (3)$$

Jitter-Bounded Digital PLL

Another technique[1] for generating a wide variety of synchronized clock frequencies with low jitter employs an accumulator Digital Controlled Oscillator (DCO) and phase and frequency comparators. The system, shown in Figure 2, can lock to any division of a reference frequency (F ref.) as selected by the data loaded into frequency divider counters.

The Successive Approximation Register (SAR) and its controller serve as a low-pass filter supplying the DCO with frequency and phase correction data. Among the three inputs to the SAR controller is the F ref. divided by a factor Q to form F q.

The other two inputs come from the phase and frequency (zero) comparators. The frequency comparator output is the DCO frequency divided by P to form F p. When the system is in lock, the following equation is true:

$$\text{F dco} = (\text{P}/\text{Q}) * \text{F ref.} \qquad (4)$$

The heart of the system is the accumulator DCO, which determines the ability to lock to a frequency and the amount of jitter allowed. The DCO consists of a four-bit accumulator whose input is fed by the SAR. The DCO input value is determined from the phase and frequency comparison feedback loops. The most significant bit of the accumulator output is the DCO output signal. It is generated by successively adding the SAR value to itself at the high-frequency system clock rate. The frequency comparator uses the value of P to divide the DCO frequency. If the frequency is out of lock during a period of F ref., the comparator asserts greater-than-zero or less-than-zero to the SAR controller to modify the value of the register. If the P counter output is zero, the DCO has the correct frequency.

The DCO latch acts as a phase register indicating the phase of the DCO with respect to F ref. The DCO phase is calculated by the N most significant accumulator output bits. When the DCO is out of phase, the jitter, or phase difference, is detected by the phase comparator and accumulates with time until it equals one period. The feedback loops then cause the SAR register controller to load a correcting value into the register or to clear the accumulator with a synchronizing pulse.

The Jitter-Bounded DPLL may be implemented entirely on an Actel FPGA. The resource requirements vary with the relationships of the system input and output frequencies, but for any F ref., system clock, and desired output frequency, the design is easily accommodated on an Actel FPGA.

References:

1. S. Walters and T. Troudet, "Digital Phase-Locked Loop with Jitter Bounded," *IEEE Transactions on Circuits and Systems*, Vol 36, No. 7, July 1989.

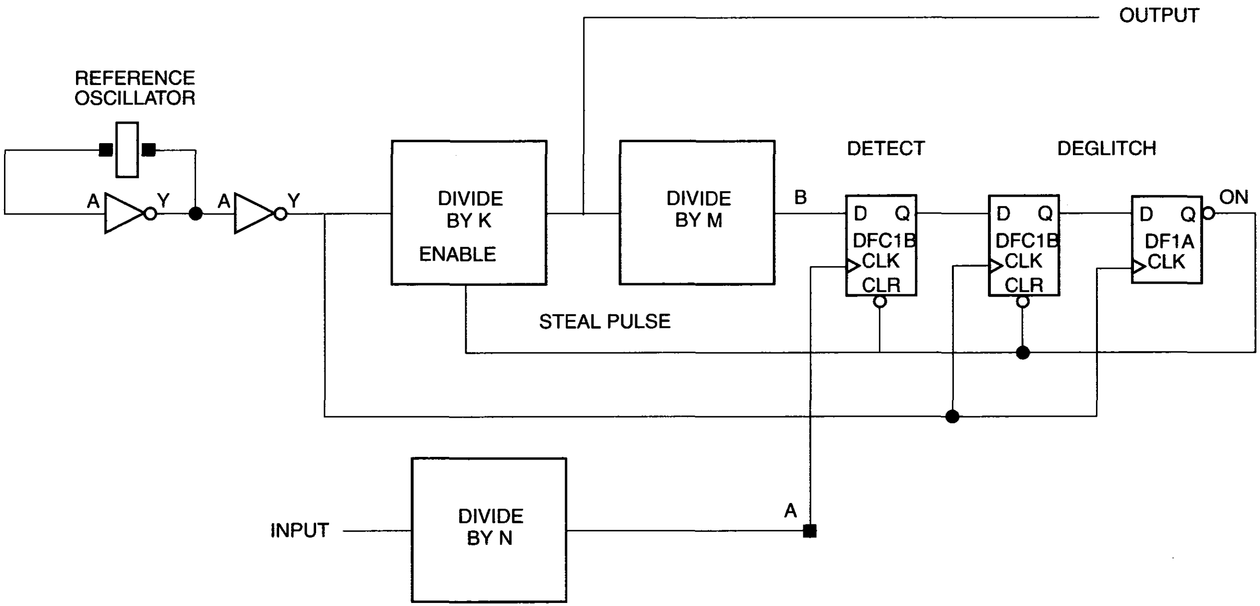


Figure 1 • Pulse Steal PLL

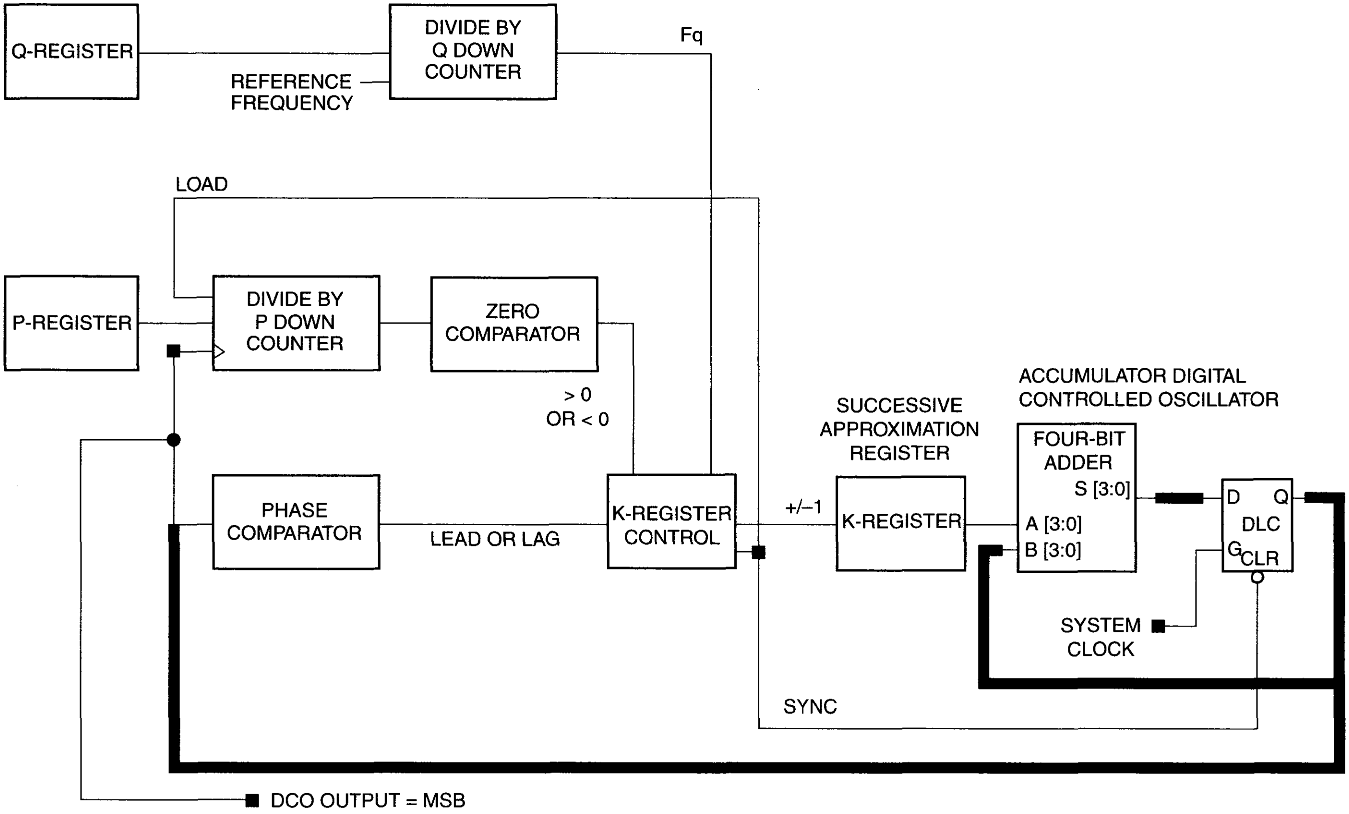


Figure 2 • Jitter-Bounded Digital PLL

Oscillators for Actel FPGAs

Crystal Oscillator

Oscillators are fundamental design circuits used to provide a reference clock signal essential for digital designs. Crystal oscillators provide a simple solution for precise, stable, and calibration-free clocks. An on-chip crystal oscillator can be implemented with Actel devices using the traditional configuration shown in Figure 1. This oscillator has been tested up to 20 MHz and is used in Actel programmers. The 10 MΩ resistor provides a negative feedback path for the inverter, which makes it behave like a high-gain amplifier. The remaining passive elements, including the crystal, form a pi network that provides a 180 degree phase inversion. The inverter will lock on to the parallel resonant frequency of the crystal, thus providing a very stable output. The RC network also acts as a low-pass filter to ensure that the crystal operates at the fundamental frequency and not a harmonic frequency. The capacitor values range from 5 to 30 picofarads and depend on the crystal frequency. Some experimentation is suggested to get an optimal value for a specific design. Generally, the two capacitors will have the same value, although the capacitor connected to the input of the inverter can be varied independently to alter the output frequency by ± 0.1 percent.

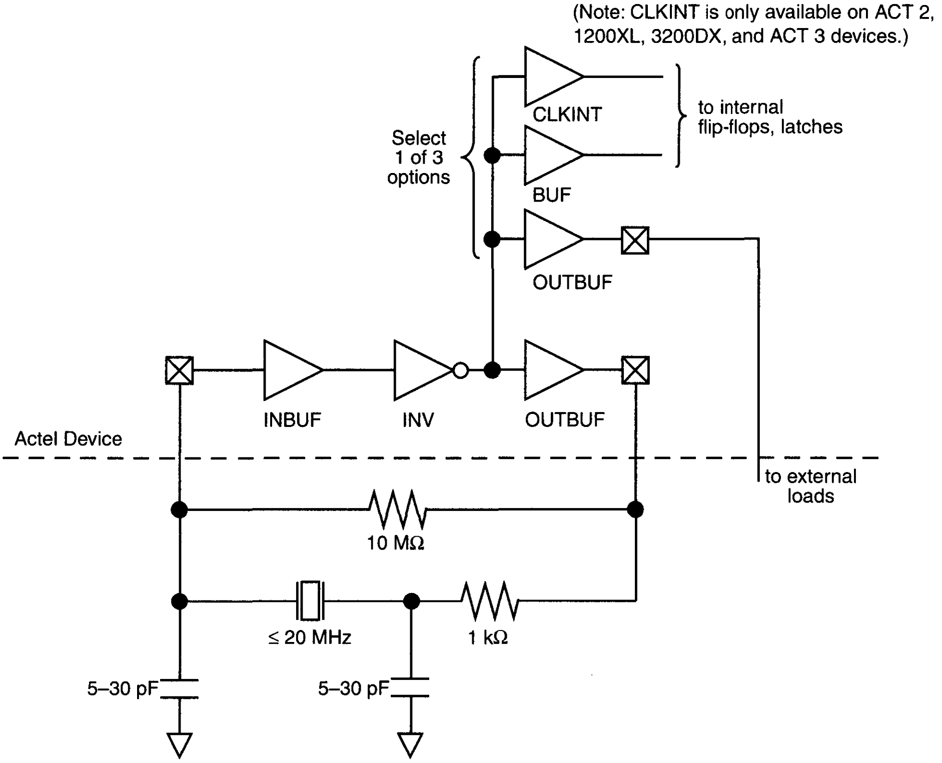

(Note: CLKINT is only available on ACT 2, 1200XL, 3200DX, and ACT 3 devices.)

Figure 1 • Crystal Oscillator Circuit

A second OUTBUF or CLKBIBUF output buffer is used to provide a sharper clock signal at full amplitude when used outside the FPGA. Alternatively, the output buffer can be replaced with an internal buffer, which will allow a direct connection to internal macros. For ACT 2, 1200XL, 3200DX, and ACT 3 devices, the CLKINT macro can be used allowing high fanout drive capability of internal macros with minimal skew.

Fix the placement of the oscillator I/O macros to adjacent package pins to minimize internal delays. Also, fixing the I/O macros near ground pins and far from other high-speed switching I/O pads minimizes noise effects.

RC Oscillator

For applications not requiring the accuracy of a crystal, there is an RC oscillator that can be used in an Actel device as shown in Figure 2. As a strong word of caution, this circuit is not recommended for system clocks, since it is heavily dependent on resistor and capacitor tolerances, process variation, and temperature. Some applications for this lower cost oscillator include LCD backplane and debounce circuits.

The circuit reaches alternate switching thresholds by charging and discharging the capacitor with resistor R2. The R1 resistor provides a better square wave output by minimizing effects of input protection diodes of the input buffer. The approximate formula for output frequency is:

$$\mathit{frequency} \cong \frac{1}{2.2\ \mathrm{R2}\ \mathrm{C}}$$

The formula is most accurate if parameters have the following limitations:

R1 > 10R2

10K > R2 > 1 M

1000 pF > C > 10 µF

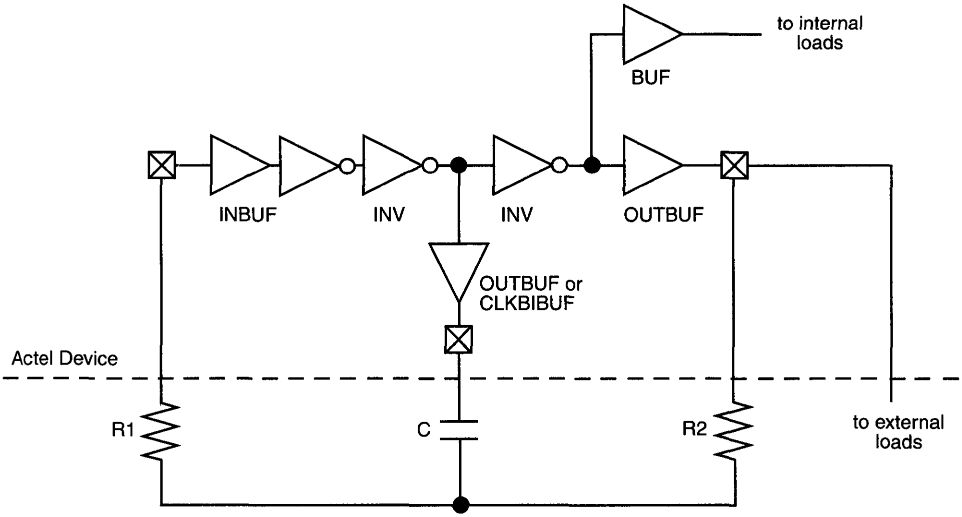


Figure 2 • RC Oscillator

Field Upgradability Using Actel One-Time-Programmable FPGAs

The conventional wisdom has been that one-time-programmable (OTP) FPGAs can't be used in applications for which logic needs to be upgraded in the field. This application note will show that in many cases, OTP FPGAs can be used in applications requiring this capability.

Introduction

In some applications, detailed specifications for functionality requirements are not available until well into the design process—in some cases, even after the design process. Usually the data path portion of the design is well defined at the beginning of a project, but some of the details of the control logic are not well defined. With a reprogrammable technology, the designer can reprogram the device in the field to adapt to different control requirements. An OTP device can also be reprogrammed in the field to accomplish similar objectives if the designer plans for it.

A typical electronic system is shown on the left side of Figure 1. A main CPU board communicates with a variety of peripheral boards, perhaps in an equipment bay for a telecommunication system. The block diagram on the right of Figure 1 shows the communication path between the CPU and an FPGA. The CPU can write data into the FPGA's on-chip SRAM or registers to alter the operation of the FPGA's control logic.

Application Example

The operation required on a data packet might be well known, but the number of data words and the size of the header might not be specified. These parameters cannot be embedded in a hardwired state machine because they are undefined. These parameters can be stored in read/write registers, however, and can be used to load counters in conjunction with the state machine. The counters can then be used to step through the header and the data packets. This capability allows complete programmability with respect to data and header size. A block diagram implementation of this application is shown in Figure 1.

The programmable header logic in Figure 2 demonstrates how to make the header size an in-system reconfigurable value. The CPU can write any desired value into the 8-bit register Reg8. The state machine will load the value into the counter and then enable the counter in the state associated with the header. When the counter has counted down and the Done signal goes active, the header state is exited. Similar logic can be used for data packet control.

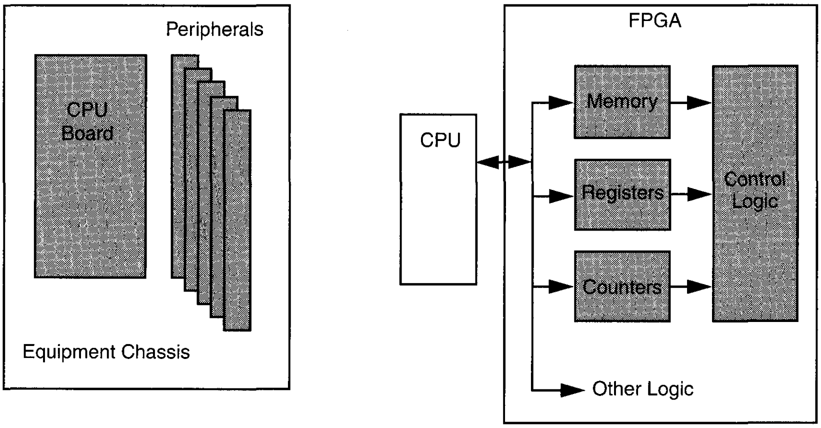


Figure 1 • Field Upgrading with OTP FPGAs: Block Diagram

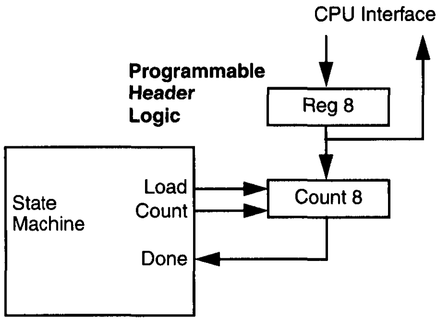


Figure 2 • Programmable Header and Data Length State Machine

The programmable state variable techniques used in Figure 1 can be applied to many different applications. Programmable delays can be important in microprocessor peripherals for wait-state generation and for burst length of synchronous processor interfaces. In networking and communication systems, values for packet size, header size, and time delays can all be programmable.

Another approach to field upgradability is to use on-chip SRAM to create state machines that can be modified once the system has been installed in the field. Actel's 3200DX family has on-chip dual-port SRAM, which can be used to create microprogrammed state machines, using the dual-port SRAM to store next-state and control-output signals. The dual-port nature makes it easy to load the SRAM from an initialization path (perhaps from an on-board processor or memory) by using the write port and to access the microcode stored in SRAM via the read port. Figure 3 shows an example application in which the dual-port SRAM is used to create a microprogrammed machine that controls a complex data path that, in turn, processes an incoming data stream.

The on-board processor loads the SRAM data from the processor interface (a simple multiplexed address and data bus). The next-state logic is predefined and can be a simple next-state address generator (perhaps similar to a bit-slice device). Test conditions are signals generated by the data path used to modify the state sequence and can be a combination of fixed function logic and programmable state variable functions. Programmable delays, for example, can use data from the SRAM to initialize compare registers or other programmable functions.

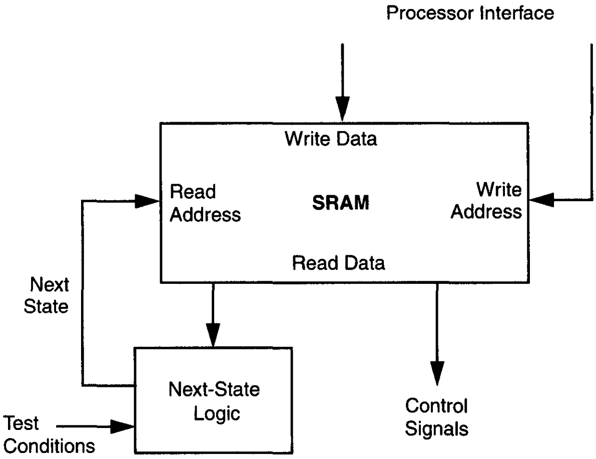


Figure 3 • Microprogrammed State Machine Using 3200DX Dual-Port SRAM

Other Programmable Functions

The following are other examples of programmable control functions that can be used in OTP FPGAs to support field reconfigurability:

- Use of comparitors and variable registers to find a programmable value in a data stream (e.g., for start and stop symbols)
- Use of dual-port RAM for programmable code conversion (for multiple-code support in the same hardware at different times)
- Use of counters and variable registers for programmable delay (when the size of a data packet is unknown)
- Use of XOR gates and variable registers for programmable signal polarity (when you don't know the polarity of an incoming signal)
- Use of multiplexers and variable registers for programmable data flow between buses (when you don't know to which bus result data will be routed)

Summary

OTP FPGAs can be used in many applications for which field reconfigurability of system logic is required. Programmable delays, RAM-based code conversions, RAM-based state machines, and programmable comparitors are all possible in OTP FPGAs.

Actel

Application Note

Predicting the Power Dissipation of Actel FPGAs

Introduction

Calculating the power dissipation of field programmable gate arrays (FPGAs) is similar to using the calculations for other CMOS ASIC devices, such as gate arrays and standard cells. The power dissipation depends on such factors as utilization, average operating frequency, and load conditions. In contrast, most PALs and PLDs have a fixed power consumption.

This application note discusses power dissipation and the concept of equivalent power capacitance. The general approach to calculating power in an Actel device will be described using equivalent power capacitance values for the devices. This general equation is useful if internal switching frequencies can be accurately determined. Since this is often difficult to do, a set of approximation curves based on average frequency rules of thumb are provided. The graphs provide an upper limit estimate for active power sufficient for most designs.

General Power Equation

$$P = [I_{CC}\text{standby} + I_{CC}\text{active}] * V_{CC} + I_{OL} * V_{OL} * N + I_{OH} * (V_{CC} - V_{OH}) * M \qquad (1)$$

Where:

I_{CC}standby is the current flowing when no inputs or outputs are changing.

I_{CC}active is the current flowing due to CMOS switching.

I_{OL} and I_{OH} are TTL sink/source currents.

V_{OL} and V_{OH} are TTL level output voltages.

N equals the number of outputs driving TTL loads to V_{OL}.

M equals the number of outputs driving TTL loads to V_{OH}.

Determining N and M depends on the design and the system I/O. An accurate determination of power dissipation comes from two components, static and active, which are considered separately.

Static Power Component

Actel FPGAs have small static power components that result in lower power dissipation than PALs or PLDs. By integrating multiple PALs/PLDs into one FPGA, an even greater reduction in board-level power dissipation can be achieved.

The power due to standby current is typically a small component of the overall power. For an ACT 3 device, the standby power is specified as 5 mWatts, worst case.

The static power dissipated by TTL loads depends on the number of outputs driving high or low and the DC load current. Again, this value is typically small. For instance, a 32-bit bus sinking 4 mA at 0.33 V will generate 42 mWatts with all outputs driving low and 140 mWatts with all outputs driving high. The actual dissipation will average somewhere between as I/Os switch states with time.

Active Power Component

Power dissipation in CMOS devices is usually dominated by the active (dynamic) power dissipation. This component is frequency dependent, a function of the logic and the external I/O. Active power dissipation results from charging internal chip capacitances of the interconnect, unprogrammed antifuses, module inputs, and module outputs, plus external capacitance to PC board traces and load device inputs. An additional component of the active power dissipation is the totem-pole current in CMOS transistor pairs. The net effect can be associated with an equivalent capacitance that can be combined with frequency and voltage to represent active power dissipation.

Equivalent Capacitance

The power dissipated by a CMOS circuit can be expressed by the equation:

$$\text{Power } (\mu\text{Watts}) = C_{EQ} * V_{CC}^2 * F \qquad (2)$$

Where:

C_{EQ} is the equivalent capacitance expressed in pF.

V_{CC} is the power supply in volts.

F is the switching frequency in MHz.

Equivalent capacitance is calculated by measuring I_{CC}active at a specified frequency and voltage for each circuit component of interest. Measurements have been made over a range of frequencies at a fixed value of V_{CC}. Equivalent capacitance is frequency independent so that the results may be used over a wide range of operating conditions. The results for ACT 1, ACT 2, 1200XL, and ACT 3 devices are given in Table 1.

Table 1 • CEQ Values for Actel FPGAs

	ACT 1	ACT 2	1200XL	ACT 3
Modules	6.3	7.7	7.7	8.2
Input Buffers	16.0	18.0	18.0	1.5
Output Buffers	25.0	25.0	25.0	2.3
Clock Buffer Loads	5.3	2.5	2.5	N/A
I/O Clock Buffer Loads	N/A	N/A	N/A	0.4
Dedicated Array Clock Buffer Loads	N/A	N/A	N/A	0.5
Routed Array Clock Buffer Loads	N/A	N/A	N/A	0.5 + fixed/ device

Finding the active power dissipated from the complete design requires solving Equation 2 for each component type. This requires the switching frequency of each part of the logic. The exact equation is a piecewise linear summation over all components as shown in Equation 3. For ACT 1, ACT 2, and 1200XL devices:

$$\text{Power} = [(m * C_{EQ} * f_m)_{\text{modules}} + (n * C_{EQ} * f_n)_{\text{Inputs}} + (p * (C_{EQ} + C_L) * f_p)_{\text{Outputs}} + (q * C_{EQ} * f_q)_{\text{clk\_loads}}] * V_{CC}^2 \qquad (3)$$

Where:

m = Number of logic modules switching at frequency f_m

n = Number of input buffers switching at frequency f_n

p = Number of output buffers switching at frequency f_p

q = Number of clock loads on the global clock network

f_m = Average logic module switching rate in MHz

f_n = Average input buffer switching rate in MHz

f_p = Average output buffer switching rate in MHz

f_q = Frequency of global clock

C_L = Output load capacitance

For ACT 3 devices:

$$\text{Power } (\mu W) = [(m \times 8.2 \times f_1) + (n \times 1.5 \times f_2) + (p \times (2.3 + C_L) \times f_3) + (q \times 0.5 \times f_4) + ((r_1 + 0.5\, r_2) \times f_5) + (s \times 0.4 \times f_6)] \times V_{CC}^2 \qquad (4)$$

Where:

m = Number of logic modules switching at f_1

n = Number of input buffers switching at f_2

p = Number of output buffers switching at f_3

q = Number of clock loads on the dedicated array clock network

A1415:	$q = 104$
A1425:	$q = 160$
A1440:	$q = 288$
A1460:	$q = 432$
A14100:	$q = 697$

r_1 = Fixed capacitance due to routed array clock network

A1415:	$r_1 = 60$
A1425:	$r_1 = 75$
A1440:	$r_1 = 105$
A1460:	$r_1 = 145$
A14100:	$r_1 = 195$

r_2 = Number of clock loads on the routed array clock network

s = Number of clock loads on the dedicated I/O clock network

A1415:	$s = 80$
A1425:	$s = 100$
A1440:	$s = 140$
A1460:	$s = 168$
A14100:	$s = 228$

f_1 = Average logic module switching rate in MHz

f_2 = Average input buffer switching rate in MHz

f_3 = Average output buffer switching rate in MHz

f_4 = Average dedicated array clock rate in MHz

f_5 = Average routed array clock rate in MHz

f_6 = Average dedicated I/O clock rate in MHz

C_L = Output load capacitance in pF

Since all of the modules or inputs or outputs do not switch at the same frequency, a weighted average can be used. For example, a design consisting of 100 modules switching at 10 MHz and 200 modules switching at 5 MHz would have a weighted average frequency of:

$$f_{ave} = [(100 * 10) + (200 * 5)]/(100 + 200) = 6.67 \text{ MHz}$$

Determining Average Frequency

Determining the exact average frequency for a design requires a detailed understanding of the data input values to the circuit. Logic simulation can provide insight into average frequency, although simulation is limited by the percentage of real-time stimulus that can be applied. Fortunately, studies based on large numbers of ASIC designs have been made to determine rules of thumb for average switching frequency in logic circuits. These rules are meant to represent worst-case scenarios, hence their use for predicting the upper limits of power dissipation is generally acceptable. The rules given in Tables 2 and 3 are for ACT 1, ACT 2, and 1200XL devices. Table 4 gives the rules for ACT 3 devices. Using these rules, we can develop power estimates.

Table 2 • *Rules for Determining Average Frequency for ACT 1 Family*

1.	Module Utilization = 90%
2.	Average Module Frequency = F/10
3.	1/3 of I/Os are Inputs
4.	Average Input Frequency = F/5
5.	2/3 of I/Os are Outputs
6.	Average Output Frequency = F/10
7.	Clock Net Loading = 45%
8.	Clock Net Frequency = F

Table 3 • *Rules for Determining Average Frequency for ACT 2 and 1200XL Families*

1.	Module Utilization = 80% of combinatorial modules
2.	Average Module Frequency = F/10
3.	1/3 of I/Os are Inputs
4.	Average Input Frequency = F/5
5.	2/3 of I/Os are Outputs
6.	Average Output Frequency = F/10
7.	Clock Net 1 Loading = 40% of sequential modules
8.	Clock Net 1 Frequency = F
9.	Clock Net 2 Loading = 40% of sequential modules
10.	Clock Net 2 Frequency = F/2

Table 4 • *Rules for Determining Average Frequency for ACT 3 Family*

1.	Logic Modules (m)	= 80% of modules
2.	Average module switching rate (f_1)	= F/10
3.	Inputs switching (n)	= # I/Os used/12
4.	Average input switching rate (f_2)	= F
5.	Outputs switching (p)	= # I/Os used/15
6.	Output loading (C_L)	= 35
7.	Average output switching rate (f_3)	= F/2
8.	Dedicated array clock loads (q)	= fixed by device
9.	Average dedicated array switching rate (f_4)	= F
10.	Routed array fixed capacitance (r_1)	= fixed by device
11.	Routed array clock loads (r_2)	= 40% of sequential modules
12.	Average routed array switching rate (f_5)	= F/2
13.	I/O clock loads (s)	= # I/Os used
14.	Average I/O switching rate (f_6)	= F

Average Frequency Example

While some portions of a logic design switch at the system frequency, F, most of the logic switches at a reduced (or divided) frequency. Consider a 16-bit synchronous counter with a system input clock equal to F as shown in Figure 1.

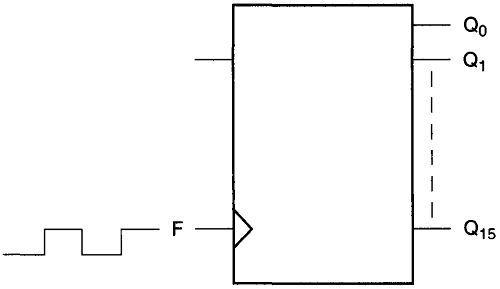


Figure 1 • 16-Bit Synchronous Counter

Where:

The Q0 output is switching at F/2 (or $1/2^1$),

The Q1 output is switching at F/4 (or $1/2^2$),

The Q15 output is switching at F/65536 (or $1/2^{16}$).

The average frequency is:

$F_{ave} = 1/16 * (1/2^1 + 1/2^2 + \ldots\ldots\ldots 1/2^{16}) \cong F/16$

Thus, the average frequency of an n-bit synchronous counter switching at F MHz is F/n.

Estimated Power

The rules in Tables 2 and 3 are applied to ACT 1, ACT 2, and 1200XL devices. The resulting power components are detailed in Tables 5 and 6, and the total device power is shown in Figures 2 and 3. The graphs provide a simple guideline for estimating power. The tables may be interpolated when your application has different resource utilizations or frequencies. Table 4 details the rules applied to ACT 3 devices.

Table 5 • Power Components for ACT 1 Family (Watts)

F (MHz)	Module Power	Input Power	Output Power	Clock Power	Total Power (watts)
A1010					
1	0.005	0.002	0.009	0.019	0.035
2	0.010	0.004	0.017	0.038	0.069
5	0.025	0.009	0.043	0.096	0.173
10	0.051	0.018	0.085	0.192	0.347
15	0.076	0.027	0.128	0.289	0.520
20	0.101	0.036	0.171	0.385	0.693
25	0.126	0.046	0.213	0.481	0.866
A1020					
1	0.009	0.002	0.010	0.035	0.057
2	0.019	0.004	0.021	0.071	0.114
5	0.047	0.011	0.052	0.176	0.286
10	0.094	0.022	0.103	0.353	0.572
15	0.141	0.033	0.155	0.529	0.858
20	0.188	0.044	0.207	0.705	1.144
25	0.235	0.055	0.258	0.882	1.430

Table 6 • Power Components for ACT 2 and 1200XL Families (Watts)

F (MHz)	Module Power	Input Power	Output Power	Clock Power	Total Power (watts)
A1280/1280XL					
1	0.011	0.013	0.021	0.027	0.072
2	0.023	0.025	0.042	0.054	0.144
5	0.057	0.063	0.105	0.136	0.360
10	0.113	0.125	0.210	0.272	0.720
20	0.227	0.250	0.419	0.544	1.440
30	0.340	0.375	0.629	0.815	2.159
40	0.453	0.500	0.839	1.087	2.879
A1240/1240XL					
1	0.006	0.009	0.016	0.015	0.046
2	0.013	0.019	0.031	0.030	0.092
5	0.032	0.046	0.078	0.074	0.231
10	0.064	0.093	0156	0.149	0.461
20	0.127	0.186	0.311	0.298	0.923
30	0.191	0.279	0.467	0.447	1.384
40	0.255	0.372	0.623	0.596	1.845
A1225/1225XL					
1	0.004	0.007	0.012	0.009	0.033
2	0.008	0.015	0.025	0.019	0.066
5	0.020	0.037	0.062	0.047	0.166
10	0.040	0.074	0.124	0.094	0.332
20	0.080	0.148	0.249	0.187	0.664
30	0.120	0.222	0.373	0.281	0.996
40	0.160	0.297	0.497	0.375	1.329
50	0.200	0.371	0.621	0.468	1.661

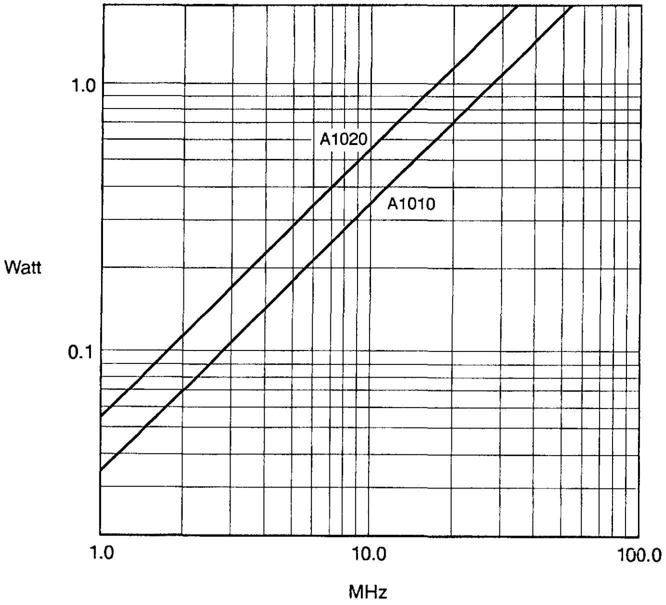


Figure 2* • *ACT 1 Family Power Estimates

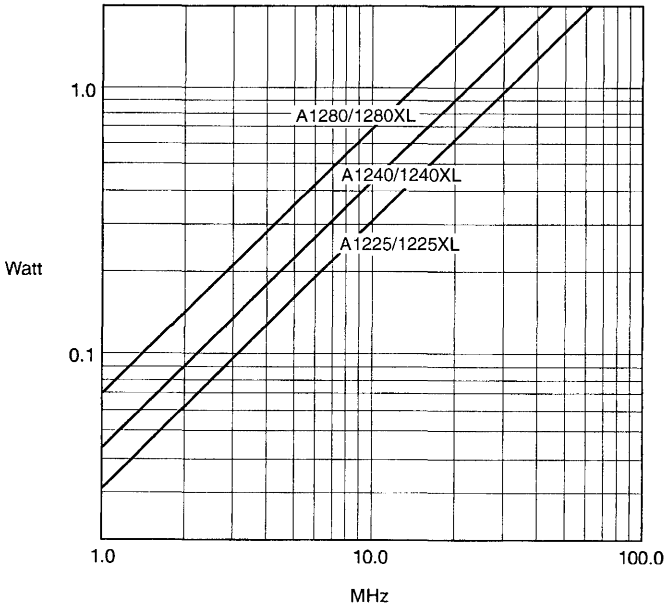


Figure 3* • *ACT 2 and 1200XL Families Power Estimates of Total Power (watts)

Application Note

Actel-Synopsys Gateway to Synthesis

As gate densities and design complexities increase, designers have turned to logic-synthesis tools to improve efficiency and productivity. When synthesis tools and methodology are used in conjunction with field programmable gate arrays (FPGAs), designers can dramatically reduce their design cycle times and produce faster time-to-market solutions.

However, selecting the appropriate FPGA architecture can be a daunting task. One may encounter difficulties meeting design requirements too late in the design cycle, and as a result, valuable time may be lost modifying the design at the gate level or performing manual placement and routing to meet design goals. Moreover, this postsynthesis processing invalidates timing predictions made by the synthesis tool. To allow the synthesis tool to provide the optimal results based on the design constraints and accurate timing models, a fine-grain synthesis-friendly FPGA architecture is highly recommended.

Architecture Overview

Actel FPGAs are based on an architecture inherently friendly to high-level synthesis. The logic module is relatively simple for efficient implementation of basic functions generated by synthesis tools, but complex enough for speed-critical functions to be synthesized into a single logic module. For example, the combinatorial module can implement up to a five-input NAND gate and a 4-to-1 multiplexer. The sequential module is designed to implement high-speed flip-flop functions, and it includes combinatorial logic that allows an additional level of logic to be combined without additional propagation delay. The register I/O modules are used for applications with fast input setups and clock-to-out timing requirements. Decoder modules are arranged around the periphery of the device and contain wide-decode circuits providing a fast decode function—for example, a seven-input NAND gate, similar to CPLDs and PALs. The dual-port SRAM module provides for a system's memory needs by supplying blocks of synchronous dual-port SRAM, which can be configured as 32 x 8 or 64 x 4 (Figure 1).

Actel devices also provide an abundance of routing resources by using antifuse interconnect elements. The routing resources available on every device ensure that logic modules can be efficiently interconnected automatically with predictable timing.

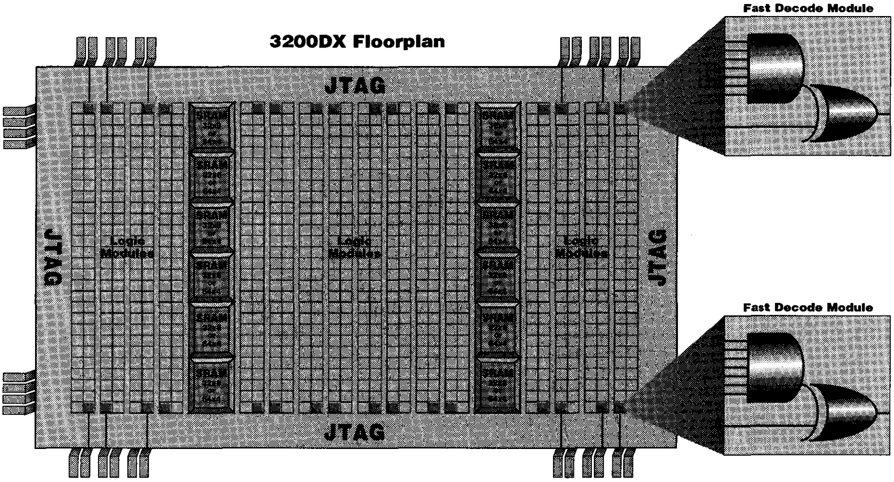


Figure 1 • *3200DX Floor Plan*

Design Methodology

Actel place-and-route utilities are tightly integrated with Synopsys tools to provide a robust high-level design (HLD) methodology for users. In the Actel-Synopsys paradigm, the design challenge has been approached with coding style, synthesis techniques, constraint-driven place and route, and verification by backannotation (Figure 2).

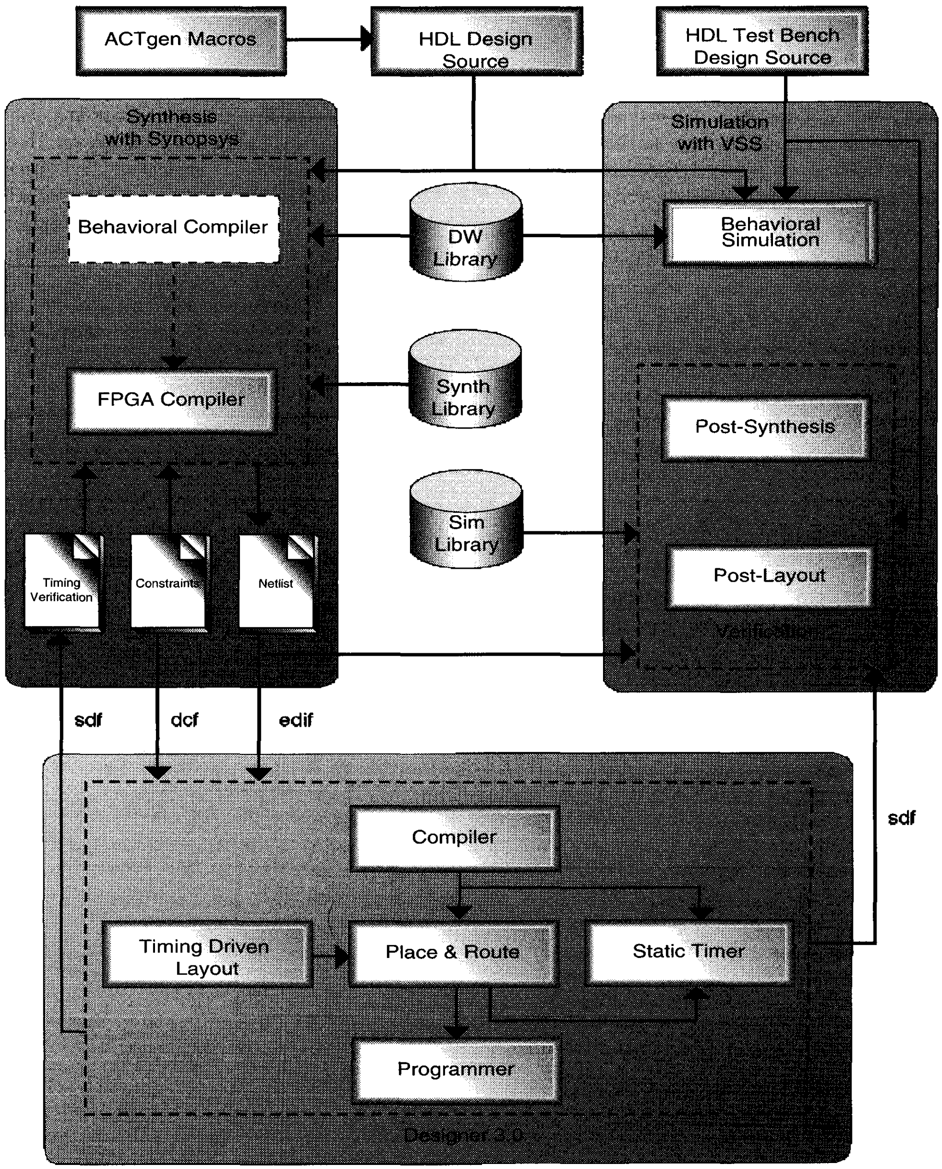
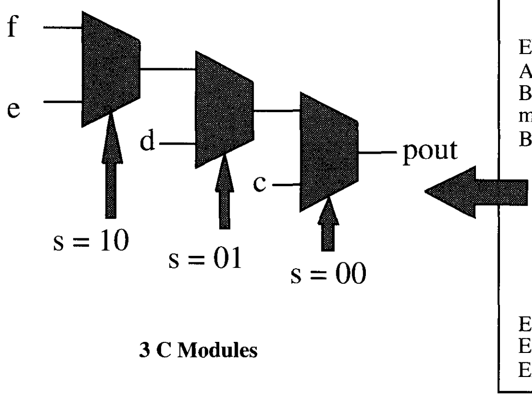


Figure 2 • The Actel-Synopsys Design Flow

Coding Style

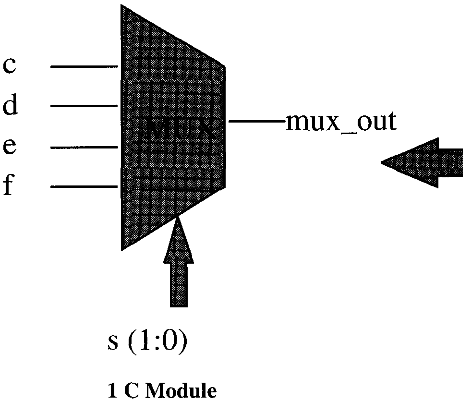
Coding styles present opportunities to explore many design implementations. However, to achieve optimal performance and area for the target technology, one must be familiar with the architecture and describe the behavior of the design for the target technology. Traditional behavioral (V)HDL, i.e., simulation friendly, is not always synthesis optimal. Hence, understanding synthesizable description is very important. Here are some coding styles preferred for the Synopsys-Actel synthesis methodology:

- Inferring multiplexers verses priority encoders (See Figure 3.)
- Complex Register synthesis (See Figure 4.)
- Keeping hardware in mind while writing HDL (See Figure 5.)
- Bitwise decoding on vectors when possible (See Figure 6.)
- DesignWare counter-directive and behavioral compiler RAMs (See Figure 7.)
- Resource sharing—area versus timing trade-off (See Figure 8.)
- Rolled and unrolled loops (See Figure 9.)
- Actel combinable modules (See Figure 10.)
- Logic duplication—area versus timing trade-off (See Figure 11.)
- Utilize CASE versus unutilized partial CASE statements (See Figure 12.)
- Timing-critical recoding—behavioral retiming (See Figure 13.)
- Timing-critical signals (See Figure 14.)



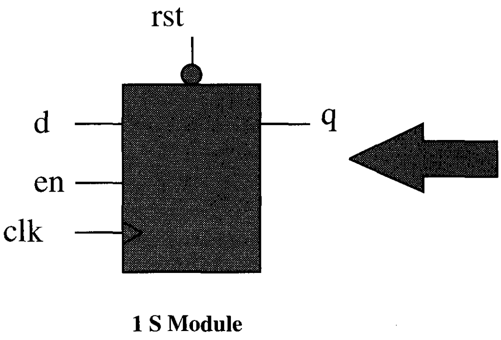
```
ENTITY my_if IS
 PORT (c, d, e, f : IN std_logic;
            s : IN std_logic_vector(1 downto 0);
            pout : OUT std_logic);
END my_if;
ARCHITECTURE my_arc OF my_if IS
BEGIN
myif_pro: PROCESS (s, c, d, e, f)
BEGIN
   if s = "00" then
     pout <= c;
    elsif s = "01" then
     pout <= d;
    elsif s = "10" then
    pout <= e;
     else  pout <= f;
END IF;
END PROCESS myif_pro;
END my_arc;
```

Figure 3 • Inferring Multiplexers versus Priority Encoders



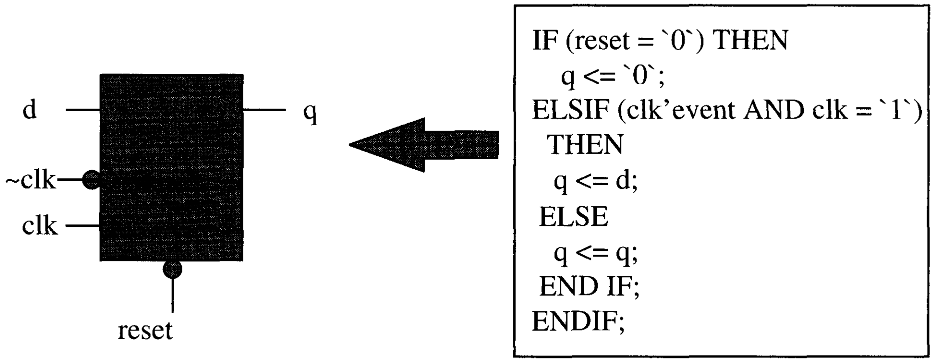
```
ENTITY mux IS
 PORT (c, d, e, f : IN std_logic;
          s : IN std_logic_vector(1 downto 0);
          muxout : OUT std_logic );
END mux;

ARCHITECTURE my_mux OF mux IS
BEGIN
mux1: PROCESS (s, c, d, e, f)
BEGIN
 CASE s IS
   WHEN "00" => muxout <= c;
   WHEN "01" => muxout <= d;
   WHEN "10" => muxout <= e;
   WHEN others => muxout <= f;
 END CASE
END PROCESS mux1;
END my_mux;
```

Figure 3* • *Inferring Multiplexers versus Priority Encoders (continued)



```
IF (rst = `0`) THEN
   q <= `0`;
ELSIF (clk'event AND clk=`1`)
   THEN
    IF (en = `1`) THEN
    q <= d;

END IF;
END IF;
```

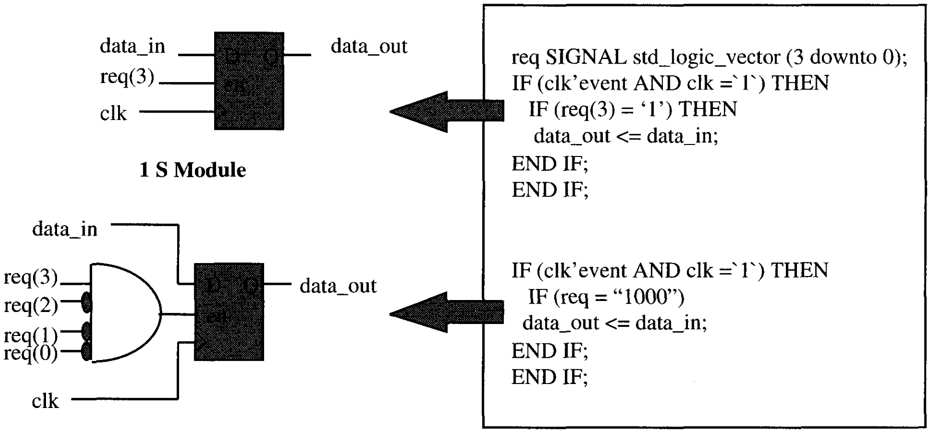
Figure 4* • *Complex Register Synthesis



```
IF (reset = `0`) THEN
   q <= `0`;
ELSIF (clk'event AND clk = `1`)
 THEN
  q <= d;
 ELSE
  q <= q;
 END IF;
ENDIF;
```

Such FF does not exist.

Figure 5 • *Keeping Hardware in Mind While Writing HDL*



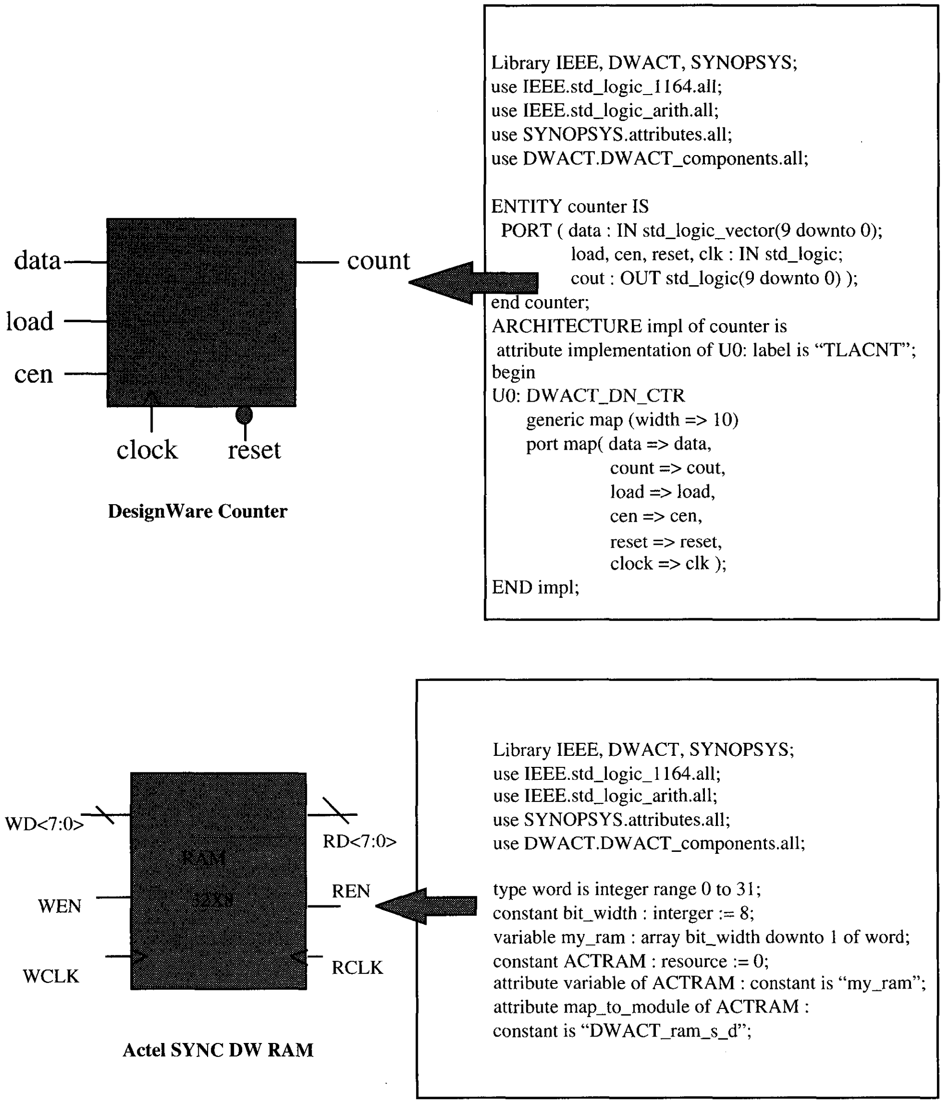
```
req SIGNAL std_logic_vector (3 downto 0);
IF (clk'event AND clk =`1`) THEN
  IF (req(3) = '1') THEN
   data_out <= data_in;
END IF;
END IF;
```

1 S Module

```
IF (clk'event AND clk =`1`) THEN
  IF (req = "1000")
 data_out <= data_in;
END IF;
END IF;
```

2 Modules

Figure 6 • *Bitwise Decoding on Vectors When Possible*



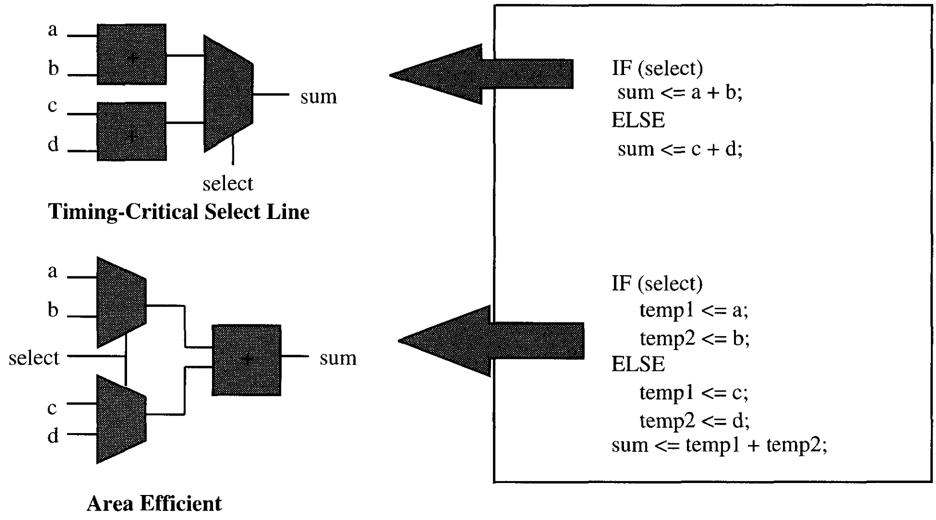
DesignWare Counter

```
Library IEEE, DWACT, SYNOPSYS;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use SYNOPSYS.attributes.all;
use DWACT.DWACT_components.all;

ENTITY counter IS
  PORT ( data : IN std_logic_vector(9 downto 0);
            load, cen, reset, clk : IN std_logic;
            cout : OUT std_logic(9 downto 0) );
end counter;
ARCHITECTURE impl of counter is
 attribute implementation of U0: label is “TLACNT”;
begin
U0: DWACT_DN_CTR
     generic map (width => 10)
     port map( data => data,
                  count => cout,
                  load => load,
                  cen => cen,
                  reset => reset,
                  clock => clk );
END impl;
```

Actel SYNC DW RAM

```
Library IEEE, DWACT, SYNOPSYS;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use SYNOPSYS.attributes.all;
use DWACT.DWACT_components.all;

type word is integer range 0 to 31;
constant bit_width : interger := 8;
variable my_ram : array bit_width downto 1 of word;
constant ACTRAM : resource := 0;
attribute variable of ACTRAM : constant is “my_ram”;
attribute map_to_module of ACTRAM :
constant is “DWACT_ram_s_d”;
```

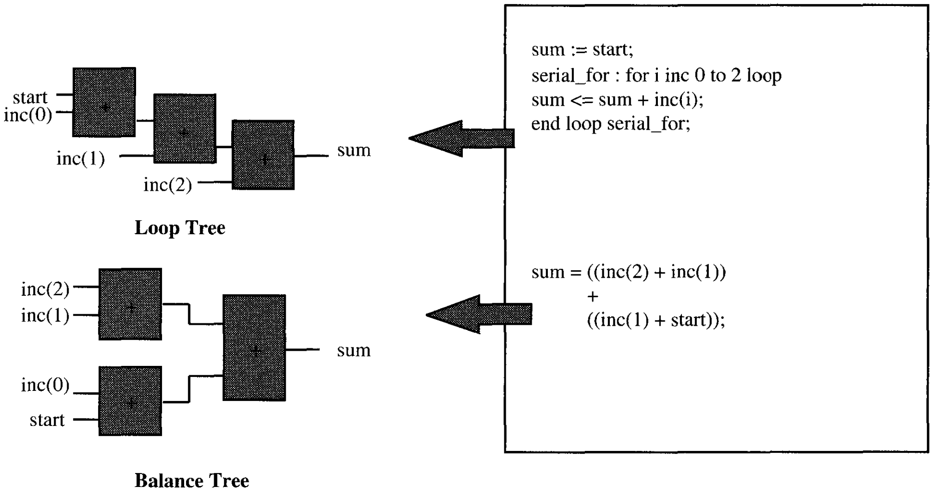
Figure 7 • *DesignWare Counter-Directive and Behavioral Compiler RAMs*



```
IF (select)
 sum <= a + b;
ELSE
 sum <= c + d;
```

```
IF (select)
    temp1 <= a;
    temp2 <= b;
ELSE
    temp1 <= c;
    temp2 <= d;
sum <= temp1 + temp2;
```

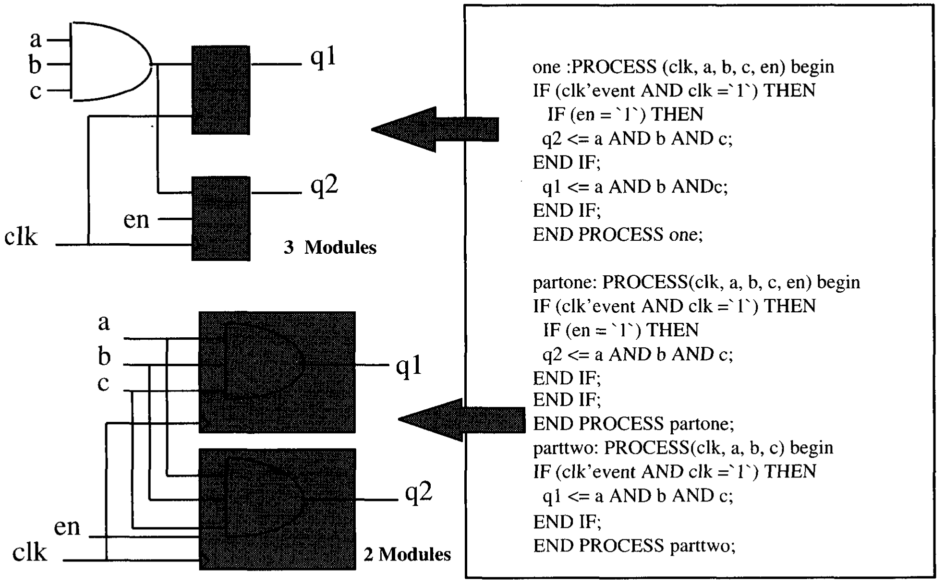
Figure 8 • *Resource Sharing (Area versus Timing Trade-Off)*



```
sum := start;
serial_for : for i inc 0 to 2 loop
sum <= sum + inc(i);
end loop serial_for;
```

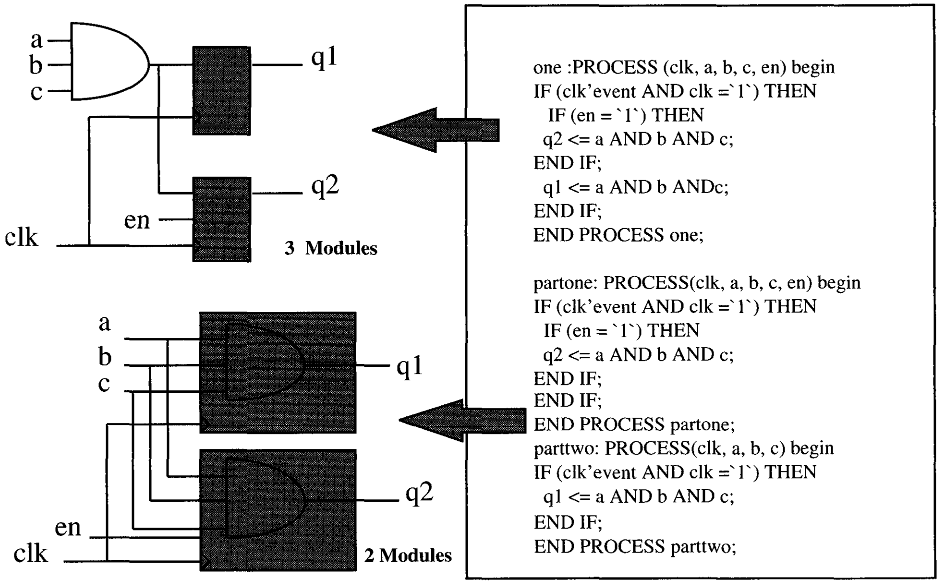
```
sum = ((inc(2) + inc(1))
        +
        ((inc(1) + start));
```

Figure 9 • *Rolled and Unrolled Loops*



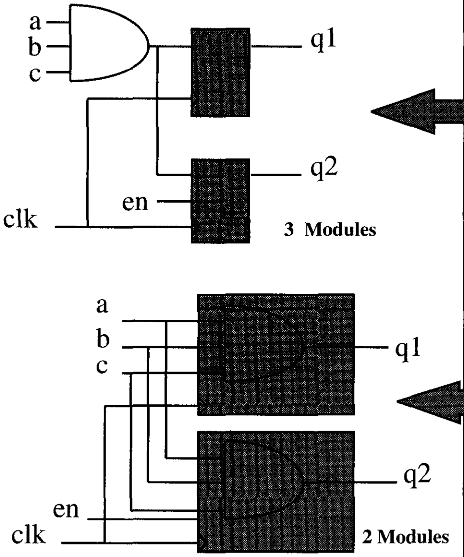
```
one :PROCESS (clk, a, b, c, en) begin
IF (clk'event AND clk =`1`) THEN
  IF (en = `1`) THEN
  q2 <= a AND b AND c;
END IF;
  ql <= a AND b ANDc;
END IF;
END PROCESS one;

partone: PROCESS(clk, a, b, c, en) begin
IF (clk'event AND clk =`1`) THEN
  IF (en = `1`) THEN
  q2 <= a AND b AND c;
END IF;
END IF;
END PROCESS partone;
parttwo: PROCESS(clk, a, b, c) begin
IF (clk'event AND clk =`1`) THEN
  ql <= a AND b AND c;
END IF;
END PROCESS parttwo;
```

Figure 10* • *Actel Combinable Modules



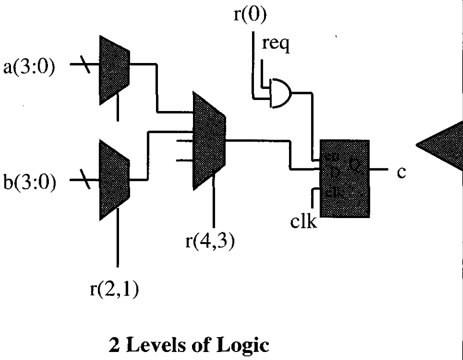
```
one :PROCESS (clk, a, b, c, en) begin
IF (clk'event AND clk =`1`) THEN
  IF (en = `1`) THEN
  q2 <= a AND b AND c;
END IF;
  ql <= a AND b ANDc;
END IF;
END PROCESS one;

partone: PROCESS(clk, a, b, c, en) begin
IF (clk'event AND clk =`1`) THEN
  IF (en = `1`) THEN
  q2 <= a AND b AND c;
END IF;
END IF;
END PROCESS partone;
parttwo: PROCESS(clk, a, b, c) begin
IF (clk'event AND clk =`1`) THEN
  ql <= a AND b AND c;
END IF;
END PROCESS parttwo;
```

Figure 11* • *Logic Duplication (Area versus Timing Trade-Off)



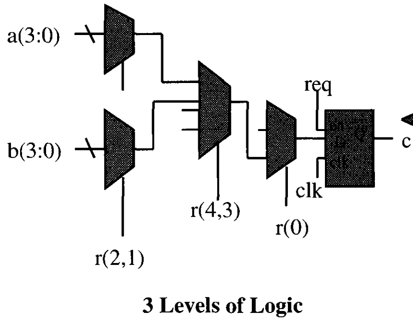
```
one :PROCESS (clk, a, b, c, en) begin
IF (clk'event AND clk =`1`) THEN
  IF (en = `1`) THEN
  q2 <= a AND b AND c;
END IF;
  q1 <= a AND b ANDc;
END IF;
END PROCESS one;

partone: PROCESS(clk, a, b, c, en) begin
IF (clk'event AND clk =`1`) THEN
  IF (en = `1`) THEN
  q2 <= a AND b AND c;
END IF;
END IF;
END PROCESS partone;
parttwo: PROCESS(clk, a, b, c) begin
IF (clk'event AND clk =`1`) THEN
  q1 <= a AND b AND c;
END IF;
END PROCESS parttwo;
```

Figure 11 • *Logic Duplication (Area versus Timing Trade-Off) (continued)*



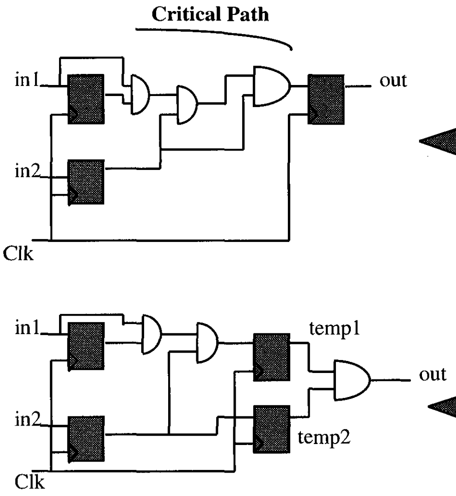
```
IF ( clk = '1' AND clk'EVENT ) THEN
IF (req = '1' AND r(0) = '1') THEN
    CASE r IS
      WHEN "0000" => c   <= a(0);
      WHEN "0001" => c   <= a(1);
      WHEN "0010" => c   <= a(2);
      WHEN "0011" => c   <= a(3);
      WHEN "0100" => c   <= b(0);
      WHEN "0101" => c   <= b(1);
      WHEN "0110" => c   <= b(2);
      WHEN "0111" => c   <= b(3);
      WHEN OTHERS  => NULL;
        END CASE;
    END IF;
```

Figure 12 • *Utilized CASE versus Unutilized Partial CASE Statements*



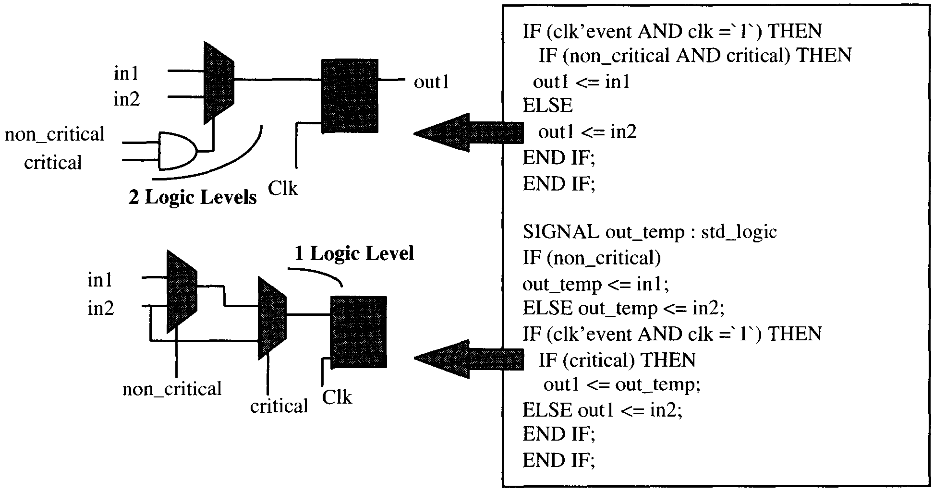
```
IF ( clk = '1' AND clk'EVENT ) THEN
IF (req = '1') THEN
     CASE r IS
      WHEN "00001" => c    <= a(0);
      WHEN "00011" => c    <= a(1);
      WHEN "00101" => c    <= a(2);
      WHEN "00111" => c    <= a(3);
      WHEN "01001" => c    <= b(0);
      WHEN "01011" => c    <= b(1);
      WHEN "01101" => c    <= b(2);
      WHEN "01111" => c    <= b(3);
      WHEN OTHERS  => NULL;
       END CASE;
     END IF;
```

Figure 12 • Utilized CASE versus Unutilized Partial CASE Statements (continued)



```
ENTITY style IS
  PORT ( in1, in2 : IN std_logic;
            Clk : IN std_logic;
            out : OUT std_logic);
end small_area;
Architecture small_area OF style IS
begin
one :PROCESS (clk, in1, in2) begin
IF (clk'event AND clk =`1`) THEN
in1_reg <= in1;
in2_reg <= in2;
END IF;
END PROCESS one;
two :PROCESS (clk, in1_reg, in2_reg, in1) begin
IF (clk'event AND clk =`1`) THEN
out <= ((in1 AND in1_reg) AND in2_reg) OR (in2_reg);
END IF;
END PROCESS two;

Architecture fast OF style IS
begin
one :PROCESS (clk, in1, in2) begin
IF (clk'event AND clk =`1`) THEN
in1_reg <= in1;
in2_reg <= in2;
END IF;
END PROCESS one;
two :PROCESS (clk, in1_reg, in2_reg, in1) begin
IF (clk'event AND clk =`1`) THEN
temp1 <= ((in1 AND in1_reg) AND in2_reg);
temp2 <= in2_reg;
END IF;
END PROCESS two;
out <= temp1 OR temp2;
```

Figure 13 • Timing-Critical Recoding (Behavioral Retiming)



```
IF (clk'event AND clk =`1`) THEN
  IF (non_critical AND critical) THEN
  out1 <= in1
ELSE
  out1 <= in2
END IF;
END IF;

SIGNAL out_temp : std_logic
IF (non_critical)
out_temp <= in1;
ELSE out_temp <= in2;
IF (clk'event AND clk =`1`) THEN
  IF (critical) THEN
   out1 <= out_temp;
ELSE out1 <= in2;
END IF;
END IF;
```

Figure 14 • *Timing-Critical Signals*

Synthesis Implementation

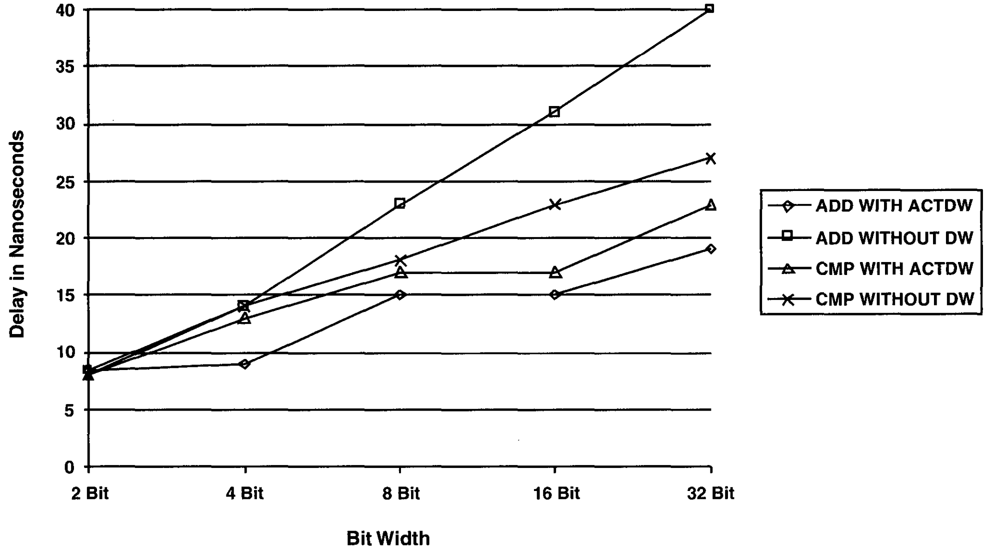
Although design validation verifies the architecture and behavioral functionality of the design, it does not guarantee the implementation and physical verification. The Synopsys synthesis optimization techniques are tools to address this issue. For example, tools such as Behavioral Compiler, HDL Adviser, Design Source, and FPGA Compiler focus on architecture-based synthesis implementations.

However, due to the complexity of these tools, they have to be evaluated and mastered with a solid design methodology. Actel has pursued a synthesis methodology that takes advantage of the power and flexibility of the Synopsys tools. Listed below are some of the major accomplishments of the synthesis FPGA methodology, a joint effort with Synopsys and Actel:

- Joint cooperation using the Synopsys Library Compiler to implement architecture-specific multiplexer mapping to utilize fully the combinatorial module. This is made possible by using the Synopsys Degeneration feature in Library Compiler, whereby library-based mapping is extended into specific Boolean functions for logic modules.
- Adapting Synopsys DesignWare standards to implement data-path and supermacro elements. This approach is completely transparent to the synthesis users and is one of the most efficient ways to take advantage of specific architecture attributes—by integrating the Actel macro generator (ACTgen) into the Synopsys DesignWare library. Actel has fully supported the existing parameterized HDL macros in DesignWare and will extend support to all future DesignWare capabilities, such as RAMs, FIFOs, and counters.

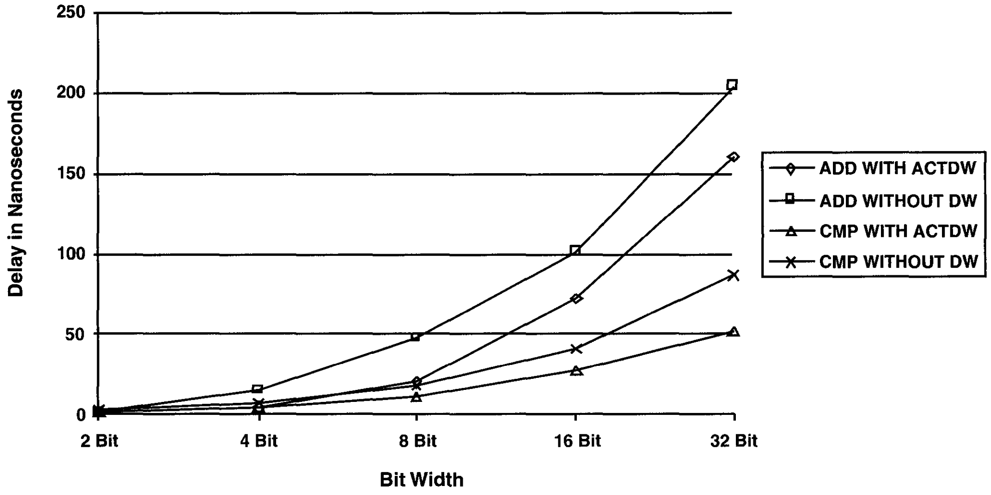
  Figures 15 and 16 show the performance of Actel DesignWare adders and comparators with Synopsys versus standard architecture-independent synthesis implementation.
- Continuing joint working relationship with Synopsys. The upcoming Synopsys releases will take advantage of complex mapping to a rich set of registers and additional multiplexer mapping by using the following hidden switches:

```
"hdl_mux_infer" : To prefer MUX mapping when-
ever possible
"actel_seq_opti" : Actel specific sequential
mapping
"actel_qbar_opti" ; To take advantage of Ac-
tel complex register varieties
```



Note: All timing numbers are in nanoseconds and all area numbers are in module counts.

Figure 15* • *Actel DW ADD/CMP Timing Performance



Note: All timing numbers are in nanoseconds and all area numbers are in module counts.

Figure 16* • *Actel DW ADD/CMP Area Performance

One of the most powerful and difficult-to-master parts of synthesis is the language-based script that gives the user complete freedom to influence the synthesis results. The Actel methodology makes this useful task as friendly and as simple as possible. An example is shown in Figure 17. The following are some examples of how one can use scripts to direct the synthesis tools for performance and improve ease of use:

- Actel-Synopsys setup scripts to install and compile libraries, to access the Actel DesignWare library, and control the netlist generation for back-end tools.

```
% setenv ACT_SYNOPDIR
/<synopsys_lib_loc>/lib/synop
% cd $ACT_SYNOPDIR/scripts
% update_all_dw (c-shell script)
dc_shell> include actsetup.scr
dc_shell> /* The rest of design script */
```

- DC shell script to assign pins automatically to a particular Actel device and package and to assign pin number 4 to a signal enable.

```
dc_shell> set_attribute find (net enable)
"ALSPIN" -type string "4"
```

- Passing Synopsys design constraints to Actel place-and-route tools.

```
dc_shell> /* Actel setup portion */
dc_shell> /* read in design and set con-
straints */
dc_shell> /* compile portion*/
dc_shell> current_design = your_design
dc_shell> write_script > your_design.sdc
dc_shell> write -f edif -h -o your_design.edn
dc_shell> sh synop2dcf  fam:<actel fam>
your_design
```

- Fast I/O timing using registered I/Os.

```
dc_shell> /* Actel setup portion */
dc_shell>  read -format verilog your_design.v
dc_shell> current_design = your_design
dc_shell> set_port_is_pad
dc_shell> insert_pads
dc_shell>   /* Constraint portion */
dc_shell> compile
dc_shell> ungroup -all flaten
dc_shell> include act3io.script */ Actel de-
veloped register IO mapping*/
dc_shell> set_local_link_library
act3io_linklib.db
dc_shell> link
dc_shell> */ report portion*/
dc_shell> write -format edif -hierarchy -out-
put your_design.edn
dc_shell> quit
```

- Actel device resource budgeting through DC scripts. As FPGA devices are targeted to solve specific problems, they are challenged with implementation of dense yet limited resources. The following example illustrates how to use a low-skew hardwired clock in the design.

```
dc_shell> remove_attribute act3/HCLKBUF
dont_use
dc_shell> set_port_is_pad
dc_shell> set_pad_type -clock clk -exact
HCLKBUF
dc_shell> insert_pads
dc_shell> set_dont_use act3/HCLKBUF
dc_shell> dont_touch HCLKBUF
```

- Actel Designer Series place-and-route software script language is an easy-to-learn macro language that lets you create powerful batch processes that link and automate separate design steps within Synopsys dc_shell. This is very valuable and productive, since it can launch the entire place-and-route process as a single command. An example of such a script that can be an initiated dc_shell is the following.

```
dc_shell> /* setup portion */
dc_shell> /* Constraints and compile por-
tion*/
dc_shell> write -f edif -h -o your_design.edn
dc_shell> sh "designer
script_file:your_design.dsf
script_mode:batch"
```

your_design.dsf

```
/* Session script file */
main()
{
new_design()
setup_design( "your_design", "ACT3" )
import_netlist( ~users/your_design.edn", "EDIF" ); set_device( "A1415A", "100 PQFP" )
compile()
import_aux_file( ~users/your_design.dcf", "DCF" ); set( "LAYOUT_MODE", "TIMING_DRIVEN" )
layout()
save_as( "~users/your_design.adb" ); }
```

Figure 17 • Session Script File

Physical Implementation

Even the best synthesis techniques that efficiently map to logic modules are only as good as the place-and-route software that interfaces with the synthesis tools—to perform a quality place and route and to ensure predictability and 100 percent automatic routing. The new Actel Designer Series suite of tools meets these requirements. The Designer Series software does a minimum amount of postprocessing on the output from synthesis tools so that the tools can provide accurate area and performance estimates. At the same time, the Designer Series tools can interface with Synopsys tools to channel information. The design constraints specified in Synopsys are automatically transferable to the Actel timing-driven place and route (DirectTime) without having to learn a new tool or introducing another data source. The DirectTime capability allows for greater timing control to achieve speed automatically during place and route.

Furthermore, the Designer Series tools can accept forward annotation or design attributes, such as pin assignments, hierarchy, or special macros, from a Synopsys-generated netlist. DirectTime Analyzer gives the user a graphical display of the timing results, showing what was achieved versus user specifications (Figure 18).

The Designer Series software has one of the easiest backannotation methods, in the form of a Standard Delay File (SDF).

Highlights of the Actel Designer Series place-and-route software and its interface with Synopsys synthesis tools are as follows:

- EDIF 200 standard reader from Synopsys
- Synopsys constraints scripts format reader for timing-driven place and route (SCR format)
- Netlist attribute reader for pin assignments, etc. (EDIF)
- Backannotation to Synopsys VSS for postlayout simulation (SDF1.0 and SDF2.1)
- 100 percent automatic place and route; no manual interaction
- Postcombiner to combine combinatorial and sequential cells if applicable
- 100 percent postsynthesis and postlayout netlist correlation
- Built-in graphical static timer
- Object-oriented database software with revision/flow control

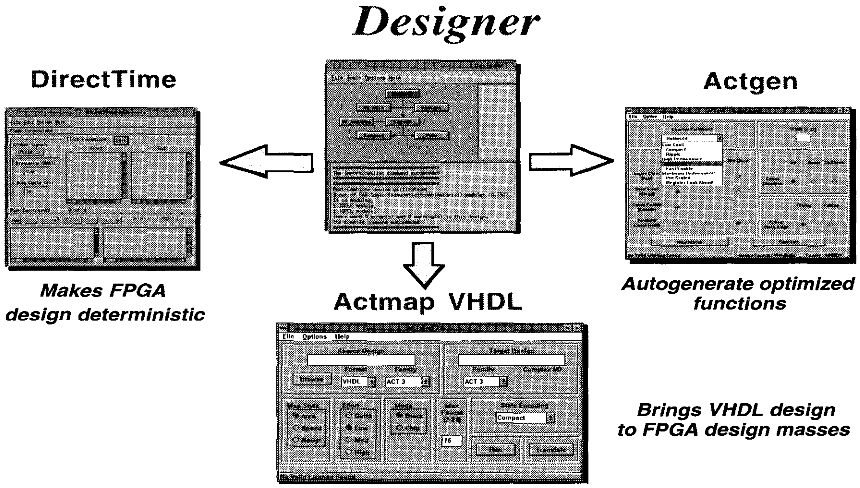

Designer
DirectTime
Actgen
Makes FPGA design deterministic
Actmap VHDL
Autogenerate optimized functions
Brings VHDL design to FPGA design masses
File Options Help
VHDL
ACT 3
Compact

Figure 18 • The Actel Designer Series Suite of Tools

Conclusion

Actel devices are inherently efficient for high-level designs because of their optimized logic modules and abundant routing resources. Experienced designers have found ways to squeeze even more efficiency out of these devices by understanding the influences of coding style and methodology on the resulting silicon implementation. The ability of high-level tools to accurately estimate capacity and performance when mapping to Actel devices also increases the usefulness of these tools. Using Actel devices and Synopsys high-level design tools, a designer has the ability to move up to higher-complexity designs.

Board Level Considerations for Actel FPGAs

Introduction

Simulating and debugging individual components is the first step towards verifying a system design. In some cases, the devices no longer behave as expected after they are integrated. Many factors, such as power, airflow, and transmission lines can introduce undesirable results in a system and ultimately impair system performance. This application note will explain how many of these factors should be treated when integrating Actel field programmable gate arrays (FPGAs) in board-level designs.

Power Up

Actel FPGAs are nonvolatile and therefore require no external configuration circuitry on power up. However, at power up it does take a finite amount of time for the device to become stable and operate normally. For a V_{CC} slew rate of ~30 ns/V, it takes approximately 250 µs for the device to become fully operational. Power up time varies with temperature, where cold is worst case. At power up, the state of all flip-flops is undefined. Refer to *A Power On Reset Circuit for Actel Devices* for more information.

Operating Environment

Actel FPGAs must not be operated outside of the recommended operating conditions as described in the Absolute Maximum Ratings section of their data sheets. Exposure to maximum rated conditions for prolonged periods may result in irreparable damage to the device.

As can be seen in the timing derating section of the data sheets, variations in voltage, temperature, and process' will affect device performance. You must take care to design systems so that the effects of these variables, from best to worst case, will not cause timing problems during interchip communications.

Thermal Considerations

An often overlooked test case examines ambient temperature effects on FPGA performance. Actel Designer Series software reports timing as derated by thermal junction temperature. This is the temperature found at the junction of the die and the package casing. Additional derating must be applied for ambient to junction temperature effects (θja). The value for θja will vary from package to package. The units for θja are °C/W. This implies that the power dissipation must first be calculated to determine the value for θja. (Please refer to the Power Dissipation section of the Actel data sheets.) Once θja is calculated, this temperature should be added at the junction to case temperature (θjc). This is the temperature value to be used for the temperature derating portion of the timing analysis. Because θja is a function of ambient temperature, the use of a fan can decrease the θja value. This can be seen by observing the difference between the θja value in still air and at 300 ft./min., as shown in the Package Thermal Characteristic section of the data sheets.

Ground Bounce and Switching Characteristics

Actel defines simultaneously switching outputs (SSO) as any outputs that transition in phase within a 10 ns window. The resultant ground bounce produced is a function of the number of outputs switching simultaneously and the capacitive loading of the outputs.

Table 1 shows example SSO limits for Actel FPGAs at 50pF for ACT 1, ACT 2, and 1200XL families. Measuring techinques and SS0 limits for additional families and capacitive loads are defined in "Simultaneously Switching Output Limits for Actel FPGAs" on page 4-125.

Exceeding the recommended limits results in a larger ground bounce. However, the width of the ground bounce pulse prevents it from being recognized by most discrete digital receivers. Refer to the specifications of the external receivers for verification.[1]

Table 1 • *Example SSO Limits for Actel FPGAs*

Device	Package	Maximum Recommended SSOs (50 pF load)
A1010A/A1020A	44 PLCC	16
A1010A/A1020A	68 PLCC	24
A1020A	84 PLCC	32
A1010A/A1020A	84 PGA	32
A1010A/A1020A	100 PQFP	32
A1280/A1280XL	PG176, PQ160	64
A1240/A1240XL	PG132, PQ144	48
A1240/A1240XL/ A1225/A1225XL	84 PLCC	32
A1225/A1225XL	100 PGA, PQFP	32

Should you have the need to switch more signals than recommended, the SSOs should be distributed evenly around the device. This will reduce the amount of mutual inductance produced between adjacent I/Os and thus reduce the ground bounce.

Buffers may also be inserted in the path before the output buffer. This will reduce the possibility of adjacent signals switching within 10 ns of each other. The Actel Designer Series software should be used to verify the delays.

Power and Ground Pins

Actel FPGAs are designed with ample power and ground pins on the device resulting in minimal ground bounce characteristics during I/O switching. A ground pin is provided for approximately every eight I/Os (please refer to package drawings found in this data book for further detail). Unused I/Os cannot be programmed to act as ground pins.

Table 2 describes the function of each of the power and ground pins found on the devices.

Table 2 • *Power and Ground Pin Functions*

V_{CC}	Device operating power.
V_{PP}	Device programming voltage. Should be tied to V_{CC} during normal operation.
V_{SV}	Super voltage supply. $V_{SV} = V_{PP} +3$. V_{SV} provides a precise lower limit on the voltage that is applied to a fuse during programming. During normal operation $V_{SV} = V_{CC}$.
V_{KS}	V_{KS} provides voltage supply for keepers (keepers maintain floating tracks at $V_{PP}/2$ during programming and prevent them from leaking down to GND). During programming, $V_{KS} = V_{PP}/2$. During normal operation, V_{KS} = GND.
GND	Supplies GND to the array (all channels).

Slew Rate

ACT 1, ACT 2, 1200XL, and 3200DX device outputs operate at high slew rate only. Figures 1, 2, and 3 illustrate the slew rate characteristics for high slew rate outputs, both loaded and unloaded, for an A1280 PG176 device.

Device I/O

The I/Os on each Actel FPGA are nonrestrictive. Any pin shown as an I/O in the pin description list can be assigned to be either input, output, bidirectional, or tri-state. ACT 2, 1200XL, and 3200DX FPGAs add latch capabilities to each of these options. Actel FPGAs do not incorporate any type of internal pullup on the I/Os. Inputs must not be left floating, since this may cause irreparable damage to the device. Any unused I/Os will be configured as active low outputs by the Designer Series System. The sink and source capabilities of individual outputs for ACT 1, ACT 2, 1200XL, and 3200DX FPGAs have been extrapolated from V-I curves and are shown in Table 3. The V-I curves for ACT 1 and ACT 2 devices are shown in Figures 4 through 9.

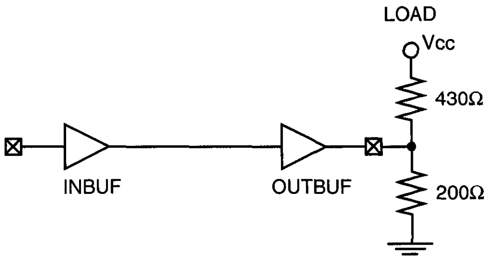


Figure 1 • *A1280 Test Circuit*

Table 3 • *Worst-Case Sink and Source Current for Actel FPGAs*

	ACT 1	1200XL and 3200DX
TTL	8 mA	10 mA
CMOS	4 mA	6 mA

I/Os may be tied together externally to increase drive capability. The outputs must switch within 4 ns of each other, thus they should be placed as close to each other as possible. (Please refer to device die bonding diagrams for pad locations.) The switching times can be verified with the Actel Designer Series software.

Decoupling Capacitors

Actel recommends the use of decoupling capacitors with all FPGA devices. We suggest a minimum of one capacitor per side, located next to power and ground. A 0.1 µF monolithic ceramic type is sufficient.

References:

1. David Shear, "EDN's Advanced CMOS Logic Ground-Bounce Tests," *EDN*, March 2, 1989, p. 88.

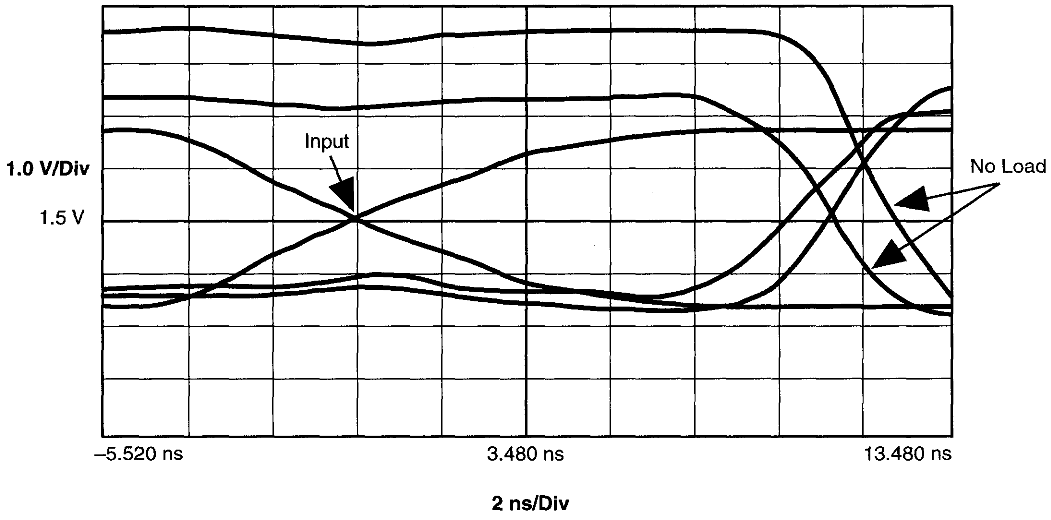
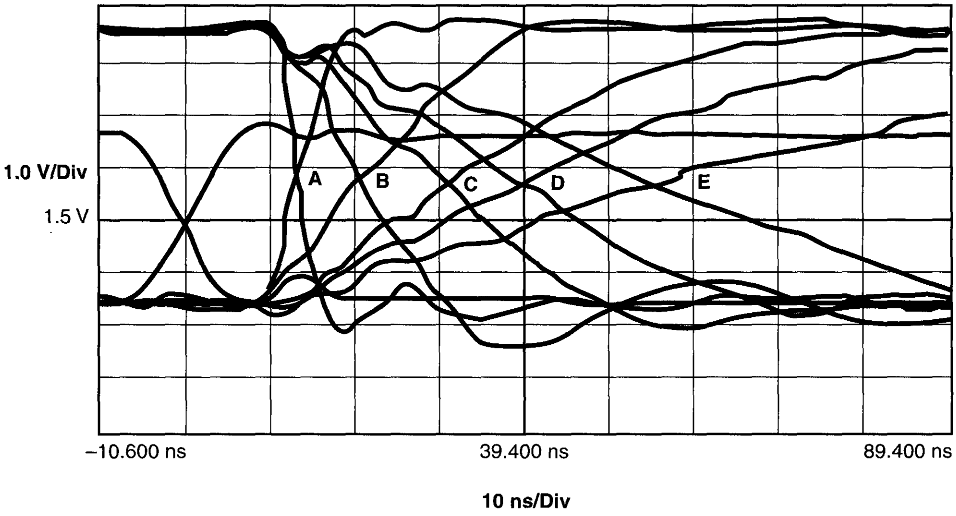
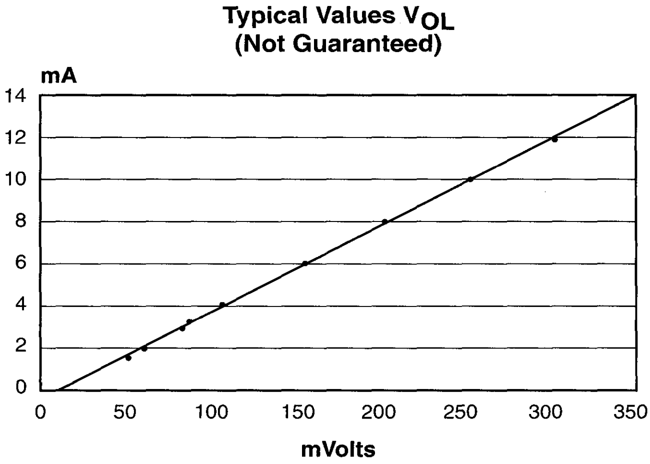


Figure 2 • A1280 No Load Versus Load—Fast Slew



A = "No Capacitive Load" (~25 pF jig + 9 pF scope)
B = 82 pF
C = 200 pF
D = 300 pF
E = 560 pF

Figure 3* • *A1280 Capacitive Loading Versus Slew



V_{CC} = 4.75 V, 25°C, Typical Process

Figure 4* • *A1020A Typical Sink Current

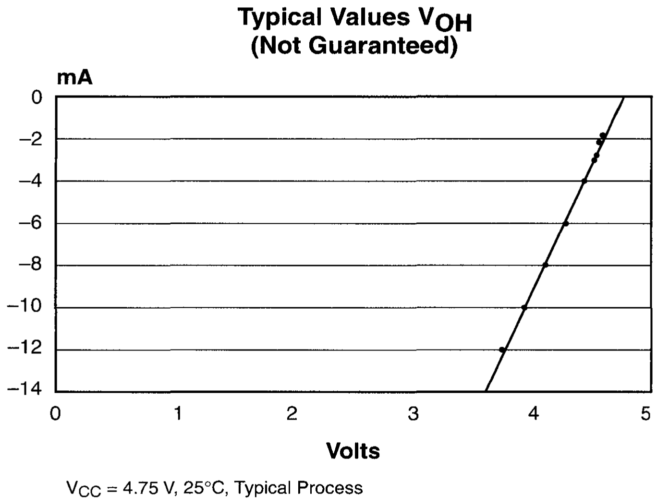


Figure 5 • *A1020A Typical Source Current*

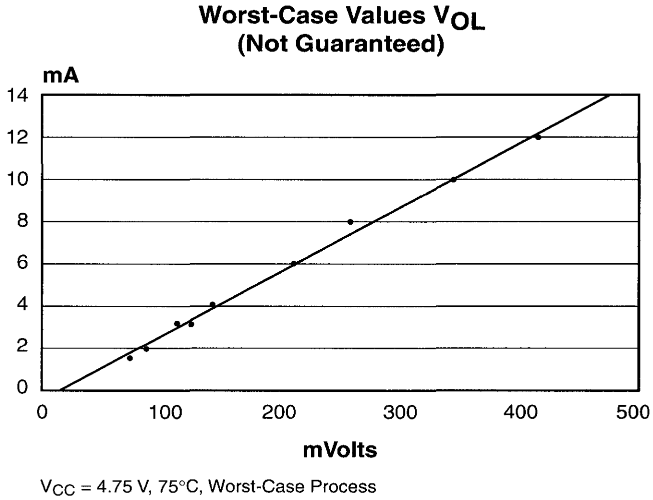


Figure 6 • *A1020A Worst-Case Sink Current*

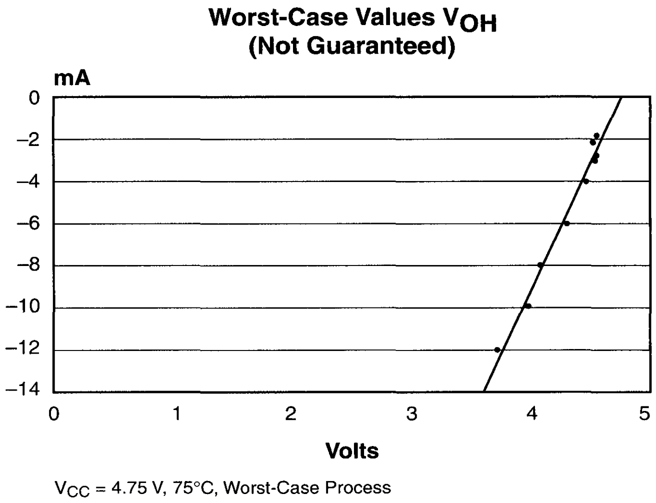


Figure 7 • *A1020A Worst-Case Source Current*

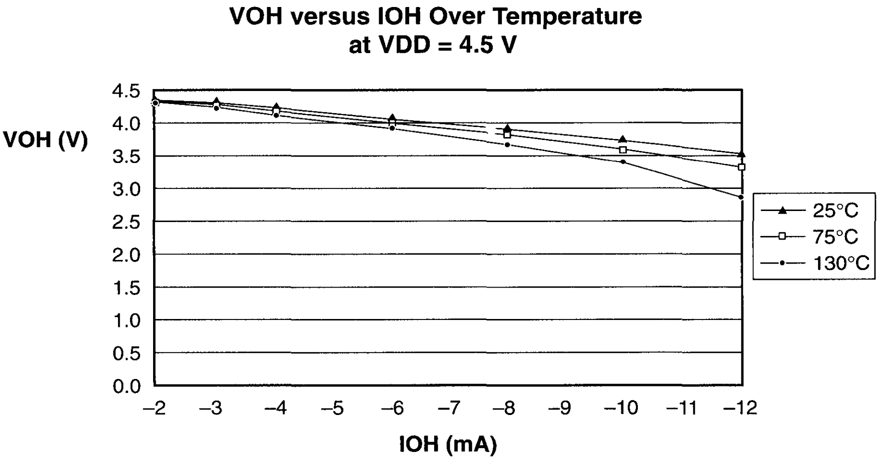


Figure 8 • *A1280 Typical Source Current*

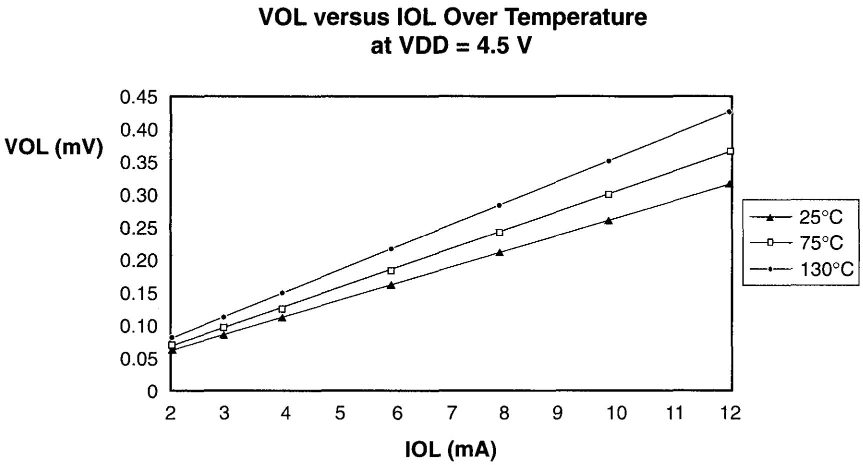


Figure 9 • A1280 Typical Sink Current

Using the Actionprobe For System-Level Debug

Problems found late in the design cycle can be time-consuming to debug and correct and can significantly impact product schedules. This problem is exacerbated by the ever-increasing complexity of designs and densities of programmable devices. The purpose of this application note is to describe Actel's Actionprobe feature and how it can be used to minimize schedule impacts resulting from functional or timing problems discovered late in the design process.

The Design Process

Much of the design process is spent in functional verification (simulation) and timing verification. Functional verification typically consists of modeling external devices that provide stimulus to the design and ensuring that the design responds properly. Timing verification can be performed either by backannotation or static timing tools like Actel's DirectTime Analyzer.

Simulation is a typical GIGO (garbage in, garbage out) function, because accurate verification of the design is completely dependent on the accuracy and completeness of input stimulus and expected response. Lack of completeness can be an issue if certain cases are not tested. Test cases are sometimes overlooked or intentionally left out because of complexity or to limit tests to corner conditions.

Timing verification presents similar challenges because correct function is dependent on other devices meeting timing expectations included in the test cases. Timing verification is typically limited to corner conditions to ensure manageability of the design time, which by definition ignores potential timing problems at non-corner conditions.

Where the Rubber Meets the Road

No matter how much effort is put into functional and timing verification, there is no substitute for powering up the system and seeing it operational. So, here you are late in the design cycle. You get your prototype boards built, you apply power to the system, and nothing happens (at least no smoke). You pull out your logic analyzers and emulators and start trying to debug the problem. You first verify connectivity and make sure clocks are wiggling. You keep working until you discover that the outputs of your FPGA are not functioning.

Now you have several choices:

- Spend a few days verifying that the inputs to the FPGA correspond to your simulations. If not, then go modify simulation inputs, and see if you can replicate and then fix the problem.
- Guess what the problem is, based on the observable failure mechanism external to the device, and then attempt to correct it.
- Place and route the design again, and hope the problem goes away, or perhaps bring out some test signals if you have unused pins.
- Or, if Actel is your FPGA solution, you can use the Actionprobe diagnostics and have your problem exactly identified and corrected in a few hours.

The Actionprobe Diagnostics Tool

The Actionprobe function in Actel devices provides the ability to probe any internal node during normal device operation. In essence, the Actionprobe extends the power of the logic analyzer, which normally stops at device boundaries. The internal node is selected by scanning a serial address from your computer into the device on the pins SDI and DCLK. Once the addressing is complete, a path from the internal node to either the PRA or PRB I/O pin is created. A logic analyzer connected to these pins can then be used to observe the internal operation of the device eliminating the guess work of system debug.

To select an internal node to observe, simply select the net name or pin name corresponding to the node. The tools will then calculate the required serial stream and communicate this information to the device. Up to two nodes can be addressed simultaneously and moved around as quickly as alternative nodes can be selected. So, starting from a failing output, you can quickly trace back through the device to determine the origin of the failure. Once the problem is found (whether functional or timing), the design can be quickly modified, allowing you to rapidly achieve a working system.

Examples

Actionprobe can be used when the simulation stimulus is incorrect or incomplete. Incorrect stimulus (accidental or intentional) can be quickly discovered by using Actionprobe to locate the earliest failing section of logic. The system stimulus for this small section can then be compared with the simulation stimulus to determine differences and corrective actions.

Large counters present another example where simulation may be inadequate. For instance, a pulse is required every second and must be generated from a 20 MHz clock. This means you would need a 25-bit counter and 20 million simulation cycles to verify the counter, which could take a while. The typical approach is to set the counter near the decode/reset point, run the simulation from there, and assume that the counter will work otherwise. If for some reason there is a failure in the counter circuitry at some unsimulated point, Actionprobe can be used to locate the problem quickly by providing the ability to observe all counter states, next states, and decode values.

Security

If security is a concern in the final product, additional fuses can be programmed to disable the Actionprobe function.

Requirements

The hardware required to support the Actionprobe diagnostic tool includes an Activator 2 or Activator 2S programmer and the Actionprobe pod, which connects the device under test to the Activator. In addition, the MODE pin on the device must be connected to ground through a 10K resistor. The software to drive the Actionprobe is built into the APS2 and APSW software, included with all Designer Series Development Systems.

Limitations

The Actionprobe diagnostic capability has a maximum observable frequency that is family dependent, as shown in Table 1. In addition, the signals PRA, PRB, SDI, and DCLK should be available (not used as I/Os) to fully take advantage of the Actionprobe capability.

Table 1 • Maximum Observable Frequencies

Device Series	Fmax
ACT 1 (1.0 µm)	25 MHz
Integrator Series	45 MHz
Accelerator Series	80 MHz

Conclusion

Increasing complexity of designs and decreasing cycle times create a significant challenge to the designer to produce a working system. Strategies to meet this challenge include simulation and timing verification tools. When these tools fall short because of a lack of accuracy or completeness, good system debug tools are required, including emulators and logic analyzers. Finally, to bare the internal operation of a device to scrutiny, Actel has developed the Actionprobe function to remove the guesswork from device debug. The result is improved productivity and products getting to market faster.

Simultaneously Switching Output Limits for Actel FPGAs

Introduction

For high performance field programmable gate arrays (FPGAs) with many I/Os, the allowable number of Simultaneously Switching Outputs (SSOs) for each device is an important issue for system designers. The limits for SSOs depend on factors such as package type and die size. Use the following guidelines to help ensure reliable system designs.

System Noise and Transients

Noise generated by off-chip drivers is a major concern in FPGA design for high-performance systems. Noise is closely related to interconnections and increases as a function of fast rise times, large total chip currents and die dimensions, and small spacing between components on-chip and onboard. As clock frequencies and die dimensions increase, signal wavelengths become comparable to wire lengths, thereby making better antennas. Reduction in the spacing between circuits leads to an increase in the capacitive, inductive, and resistive couplings. The voltage (IR) drop across the power lines becomes more significant as the current densities increase and the feature size decreases. With larger amounts of current switching, the inductive noise associated with the power lines also increases. These current transients may generate large potential drops because of the inductance of the power distribution network and is referred to as simultaneous switching (ΔI) noise.

When several circuits switch simultaneously, the current supplied by the power lines can change at a very fast rate and the inductive voltage drop along the line can cause the power supply level to fall. This voltage drop is proportional to the switching speed, the number of drivers switching simultaneously, and the effective inductance of the power lines. This effect can be summarized by Faraday's Law, which states that any change in the magnetic flux will be confronted by an opposition, a self induced EMF, determined by the rate of change in the total magnetic flux, represented by this equation:

$$\mathrm{EMF} = d\Phi/dt = LdI/dt$$

This reduction in the supply level diminishes the current drive of a circuit, increases the delay, and may lead to the spurious switching of the receiver.

SSO Recommendations

Actel defines SSOs as any outputs that transition in phase within a 10-ns window. The output current of these drivers is shown in device data sheets. The amplitude and duration of the ground bounce is a function of the number of outputs switching simultaneously and the capacitive loading of the outputs.

Table 1 shows the recommended SSO limits for Actel FPGAs. These may vary because the amount of ground bounce that an application can tolerate is difficult to determine. Worst-case conditions are simulated by placing the I/Os adjacent to each other while driving 20 pF, 35 pF, and 50 pF loads. The observed ground bounce is less than 1.5 V, with a pulse width of less than 2.0 ns. This is within the acceptable limits of the dynamic threshold of 74F, 74LS, and ACT (Advanced CMOS Technology) families. Results from the EDN Special Report on Ground Bounce Tests by David Shear[1] show a plot of the ground bounce pulse amplitude versus pulse width duration (ns) for different logic families. This article shows that as the input pulse width gets shorter, the voltage must be higher to affect the output of the device. Exceeding the recommended limits may result in larger ground bounce. The output drivers should be placed separately if more outputs must be switched simultaneously. This arrangement reduces the mutual inductance produced between adjacent I/Os resulting in a lower ground bounce. If necessary, buffers may also be inserted in the path before the output buffer. This will reduce the probability of adjacent drivers switching within 10 ns of each other.

References:

1. David Shear, "EDN Special Report on Ground Bounce Tests," *EDN*, April 15, 1993, p. 120–134.

Table 1 • Recommended SSO Limits for Actel FPGAs

		Maximum Recommended SSOs for Loads		
Device	**Package**	**20 pf**	**35 pf**	**50 pf**
A1010A/A1020A	44 PLCC	40	22	16
A1010A/1020A	68 PLCC	60	34	24
A1020A	84 PLCC	80	45	32
A1010A/1020A	84 PGA	80	45	32
A1010A/A1020A	100 PQFP	80	45	32
A1280/A1280XL	PG 176, PQ 160	160	90	64
A1240/A1240XL	PG 132, PQ 144	120	68	48
A1240/A1225/A1225XL	84 PLCC	80	45	32
A1225/A1225XL	100 PGA, PQFP	80	45	32
A1400 Family[1]	84 PLCC	64	48	42
A1400 Family[1]	All other packages	128	64	58

Note:

1. *The recommended SSO value for the A1400 family can be doubled for outputs using low slew drivers.*

Application Note

A Power-On Reset (POR) Circuit for Actel Devices

The state of a system at start-up is an important consideration in designing a circuit. It is usually desirable to provide an input signal at start-up to reset synchronous circuitry. Otherwise, the system may initially operate in an unpredictable fashion because flip-flops are not designed to power-on in any particular state. Figure 1 shows a typical power-on reset (POR) circuit from which the series resistor, R1, is omitted for TTL circuits.

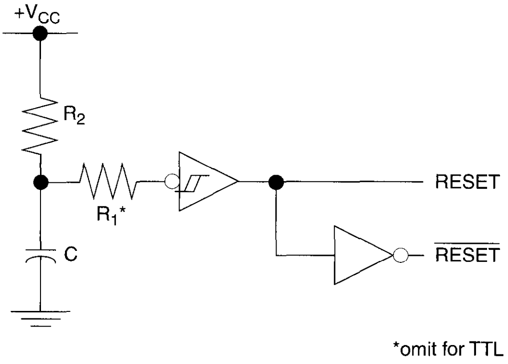


Figure 1 • *Power-On Reset Circuit*

This resistor is necessary with CMOS implementations to prevent damage to the device when power is removed from the circuit. Otherwise, the capacitor would try to power the system via the CMOS input gate protection circuit. A Schmitt trigger (40106, 74LS14) may be advantageous in making the RESET signal switch off cleanly. The hysteresis symbol shown in Figure 1 indicates an inverter with a Schmitt trigger input such as the CMOS 40106 hex inverter. The following sections describe the power-up conditions of an Actel device and a recommended POR circuit.

Behavior of ACT™ 1 FPGA Inputs

During power-on, the +5 V logic supply rail of a system typically rises from 0 to +5 V in 50 ms or less. Because regulator outputs are usually current limited during this transition, the rise time is more or less linear, with a slope in the range of 0.1 V/ms to 5 V/ms. Each Actel FPGA has a universal pad driver design that may be configured as an input, output, three-state output, or bidirectional input/output. This configuration of the pad driver is accomplished by programming antifuses in the pad driver circuitry.

As the +5 V logic supply rail passes through the region from approximately +2.2 V through +2.5 V, pad drivers that have been programmed as inputs may behave temporarily as outputs that are in the logical '1' state. Thus, these input pins will temporarily source current (approximately 8 to 10 mA, if not otherwise limited) into whatever driver is connected to them. They will be sourcing this current from the +5 V logic rail, which at this time is at +2.2 to +2.5 V. This duration is a function of the power rail rise time. For +5 V rails that come up quickly, at 5 V/ms, the duration of the behavior will be approximately 60 µs. For supply rails that rise slowly, at 0.1 V/ms, the duration of the current sourcing behavior will be 3 ms. In the former case, the Actel input can deliver as much as 0.6 µC to the circuit that drives it; in the latter case, the charge is as much as 30 µC. For many driver circuits, this amount of charge is insignificant; however, for others it may be unacceptable.

Inserting a series resistance of sufficient size into the Actel input line can limit the effect of this behavior. In the case of the POR circuit in Figure 2, the series resistance must be chosen to keep $\Delta V \leq 1V$. This guarantees that the POR remains at a logic '0' following the irregularity when the logic supply rail is at approximately 2.5 V, where $\Delta V/\Delta t = i/C$ is the voltage rise time of the capacitor. The capacitor is charged through a resistor to the 5 V logic supply rail, and the diode across the resistor is used to discharge the capacitor at power-off. For a power rail rise time of 0.1 V/ms, the duration of this behavior will be approximately 3 ms. This means that for a POR capacitance of 0.1 µF, the current out of the Actel device input must be limited to

$$i = C\Delta v / \Delta t = (0.1\ \mu F * 1\ V) / 3\ ms = 33\ \mu A$$

which can be achieved using a resistance of 2.24 V/33 mA = 68 kΩ.

Furthermore, for drivers that cannot accept a source of current at their outputs or for a multiple-source data bus, it is strongly recommended that the bus drivers be three stated during POR.

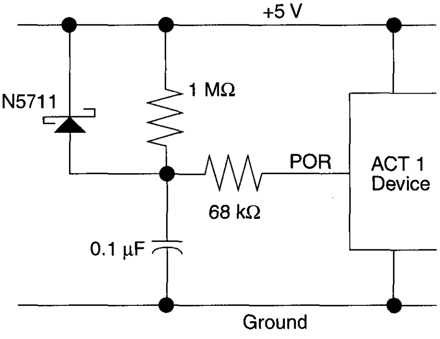


Figure 2 • Power-On Reset (POR) Circuit with Current-Limiting Resistor

Device Behavior during Power-Up

Actel devices have been characterized with two V_{CC} rise time conditions—fast power-up of approximately 0.5 V/µs and slow power-up of approximately 0.2 V/ms. During power-up, the following I/O conditions were characterized: input characteristics, tristate outputs, output driving low, output driving high, and output toggling. Normal input behavior is defined as the input being high impedance (<10 µA leakage). Normal output behavior means the output will either be tristate or be in a predictable state as defined by the logic of the user's design. The description that follows summarizes what happens to the Actel devices during power-up.

ACT 1 Devices

Inputs and outputs will behave normally within 100 µs after V_{CC} reaches 4.75 V with fast power-up and 3.5 V with slow power-up. With fast power-up, inputs remain in a high impedance state. With slow power-up, prior to reaching 3.5 V, inputs will momentarily behave as outputs driving high or low. This happens in about a 0.3 V window between 1 V and 2.5 V. The length of time this high or low level is exerted depends on the rise time of V_{CC}. At 0.2 V/ms, the input went high for about 1 ms and twice went low for about 150 µs each time. Tristated outputs will behave about the same as inputs. Outputs driving high or low will sometimes go to the opposite state (or tristate). Toggling outputs may not perform as expected until V_{CC} reaches 3.5 V. When V_{CC} is between 1.5 V and 3.5 V, I_{CC} will increase to 10–60 mA and then return to normal. The duration is dependant on the V_{CC} rise time, slow power-up being the worst case.

The inputs and outputs of the 3.3 V devices will behave normally within 200 µs after V_{CC} reaches 3 V for fast power-up and within 100 µs after V_{CC} reaches 2 V for slow power-up.

ACT 2/1200XL/3200DX Devices

Normal operation will occur within 100 µs after V_{CC} reaches 3.5 V for both fast and slow power-up. With slow power-up, inputs and tristated outputs will behave as outputs driving high for approximately 200 µs during a 0.3 V window where V_{CC} is between 1.5 V and 3.0 V. Inputs and tristated outputs stay at high impedance with fast power-up. I_{CC} will rise to 10–50 mA when V_{CC} is between 2 V and 4.5 V and then return to normal.

The 1200XL and 3200DX devices are expected to reach normal operation within 100 µs after V_{CC} reaches 3 V.

ACT 3 Devices

Normal operation for inputs and outputs will occur within 100 µs after V_{CC} reaches 2.75 V. Before reaching the point of normal operation, all inputs and outputs are in a high impedance state (tristate) regardless of V_{CC} rise time. I_{CC} rises to 10–60 mA when V_{CC} is between 2 V and 3 V and then returns to normal.

Summary

Use these methods for avoiding POR problems. The transistors for these devices are turned on at approximately 0.7 V, their threshold voltage, while the circuit is functionally operational at a voltage level of approximately 3.3 V. The global routed clocks in Actel devices can also be used as resets for synchronous circuits when connected to either CLEAR or PRESET inputs of synchronous macros for the ACT 1 family, and to the CLEAR input for ACT 2, 1200XL, 3200DX, and ACT 3 families.

Application Note

Three-Stating Actel Device I/O Pins for Board Level Testing

Introduction

During board testing and debugging, it is frequently desired or necessary to place all device I/O pins into a three-state, high-impedance condition. This isolates the device from other devices that have common signal paths on a printed circuit board. The three-state condition also allows board testing for trace integrity or insertion damage to pins. It is usually more convenient to simultaneously three-state all I/O pins of a device with a built-in procedure rather than to create test vectors to force this condition by circuit function.

All Actel field programmable gate arrays (FPGAs) include a special operating mode to place all I/O pins in a temporary, three-state condition. Thus, each Actel device may be easily isolated from I/O signal paths of other devices on a circuit board, enabling a convenient board testing procedure. This capability, combined with Actionprobe® diagnostic tools, allows both single device and system board testing with a power and ease previously unavailable in programmable devices.

ACT 1 Family Three-State Procedure

Three-stating an ACT 1 device is easy using the unique debugging features of the Actel architecture. Three special pins on Actel devices are used for this function: MODE, SDI, and DCLK. To maintain the three-stating feature on ACT 1 devices, no user-defined output or bidirectional pins may be assigned to the SDI and DCLK locations. These pins should remain unassigned or should be defined as input-only locations. All other user-defined pins have no restrictions for their function; they may be assigned as input-only, output-only, or bidirectional pins. These pins can be temporarily three-stated for testing and debugging.

Figure 1 shows the sequence for three-stating an ACT 1 device. Seven data bits are clocked into the device using the SDI pin as data input and DCLK as the clock signal. The MODE pin distinguishes "test" mode from "normal" mode. The data sequence is {0001011}. After clocking the seventh bit, all user-defined pins are placed in a three-state condition until MODE is returned to logic low.

ACT 2, 1200XL, 3200DX, and ACT 3 Families Three-State Procedure

The same special purpose pins are used to three-state ACT 2, 1200XL, 3200DX, and ACT 3 devices: MODE, SDI, and DCLK. In contrast to the ACT 1 family, there are no restrictions on the assignment of user-defined functions to these pins. All user-defined pins have no restrictions for their function; they may be assigned as input-only, output-only, or bidirectional pins.

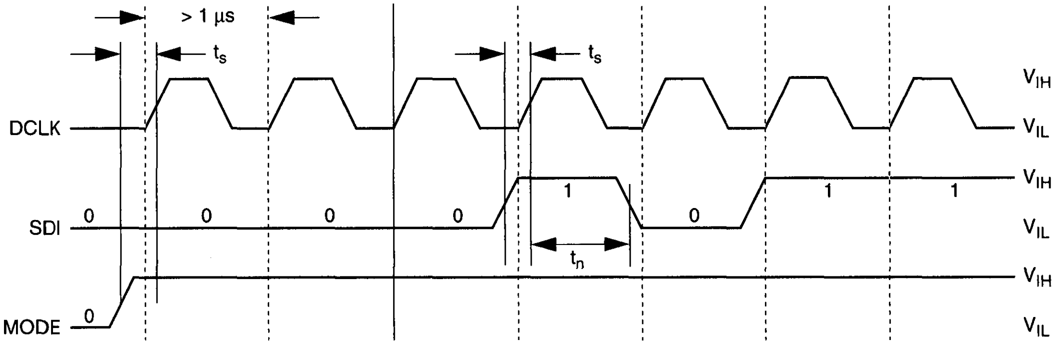
Thirty-two data bits are clocked into the device using the SDI pin as data input and DCLK as the clock signal. The MODE pin distinguishes "test" mode from "normal" mode. The data sequence for ACT 2, 1200XL, 3200DX, and ACT 3 devices is determined by the programmed state of the device. For unprogrammed, blank devices, the data sequence is

(LSB) 0000000000 0000000000 0000000111 10 (MSB)

For programmed devices, the data sequence is

(LSB) 0000000000 0000000000 0000000110 00 (MSB)

Load the data beginning with the LSB. After clocking the 32nd bit, all user-defined pins are placed in a three-state condition until MODE is returned to logic low. Use the same relative timing for the MODE, SDI, and DCLK inputs as shown in Figure 1.



Notes:

1. *$0 V \leq V_{IL} \leq 0.5 V$; $3.0 V \leq V_{IH} \leq VCC$*
2. *Test mode configuration is a low frequency (< 1.0 MHz) operation.*
3. *All setup and hold conditions (t_n, t^s) ≥ 250 ns.*

Figure 1 • *ACT 1 Device Three-State Timing Diagram*

Application Note

Using Actel Devices in Hot Socketing Applications

Hot socketing refers to the practice of inserting and removing a module from a system board while system power is active. Hot socketing is a requirement for systems for which turning off the machine would be unacceptably disruptive. Telecommunications switches and routers are two examples.

The biggest concern regarding hot socketing devices is potential latch up—a well-known cause of failure in CMOS devices such as FPGAs. When a subsystem is plugged into active hardware, a low impedance path occurs between Vcc and ground. A large amount of current flows through this path and can temporarily make the device nonfunctional. Although latch up with Actel devices is not likely to occur, due to the robust internal protection diodes, there is a recommended technique to improve circuit integrity.

Actel recommends that the connector of hot socketing modules be modified to guarantee that Vcc and the Gnd connection be established prior to all other signals. This can be achieved by making the Vcc and Gnd contacts (or fingers) longer than all others. In this way, power will be applied to the FPGA prior to any logic signals, thereby preventing latch up.

The circuit shown in Figure 1 is an added precaution for hot socketing. This added circuit is recommended for FPGA input signals that are connected directly to the connector. It essentially duplicates the internal protection circuit with the input pad of the FPGA. The diodes prevent the input voltage from straying beyond one diode drop of Gnd or V_{CC}, which is within maximum device specifications. The protection resistor is used to limit safely peak current flowing in the diodes.

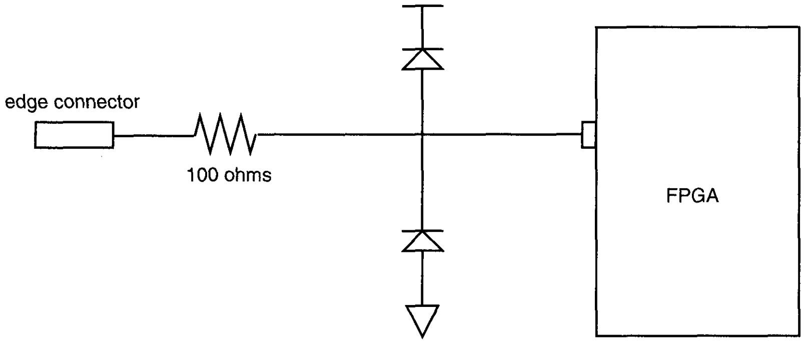


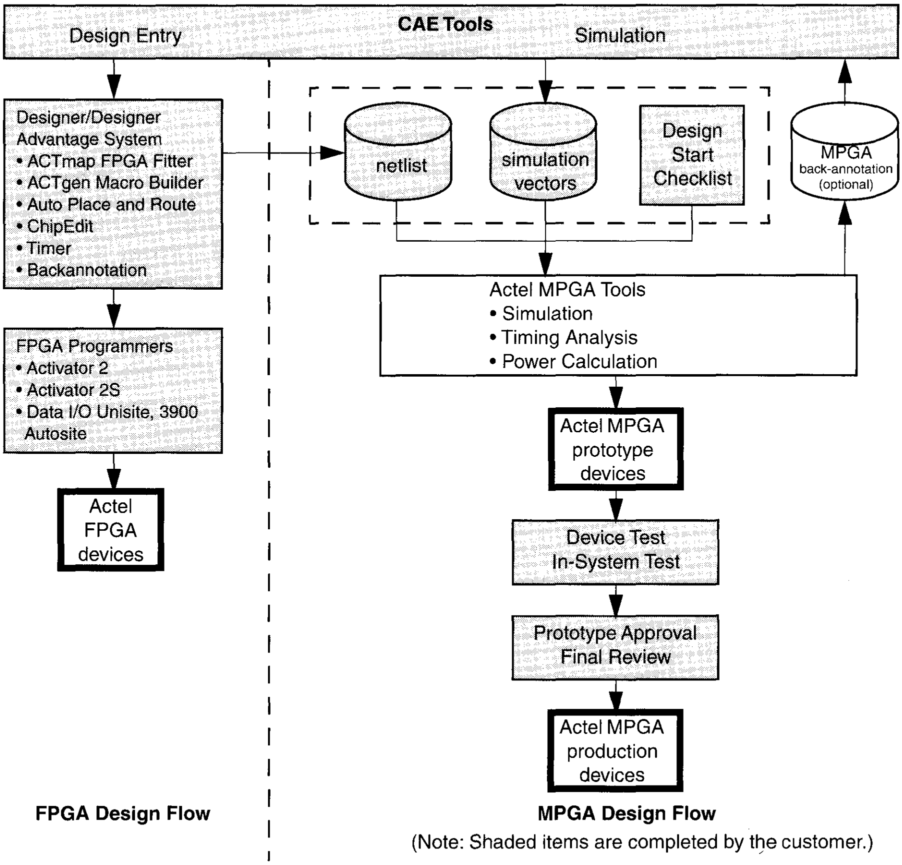
Figure 1 • *Added Circuit Recommended As Precaution for Hot Socketing*

Application Note

Designing for Migration to Actel MPGAs

The Actel Mask Programmed Gate Array (MPGA) family of high-density, high-performance devices comprises mask programmed versions of the ACT 1, ACT 2, 1200XL, 3200DX, and ACT 3 Field Programmable Gate Array (FPGA) families. The MPGA family provides a low-risk conversion path from programmable gate arrays to production quantity devices. By significantly reducing the production costs of a product without technical or time-to-market risks, MPGAs prolong the life cycle of a finished design.

The MPGA products are specifically designed to make conversion from programmable devices virtually transparent to all members of a product team, from initial system designers to final production manufacturing personnel. Virtually any successful FPGA design can be successfully converted to an MPGA device without difficulty. Figure 1 shows the design process for Actel FPGA and MPGA devices. The design flow is tailored to ensure that the conversion process is as seamless as possible, but adhering to some simple guidelines will further ease the transformation. Use the following suggestions and design hints during the FPGA development process to make each design as convertible as possible.



FPGA Design Flow

MPGA Design Flow

(Note: Shaded items are completed by the customer.)

Figure 1 • Actel Device Design Flow

General Design Guidelines

Fundamental principles should be applied to all digital designs regardless of the intended target devices. Actel FPGA designs and their corresponding MPGA versions are no different; reliable systems result from solid circuit designs. Design practices such as reducing unnecessary logic (for example, inverters) using DeMorgan's Law to combine logic, and balancing logic levels (and their corresponding delays) between registers will all help to produce stable circuits. The *1996 Actel FPGA Data Book and Design Guide* includes many application notes to assist in the design process.

MPGA versus FPGA Performance

Actel MPGAs are built using a "sea-of-gates" architecture with 0.8 and 1.0 micron processes. This architecture is similar to the FPGA architecture but has more compact spacing between logic modules and tighter metal-only interconnections. It enables MPGAs to provide performance equal or superior to their FPGA counterparts for almost every design. On average, both logic module and routing speeds are significantly faster for MPGAs. This usually translates into superior system performance for the same design.

Some designs that use ACT 3 devices to satisfy very high on-chip and off-chip performance requirements may experience a small decrease in speed. The ACT 3 I/O modules are especially fast and are not implemented identically in the MPGA versions. However, in most cases, the MPGA is still fast enough as a drop-in replacement for any Actel FPGA. Check the MPGA data sheet to verify that its I/O performance satisfies your particular system requirement.

Power

Due to its relatively smaller die size and metal-only interconnections, MPGA power dissipation is usually significantly lower than its FPGA counterpart. The active power dissipation of a CMOS device is a function of its equivalent capacitance. On average, the removal of programmable elements from the device reduces its equivalent capacitance and, in turn, its active power dissipation. Upon receipt of the Design Start Checklist and associated materials, Actel calculates the MPGA active power dissipation for each design based on this formula. This calculation is immediately relayed to you so that you can update system power specifications accordingly.

Use Synchronous Design

Asynchronous designs are often the cause of many problems, especially when a change is made among target devices to implement a design. An asynchronous design that works with SSI/MSI devices may translate poorly to other technologies such as ASICs because of significant differences in timing between devices. Designs using the same technology from a single manufacturer still may fail because of process differences from die to die. Even similar devices such as Actel FPGAs and MPGAs may not automatically implement an asynchronous design without problems. Generally, most of the problems come from small timing differences that lead to functional errors and days of debugging.

Synchronous design significantly increases the reliability of a circuit. It also makes systems easier to test and debug. The odds of successfully migrating to Actel MPGAs increase directly with the elimination of asynchronous elements in the design. In particular, try to avoid the following types of potential problem circuits:

1. **glitch generators**—Avoid circuits that may create pulses or glitches whose widths depend on variations in propagation delay. For example, an asynchronously generated clock signal generated by decoding the output of a counter with combinatorial logic may work in one technology but may fail to generate a proper rising edge in another. The same decoder may create extra, unwanted clock pulses if the decoder has uneven levels of logic from input to output.
2. **gated clocks**—Do not gate clock inputs to edge triggered storage elements. Use flip-flops with enable inputs instead. Not only is this more reliable, but it also saves logic resources.
3. **asynchronously generated reset signals**—Flip-flops or latches implemented to create asynchronous one-shot signals are unreliable and untestable. Whenever the output of a storage element is fed back to its asynchronous input, the resulting pulse varies widely in behavior depending on process and environmental variations.
4. **internal delay lines**—Strings of buffers or inverters used to create ring oscillators or internal delay lines have unpredictable delay and frequency. They are also extremely difficult to test. Use stable, external oscillators and synchronous pulse generators as alternatives.

Sometimes asynchronous elements must be included in a design. In that case, be sure to understand the effects that technology, process, and environmental variations have on each circuit. Refer to industry standard ASIC logic design manuals for more complete information about creating reliable, testable designs.

Testability

Actel FPGAs are fully testable after programming using two Actionprobe circuits built into every device. Any node may be observed in-circuit, in real time to determine its behavior under any stimulus. The Actionprobes help to verify the results of static timing analysis along with functional and backannotated simulation. Actel MPGAs do not include

Actionprobe circuits, so it is important to design testable circuits into the FPGA from the beginning before converting to an MPGA. A complete simulation procedure for thorough testing is essential to ensure the success of FPGA and MPGA designs. Make sure that the simulation procedures test each section of the chip thoroughly for proper functionality under a wide variety of inputs.

Some common methods to make each design more testable include the following:

- Add global reset inputs to all sequential logic.
- Use global, minimal skew clock signals.
- Use unused I/O pins to observe important internal nodes.
- Use loadable synchronous counters.
- Remove asynchronous logic loops.
- Remove asynchronous one-shot pulse generators.

Use ACTgen Macro Builder or Actel Soft Macro Library

The Actel Designer and Designer Advantage systems include ACTgen Macro Builder software to create fast, reliable soft macros automatically. You can use ACTgen to generate customized macros such as counters, registers, decoders, adders, and multiplexers according to precise specifications. For other types of high-level functions, use the Actel soft macro libraries. Whether coming from ACTgen or the Actel libraries, Actel soft macros are easily migrated from FPGA to MPGA devices because of their built-in reliability and testablility.

I/O Pin Assignment

The performance and routability of Actel FPGA designs are largely determined by the effectiveness of the I/O pin assignments. Likewise, corresponding MPGA conversions have an equally important dependence upon pin placements. Actel's Designer and Designer Advantage development systems include I/O placement software to determine automatically excellent locations for the I/Os, depending on circuit topology and device resources. Since automatic pin placement generally produces superior results versus a design with manual placements, it is important to limit the number of manually placed pin locations as much as possible. The Actel application note "Selecting and Modifying I/O Assignments" thoroughly discusses the issues concerning this subject.

Since MPGA devices are specifically designed to be drop-in replacements for FPGAs, there is no need to change the I/O pin assignments as part of the conversion process. If at all possible, it is best to retain an identical pinout for all versions of a design. The performance of the MPGA may be adversely affected if many changes are made to the pin placements. However, Actel will modify the MPGA pin placements if any changes are necessary.

Summary

By following these design hints and guidelines, the conversion from programmable devices is virtually transparent to all members of a product team. Virtually any successful FPGA design can be replaced by a drop-in replacement without difficulty.

System Level Applications

Section 5: System Level Applications

Application Note

Using Actel FPGAs to Implement the 100 Mbit/s Ethernet Standard

One of the more recent entrants into the high-speed networking standards battle is 100Base-X—Ethernet operating at 100 Mbit/s. This standard is supported by the Fast Ethernet Alliance and sponsored by several key networking companies such as Intel, National Semiconductor, Sun Microsystems, and 3Com. This proposed standard involves many of the types of digital logic functions facing high-speed network designers and, as will be shown in this application note, can be readily implemented using Actel FPGA devices.

The emerging 100 Mbit Ethernet market is expected to mushroom as network performance requirements continue to grow. Network users are expected to have almost doubled between 1991 and 1994, and networks will need to provide these new users with just as much (if not more) bandwidth. A high-speed Ethernet network could solve the bandwidth problems for many classes of users while maintaining compatibility with current equipment and software.

100Base-X Network Standards

The 100Base-X proposal uses two established networking standards to support the 100 Mbit data rate required to implement the tenfold increase in the 10 Mbit rate of the current Ethernet standard. The 100Base-X standard keeps the Media Access Control (MAC) layer the same as the current Ethernet standard, but it raises the data rate to 100 Mbit/s. Since the MAC layer was defined independently of performance level, this increase can be accomplished relatively easily, and the well-proven behavioral dynamics of the Ethernet MAC can be retained. The only change required is to reduce the physical network span to 1/10 of the 10 Mbit/s distance, resulting in a span of about 250 meters.

This reduced span fits well within current structured wiring methodologies. Building-floor wiring in modern installations of Ethernet, such as 10Base-T, are organized as physical stars with a centralized wiring closet and cable runs of less than 100 meters. For LANs, this results in a hub-station architecture with interconnections of less than 100 meters.

At the physical layer, 100Base-X leverages off the proven FDDI standard for 100 Mbit/s communications using a full-duplex 125 Mbit/s Physical Media-Dependent (PMD) sublayer. This supports fiber optic, shielded twisted-pair (STP) and unshielded twisted-pair (UTP) wiring. Combining the MAC layer of Ethernet to the PMD layer of FDDI requires a convergence sublayer (CS) between them. Using the CS, 100Base-X maps the PMD's constant signaling system to the packet-oriented half-duplex system imposed by the Ethernet MAC.

Convergence Sublayer Interfaces

The MAC transmits data to the convergence sublayer in the form of 4-bit words (Figure 1). This data is then encoded into 5-bit groups, serialized, and transmitted by the CS to the PMD sublayer as the transmitPMD signal.

Received data is sent from the PMD to the CS as the receivePMD signal and is synchronized with the 125 MHz clock. Note that the PMD also generates signalDetect when data is detected on the line. The CS decodes the serial data, converting the input 5-bit code groups into 4-bit hex characters and sends it to the MAC as the receiveMAC signal. Note that the PMD extracts the clock from the serial bit stream input. The 125 MHz frequency is recovered from the input data stream by the PMD clock circuits in the CS. In addition, receiveError is generated by the CS to indicate to the MAC that an error has occurred during reception. The carrierSense signal is provided to the MAC to indicate that the line is active. The collisionDetect signal notifies the MAC if a collision has occurred.

This application note will show you how to use Actel FPGAs to develop a complete convergence sublayer. It will subdivide the CS into its functional divisions and will show you how each can be implemented using Actel ACT 3 FPGAs.

Convergence Sublayer Functions

Figure 2 shows the basic data flow in the convergence sublayer. The CS receives transmit data from the MAC as 4-bit words designated transmitMAC. These 4-bit words are encoded into 5-bit symbols (designated TxSYM) that are shifted out to the PMD at the 125 MHz clock rate.

Received data at a 125 Mbit/s rate is sent from the PMD to the CS as the receivePMD signal. The CS formats input data to produce 5-bit symbol groups. Detection of the two-symbol sequence, J and K, marks the beginning of a packet and starts the synchronization of the input data stream. The 5-bit groups are then decoded by the 5B4B decoder and sent to the MAC as a stream of 4-bit words until the packet's end is detected by the reception of the end-of-packet delimiter characters, T and R.

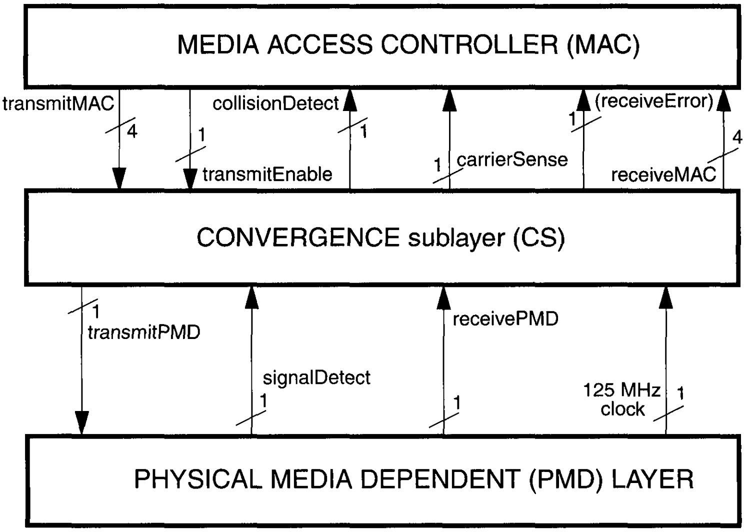


Figure 1 • Convergence Sublayer Interfaces

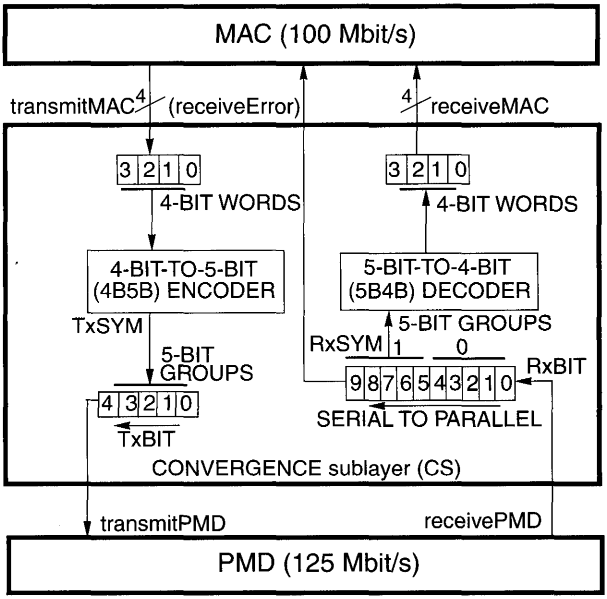


Figure 2 • Data Flow in Convergence Sublayer

The 4B5B encoding/decoding method, which is a subset of the standard FDDI 4B5B encoding method, employs 5-bits to encode/decode both 16 data (hex) characters and the signaling symbols required to indicate the start and end of the data packet. In addition to the 16 valid 4-bit-binary code groups shown in Table 1, there are five special control signals used to indicate start of packet (J followed by K), end of packet (T followed by R), and idle (I). A number of other 5-bit combinations are designated as invalid and represent channel errors or repeater collision artifacts. Thus, the physical line idles until the start of a data packet is indicated by a J symbol followed by a K. Data symbols then follow with the end of data being indicated by a T symbol followed by an R. Idle symbols immediately follow. The job of the convergence sublayer is to extract the control characters or idles from the packet and then send a data-only packet to the MAC. Thus, the MAC never receives idle, JK, or TR symbols. When receiving, the CS reverses the process, encapsulating and encoding the data from the MAC for transmission by the PMD.

Convergence Sublayer Data Flow

The block diagram of the convergence sublayer is shown in Figure 3. The receive state machine generates receiveMAC data and receiveError for the MAC based on the receivePMD data input from the media. The transmit state machine accepts transmitMAC and transmitEnable from the MAC and generates the transmitPMD data to the physical layer. The collisionDetect function is generated by the transmit state machine, based on transmitEnable and the receive state machine's receiving signal.

The Carrier Sense function asserts the carrierSense signal when the convergence sublayer is either transmitting or receiving, based upon the two corresponding internal signals generated by the Transmit and Receive functions. The Link Monitor function generates linkTestFail based on the PMDs signalDetect. LinkTestFail is an internal signal unused by the MAC and can optionally be used by your network management entity.

Transmitted data, shown as the transmitMAC signal in Figure 3, indicates that MAC data is available and is registered in the convergence sublayer logic. Groups of 4 data bits in the transmit bit stream are converted to 5-bit code groups by 4B5B encoding prior to transmission on the 125 Mbit/s PMD. The TxDATA, TxSYM, and TxBIT signals are all different views of the same data, but at different data rates.

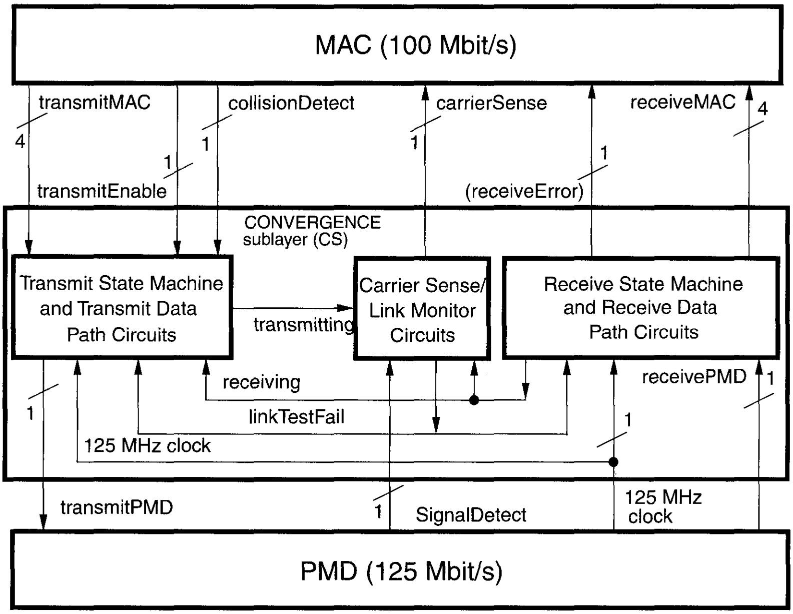


Figure 3 • *Convergence Sublayer Functional Block Diagram*

Using FPGAs to Implement a 100Base-X Convergence Sublayer

As will be seen in the following sections, the 100Base-X convergence sublayer can be implemented as eight functional blocks, each of which forms the subject of a separate application discussion. These are listed under the three main headings: the transmit function, the receive function, and the carrier-sense and link-monitor circuits.

Convergence Sublayer Transmit Function

The design of the transmit function shows some common design techniques used in high-speed FPGA applications. The main function of this block is to provide the requested symbol data to the PMD at the 125 Mbit/s serial rate. This requires the 4-bit MAC data words to be 4B5B encoded and then shifted out using the 125 MHz clock. In addition, the serial data stream needs to be framed using the leading /J/K/ symbol pair and trailing /T/R/ symbol pair. When data is not transmitting, it is replaced by the constant transmission of idle symbols. The transmit function is divided into two blocks as shown in Figure 4:

- The Transmit State Machine
- The Data Path

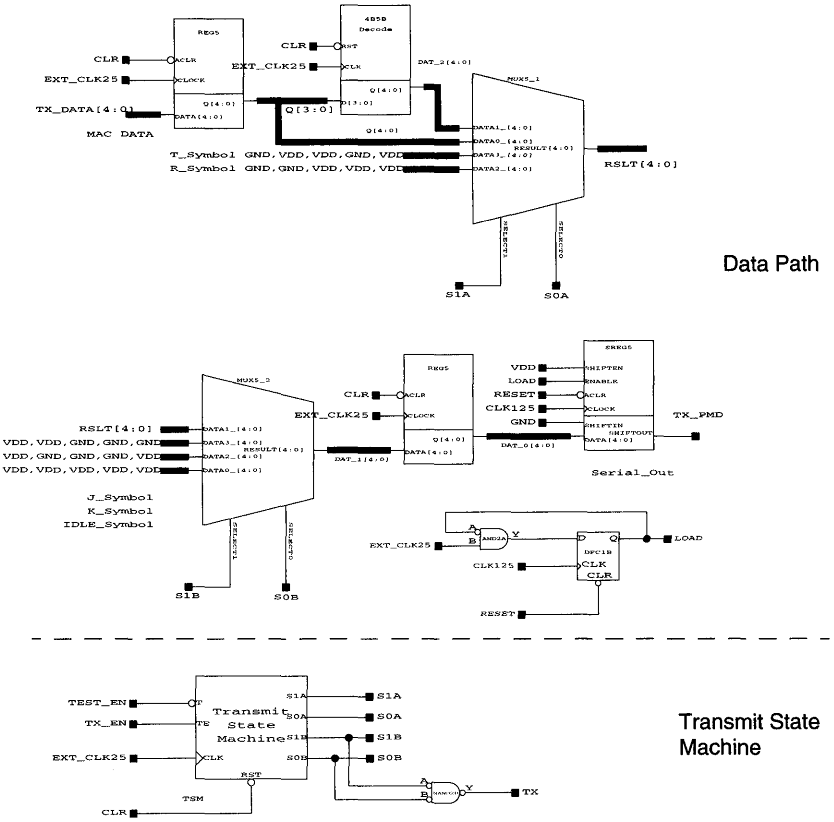


Figure 4 • Convergence Sublayer Transmit Functions

Data Path

The data path portion of the transmit block is shown in Figure 4. The main flow of data comes from the MAC at 25 MHz with a 4-bit-wide data word and a control signal that enables transmission. When MAC data is not being transmitted, the convergence sublayer sends continuous idle (I) symbols to the PMD. When the transmitEnable signal becomes active, a /J/K/ symbol pair is transmitted to indicate the beginning of a data packet. MAC data symbols then follow and are encoded into 5-bit symbols using the 4B5B encoding scheme shown in Table 1. The end of MAC data is indicated by the transmitEnable signal going inactive and a /T/R/ symbol pair is inserted at the end of the data packet. Finally, the CS logic returns to transmitting idle symbols.

Table 1 • *4B5B Symbol Coding*

Symbol	5-bit Code Group (in Convergence sublayer)	4-Bit-Binary Code Group (in MAC)	Interpretation/Function
0	11110	0000	Data character: 0H
1	01001	0001	Data character: 1H
2	10100	0010	Data character: 2H
3	10101	0011	Data character: 3H
4	01010	0100	Data character: 4H
5	01011	0101	Data character: 5H
6	01110	0110	Data character: 6H
7	01111	0111	Data character: 7H
8	10010	1000	Data character: 8H
9	10011	1001	Data character: 9H
A	10110	1010	Data character: AH
B	10111	1011	Data character: BH
C	11010	1100	Data character: CH
D	11011	1101	Data character: DH
E	11100	1110	Data character: EH
F	11101	1111	Data character: FH
I	11111	--	Idle character transmitted between packets
J	11000	--	First control character in start-of-packet delimiter
K	10001	--	Second control character in start-of-packet delimiter
T	01101	--	First control character in end-of-packet delimiter
R	00111	--	Second control character in end-of-packet delimiter
V	00000	--	Invalid character
V	00001	--	Invalid character
V	00010	--	Invalid character
V	00011	--	Invalid character
V	00100	--	Invalid character
V	00101	--	Invalid character
V	00110	--	Invalid character
V	01000	--	Invalid character
V	01100	--	Invalid character
V	10000	--	Invalid character
V	11001	--	Invalid character

The implementation of the described functions involves selecting six different symbol sources for PMD data: the I (idle), J, K, T, and R symbols and the 4B5B encoded MAC data. In addition, a TestData input can be used to provide raw unencoded data to the PMD for use in diagnostics and testing. This selection is accomplished via the multiplexer in front of the output shift register. One multiplexer selects from the I, J, and K symbols or from another multiplexer output. The other multiplexer selects from the T and R symbols and the encoded MAC data. Note the additional path around the encoder, which allows raw (unencoded) data to be provided to the PMD. This is used for system test and diagnostics and is the only way to inject known errors into the system, simulating collision remnants and exercising the boundary conditions of the standard.

The Actel logic implements multiplexers directly in a single logic module so, by inspection, the path through the multiplexer tree requires only two module delays and can easily meet the 25 MHz performance requirement. The 4B5B encode block also requires only two logic levels and can be designed via schematics or via equations. The equations can be automatically compiled using Actel's ACTmap VHDL Synthesis tool and then incorporated into the schematic.

4B5B Encoder

Symbol encoding of the 4-bit data words transmitted from the MAC into the 5-bit coded groups required by the convergence sublayer and the PHY layer employs a modified version of the coding used in FDDI-based systems. The differences from FDDI are that the symbols S, Q, and H are not used and that R is now used as part of the /T/R/ end-of-packet delimiter character group.

Table 1 lists all 32 5-bit data- and special-symbol codes that the PMD can send to the convergence sublayer. The 16 data characters—0 through F (hex)—are shown in Table 1, both as 5-bit code groups and as their 4-bit binary equivalents, as sent by the CS to the MAC. The idle character I and the control characters J, K, T, and R are shown in Table 1 in 5-bit form only, because they are not used in the MAC. The same applies to the remaining 11 possible 5-bit combinations that might be received on the media, all of which have no meaning to the decoder and hence are treated as invalid. For simplicity, each of the 11 invalid symbols is designated as V.

Encoding of 4-bit data words into 5-bit symbols can be accomplished in a few simple logic equations, as shown in the PALASM2 entry format shown in Figure 5.

The above equations translate each bit in sequence. Bits D0-D3 are the 4-bit data word input to the decoder, and B0-B4 define the 5-bit output symbols from the decoder. Thus, in the first equation, bit D0 is always the same as the bit B0, as can be seen by inspection of Table 1. The decoding equations for the remaining output bits (bits B4 through B1) are derived in a comparable fashion.

ACTmap VHDL Synthesis

These equations are then processed by ACTmap VHDL Synthesis, a computer-aided design tool for working with the Actel families of FPGAs. It performs three basic functions:

- PALASM2, VHDL to netlist translation
- Netlist-optimized mapping
- I/O insertion

ACTmap reads the PALASM2 or VHDL source file and translates it into either an EDIF or an ADL (Actel Design Language) output file or Verilog netlist. The output file that it generates is optimized for a specific family of Actel FPGAs (ACT 1, ACT 2, 1200XL, 3200DX, or ACT 3).

You can specify whether the design should be optimized for area or speed. The PALASM2 description for the 4B5B encoder shown in Figure 5 was processed by ACTmap VHDL, and the following results where achieved:

- Area = 9 modules
- Estimated worst-case delay = 8.80 ns

```
;Encoder for 4B to 5B
;Used in 100 Mbit Ethernet application
CHIP 4b5b generic
clk rst d3 d2 d1 d0 q4 q3 q2 q1 q0

EQUATIONS

q4 := d3 + (/d2 * d1) + (/d2 * /d0)
q3 := d2 + (/d3 * /d1)
q2 := d1 + (/d3 * /d2 * /d0)
q1 := (/d1 * /d0) + (d3 :+: d2) + (d2 * /d1)
q0 := d0

q4.clkf = clk
q3.clkf = clk
q2.clkf = clk
q1.clkf = clk
q0.clkf = clk

q4.rstf = /rst
q3.rstf = /rst
q2.rstf = /rst
q1.rstf = /rst
q0.rstf = /rst
```

Figure 5 • PALASM2 Description for the 4B5B Encoder

These results easily meet the 25 MHz requirements of the transmit function and show the speed and capacity capabilities of the ACT 3 architecture. The schematic logic implementation of the 4B5B encoder (see Figure 6) shows the compact nature of the final implementation.

Transmit Operation

Transmit operational states are shown in block diagram form in Figure 7.

The actions shown in Figure 7 are assumed to be instantaneous, although, for simplicity, some time-sequenced events are contained in single states. Unconditional state transitions are unlabeled. Conditional state transitions occur when explicitly shown by the accompanying condition; a state is repeated until some transitional condition is detected. States are atomic in that conditions are evaluated only at the completion of the state's actions. Transitions shown without source states, notably linkTestFail, are evaluated at the completion of every state and take precedence over other transition conditions.

Convergence Sublayer Transmit Operation

The transmit state block diagram begins with the IDLE state. The transmitting and collisionDetect signals are initialized as FALSE and the IDLE symbol is continuously supplied to the PMD. Once the MAC has data to transmit, it asserts transmitEnable and the START state is entered. The transmitting signal is asserted (set to TRUE) to indicate to the Carrier Sense function that data is being transmitted. In addition, collisionDetect is set to the level of the receiving signal. The receiving signal comes from the Receive function; if it is also asserted, a collision has occurred. The waitNibble function synchronizes the MAC data with the PMD clock. The first 8-bits of the MAC preamble are replaced with the /J/K/ symbol pair. If transmitEnable becomes FALSE, the machine makes a transition back to IDLE. If transmitEnable stays asserted, the next state becomes TRANSMIT. During TRANSMIT state, collisionDetect is still set to receiving. The MAC data (TxDATA) is encoded using the 4B5B function, and encoding continues until transmitEnable is disabled. Once transmitEnable is deasserted, the machine makes a transition to the END state. In the END state, transmitting and collisionDetect are both FALSE. The /T/R/ symbols are transmitted to indicate the end of data, and the machine moves to the IDLE state. The assertion of linkTestFail (by the Link Monitor function) causes an immediate transition to the IDLE state and takes precedence over any MAC request.

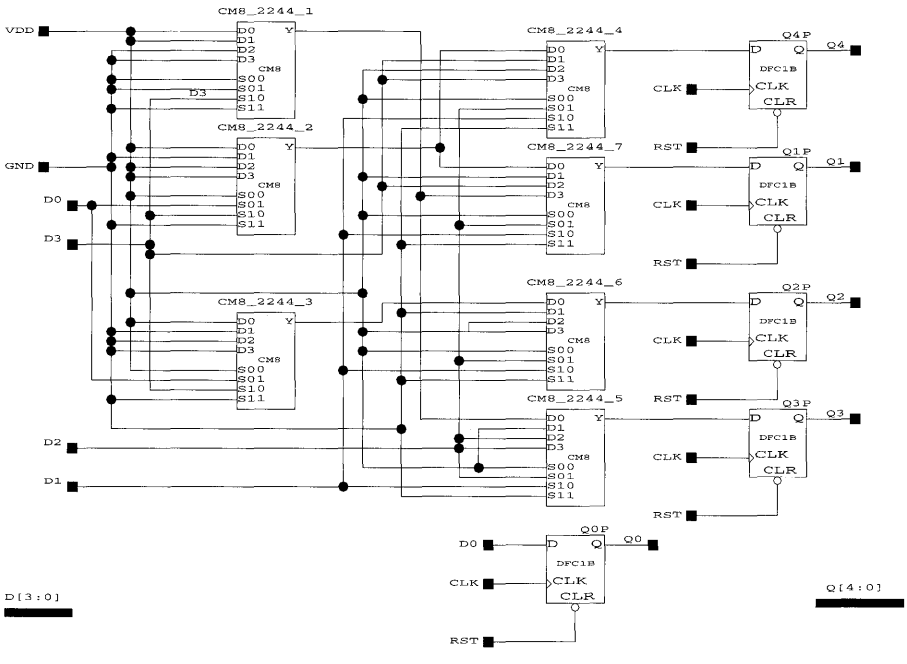


Figure 6 • *Transmit Path: 4B5B Encoder FPGA Logic*

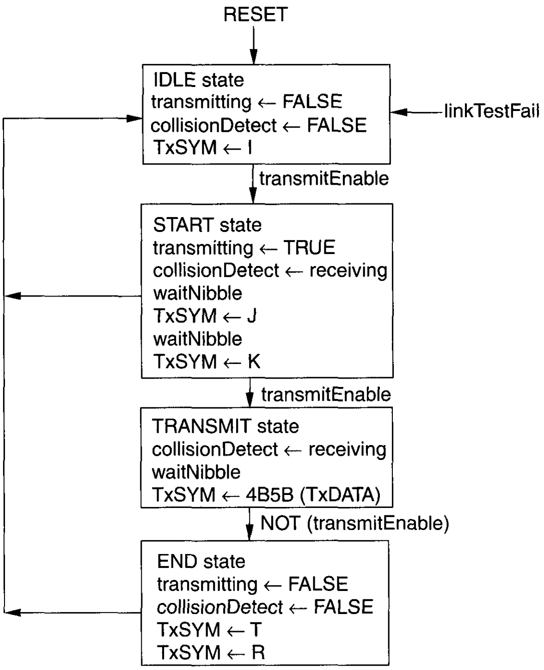


Figure 7 • Convergence Sublayer Transmit Operation

Transmit State Machine

The state-diagram implementation of transmit operation is shown in Figure 8. The state machine starts in the IDLE state and transmits the idle symbol (I) until transmitEnable (TE) is TRUE. As long as TE is TRUE, the machine proceeds through the J and K states, sending first the J symbol and then the K symbol to indicate the start of a data packet. The machine then transmits data until TE goes FALSE (i.e., transmit not enabled (/TE)), after which a T and an R are transmitted, indicating the end of the data packet. The state machine then returns to the IDLE state and waits for the next data packet. The test mode may be entered from the IDLE state by asserting Test mode (TM). In this mode, any 5-bit code symbol may be transmitted, thus allowing known error conditions to be injected onto the network.

The logic implementation of the transmit state machine is shown in Figure 9. Each state is encoded into the transmit state machine flip-flops to allow symbol selection in the transmit multiplexer. These transitions are controlled by the input logic for each flip-flop and depend only on the TE signal and the current state.

The resulting design employs only 11 logic modules and runs well in excess of the required 25 MHz speed.

Note: *This state-machine implementation differs from the commonly seen one-hot approach in that the states are encoded into four D flip-flops rather than a single flip-flop per state. Also, the state-machine encoding shown in this application is more efficient than the one-hot approach because the state flip-flops can drive the data-path multiplexer directly, eliminating the additional encoding logic that would be required to handle the one-hot state variables and to select desired multiplexer sources.*

Convergence Sublayer Receive Function

The receive functions of the convergence sublayer are shown in the block diagram in Figure 10. These functions are discussed in the following sections.

- Shift register, sync detect, and squelch
- Clock generation
- 5B4B symbol decoder
- Receive state machine

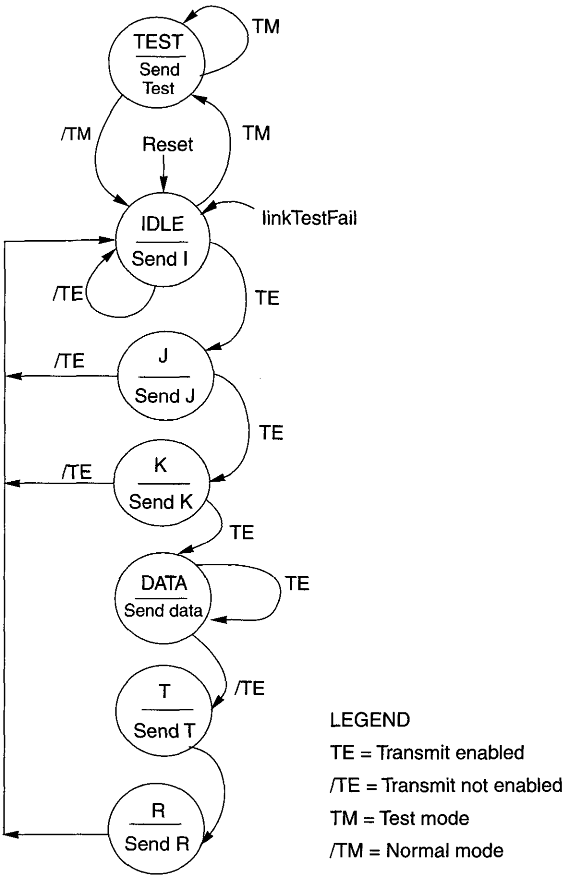


Figure 8 • *Transmit State Machine Diagram*

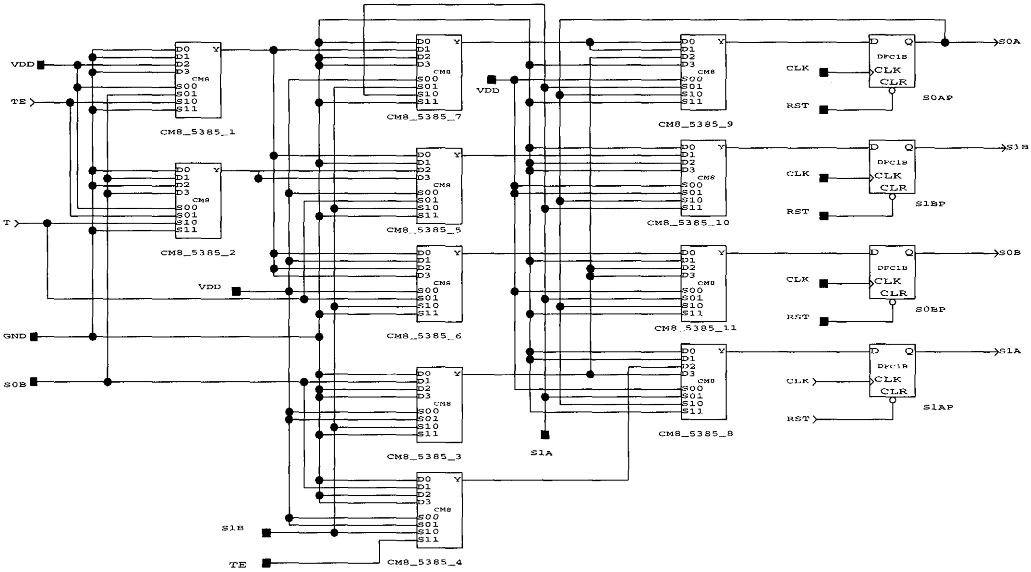


Figure 9 • Transmit State Machine Schematic

Receive Operation

The sequence of receive states is shown in the receive-operation diagram, Figure 11. The receive state machine tracks the received symbols to ensure that a complete packet has been received and indicates the current line state to the next layer of the protocol. The receive process (see Figure 11) involves two separate sets of states. The constituents in the first set—the IDLE, SCAN, CARRIER, and ALIGN states—are prealigned and operate on the raw input bits using RxBIT. The remaining states are aligned and operate on the input data stream as symbols (RxSYM). Output data, designated RxDATA, is sent directly to the MAC in these states.

The RECEIVE state sequence begins with the IDLE state. The receiving signal and the optional receiveError signal are initialized to FALSE. The SCAN state is entered next and the waitBit function synchronizes the machine to the received data stream. At this point, the squelch function filters out noise events by not allowing a transition to the CARRIER state unless two nonconsecutive zeros are detected. Because carrierSense is used by the MAC for deferral purposes, it must be asserted on the detection of any received signal (i.e., received energy, or non-IDLE input) whether or not it's an actual packet. Since carrierSense is also used to detect collisions, it's important to avoid triggering on noise, specifically a single-bit event. If CARRIER is entered, RxDATA is initialized to all zeros (0000) and receiving is set to true.

The system enters the ALIGN state next. In ALIGN, the start of packet symbols /J/K/ is searched for. If at least two idle symbols (1111111111) are found instead, no start of packet has been detected and the machine moves to the IDLE state. If the /J/K/ symbols are successfully found, the START state is entered. In START, the MAC preamble data (55) is substituted for the received /J/K/ symbols. The waitQuint function assures that MAC data is not overwritten. The RECEIVE state is entered next. Usually, in the RECEIVE state, valid data is received and a transition to the DATA state is made. In the DATA state, receiveError is deasserted and 4B5B decoded data is sent to the MAC.

From the DATA state, the machine returns to the RECEIVE state. If, during the RECEIVE state, two idle symbols are received, the PREMATURE END state is entered, receiveError is asserted, and IDLE is reentered. If, in the RECEIVE state, invalid data is received, the DATA ERROR state is entered, receiveError is asserted, and RECEIVE is reentered. Invalid data is not transmitted to the MAC. If, in the RECEIVE state, a /T/R/ symbol pair is detected, the END state is entered, receiving is deasserted, and IDLE state is reentered.

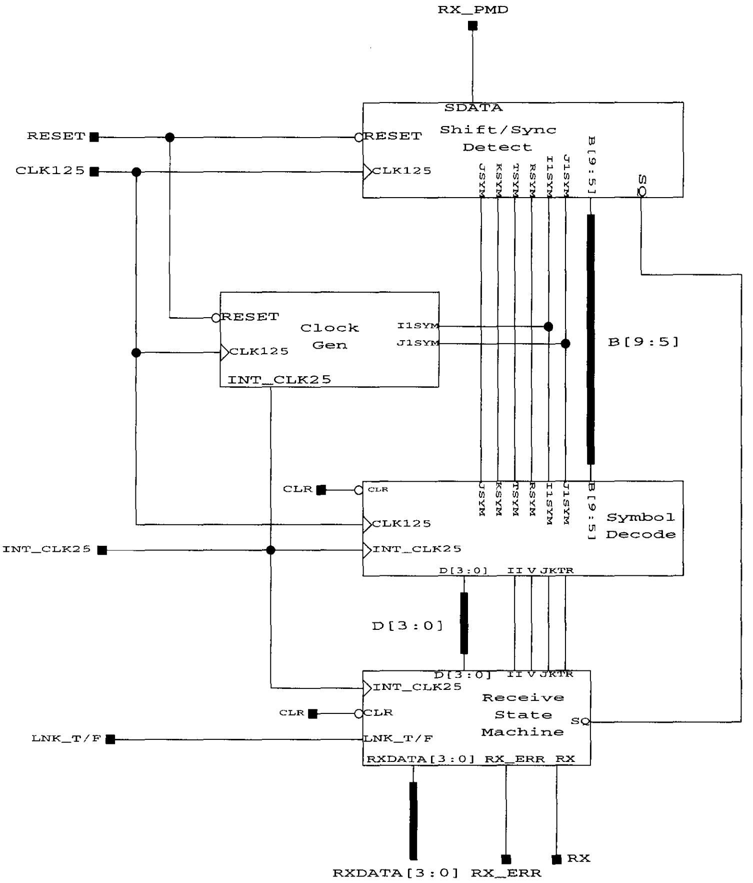


Figure 10 • Convergence Sublayer Receive Functions

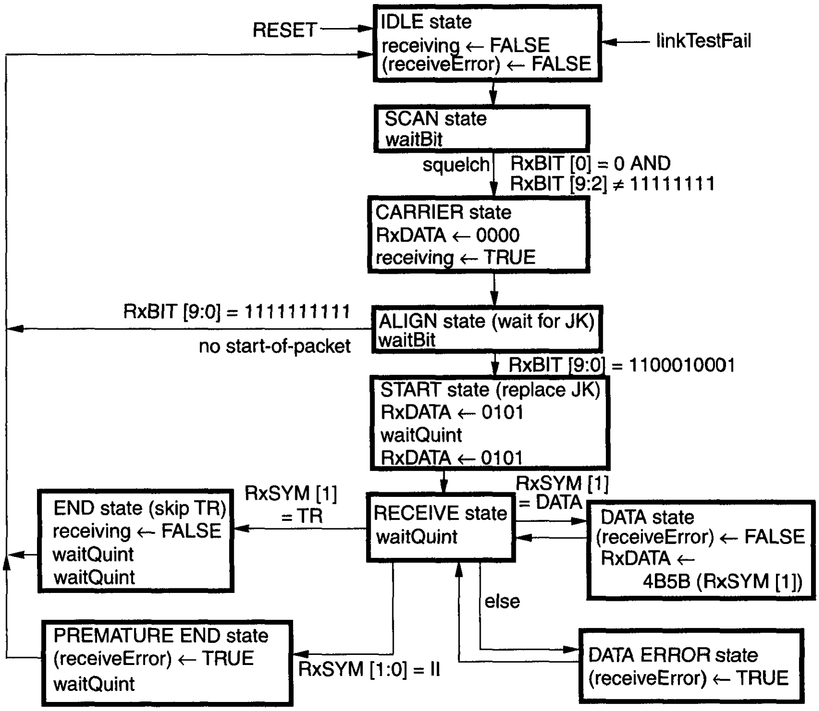


Figure 11 • Convergence Sublayer Receive Operation

Shift Register, Sync Detect, and Squelch

The shift register, sync detect, and squelch circuits (Figure 12) are responsible for shifting serial data at the 125 MHz line rate and detecting clock synchronization symbols. Once a sync symbol is detected, the clock generation state machine adjusts the 25 MHz symbol clock by stretching it the required number of 125 MHz clocks to align it with an input symbol. Control symbols in the input data stream can then be captured correctly by the 25 MHz clock and decoded by the 4B5B decode block.

Serial data is clocked into the shift register and sync detect block by using the 125 MHz clock. Sync symbols are detected as the data shifts. As shown in Figure 12, only a single logic level is required to detect each of the five important sync signals (J, K, T, R, and I). The code groups corresponding to each of these symbols are shown in Table 1.

Note: *The shift register generates both the true and complemented versions of bits B3, B4, B8, and B9. This is required to implement single-level decode for the I symbol because the ACT 3 logic module implements five-input AND/NAND gates with at least two inverted inputs. The technique of providing additional registers with inverted outputs is common when implementing logic functions using fine-grained antifuse FPGAs. The additional registers cost little because of the fine-grained logic module, and they can be used where needed to provide additional logic signals. The abundant routing resources available with antifuse FPGAs also supply the additional routing required to create these additional logic signals.*

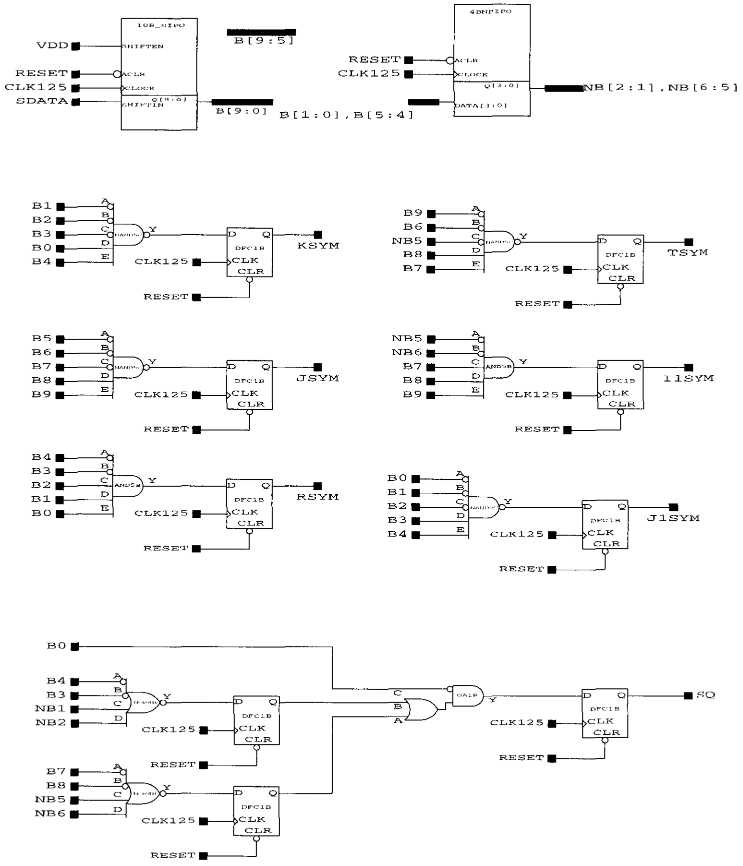


Figure 12 • Shift Register, Sync Detect, and Squelch Schematic Diagram

The squelch function filters out noise events from the received data stream. Zeros are ignored unless there are two noncontiguous zeros within the first 10 bits. At first glance, it would appear that the logic to detect two noncontiguous zeros in a 10-bit word should be quite extensive. However, once it is observed that this function is used only on a serial data stream, several simplifying logic reductions can be made. First, check the least significant bit (B0) for a zero, and then check that at least one of the higher-order bits (only B2 through B9, since B1 is contiguous) is also zero. Any other combination is simply a shift from B0. However, the resulting logic equation for a squelch state (S),

S:= /B0*(/B2+/B3+/B4+/B5+/B6+/B7+/B8+/B9)

is too large to implement in a single FPGA logic level.

To simplify this approach, note that because data is being serially shifted in, higher-order terms can be precomputed and then combined with the critical B0 signal using a single logic level. The logic that results can be expressed by the following three equations:

S1:= /(/B1+/B2+/B3+/B4)

S2:= /(/B5+/B6+/B7+/B8)

S:= /B0*(/S1+/S2)

These three equations are the ones actually implemented in FPGA form. (See Figure 12.)

Note: *As shown in Figure 12, bits B1–B4 and bits B5–B8 are used with registers to develop the two intermediate terms S1 and S2. These two are then ORed with bit B1 to develop the final squelch function (S). This form of pipelined operation works well in serial data applications and will almost always result in faster and more area-efficient FPGA designs.*

Also, notice that the extra inversions on the S1 and S2 terms (Figure 12) are used because NOR functions with inverted inputs map more easily into a single ACT 3 logic module. Synthesis software like Actel's ACTmap VHDL Synthesis program figures this out automatically, allowing the designer to focus on architectural and functional issues instead. Thus, what initially looks like a difficult decoding problem can be significantly simplified to only three logic modules that operate easily at the serial data rate.

Clock Generation

The clock generation state diagram and the clock generation schematic diagram are shown in Figures 13 and 14, respectively. The clock generation logic divides the 125 MHz serial clock by 5 to generate the 25 MHz symbol clock, Clock25. The Clock25 signal (Figure 13) is stretched when a sync symbol is detected, to align it with the 5-bit symbols. This is accomplished by a transition to the Q0 state when the JK signal (start of packet) is active. Entry into Q0 synchronizes the 25 MHz clock (Clock25, the output from states Q2 and Q3) and the load signal. The load signal is active every 5 clocks after synchronization, which captures the 5-bit symbol from the aligned data stream. The symbol can then be safely captured by the 25 MHz clock, Clock25 in the aligned symbol register (ASR). The schematic implementation for this process is shown in Figure 14.

Note: *Each state in the machine uses a single register. This one-hot (i.e., one register at a time) type of state machine design uses the register-intensive nature of fine-grained, antifuse-based FPGAs to reduce the logic complexity required to determine next-state transitions. In traditional encoded designs every state bit is needed to determine which state the machine is in. This can make for large transition terms in complex state machines. FPGAs, on the other hand, can use the additional register available to reduce the logic complexity, because only a single register output is required to determine the state of the machine. Thus, the FPGA's narrow, high-speed logic module can be used to generate the transition terms efficiently. In fact, on closer examination, the implementation of the state machine maps very closely to the state diagram. Transitions from one state to another result in a connection from the starting-states register to the entered-states register. Logic complexity can easily be estimated directly from the state diagram. Because only a single logic module is required to implement even the most complex transition, the entire machine runs easily at the 125 MHz clock rate.*

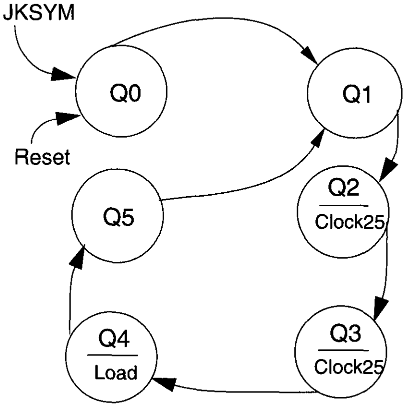


Figure 13 • Clock Generation State Diagram

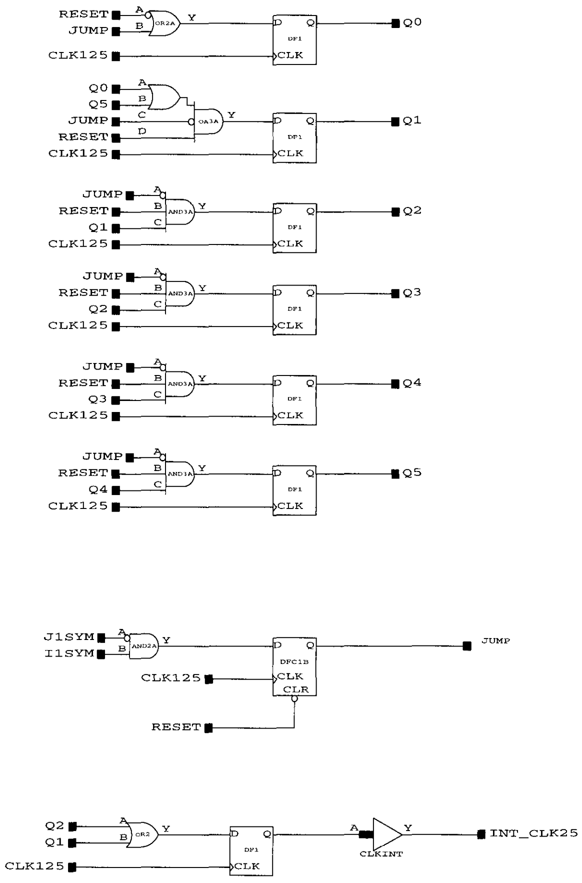


Figure 14 • Clock Generation State Machine Schematic

5B4B Symbol Decoder

Once a symbol has been aligned, the data must be extracted by converting the 5-bit input from the PHY into a 4-bit data word that is sent to the MAC. The logic diagram for this decoder is shown in Figure 15. Symbol conversion is done in accordance with the 4B5B decode table, Table 1. Implementation of the decoder in the ACT 3 family is automatically generated from the logic equations developed from the encoding table by using the ACTmap VHDL tool. As shown in Figure 15, the full decode requires only 24 modules and only two levels of logic, easily meeting the speed required for the 25 MHz clock.

Receive State Machine Logic

As shown in Figure 17, the schematic implementation of the RECEIVE state machine requires only 4 logic modules and two levels of logic. It easily meets the 25 MHz clock rate required for this portion of the design.

Receive State Machine Diagram

The RECEIVE state diagram is shown in Figure 16. The machine begins in the START state and waits for the reception of a JK symbol pair. The RECEIVE state is entered upon the reception of this pair and exited only under one or more of the following conditions:

- Reception of a TR symbol pair (end of data packet)
- Reception of an idle (I) symbol (premature packet end)
- Reception of an invalid symbol (error condition)

Note that if an invalid symbol is received, the ERROR state is entered to capture the event. This state is cleared only by resetting the state machine.

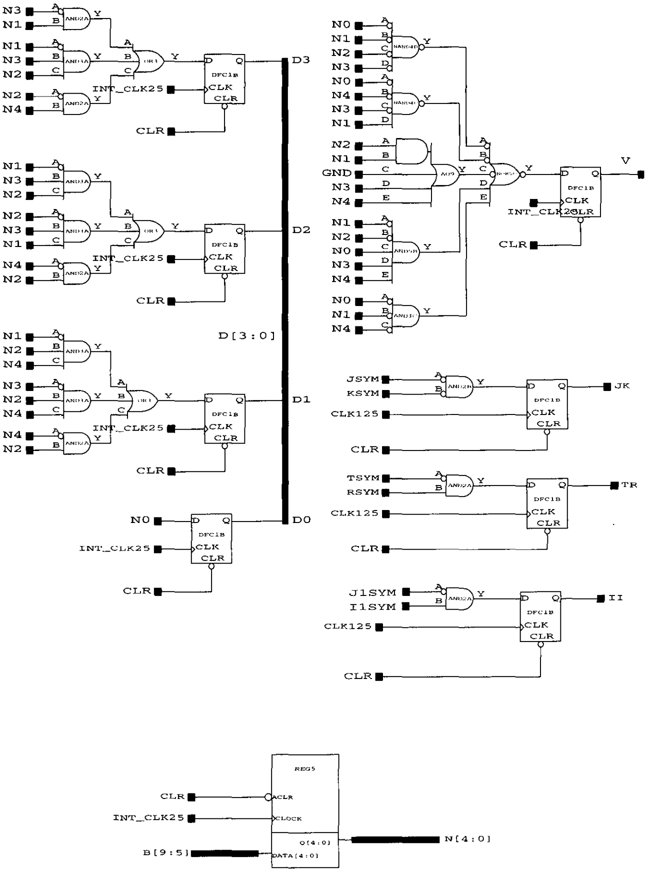


Figure 15 • *5B4B Decoder Schematic Diagram*

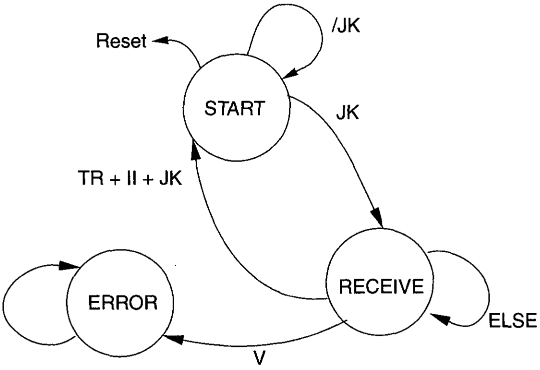


Figure 16 • *Receive State Machine Diagram*

Receive State Machine Logic

As shown in Figure 17, the schematic implementation of the RECEIVE state machine requires only 4 logic modules and two levels of logic. It easily meets the 25 MHz clock rate required for this portion of the design.

Carrier Sense and Link Monitor Circuits

The carrier sense and link monitor circuits combine outputs from the transmit, receive, and PMD blocks to develop the receiveError, carrierSense, and collisionDetect signals. The logic for implementing this process is shown in Figure 18.

Conclusion

This application note has described the complete design of a 100Base-X convergence sublayer. Each building block has been fully tested and documented and is available on disk to those contemplating similar or related applications.

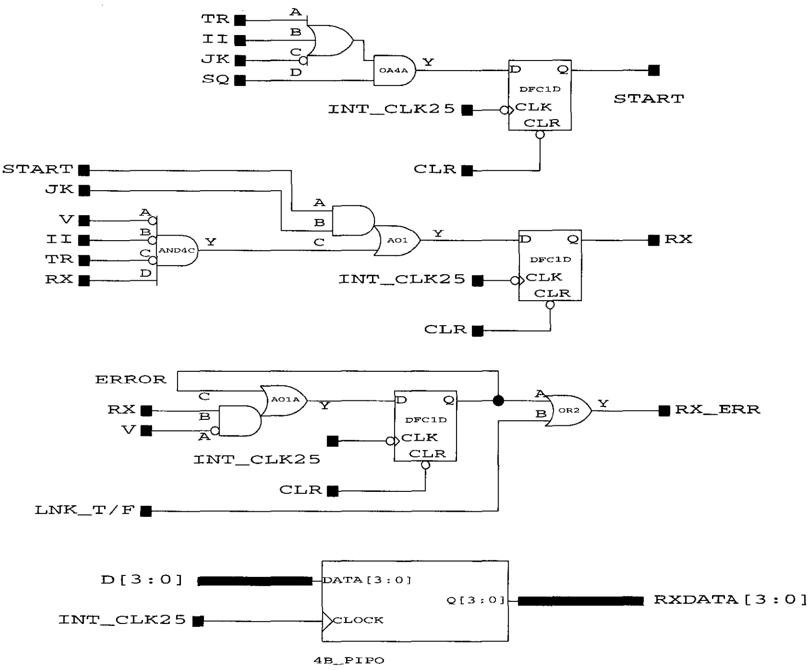


Figure 17 • Receive State Machine Schematic Diagram

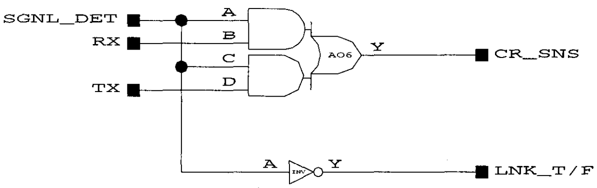


Figure 18 • Carrier Sense and Link Monitor FPGA Logic Schematic

Designing High-Speed ATM Switch Fabrics by Using Actel FPGAs

The recent upsurge of interest in Asynchronous Transfer Mode (ATM) is based on the recognition that it represents a new level of both speed and simplification in telecommunication networks. The most significant characteristic of ATM is that it requires minimum cell processing in network nodes and in links such as repeaters, bridges, and routers. This means that ATM allows systems to operate at rates much higher than current packet-switching systems allow. This improved performance is due to higher media quality and to ATM operation in a connection-oriented mode that guarantees minimum packet loss. This low packet loss is the result of not granting entrance to the network until completion of a setup phase that allocates all necessary network resources.

To reduce the size of the internal buffers in switching nodes, and thus to reduce the queuing delays in these buffers, the information field length in ATM packets is kept relatively small. As a result, as packet size goes down, the speed requirement for each switching node on the network goes up. In general, to keep packet loss to a minimum, the throughput of ATM switching nodes must be in the 1-gigabit-per-second range.

This application note describes how to design typical high-speed switch fabrics that route ATM packets on broadband networks. *Switch fabric* is a term used to denote a large group of basic switching building blocks connected in a specific topology. The design, analysis, and implementation of these building blocks will be described.

ATM Switching Applications

One of the main tasks of an ATM switching node is to transport ATM cells at high speed from its input ports to its destination output ports. This task is performed by the switch fabric. The switch fabric establishes a connection between an arbitrary pair of input and output ports. Switch fabrics usually consist of identical basic units called *switching elements*. The switching elements are interconnected in a specific topology to create the switch fabric.

The Actel ACT 3 family of FPGAs, with their high-speed multiplexer-based architectures, are, as will be seen in this application note, an excellent fit for applications, such as ATM switching, that stress the heavy use of multiplexing. This application note will describe in detail the designs of two typical high-speed ATM switches:

- A pipelined 16:16 switch fabric
- A 16:16 multipath interconnect (MIN) switch fabric

Pipelined 16:16 FPGA Switch Fabric

One of the simpler FPGA approaches to ATM switch fabric design is shown by the straightforward 16:16 multiplexing scheme in Figure 1. This switch fabric design is known as a *single-path network,* because the same path is always used from any given input to a given output. In this example, the switch fabric has 16 input ports and 16 output ports. To achieve connectability, it employs 16 16:1 multiplexers called FFMX16's (Figure 2). Each of the 16 FFMX16s uses five Actel DFM6A basic 4:1 multiplexed flip-flops connected in two pipelined stages.

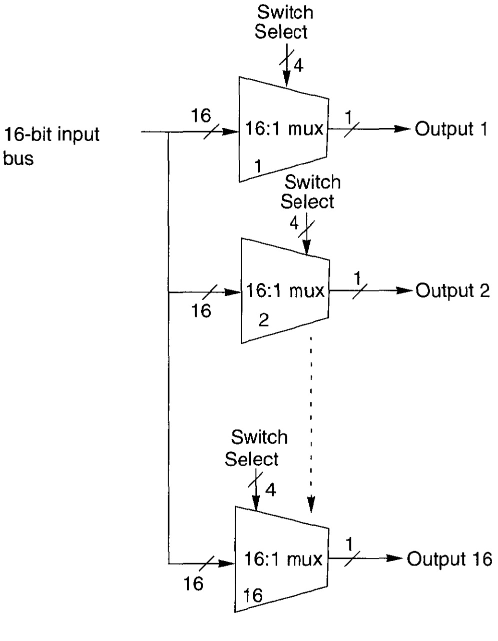


Figure 1 *• 16:16 Basic Multiplexer Switch Fabric*

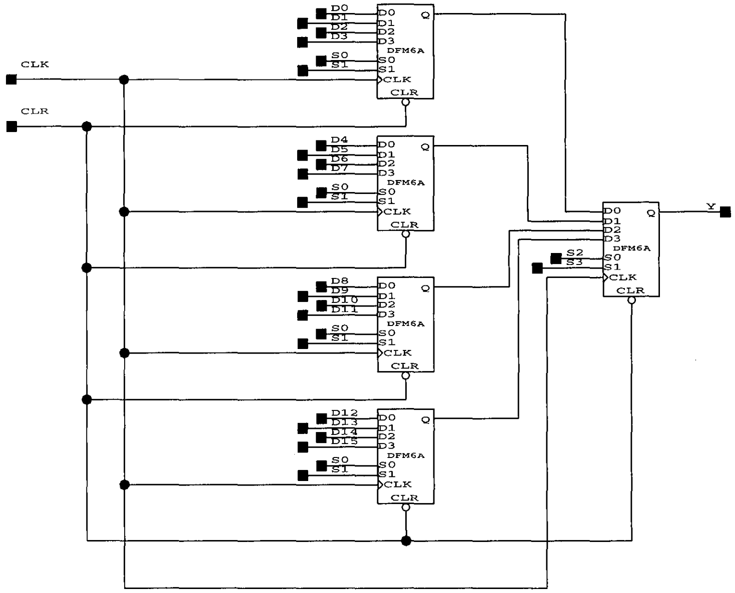


Figure 2 • Two-Stage Pipelined 16:1 Multiplexer (FFMX16)

Each DFM6A (Figure 3) is a 4:1 multiplexer driving a flip-flop and occupies a single Actel ACT 3 sequential logic module. This multiplexed flip-flop introduces only one level of logic delay in the design. In this implementation, once each of the two stages of the 16:1 multiplexer is full, the pipeline outputs data on every subsequent clock cycle. Thus, these 16:1 multiplexers are effectively implemented in one logic level, providing improved throughput.

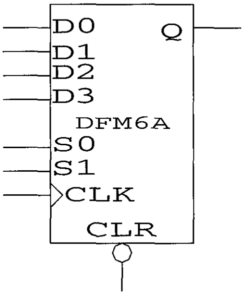


Figure 3 • Multiplexed Flip-Flops DFM6A

Switch Fabric Driving Circuit

The driving circuit for each of the FFMX16s in the 16:16 switch fabric is shown in Figure 4. As shown in the figure, selecting data for the output of each FFMX16 requires 64 signals. This number is based on the need for four switch-select inputs (S0, S1, S2, and S3) for each of the FFMX16s. Switch-select signals S0 and S1 operate with the first stage of each multiplexer, and select signals S2 and S3 operate with the second stages. A drive signal is received as a serial bit stream (SWSEL) that is converted to parallel form by a 64-bit serial-to-parallel shift register (SIPO64). Input data to the switch is received on the lines D[15:0]. Notice that the FFMX16 shown in Figure 4 represents 16 FFMX16s, so the inputs are 16 bits wide.

It takes two clock cycles to fill the FFMX16 pipeline, after which data is present on the OUTP[15:0] bus at each clock cycle. The 64-bit data selection word S[63:0] from the shift register is divided into four groups of 16 bits each, which are used to select the appropriate routing through the switch fabric. Note that there are separate clock lines for both the shift register and for the switch fabric so that data can be clocked to the output bus at a rate different from that of the shifted control bits.

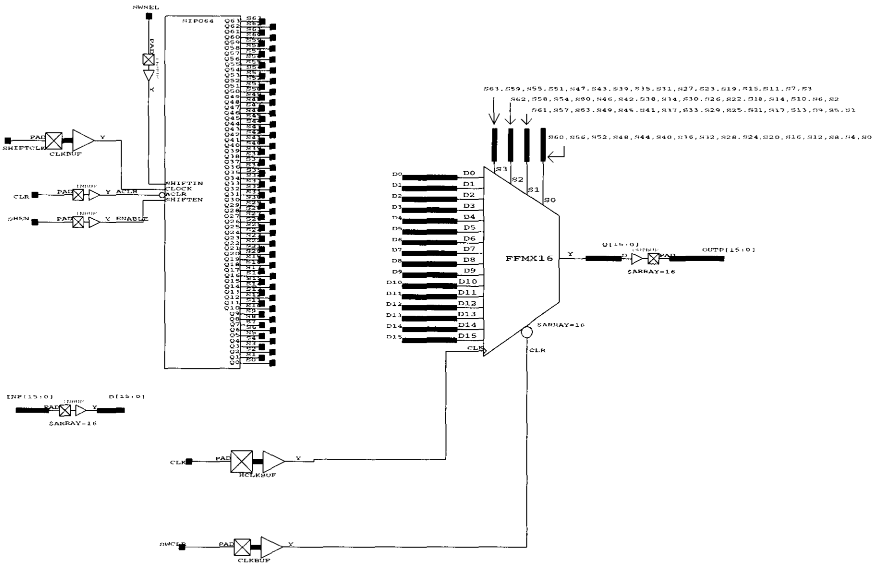


Figure 4 • Top Level View of the 16:16 Switch Fabric Design

Using ACTgen Macro Builder

The 64-bit serial-to-parallel shift register (SIPO64) is generated by the Actel macro generator, *ACTgen* included with Designer Series software packages. With ACTgen's graphical user interface, you can build structured macros (counters, adders, etc.) by simply clicking on a few menu choices. The ACTgen Macro Builder then creates functions that effectively use the Actel architecture. Each macro is developed with the goal of limiting module count, maximizing performance, and restricting loading to acceptable levels.

In this design, the SIPO64 is generated by simply choosing the desired parameters from ACTgen's graphical user interface (64-bits, serial-to-parallel, active-low clear, active-high shift enable, and positive-edge triggered clock). The created shift register is then instantiated in the design with no need for simulation. The ACTgen macros are already tested to guarantee correct functionality.

Note: *In the N:N multiplexed structure shown here, any given input may be broadcast to all outputs simultaneously. Also, an advantage of this approach is that after only two clock cycles, the pipeline is full and ready to output data. A disadvantage of this scheme is that it allows only one possible path, and no alternative paths, between each input and output. To implement a multiple-path capability, multiples of this switch fabric can be cascaded together. However, a better solution is the MIN switch fabric.*

16:16 Multipath Interconnect (MIN) Switch Fabric

The primary advantage of a multipath interconnect network (MIN) is that it permits the creation of alternative paths between a given input and a given output in order to avoid possible packet collisions. One implementation of a MIN switch is the two-section Banyan network shown in Figure 5. This network consists of four stages that drive a second group of four. The second group is made up of the first four stages with a reversed topology—it is the mirror image of the first. Adding this second half produces a complete MIN switch fabric in a minimum number of stages.

The first four stages (see Figure 5) enable any output to be reached from any input via one specific path. This is the standard Banyan configuration. The second four stages use a reversed Banyan topology. Together, the two sections provide the multiplicity of paths required for a MIN switch fabric. That is, in an N:N MIN switch fabric, N internal paths are available to reach any output from an arbitrary input.

Each basic switching element used in this switch fabric is a 2:2 switch. (See Figure 6.) Depending on the value of switch line SW, the data will be either passed or crossed between the input and output lines. Figure 7 shows the implementation of the basic switching element using Actel DFME1A ACT 3 multiplexed flip-flops. As shown in the figure, two multiplexed flip-flops are required to implement the switching element. The truth table for the basic switching element shown is given in Table 1.

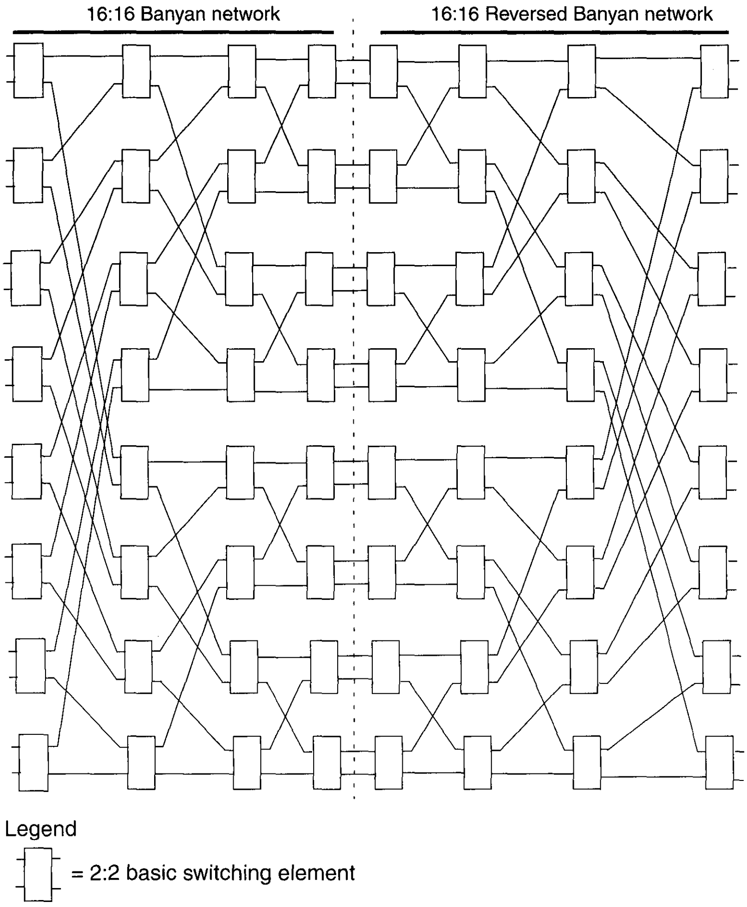


Figure 5 • 16 X 16 Mulltipath Interconnect Network (MIN) Switch Fabric

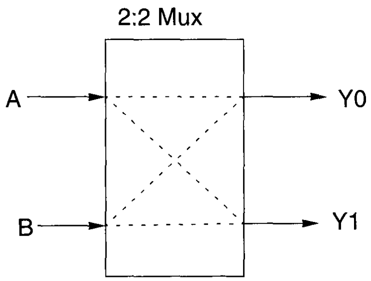


Figure 6 • Possible Signal Paths in a 2:2 Basic Switching Element

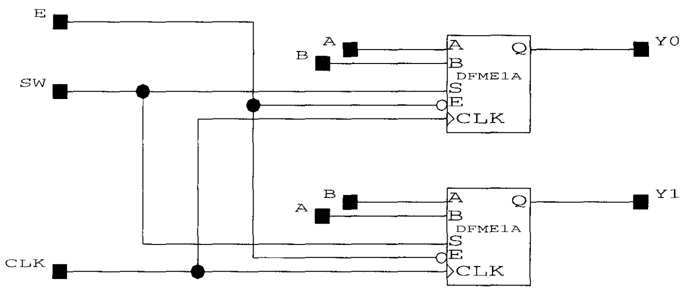


Figure 7 • *2:2 Basic Switching Element SWCELL*

Table 1 • *Truth Table for 2:2 Basic Switching Element SWCELL*

Clock CLK	Enable E	Switch SW	Y0	Y1
X	1	X	As in previous state	As in previous state
↑	0	0	A	B
↑	0	1	B	A

Notes: ↑ = Triggered on positive edge of clock
X = Don't care

Assuming an N:N MIN switch fabric, the number of stages in the network is $2\log_2 N$. If N = 16 (as in the present case), the MIN switch fabric is implemented in eight stages. Thus, a 16:16 MIN can be constructed of eight stages, with each stage consisting of eight 2:2 basic switching elements.

Switch Fabric Driving Circuit

The driving circuit for the MIN network is similar to the one used for the multiplexer-based switch fabric described in the previous section. As shown in Figure 8, the switching elements (SWCELL) are connected to each other to implement the topology shown in Figure 5. The SW lines of the switching elements are driven by the outputs (S[63:0]) of a 64-bit shift register. The S[63:0] signals are received as a serial bit stream (SWSEL) and are converted to parallel form by a 64-bit serial-to-parallel shift register (SIPO64). Input data to the switch is received on the lines IN[15:0] and is clocked to the outputs once the SWCELLs are enabled.

It takes eight clock cycles to fill the switch fabric pipeline, after which data is present on the OUTP[15:0] bus at each clock cycle. The 64 bits of the data selection word (S63:S0) from the shift register are used to select the appropriate routing through the switch fabric. Note that there are separate clock lines for the shift register and for the switch fabric so that data can be clocked to the output bus at a rate different from that of the shifted control bits S[63:0].

Note: *The complete multipath interconnect switch fabric is implemented by using 128 (8 x 8 x 2) multiplexed flip-flops of the type DFME1A. The straight multiplexed switch structure discussed in the previous section requires 80 (5 x 16) DFM6A multiplexed flip-flops. However, this size differential does not translate for larger values of N (where N is the number of input and output ports). As N gets larger, the number of modules required to implement the MIN network does not increase as rapidly as it does for the simple mux-structured network. Also, notice that the multiplexed flip-flop used in the MIN network is a 2:1 type, whereas the straight mux switch fabric requires a 4:1 type. The 2:1 multiplexed flip-flop offers the advantage that, because of its lower fanin, it is easier to route on the Actel software.*

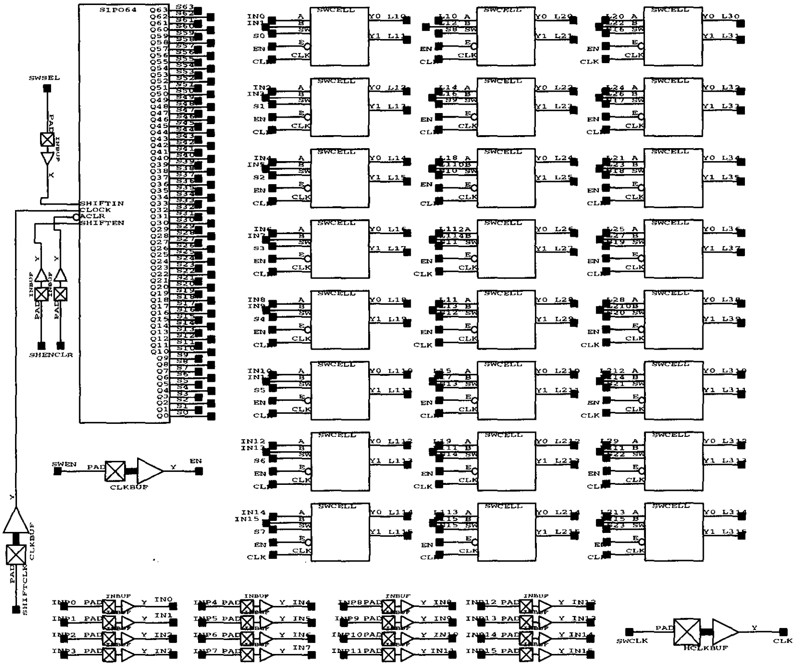


Figure 8 • Top-Level View of the MIN Switch Fabric Design

Timing Analysis

The MIN switch fabric discussed here can be implemented in most Actel ACT 3 devices, such as the A1425A, the A1440A, A1460A, and the A14100A. The timing analysis given in this section was obtained from the A1440A-2.

The MIN switch fabric can be operated as fast as the slowest switching element can switch its data from input to output. The basic switching element has a 5.7 ns clock-to-q (input-to-output) delay, along with 0.7 ns of setup time. Hence, the maximum frequency of the switch fabric clock is 156 MHz. In a 16:16 switch, this provides a maximum throughput of 2.5 gigabits per second. Note that it takes eight clock cycles for data to move from the switch fabric input to its output (about 51.2 ns). However, as in all such pipelines, once all stages of the network are filled up, data is output at every subsequent clock cycle.

Conclusion

The basic concept behind switch fabrics is multiplexing data from input ports to output ports. The multiplexer-based architecture of Actel FPGAs fits this requirement. High-speed switching networks of almost any topology can be implemented efficiently using the multiplexed flip-flops in the Actel library. Each of these flip-flops is mapped to only one sequential module within the FPGA to take maximum advantage of the die area within the chip.

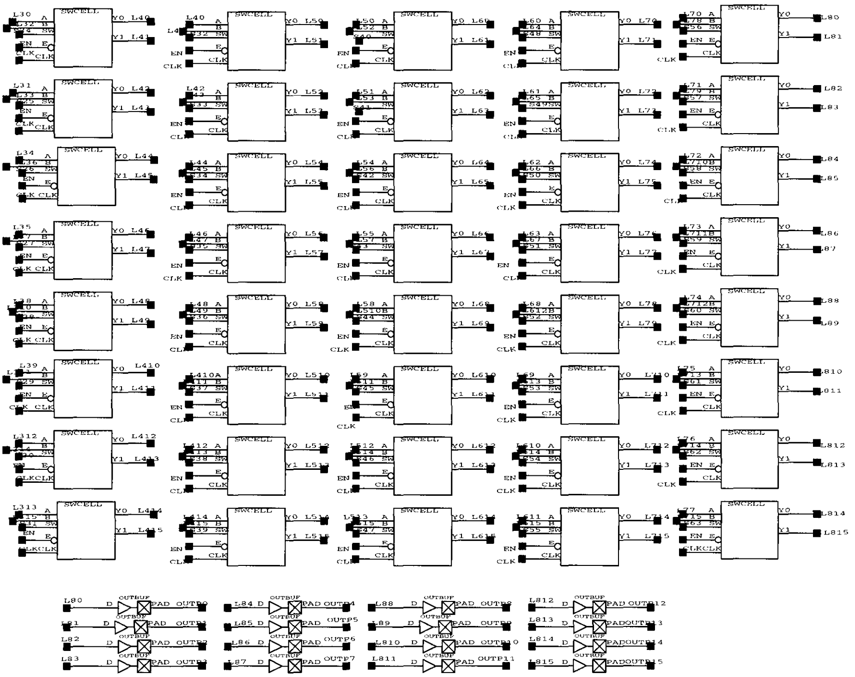


Figure 8 • Top-Level View of the MIN Switch Fabric Design (continued)

Actel

Application Note

Universal Serial Bus Host Interface on an FPGA

For many years, designers have yearned for a general-purpose, high-performance serial communication protocol. The RS-232 and its derivatives have long since been eclipsed by the requirements for speed and integrity of data. The emerging Universal Serial Bus (USB) specification may well prove to be the answer to the needs of system designers for now and the future.

USB Background

The USB was developed by a consortium of companies, including IBM, Microsoft, Intel, and others. These companies saw a need for a robust link between PCs and telephones. They also sought to make PC peripheral ports more expandable to support more devices outside the card cage. Concomitant with the expandability requirement are support for a range of transmission rates for different devices, maintenance of data integrity, and automatic PC reconfigurability. Overall, the goal of the USB is to provide a low-cost, flexible, easy-to-use bus that can support PC expansion at real-time audio/video data rates.

USB Specification

The USB is defined as an industry standard, which can be implemented by any company that produces a device to communicate with other USB devices. The standard defines communication protocols, transactions, electrical characteristics, and bus management.

Some of the applications for the USB are shown in Table 1, which categorizes them according to performance requirements. The USB consists of a host and its devices, which are connected in a hierarchical topology, as shown in Figure 1.

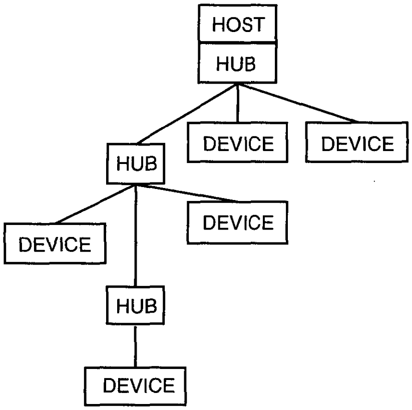


Figure 1 • USB Topology

Table 1 • USB Applications

Performance	Applications	Attributes
Low speed Interactive devices 10–100 KB/s	Keyboard, mouse Stylus Game peripherals Virtual reality peripherals Monitor configuration	Lower cost Hot plug-unplug Ease of use Multiple peripherals
Medium speed Phone, audio, compressed video 500 KB/s–10 Mb/s	ISDN PBX POTS Audio	Low cost Ease of use Guaranteed latency Guaranteed bandwidth Dynamic attach-detach Multiple devices
High speed Video, disk 25–500 Mb/s	Video Disk	High bandwidth Guaranteed latency Ease of use

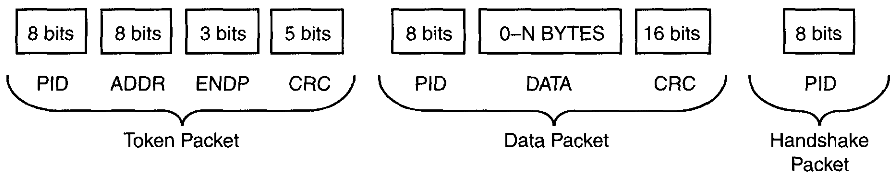


Figure 2 • Three-Stage Transfer Packet Fields

The USB protocol supports three types of communication packets: token, data, and handshake. Depending on the purpose of the communication, a transfer may consist of all three packet types, a token and handshake, or a token alone. A three-packet transfer is for normal data transfers with handshaking. A simplified view of the fields in each packet in a three-packet transfer is shown in Figure 2. The fields of a single-stage or token-only packet are shown in Figure 3. (Not shown are the 1-byte synchronizer fields that head each packet and are filtered out by the data synchronizer hardware.)

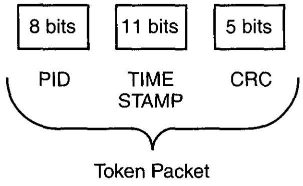


Figure 3 • Start-of-Frame Packet Fields

A USB Interface on an FPGA

Developers of interfaces to buses, such as the USB, specified without a carrier clock have many asynchronous events to contend with. For example, the USB host must phase lock to asynchronous data transmitted from any one of several possible frequencies within eight transmitted bits. Designing a system that can asynchronously detect a valid data stream and phase lock to it throughout the transmission is a delicate task. Debugging such systems can only be done in hardware at actual clock and data rates. The debugging usually requires several iterations of the design, making an ASIC implementation impractical.

Using an FPGA to prototype the design for system verification is the safest and lowest-cost path to validating the design. For the design example described here, we use a member of the Actel 3200DX family of devices. These devices have the resources and the performance to implement a working system.

The low granularity of the logic modules on the 3200DX devices make them good candidates for efficient design synthesis because the synthesizer can map library components directly to the logic modules.

The design description presents a somewhat simplified view of the USB specification so as to focus on the role of the FPGA in the host system. We do not discuss operating system issues such as how to process erroneous or incomplete data transmissions. The status for such occurrences are encoded according to the requirements and passed to the operating system running on the CPU.

USB Host System Architecture

The architecture of the USB host interface is shown in Figure 4. There are data paths between the bus and the host supported by a FIFO attached to bidirectional data paths. On the host system side, the FIFO sends received data to the main memory using DMA. (The DMA controller is system specific and will be left as a block.) In data transmissions, the host loads data to the FIFO from main memory.

The host establishes and maintains a connection table that contains target data for token formation and time stamps. The table contains all the device-specific information required to make transfers between the bus and the host.

The operating system initiates transfers by loading the Packet Type register with the packet type code and with the connection table starting address to point to the PID code and to the target device description in the connection table.

Transactions requiring acknowledgment are terminated by an overflow of the time-out counter. The Status register provides feedback to the host operating system as to the results of bus transfers.

On the USB side of the data path, a serializer and a Shift register convert data between serial and parallel formats. Data is pushed or popped from the FIFO according to the direction of the data. The CRC is used to append a code to outgoing packets and to compare against the codes of incoming data.

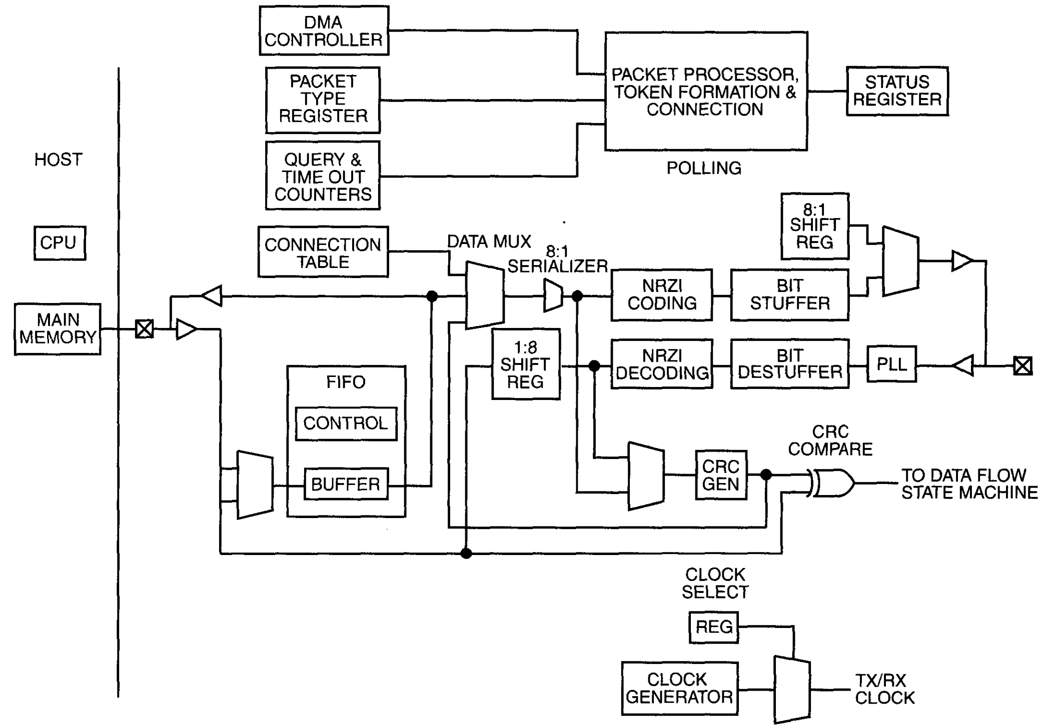


Figure 4 • System Architecture

All data transfers are NRZI coded and decoded. NRZI encoding represents a logical 1 as no change in level and a logical 0 as a change in level. The NRZI coding state graph is shown in Figure 5.

Transmissions are also bit stuffed to retime the receiver logic. Stuffed bits must be inserted by the transmitter and discarded by the receiver. The bit insertion state graph is shown in Figure 6. Both the NRZI and Bit-Stuffer state machines are alerted to the presence of data on the next clock edge by the Data Flow Control state machine.

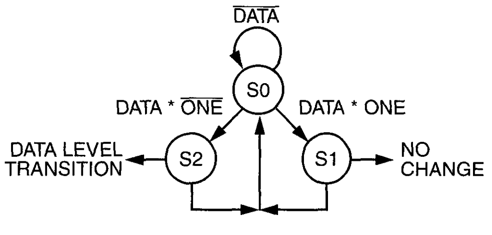


Figure 5 • State Graph of NRZI Coding

At the edge of the USB and the interface is the synchronizing logic. Eight bits of synchronization data are sent by the transmitter prior to each packet. The synchronization data are discarded by the receiver and is not shown in the packet field diagrams.

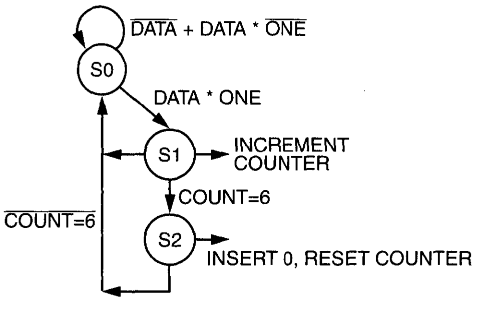

$\overline{\text{DATA}}$ + DATA * $\overline{\text{ONE}}$
S0
DATA * ONE
S1
INCREMENT COUNTER
COUNT=6
$\overline{\text{COUNT=6}}$
S2
INSERT 0, RESET COUNTER

Figure 6 • State Graph of Bit Stuffing

The transmit side of the system described multiplexes outgoing synchronization and communication data so as not to NRZI code or bit-stuff code the synchronization byte. Data transfer frequency is selected through the clock multiplexer according to the target frequency. On the receive side, there is a phase lock loop to detect a synchronization clock and align incoming data to the local clock.

Operation

The state machine assembles packets under the direction of the operating system, which determines the schedule composed of slots, frames, and superframes. Frames consist of a start-of-frame (SOF) token and successive transfers. The Assembly state machine is keyed to the beginning of a frame by the operating system and inserts an SOF token at that point. A superframe is made up of several frames, each of which has its own SOF token.

Packets can consist of tokens only when all transmitted data comes from the connection table. Other packets include a token and data packet, with the latter coming from the FIFO and being appended to the token by the state machines.

There are state machines to control the packet and data transfer processing. The packet processor, shown in Figure 7, calls the Token Formation state machine, shown in Figure 8.

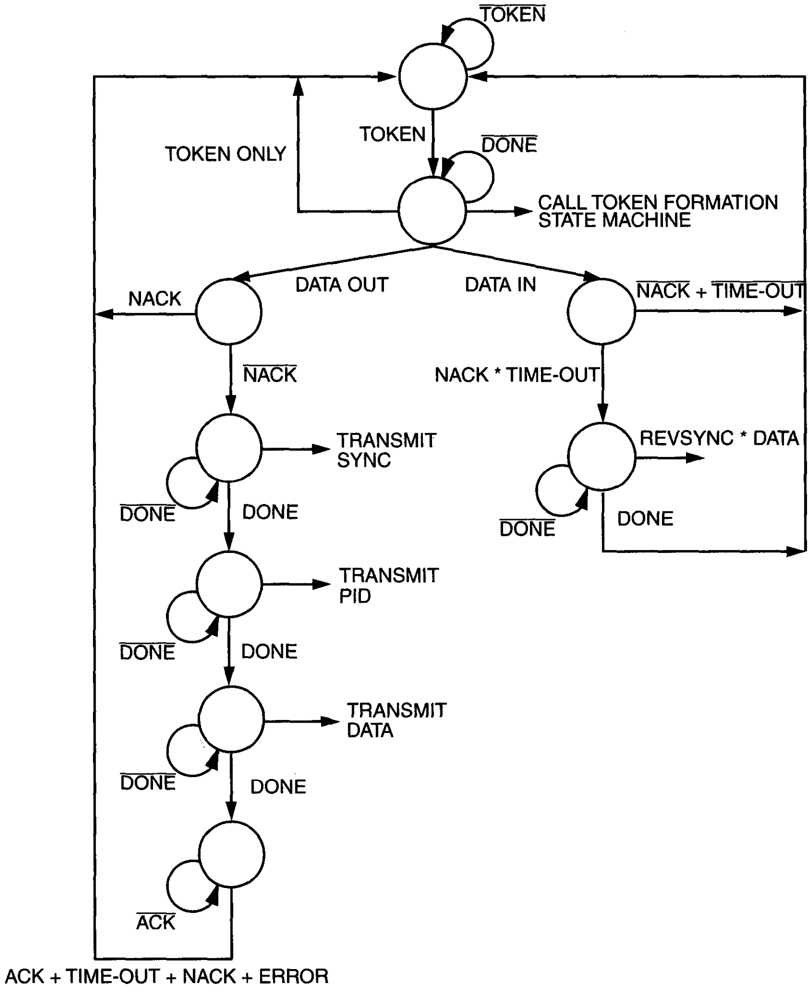


Figure 7 • Packet Processing State Graph

The query timer activates the Connection Polling state machine, shown in Figure 9. The polling broadcasts a default address for response by any device new to the network. A response is relayed to the host so that the device can be configured and its entry made in the connection table.

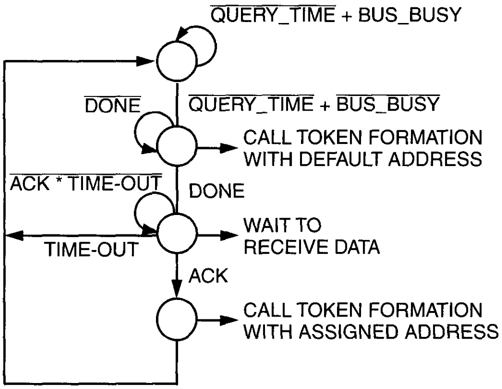


Figure 8 • *Token Formation State Graph*

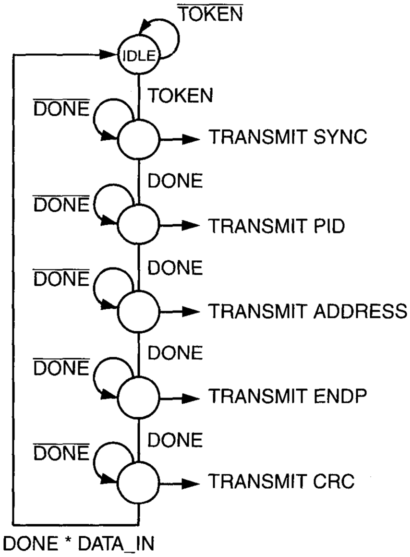


Figure 9 • *Connection Polling State Graph*

The operating system uses the configuration table to store the token bytes (except for the last 5 bits of the last byte, which come from the CRC as calculated during transmission). Transmission control is divided into two parts; Packet processing has overall responsibility, and the Token Formation state machine controls the flow of bytes making up the token.

The Token Formation state machine increments the token buffer address counter to point to the next byte so that it is available for transmission before the last bit of the previous byte is shifted out (indicated by the DONE condition in the state graph). On the next clock, the first bit of the next byte begins transmission.

The USB host must support different data clock rates from its devices. The system contains a clock generator counter whose outputs decode to the USB data rates. Data describing clock rate comes from the connection table and is registered to select the appropriate clock frequency for the session.

FPGA Design

All state machines are implemented in the FPGA by using bit-per-state encoding. This encoding scheme uses one flip-flop for each state, minimizing combinatorial logic resource requirements. Bit-per-state encoding takes advantage of the balance of sequential and combinatorial resources on the device. It is also the highest-performing encoding scheme.

Data is clocked in and out of the device at a rate determined by the device definition. The frequencies of the possible clocks have not been defined as of this writing, but the variability will require the clocks to be generated within the FPGA, making the USB interface a multiple-clock design.

The 3200DX family features quadrant clocks, which are suited for the multiple and dynamic frequencies requirements described here. Logic common to some clock frequency may reside in one or more quadrants and be clocked by those quadrants' clocks. Other logic using different clocks may reside in another quadrant.

The existence of dedicated, dual-port, synchronous SRAM resources on the device allows easy implementation of FIFOs for bidirectional communication with the USB and the host system main memory. The FIFO buffer storage uses the dedicated, dual-port SRAM for message storage. Because reads and writes are completely separate, the FIFO controller may allow data from one side of the buffer to be transmitting while the other is receiving.

The FIFO design need not be specified, since it is generated automatically by the Actel ACTgen Macro Builder. The tool will create the FIFO once you enter its size and features. The netlist will be integrated with the top-level design during compilation. The netlist may also be converted to the design in HDL and referenced as a component for simulation.

The connection table and other register functions may also be implemented in synchronous SRAM where addresses are registered and outputs change on clock edges. That allows built-in latency on accesses so that the data in a token field can be shifted out from the Memory Output register while the address for the next is used to access the memory. By this mechanism, the last bit from 1 byte is succeeded by the first bit from the next byte on the next clock edge.

The entire USB interface fits easily into a midsize 3200DX device, leaving resources for the DMA and other host interface functions.

Conclusion

New interface standards are popular targets for implementation in FPGAs. The Universal Serial Bus (USB) is a popular emerging interface standard and is easily implemented in Actel FPGAs. In particular, the 3200DX family offers the speed, capacity, and on-chip FIFO capability required by USB. This application note along with the complete design files and test cases (in both schematic and HDL) are available from Actel. Visit the Actel World Wide Web site (www.actel.com), or contact Actel Technical Support.

A 155 Mbps ATM Network Interface Controller Using Actel's New 3200DX FPGAs

Given that the asynchronous transmission mode (ATM) peripheral market is highly competitive and time-to-market is critical, logic designers must meet shrinking design cycles. Until recently, designers had to rely on gate arrays to deliver the performance and features required for network peripherals. The current generation of FPGAs, however, allows the designer to achieve the performance and capacity required for ATM applications and to meet the key time-to-market goals of these fast-evolving applications.

Introduction

An ATM network interface card (NIC) requires high-speed system logic functions such as DMA controllers, memory (DRAM, SDRAM, VRAM) controllers, FIFOs, and bus interfaces. This paper describes a 155 Mbps ATM NIC controller that can interface to any standard microprocessor bus. This design, in conjunction with a microprocessor controller and a standard SRAM device, is capable of implementing the segmentation and reassembly (SAR) functions for AAL3/4 communications. This includes the control and interface to the ATM physical layer interface (Utopia), a 1 MB dual-port SRAM, and a host bus. The host bus was chosen as a generic bus for simplicity; however, the discussion applies to any of the popular busing architectures such as PCI, VME, EISA, and SBUS.

Controller Architecture

A block diagram of an ATM NIC is shown in Figure 1. The Utopia interface provides a standard data-path protocol for interfacing to physical layer components. A received cell is written into a 64 x 8 FIFO, which can store an entire 53-byte ATM cell, and is clocked at the 25 MHz Utopia clock rate.

The received ATM cell is read from the FIFO at 66 MHz and the cell is written into system memory (dual-port RAM). The dual-port RAM temporarily stores received data packets while they are being reassembled. The RAM space can be organized such that cells with different virtual channel identifiers (VCI) and virtual path identifiers (VPI) addresses are stored in separate memory areas for reassembly. Thus, the RAM can be accessed based upon the VCI/VPI fields. The addressable contents contain pointers to memory space where reassembly takes place. The DRAM controller reads and writes data to the dual-port RAM device.

The first 5 bytes of an ATM cell contain overhead information, such as destination address, and an error control byte. When reading the ATM cell from the receive FIFO, the first 5 bytes of the cell are directed to the header error control (HEC) block. The HEC is an 8-bit cyclic redundancy code (CRC), which detects transmission errors in the ATM cell header.

ATM Cell Definition

Figure 2 shows the definition of the ATM cell. The cell is composed of 53 bytes, with 5 header bytes and 48 data bytes. The header contains 4 bytes of addressing and 1 byte of error checking information. The addressing information is one of two different types, the user-network interface (UNI) and the network node interface (NNI).

The UNI has one significant difference from the NNI. The first byte of the UNI contains a generic flow control (GFC) field, whereas the GFC is not required in the NNI. The GFC designator is used only for traffic traversing the UNI interface, where the operation, administration, and management (OAM) functions are required.

The remainder of the UNI and NNI fields contain VCIs and VPIs. VCIs and VPIs are used to establish connections within the network and can be thought of as the user's connection address. Unlike Ethernet, these addresses are assigned on a connection-on-demand basis, not a fixed number for each user.

The HEC field is an error check field and can also correct for single bit errors. It is calculated only on the header; the 48-byte payload is not included.

Receive FIFO Design

The receive FIFO is implemented in a dual-port SRAM internal to the FPGA. Data from the Utopia interface is written into the FIFO until the entire cell is available. The FIFO is implemented in the dual-port SRAM and uses the synchronous operation mode. The controller provides the data to the FIFO and activates the write enable signal. On the next write clock, the data is written into the FIFO.

Data can be read out of the FIFO independently from the read port because of the dual-port nature of the SRAM. The read port can operate at the required 66 MHz to achieve the required bandwidth on the 32-bit bus.

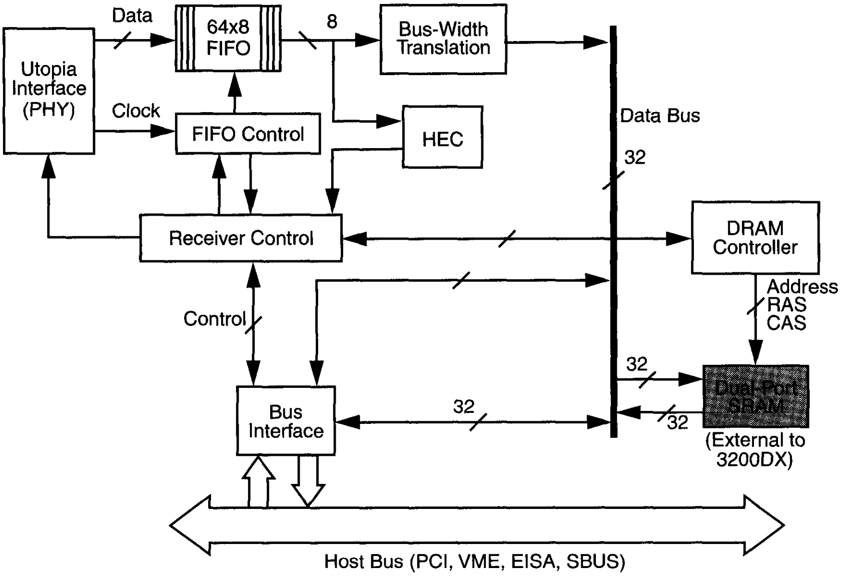
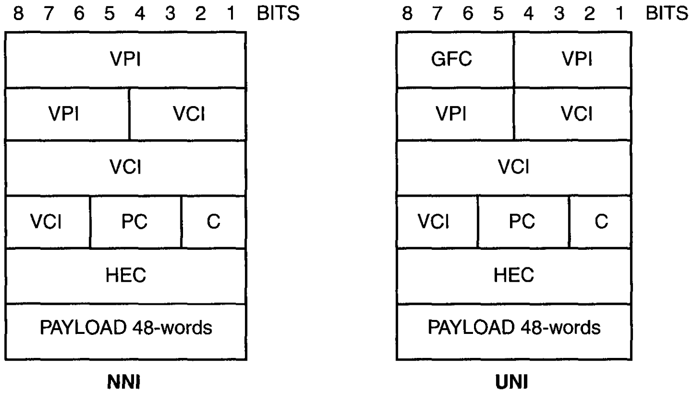


Figure 1 • Controller Architecture



NNI

8 7 6 5	4 3 2 1 BITS	
VPI		
VPI	VCI	
VCI		
VCI	PC	C
HEC		
PAYLOAD 48-words		

UNI

8 7 6 5	4 3 2 1 BITS	
GFC	VPI	
VPI	VCI	
VCI		
VCI	PC	C
HEC		
PAYLOAD 48-words		

Figure 2 • ATM Cell Definition

A variety of FIFO can be automatically generated by the ACTgen Macro Builder tool of the Actel Designer Series software. The user can specify the size, features, and other key functions of the FIFO and ACTgen automatically constructs the requested FIFO "flavor."

DRAM Controller

The DRAM control circuitry is shown in Figure 3. The key blocks of the design are the address multiplexer, the refresh counters, and the RAS/CAS select logic. Timing on RAS and CAS is preserved by using bidirectional buffers on the RAS and CAS select lines, a common technique when interfacing to DRAMs.

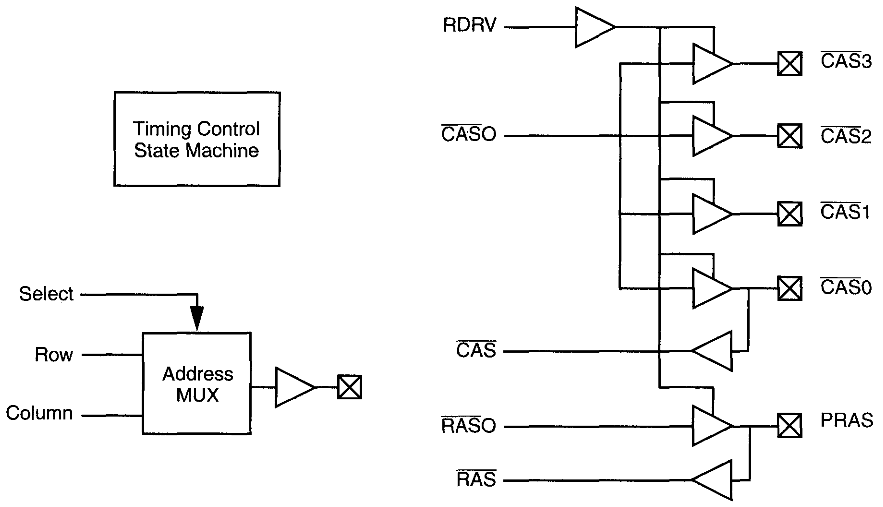


Figure 3 • DRAM Controller Design

A simple DRAM controller is available from Actel as a predefined function for the 3200DX family.

Packet Assembly

Packet assembly is accomplished in the DRAM. The typical contents of the DRAM are shown in Figure 4. Packets are assembled in the DRAM as the data is emptied from the FIFO. A linked list data structure is used to distribute the data to the required packet. Once a packet is completed, the receive buffer link list is updated. The output DMA controller processes the linked list structure in hardware, keeping processor overhead low. The processor can use its processing power on managing the higher-level protocol algorithms because the low-level data transfers are handled in the FPGA.

A simple DMA controller is available from Actel and can be customized by the user to implement more complex transfers. The design uses VHDL to make it easy to customize. Changes to the state machine make possible additional features, like automatic chaining, burst processing, and bandwidth management.

Header Error Control Design

The header error control logic determines if an error has occurred in the header and can correct for any single bit error. The generator polynomial for the header is $X^8 + X^2 + X + 1$ and is implemented with XOR gates, as shown in Figure 5. A parallel implementation of the CRC is also possible. It uses a larger number of components but can compute the full 8-bit CRC in a single clock cycle. A VHDL code fragment is shown in the following display. The X[7-0] bits are the contents of the CRC register, and the D[7-0] bits are the 8-bit data input. The full design is available from Actel and can be implemented in any Actel FPGA family.

```
X[7] <= D[7] xor D[1] xor D[0] xor X[7];
X[6] <= D[6] xor D[0] xor X[7] xor X[6];
X[5] <= D[5] xor X[7] xor X[6] xor X[5];
X[4] <= D[4] xor X[6] xor X[5] xor X[4];
X[3] <= D[3] xor X[5] xor X[4] xor X[3];
X[2] <= D[2] xor X[4] xor X[3] xor X[2];
X[1] <= D[1] xor X[3] xor X[2] xor X[1];
X[0] <= D[0] xor X[2] xor X[1] xor X[0];
```

Processor Interface Design

The processor interface can be a generic high-speed synchronous interface using a block transfer mechanism to keep bus bandwidth high. Since ATM is a packet-oriented protocol, a block-oriented transfer mechanism to the processor memory is most efficient. The DMA controller can be a complex controller with features like chaining and threading, bus throttling, and other bandwidth-optimizing features—or a simple, fixed-size, block fill controller. The optimum design will depend on the processing needed by the CPU and the performance requirements. FPGAs can be very effective at accelerating processing requirements by getting data set up prior to CPU processing. Many algorithms can be significantly sped up if the data is first organized by a smart DMA controller. This allows the CPU to focus on the processing portion of the algorithm.

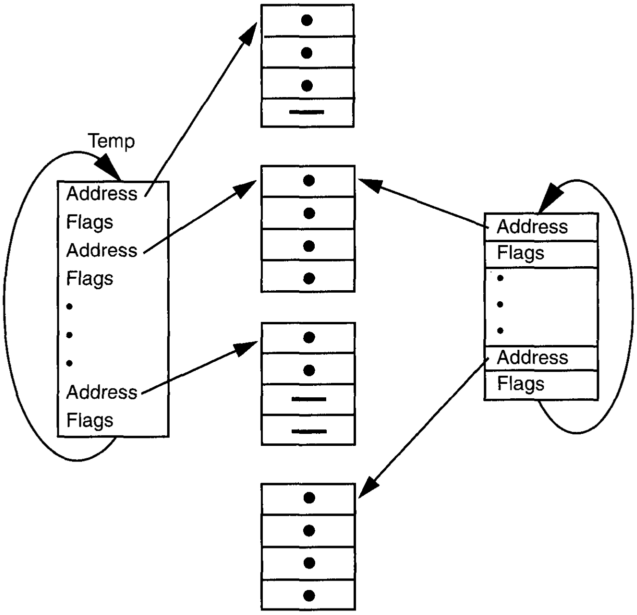


Figure 4 • *Packet Reassembly*

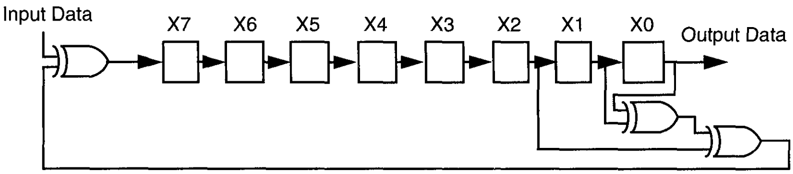


Figure 5 • *Header Error Control Design*

FPGA Implementation

The design can be implemented in an Actel 32200DX FPGA and a single bank of DRAM, and it would operate in excess of 66 MHz internal to the FPGA with a 33 MHz interface to the processor. Actel-provided designs for the FIFO, DRAM controller, and DMA controller help speed the design and ensure that key timing constraints can be met, prior to completion of the entire design.

Applications like this, with complex interactions at the system level, may require additional logic to be designed for use during debugging. The Actel 3200DX family, with its Actionprobe capability, allows the designer to observe all internal signals during device operation, making it easy to identify design errors without needing to change the design.

Summary

ATM network interface cards require high-speed system logic functions such as FIFOs, memory controllers, DMA controllers, and decoders. The 3200DX family, with its system logic integration features of high-capacity, fast dual-port SRAM and wide-decode function, provides just the right mix of capabilities for applications like ATM network interface cards. These system logic functions, along with the predesigned functions for FIFOs, DMA controllers, and DRAM, make the 3200DX family an ideal quick-time-to-market solution.

Memory Page Translation On a Single FPGA

Actel's 3200DX family includes devices that have high-speed dual-port SRAM blocks on-chip. These blocks can be used for a variety of system functions, including FIFO, register files, control store for state machines, and table lookup. Table lookup algorithms are common in many applications like networking, communications, graphics, and computer peripherals. Memory page translation algorithms provide a good example of how a table lookup function can be implemented in Actel 3200DX family FPGAs.

Overview

Powerful workstations have memory systems that support multitasking operating systems. Such operating systems use virtual memory addresses for memory accesses that must be translated into physical addresses. Workstations and mainframe computers employ sophisticated hierarchical memory systems with accesses beginning at the level of a virtual, variable-size memory segment and ending at a physical, fixed-size page.

Memory access requires that a virtual address be translated into a physical address. A memory translation system must first decode that a processor cycle is accessing memory. Once a memory access is recognized, the address translation system must translate the virtual address to a page-level address. At the same time, it has to verify that the page is valid and accessible while quickly applying the physical address to complete the access cycle.

Storing the translations of the most recently used pages in fast SRAM greatly improves main memory access time because the translation requires little overhead. During the page address translation, the bank-select address field can enable the bank, and the page offset can be applied as the RAS address to main memory. The page (CAS) address will be available from the translation buffer when the memory is ready for it.

For efficient address translations, it is important for the translation buffer memory address and data ports to be well integrated with their control logic. Translation memory data must also have rapid, deterministic access times and fast paths to I/O pads. An FPGA integrating logic, dedicated SRAM, and decode is an ideal vehicle for a memory translation system.

Unlike the memory found on other FPGAs, the 3200DX family memory is a dedicated resource. Like a discrete SRAM device, the 3200DX memory provides a guaranteed access time on reads or writes. Integrating logic and memory on one device greatly speeds the memory translation process.

Memory System Organization

Memories in sophisticated, multitasking computer systems must store and keep track of data being used by different processes. The operating system has to allocate memory so that a process does not overwrite an other's space.

The operating system kernel and each process are assigned to a portion of memory called a *segment*. Segments are large areas of virtual memory of variable sizes. Individual segments can be subdivided into fixed-size physical pages. Segmentation allows process address space to be dedicated to procedure or data. It also allows application of access control to prevent undesired interaction between the data of different processes.

CPU addresses are made up of fields such as those shown in Figure 1. The MSBs of the address point to a register file containing segment description data. The middle field contains the virtual page address, which must be translated, and the LSB field is the physical page offset.

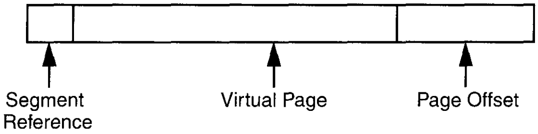


Figure 1 • *CPU Memory Address*

As the CPU address is issued, the segment description file and translation buffer are addressed simultaneously. The fields of a segment description are shown in Figure 2. The lower fields tell whether the segment is used, paged, or present in the translation buffer, and they indicate the segment's bank address, access lever, write level, and page table base.

The segment reference can be stacked in two register file locations such that all the fields, except the page table base, are accessed first and the page table base is read subsequently. The latter field is used only if the data in the earlier fields qualify the translation and the translation misses.

Should the data in the first segment description access abort because, for example, the segment is invalid, the information is passed immediately to the CPU. Otherwise, the translation begins with the virtual page address used to access the translation buffer. At the same time, the bank address is passed to access the bank in main memory. The highest level of memory reference is the bank address. Banks can contain segments, but a segment cannot be spread across bank boundaries.

A translation buffer entry is shown in Figure 3. It contains the physical address of the page and some bits indicating access privilege levels. There is also a field to compare the segment with the register file segment reference and (in some systems) with some of the upper bits of the virtual page address. A comparison failure means that the translation entry is invalid and should be updated.

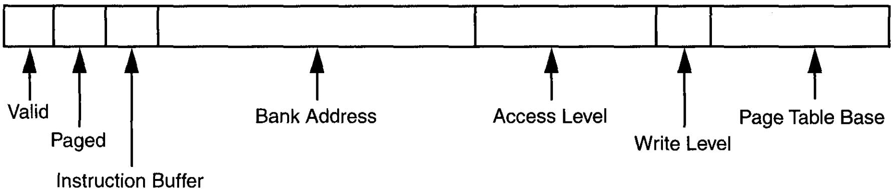


Figure 2* • *Segment Description

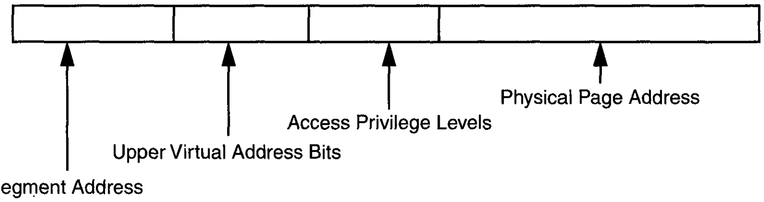


Figure 3* • *Translation Buffer Entry

When there is a translation miss, the page table must be addressed to fill the buffer line with the correct physical page address. The page table contains the physical address of all the pages.

The operation, shown in Figure 4, can be performed by shifting the segment page table base up some number of bits and adding it to the virtual page address (generally shifted down for positioning relative to the page table base) from the original CPU reference address. The sum is used to address the page table, which yields the physical page address. That address is then used to complete the original CPU memory request and fill the line in the translation buffer for the next reference to the page simultaneously.

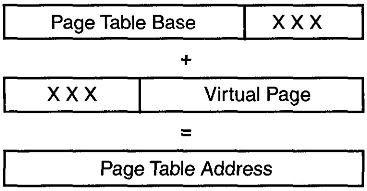


Figure 4* • *Page Table Address Calculation

Operation

The architecture of the address translator is shown in Figure 5. The most significant address bits drive the register file to access the segment description while also accessing the translation buffer. In some systems, the highest bits can be used by a comparator to determine whether the CPU address is a main memory reference at all. The comparator output enables the storage element in the translation system.

The translation controller must consider the various status fields from the description and the privilege field from the buffer to see if the main memory access is allowed. It must also note the output from the comparator of the segment address field from the CPU with that found in the translation buffer line to ensure that the buffer entry is valid. Should these checks prevent the memory access, the controller passes the access status back to the CPU.

If the buffer entry is invalid, an adder sums the page table base and the page address from the CPU to retrieve the page address from the page table.

Main memory accesses are complete when the translation completes successfully, when the bank address from the segment description is concatenated with the physical page from the buffer, and when the page offset from the original CPU address is concatenated with the address in main memory.

Memory

The 3200DX family devices contain a number of features, in addition to logic and I/O resources, that are used in the translator design. The 32300DX, for example, has 12 SRAM blocks, each of which can be configured into a 32 x 8 or 64 x 4 memory during device programming. In addition to the individual block address lines, memory blocks are four-way cascadable in depth without additional logic. Width cascading requires the use of a multiplexer on data reads. Four-block depth cascading uses up to two of the existing enable signals (BLKEN and WEN) on the SRAM block. Additional cascading is also possible, but, due to the placement of the memory block on the device, it would entail dispersion of the address and data, and therefore different delays.

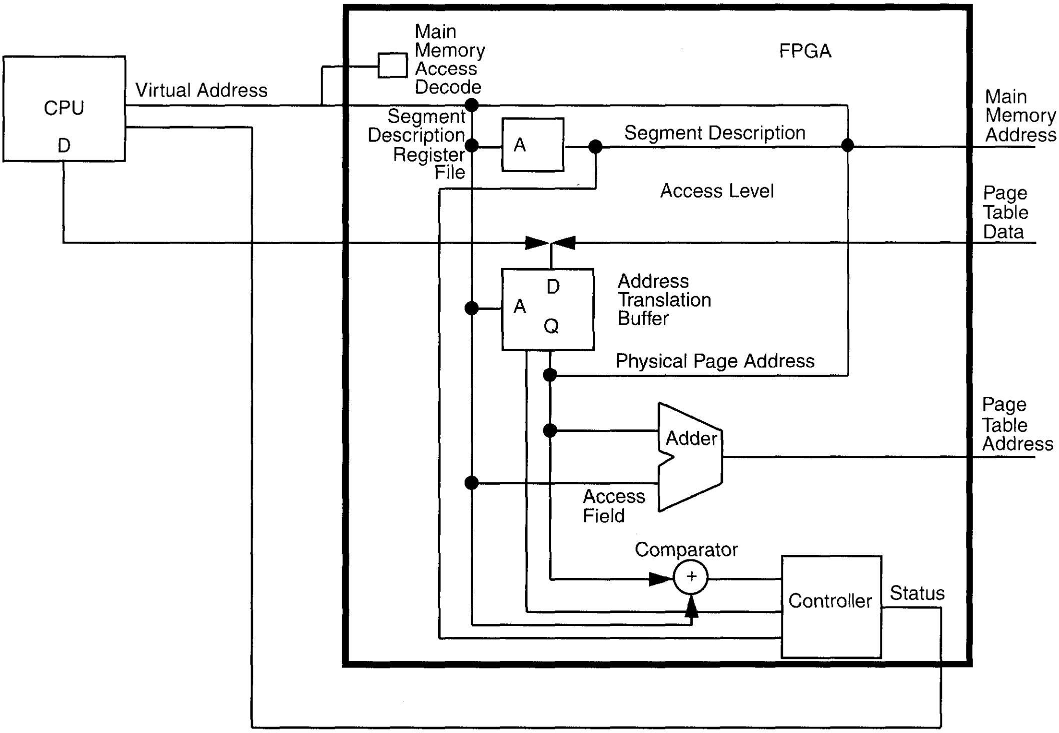


Figure 5 • *Address Translator Architecture*

The memory is dual port, with separate address and data ports for reads and writes. The separate ports allow for simultaneous reads and writes.

Writes are synchronous with the write clock (WCLK). The memory can be operated in synchronous or asynchronous read mode, selectable at device programming. In synchronous read mode, data is registered internally and remains valid on the SRAM outputs until the assertion of the next read enable (REN) and read clock (RCLK) signals. Asynchronous reads require only the address and the read enable. Unlike the memory resources offered on other FPGAs, the 3200DX memory has guaranteed access times regardless of the width or depth of the configuration.

Depending on the system requirements, the SRAM resource can be used in different ways. The memory could be entirely devoted to the translation buffer of the dimensions 128 x 32 or 256 x 16, for example, where the width requirements of a translation entry would determine the number of entries possible. Another alternative would be to use two blocks to store sixteen segment descriptions of four bytes each and to use the remaining SRAM resources for the buffer. Other systems, as mentioned earlier, might divide the buffer into kernel and process sections where the entire contents of the process side are flushed on a context switch.

Dedicated Decode

The family also features dedicated decode logic modules. The 32300DX, for example, has 32 such decoders. Each decoder has seven inputs and a polarity-selectable output. The decoders are located around the periphery of the device, allowing fast routing from inputs. Alternatively, the decoders can be driven from internal AND gates to expand their widths up to 35 inputs. Decoder outputs are available to internal logic and have a hardwired fast path to an output pad.

Implementation

The sample design shown in Figure 5 uses a register file made from a segment descriptor with a single translation buffer. A decoder qualifies the CPU address as a main memory access and enables the register file and the translation memory. Two clocks are required to access the entire description. On the first clock after the CPU address is issued, all the description information is available, except the page table base.

During the initial clock, data from the description and the translation buffer is used by the controller either to abort the access and return status to the CPU or to complete the access. If the access cannot proceed because the buffer line is invalid, then on the second clock, the calculated page table access is used to address the page table. Data from the table is written to the translation buffer by the controller at the same time it is used as part of the address to main memory.

The controller is a conventional one-hot state machine. The segment address comparator can be implemented with the dedicated decoder modules. The adder can be synthesized, or it can be a modified library macro.

Conclusion

Actel 3200DX FPGAs contain on-chip high-speed SRAM blocks that can be used to implement a variety of table lookup algorithms. A memory translation example was used to demonstrate this technique. Details on this design are available from Actel.

Customer-Authored Application Notes

Section 6: Customer-Authored Application Notes

Customer-Authored Application Note

HDL Methodology Offers Fast Design Cycle and Vendor Independence

Joseph Cerra, Senior Design Engineer
Wellfleet Communications Inc.

In the highly competitive data communications field, the ability to bring a product to market quickly is essential for success. Using FPGAs with an HDL methodology offers this fast time to market by providing the flexibility to design, debug and verify, all within the same environment. In addition, the use of HDL allows the relative ease of re-targeting a design between vendors/technologies.

The design consists of a complex multiple port DMA controller for a data communication card that resides in a high speed router environment. The objective of the design was to control the flow of data from two TI TMS380 token ring controllers to a proprietary high speed bus interface utilizing FPGA technology and high speed FIFOs. The design contained two DMA engines, one that controlled the data flow to and from the TMS380's and the other that controlled the data flow to and from the bus interface. The data within the FIFO was monitored by the use of multiple address descriptors and transfer counters within the FPGA.

During the course of the design process, several FPGA vendors were considered and implemented based on the speed, size and architecture of their product offerings. Additionally, there was a future consideration of converting the design to a conventional gate array. By carefully writing scripts files and a "Verilog Wrapper" the core Verilog code itself was "untouched"(i.e., vendor independent) while taking optimal advantage of the architecture offered by the various vendors.

The design process itself had three steps to achieve the desired target. First, the core Verilog code was written and simulated at the behavioral level to ensure circuit functionality. Once this process was completed this would become the baseline that would remain unchanged throughout the design process to ensure vendor independence. Second, Synopsys script files were written based on the vendors architecture to keep reasonable constraints on parameters such as clock period, area, clock trees and internal delays. These constraints would keep logic levels and excessive fanout in check. Third, a "Verilog Wrapper" was written around the Verilog core to take advantage of architectural differences in the pad logic offered by the various vendors.

At the beginning of the design cycle it was estimated that the target design would need approximately 8000 gate array gates. A gate was agreed upon to be a four transistor two input NAND gate (gate array equivalent) and not a FPGA equivalent gate. The desired clock speed of the design was 16 MHz. The toolset for the design was Verilog along with Synopsys FPGA compiler version 3.2 all running on a Sun Sparc 20 workstation. The first FPGA chosen was a Xilinx XC4025 based on its vast amount of gates with additional benefits of re-programmability and on-board RAM. After working on the design for several months with help from the vendor's application staff, it was realized that the implementation of this particular design would only yield a clock speed of about 12.5 MHz.

The benefits of this design flow were realized when a second FPGA vendor was selected. As stated before, the short design cycle in this highly competitive market required that the vast amount of work done on the first FPGA be transferred to a second FPGA as seamlessly as possible. The second FPGA chosen was the Actel ACT 3 A14100 because of its fine grained and "Synthesis Friendly" architecture. The Actel part had an additional benefit of flip-flops in the pad ring. The I/Os were hand instantiated ("Verilog Wrapper") to take advantage of these features. The Synopsys script files also needed to be modified to take advantage of technology specific features like various clock drivers and inter-connect delays.

When the core Verilog was about to be written, the choice was made to use Verilog because at a high level of abstraction the user can conceptually design a system without regard to a specific technology. There was also the future consideration of turning this design into a gate array when the volumes ramped up and it was desired to keep the core Verilog as stable as possible. During this design it was discovered that it was not necessary to select the target technology before the system design was fully functioning and simulated through high level Verilog simulation. The desire was to avoid technology specific code. This would allow the design to be migrated from one technology to another without core Verilog changes. The core Verilog was written as generic as possible in hope that the design tools would ultimately make smart technology specific choices.

In general, technology mappers want one thing—small homogenous building blocks. It is for that reason that most ASICs are mapped reasonably well when it comes to speed and density. The basic building block is typically an equivalent of a 2 input NAND gate. Also, the ASIC interconnect is a metal to metal "via" that doesn't represent a significant amount of circuit delay. Therefore the mapper can produce less than optimal solutions and the ASIC technology will be much more forgiving. FPGAs, on the other hand are much less forgiving. This FPGA design pushed both the speed and density envelope and required two additional steps: The first was the Synopsys script file, the second was the hand instantiation of the complex I/Os.

The power of the Synopsys compiler is it's flexibility through scripts. The core Verilog code need not be modified to change a design from a small, medium speed, compact design to a much faster but larger design. This was accomplished through design constraints. In this design two types of constraints were set for each of the chosen technologies: "design rule" and "optimization." In general, "design rule" constraints reflect technology-specific restrictions that must be met for a functional design (such as maximum loading on a net). "Optimization" constraints represent design goals that are desirable, but not crucial to the operation of a design (such as maximum circuit area or delay).

The Synopsys Compiler tries to meet both types of constraints with an emphasis on "design rule" constraints as they are requirements for a functional design. The Synopsys Compiler uses these constraints to guide optimization and implementation of a design. Constraints define the goals of the synthesis process. The constraints for the technology-specific target can be put into a script file to help keep the core Verilog code untouched.

The amount of constraints that can be put into a Synopsys script are virtually unlimited. However, with the described design task in hand, the following assortment of constraints were found to be quite useful. The "max_area" constraint specifies the maximum allowable area for the current design. When the Synopsys Compiler sees this "max_area" constraint it computes the area of a design by adding together the areas of each of its components on the lowest level of the design hierarchy. The area of a cell is obtained from the specific technology file. The constant put after the "max_area" constraint represents the total amount of cells available in the target part.

The most important optimization constraint is maximum delay (max_delay). There are four types of delay categories: Clock to Q (Tcq), Set Up (Tsu), Clock to Out (Tco), and In to Out (Tio). Since the I/Os were hand instantiated, the only delays of concern were internal synchronous paths (register to register). The "max_delay" constraint was carefully set so the longest path from clock to out of one register, through logic to the input setup time of the next register was less than the target clock period of 62.5 ns (16 MHz). The Synopsys Compiler has a built-in static timing analyzer for evaluating timing constraints. A static timing analyzer calculates path delays from local gate and interconnect delays but does not simulate the design. That is to say, it does not check the design for functionality. The Synopsys Compiler timing analyzer performs critical path tracing to check minimum and maximum delay for every timing path in the design.

The last constraint that was found useful was the "dont_touch" constraint. FPGAs have special features like high drive resources (e.g., CLKBUF macro) used for driving high fanout clock nets and special complex I/Os that were hand instantiated in this design. To prevent the Synopsys Compiler from breaking up and adding buffers to the clock net the "dont_touch" constraint was added to this net. After instantiating the I/O cells, the "dont_touch" constraint was added to keep these elements intact.

The final step of the design process was to hand instantiate the complex I/Os. The idea of this final step was to hand instantiate the I/Os and wrap them around the simulated core Verilog. This is known as a "Verilog Wrapper." When it came time to change technologies, it was a simple matter to change "wrappers". The I/O ring was manually instantiated and connected to the top-level of the core Verilog code. When the "Verilog Wrapper" was written, "Direct Instantiation" of the complex I/Os was used and the port order was automatic. For example, when it came time to instantiate a high speed I/O flip-flop like the BRECTH cell (shown in Figure 1) from the Actel ACT 3 family, it was instantiated as follows:

```
BRECTH
I2(.D(<n1>),.PAD(<p1>),.E(<n2>).Y(<n3>),.
CLK(n5>),.IOPCL(<n6>));
```

When all the I/Os were instantiated, the "dont_touch" constraint was added in the Synopsys script file.

When all the above mentioned steps were performed, the design was smoothly transitioned from one FPGA technology to another and the desired speed of 16 MHz was achieved. The next step of this process is to convert the design to a conventional gate array. Although this process has not yet begun, it is certain that by following the process described here, the transition will go smoothly.

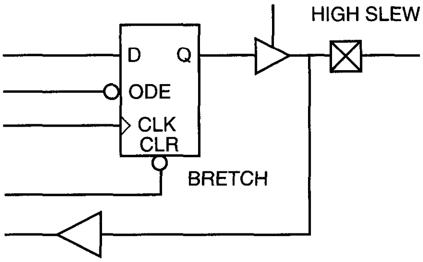


Figure 1* • *BRETCH Actel ACT 3 I/O Macro

Actel

Customer-Authored Application Note

Using FPGAs for 100 Mbit/sec Imagesetter Application

Thomas A. Everett, Electronic Design Engineer
ECRM Trust

In the magazine and newspaper industry, imagesetting speed is critical. Ten years ago, images were outputting 12 inches at 1016 dots per inch (dpi) in 5ms resulting in a peak data rate of 2.4 Mbits/s. Today's imagesetters output 18 inches or more at resolutions exceeding 2540 dpi. Peak data rates of 100Mbits/sec are now necessary.

The speed of imagesetters as well as many other business machines follows the trend of computers, lower cost, and higher speed. The FPGA facilitates the design in both regards. The cost of a design is directly related to the complexity, and increased speed adds to the complexity of producing a reliable printed circuit board (PCB). So any tools or devices that can aid the engineer in completing a complex design are welcome. Designing with the FPGA is made easier with the use of new computer aided design tools and speed is only a matter of selecting an FPGA that meets the design requirements. A wide variety of FPGA parts are available whether the speed requirements are for fast pin to pin delays, or fast internal clock and propagation delays.

FPGAs allow designers to continue meeting the high speed requirements without switching to GaAs or ECL technologies. Wide counters, registers, address, and data paths can be the most difficult to implement in a fast design. Incorporating these critical functions (such as a 16+bit counter running at 100 MHz) on one IC eliminates most of the more difficult high speed design issues usually encountered at the PCB level. Design issues such as: clock distribution, propagation delay, transmission line effects, and power supply noise are all significantly reduced for an FPGA design. If a design can be implemented with clear and predictable performance the amount work required between design and production is minimized.

With continued shrinking time to market cycles and the need to remain competitive, logic designers are constantly looking for methods to maximize the performance of current technologies. One way to do this is to make use of the FPGA, incorporating large amounts of circuitry into one package. Reductions in PCB size and power consumption can also be realized when designing with the FPGA. Programmability makes the design flexible, the tools make the designs easy to implement, and the current speeds of the FPGA make it a logical choice for high speed designs.

FPGAs were first used at ECRM in the original design of a 50 MB per second imagesetter. At the time the major purpose of the FPGA was to consolidate the design onto a single PCB of reasonable size and cost. Speed was of secondary concern as host computers could not support much more than 50 MB per second even if the imagesetter could. Faster systems were much larger and required high performance dedicated host computers to support the higher data rates. In time the speed of host computers would increase and the cost would decrease. One only had to watch the trend of computers to know the future of all related equipment.

The new imagesetter design did not require an excessively large number of gates but did have a speed requirement that was originally thought not to be possible with existing FPGA technology. Originally the design used a 50 MHz data path and a 100 MHz synchronization circuit. The new design was specified to be 2 times the speed of the original. This meant that the data path would be 100 MHz and the synchronization circuit would be 200 MHz. Since 200 MHz was not possible with any existing CMOS technology the alternatives (ECL or GaAs) were not very attractive. This portion of the circuit would definitely have to be replaced. The data path was considered optimized for a theoretical 60 MHz using a 75 MHz FPGA. If the same ratio applied to the new design the data path would need a part specified to 160 MHz, also not available in CMOS at the start of this project. Only preliminary data was available for parts approaching that speed. If only we could make the data path run at the 105 MHz rate.

With Actel being the source of the original data path design we decided to reevaluate the possibility of using Actel parts in the new design. The highest speed grade of the size part required was 105 MHz. All we had to do was fit the existing design into the new speed grade and be able to achieve better than 95% of the manufacturers specified speed limit. Timing measurements showed the original design would not run at the rated speed and would have to be redesigned, or replaced. The data sheets as well as some very helpful application notes in the Actel data books indicated the design could be made to operate at the rated speed of 105 MHz. We decided to test the manufacturer's specifications and see how fast we could make the 75 MHz part operate. If we could get the critical parts running at the maximum speed in the standard parts we would gain the confidence needed to pursue the design using the highest speed grade.

Paper published in the 1995 PLD Conference Proceedings

There were many parts in the data path that required tweaking for speed but three in particular needed a new design approach; a 16 bit up counter, a 4 bit counter/synchronizer and an 8 bit shift register. Functionally each piece did not require a great deal of thought. It was the speed requirement that made the design more difficult. The two counters had to be capable of running at the limits of the rated speed without fail even under a battery of stress tests. The circuit would have to be subject to extreme variations in both power supply rails, and operating temperatures and still perform to the design specifications. With these limits in mind we made or first attempt at optimizing the critical functions.

We used the Viewlogic schematic capture package to enter the design and through the combination of Actel's design tools and Viewlogic's simulation we were able to simulate the design timing. After a few attempts at simulation we felt we had a design that would perform as required. System tests at this point did not quite reach the limit of the device specification but were far beyond the speed attained in the original design. After a more detailed performance analysis we determined there was still some room for improvement. Logic levels where reduced further and routing delays where trimmed using redundant logic. Being extremely thorough in the analysis and maximizing the capabilities of all the available tools proved to be worth all of the effort. One more simulation and we were back to system testing. This time we were able to run the standard speed device right up to the specified limit of 75 MHz. Under high temperature and low power supply voltage the device performed reliably right up to the maximum rating.

The counter is a 16 bit up counter with synchronous load, and clear. It is primarily based on the application note "Implementing Load Latency Fast Counters with ACT 2 FPGAs", Actel FPGA Data Book and Design Guide, 1994, page 9-43. The two LSB's of the counter are the fastest part of the design as they must toggle at the highest rate. Therefore the choices for implementation are limited. Also if the counter is more than four bits the lower two must be duplicated to keep the routing delays at a minimum. The Q0 bit must have feedback from the Q output with zero levels of logic to be able to toggle at the period of 1/fmax. The feedback becomes a select line to a mux that puts the inverted logic level at the D input to the flip-flop before the next clock edge. Actel provides macros that implement this type of function with effectively zero levels of logic needed for the mux portion. The toggle rate is then defined as; the Tsu (setup time of the flip-flop) plus the Tco (clock to output delay of the flip flops pad to pad).

With a high speed data path design in place all that was left was to fit the design into the higher speed grade part. It was now time to design the clocking and synchronization circuit for without a high speed clock we would not have a high speed design no matter how fast the parts we used.

We came up with a method of synchronizing the system with a 1x clock instead of the 2x or 200 MHz clock. For this we chose a device that had faster pin to pin delays as we would have to use feedback around the part for synchronization. After prototyping the design we were ready to test the performance of the 105 MHz design. System tests now proved the performance of the design right up to the maximum 105 MHz. Timing simulation results closely matched the manufacturer's specifications and the actual performance matched the simulation. Current FPGA technology is capable of performance levels that can satisfy almost any requirement. A system design engineer need only be aware of the products that are available and be able to determine when and where to apply them.

Biography

Thomas A. Everett, Electronic Design Engineer, ECRM Trust, Tewksbury, MA. I have been with the engineering group at ECRM for nine years helping to develop imagesetters for the newspaper and magazine industry. I have contributed at least in some part to all aspects of design work at ECRM including, optics, mechanics, digital and electronic design. I have most recently been assigned the task of increasing the speed of existing product lines to keep up with todays customers high speed requirements.

Customer-Authored Application Note

Design of a High Speed Communications Link Using Field Programmable Gate Arrays

Amy Lovelace, Technical Staff Engineer
Alcatel Network Systems

Introduction

A communication mechanism has been developed using two 8000 gate Field Programmable Gate Arrays (FPGAs). This mechanism was developed to provide high speed, serial communication between shelf processing units in a hierarchical control system. The mechanism provides the higher level processing board with a virtual memory link to the lower level processor boards. The link provides memory-to-memory transfers between the two processing tiers. The FPGAs are master/slave devices which provide the physical link across the backplane and access to the local memory on each unit. Multiple types of transactions can be sent across the link from the master to one or all of the slave devices on the bus. Each FPGA contains internal registers used to configure the devices for the desired transaction, a mechanism to do DMA transfers to and from the unit's memory, and a mechanism to transfer and receive serial data over the link. This paper concentrates on the board where the master FPGA resides, Satellite Processor (SPB).

System Description

SPB is a second tier module that provides the communication interface between the top level Processing System and the third tier Processor Elements for I/O modules and for matrix modules. The satellite processor is the second-level controller that resides in each shelf of the SI48 end stage and center stage shelf. It functionally provides the necessary Operation, Administration, Maintenance, and Provisioning (OAM&P) and protective switching. It distributes control to the third level (smart circuit cards) and controls the functions required of the dumb circuit packs, such as the Internal Protection (IPB 101) board. Figure 1 is a typical 1631 SX application system diagram.

Functional Overview

Figure 2 is a functional block diagram of the satellite processor. Each component is briefly described in the following paragraphs.

Central Processing Unit and Memory

The main processor is a 32-bit embedded microprocessor. A multitask real-time operating system handles the primary processing functions. The processor controls all functions and supports memory read/write access, fault detection, and interrupt control. The processor contains three types of memory: EPROM, local RAM, and system parameter RAM. The flash EPROM has asynchronous access via 16 bits of the available 32-bit data bus. It is used for power-up operations. Local RAM is addressed at 20 MHz over a 32-bit address bus, and data is exchanged over a separate 32-bit data bus. Local RAM includes real-time operating system, communications driver, application program, application data, and exception vector. The SPB has two Communication Controllers (CC) with system and parameter RAM. The main processor accesses both CCs via 16 bits of the available 32-bit data bus. Each CC has an associated parameter RAM for internal use.

Communication Controllers

The Communications Controller each provide a redundant HDLC interface (A and B) to the lower level processors and a redundant ACL bus to the top level processor. This includes fault recovery software that uses the other CC as a communication path to check the database of a failed satellite processor to reconfigure the new active partner processor if a failure occurs. CC1 and CC2 are initialized in the asynchronous mode to provide debug capability via a front-panel connector. Also, an HDLC communications link exists between the partner processors for data synchronization.

A system Integration Block (SIB) is part of each CC. They are called SIB-CC1 and SIB-CC2. SIBs control the following functions:

- Parallel I/O ports
- Timer
- Watchdog timer
- DMA interrupt controller
- Chip-select lines and wait-state generator logic
- On-chip clock generator with output signal

The SPB contains up to 12 general purpose inputs and 13 general purpose outputs, including the Card Presence Indicator (CPI), which is used to control dumb boards and provide alarm detection or activation of service control. The

Paper published in the 1995 PLD Conference Proceedings

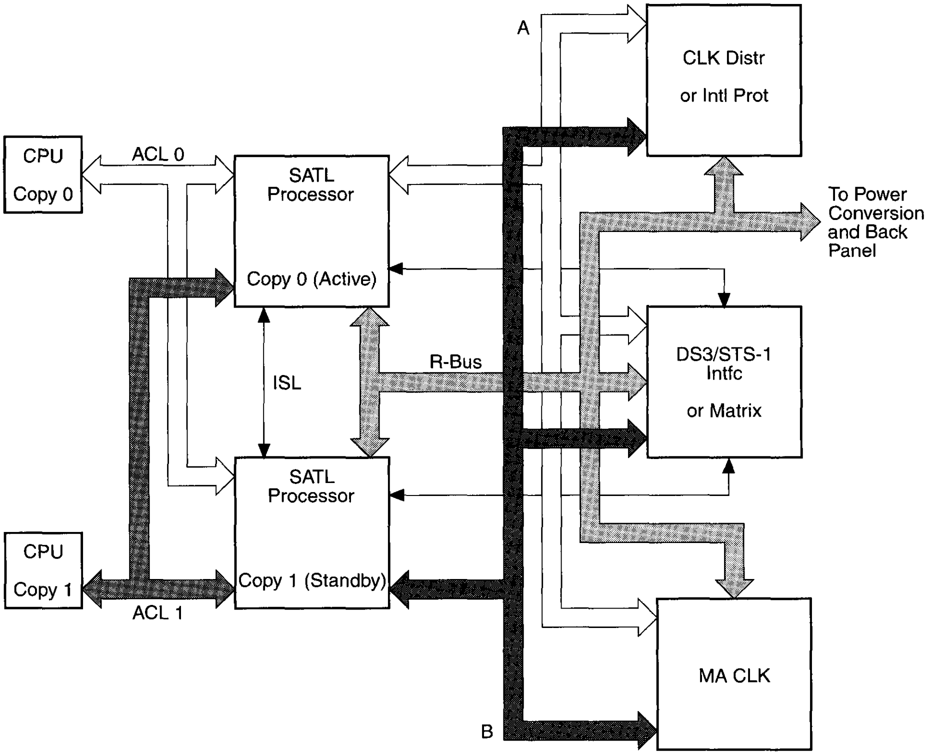


Figure 1 • System Application Block Diagram

CPI allows the top level processor to poll a missing satellite processor. This allows the top level shelf to determine if any satellite processors are not responding with an out-of-state switch failure or fuse failure response.

Arbiter

The bus arbiter prioritizes CC1s and CC2s Serial Direct Memory Access (SDMA). The arbiter allows multiple SDMA transfers to be multiplexed onto the common processor bus to access local RAM and different system and parameter RAM. It also allows Independent Direct Memory Access (IDMA) from the CCs and from the master FPGA.

CPU-CC Bus Adapter

The CPU-CC bus adapter logic circuit ensures full local RAM access for SDMA and IDMA of the CC controllers. It provides a 16- to 32-bit-wide bus adaption. The master FPGA has a 32-bit-wide interface to the main processor.

Interrupt Request Handler

The interrupt request handler responds up to 48 service request sources that are generated on the various user boards within a twin shelf. Any interrupt request within a shelf group of 24 board elements generates a fast board protection switch.

Memory Transfer Controller

The Memory Transfer Controller uses two redundant buses (copy 0 and copy 1) to access user boards and support protection switching. The Memory Transfer interfaces bus interfaces with third tier processors in the I/O and matrix shelves. This bus is a high-speed serial communication interface between the satellite processor and transmission circuits packs. The bus provides serial DMA transfers for a data rate faster than the HDLC link. This bus is used primarily for protection switching in the event of a system failure.

IIC Controller Section

The Remote Inventory link (called the RI-bus) provides access onto the remote inventory data bank (256- by 8-bit Electrically Erasable Programmable Read Only Memory [EEPROM]) implemented on each circuit pack, such as

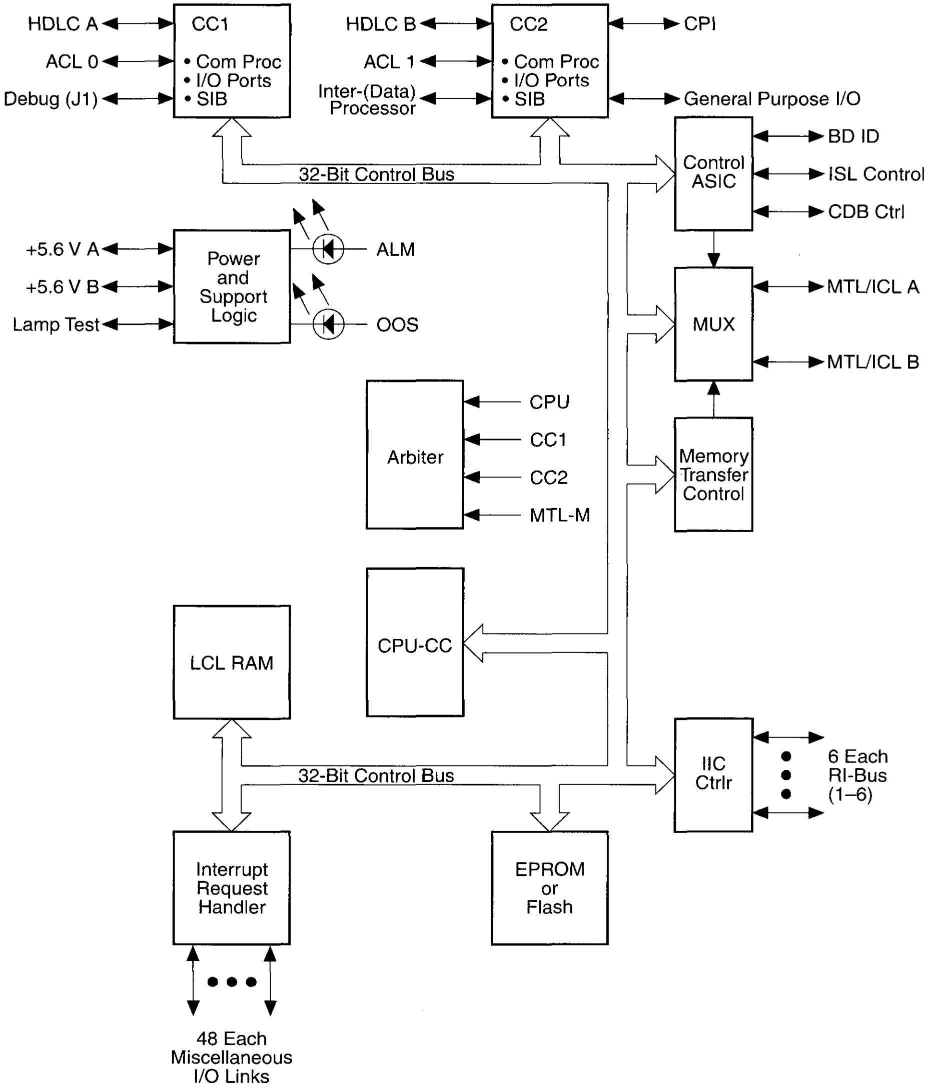


Figure 2 • SPB Functional Block Diagram

power converters, back panels, and user boards (with or without processors). Each SPB has one controller for each of the six bus blocks on the module. The IIC controller has an 8-bit processor interface and provides the R-bus master and slave function. It controls all sequencing, protocol, arbitration and timing.

Power Supply Circuit and Support Logic

The processor circuit pack is connected to two redundant 5 V power converters (copy 0 and copy 1). Each is C-sourced and fused. A reset circuit monitors the voltage at the input. In the event of low voltage, the ALM indicator lights and a power-on reset is generated. A blown fuse on the processor does not cause loss of service in the twin shelf.

The support logic provides the functions to access memory, level shifting to external interfaces and voltage supervision.

FPGA Implementation

Figure 3 is a block diagram of the circuitry implemented within the master FPGA. This FPGA is an Actel A1280-1 in a 160 pin Plastic Quad Flat (160PQFP) package.

The A1280 is an 8000 gate device which provided the design group with the capacity to fit a large amount of circuitry in a single chip and a small board space. Also, the low power consumption of the A1280-1 was an attractive feature. The slave FPGA on the third tier processing card is another A1280 with similar functionality.

The FPGA implementation started with drawing the schematic of the circuit using Viewlogic's schematic capture tool. Another tool used during FPGA implementation was ACTgen. ACTgen is a computer-aided engineering (CAE) tool, included with Actel's Designer software. With ACTgen's graphical user interface, we were able to build structured macros (counters, adders, etc.) by simply clicking on a few menu choices. ACTgen then creates functions that are optimized for Actel architecture. The shift registers and the 24-bit counters in this design were generated with minimal effort and no simulation effort using ACTgen.

Once all the blocks were generated and the top level hierarchical schematic was drawn, a functional simulation verified correct functionality of the design. The design was then netlisted into Actel Design Language (ADL). The pin assignment for this design was fixed before the FPGA design was initiated. This meant that the Actel's Automatic Place and Route program was not allowed any flexibility to move the pins around. Additionally, the A1280 was used to its near maximum capacity at 98% utilization. Nevertheless, the Automatic Place and Route program completed the FPGA layout. Once the chip layout was finalized, the delays were extracted and a post-layout simulation was performed. The same test vectors used in functional simulation were used in the post-layout simulation. The required system speed of 25 MHz was easily achieved and the FPGA was sent to be programmed for production.

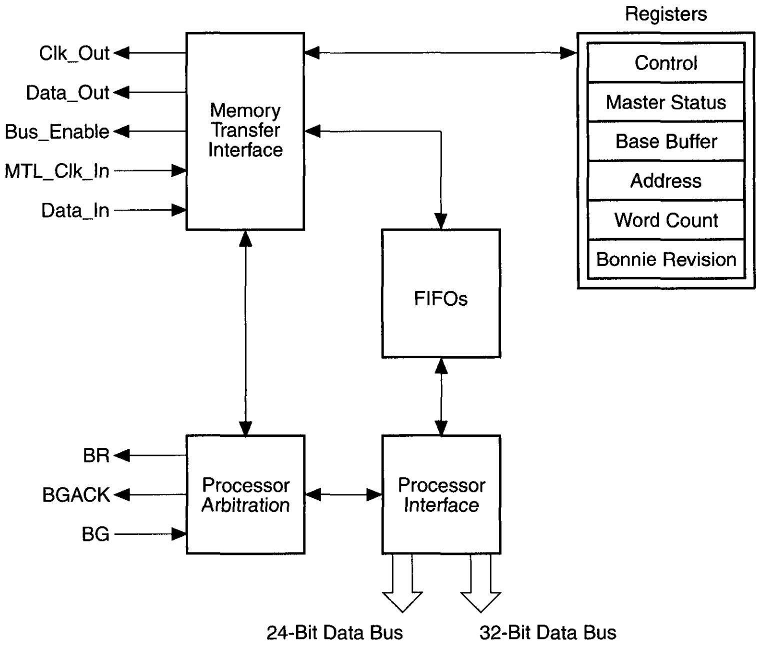


Figure 3 • FPGA Implementation Block Diagram

Conclusion

The Memory Transfer link was designed to provide high speed communication between processing units in a control system. FPGA technology was used in this project which provided a number of advantages. The density of the FPGA allowed considerable savings in board area. Most importantly, the FPGA provided tremendous flexibility for the design group. The FPGA chip layout was not even completely defined when the SPB board was layed out. A1280 allows this flexibility due to its architecture and abundant routing resources. Even at 98% utilization and with locked pin assignments and a few design change iterations, the FPGA was placed and routed with no problems.

A 256 Channel Control System Using FPGAs and a PLD

Dave DeLauter, Consultant
DeltaT

This paper describes the development process and the latest design iteration of a simple Pulse Width Modulation (PWM) Digital to Analog Converter (DAC) system. From an original micro based design to the high speed serial implementation and the extensions described in this paper, the design seems to continuously evolve with each step opening new possibilities. The unique flexibility of PLDs and FPGAs allows the design to have the capability of implementations from a few 16 bit PWM DACs in an FPGA up to 256 DACs on a printed circuit board. It also allows the use of more DACs of less resolution per FPGA such as 8 bit or even smaller. Using a fast PLD for the microwire-like interface and then dispersing data to up to 256 DACs implemented in slower FPGAs is the current level of evolution.

Background

Since we knew that PWMs would work in our application, our original design called for using a micro and its PWM capabilities to control few analog signals. As the design continued we found more places where PWM control would be adequate and would allow software control of sections of the product that were not even considered earlier. This additional level of control would allow software customization of each implementation while increasing the reliability of the system.

As the number of possible PWMs being considered grew, we also decided that it would be useful to drop the micro from certain implementations of the design and use a parallel PC port interface with the PWMs in an FPGA. When we approached our customer with this idea, he reacted with more possible applications which would include as many as 256 PWM DACs.

Design

The design was then changed to have one "master" FPGA controlling the parallel port and distributing data to as many "slave" FPGAs for the PWM DACs as the implementation required (Figure 1). In some previous designs we had used PWM DACs in an Actel 1280. In this case we decided initially to put as many 12 bit PWM DACs in an Actel 1020 as possible and then clone that part several times on the board. This structure was developed for the prototype. The actual design for PWM DACs is well known as is the design of various parallel port interfaces. The use of the multiple 1020s to add as many DACs as needed for a particular customer application allows us to use one printed circuit board for this design. The Actel 1020s change to support specific implementations.

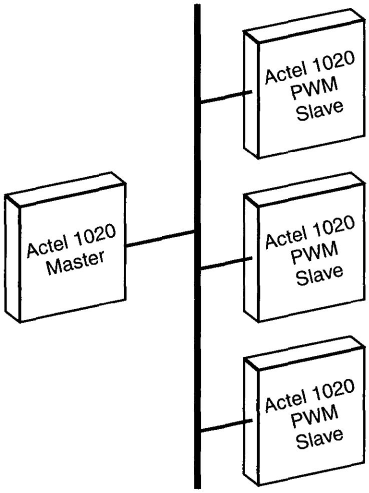


Figure 1 *• Parallel Port Diagram*

PWM Design

As indicated earlier there is nothing unique about the PWM structure we are using. Our PWMs consist of a counter, a latch, a comparator, and a flip flop. The latch data is compared to the counter on each clock cycle and when they are equal the flip flop is set (Figure 2). The flip flop gets reset by the carry out of the counter and the cycle begins again. In the Actel implementation, setting the flip flop drives the enable of a tristate pin that has ground as its input. This pin in the simplest case drives a resistor-capacitor combination to convert the asymmetry of the waveform into a DC level with a 50 percent duty cycle being the nominal midpoint. The latches we use are reset to a 50 percent duty cycle for convenience. With this structure, a zero in the latch will force the output pin to always be ground, so that the external resistor-capacitor combination (i.e., the DAC output) will

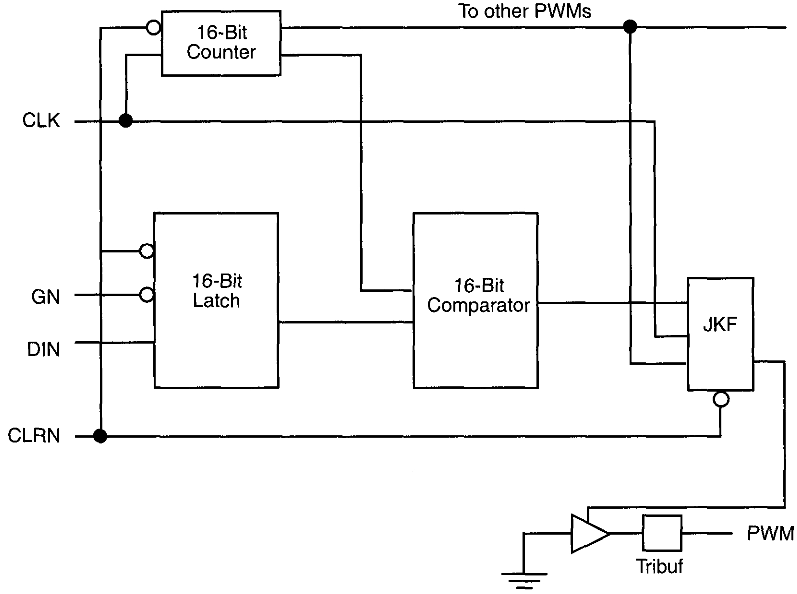


Figure 2 • PWM Structure

also be ground. As the latch increases in value, the output pin stays high longer which causes a corresponding increase in the DAC output voltage. The frequency of the counter sets the basis of the ripple of the DAC and also the response time.

Only one counter is required to support as many comparators and latches as the FPGA will hold. There are also tricks that can be done such as in a mostly 16 bit FPGA adding a carryout from the counter at 8 bits to have some faster frequency 8 bit DACs. In our implementations, we have not used a loadable counter since this changes the range of the DAC. There are more minimal gate implementation, but the software for them is more complicated as well. In this implementation, an 8 bit PWM requires 21 modules, a 12 bit, 30 modules, and a 16 bit 39 modules. The latch-comparator-flip flop combination is replicated usually to a nice number or if only a few parts are used to the maximum per FPGA. In some systems the input muxes are added to allow reading the latch data and in some cases the counter output by the software. No attempt has been made yet to read in the microwire design.

Microwire

The latest evolution of the design is to allow this customer to move his system or systems further from the computer by using a high speed dedicated microwire interface to replace the parallel port interface. The original idea was to have a simple 8 bit microwire type port and then feed data to the Actel 1020s for the PWM control.

When we designed the original microwire version we simply converted the parallel port FPGA to microwire. This we realized would have speed limitations which would slow down the control rate. This paper is the result of rethinking that portion of the design and using a high speed PLD to allow a faster microwire for those situations. In this sense it also reflects a design that is evolving. We could have used a shift register for the very high speed section and then moved bytes at the byte rate to the FPGA. Instead we are using a PLD which will allow us to include more control and sync logic as well as quickly changing the data path width without changing our board.

When we were thinking about this paper, we realized that the whole structure can be applied in more general applications with each FPGA being different if that is required. Some PWMs could be 8 bit which would allow more per part and/or a higher frequency versus others at 16 bit for better resolution. We have only begun to look at that structure, but the inherent nature of the FPGA allows for that kind of change without changing the pcb every time. Also, for this paper we have looked at using a pinout for the ACT 1 PLCC84 that would also work for the Actel 1240 or even 1280 for some more complex design changes by adding cuts and jumpers. This has not been tested.

The initial high speed version of the board will use a PA7024 for the microwire. The pcb will also be set to allow jumpering directly to the master control FPGA allowing for a slower speed minimal implementation with a master Actel 1020. The microwire version of the design is completely new and has yet to be tested outside the lab.

The initial microwire design is set up so that an ICT PA7024 receives data from the high speed serial bit link and then transfers bytes to an ACT 1 FPGA that does chip select, control, and buffering of the DAC data and address (Figure 3). This data and address is then passed to one of the individual slave PWM FPGA modules on the board. Each of the FPGA slaves contains its own counter which is run from the master clock on the board.

The PA7024 fully buffers the high speed serial data. The FPGAs run on a much slower independent clock. The speed of the FPGA clock is a function of the desired frequency of the PWMs. Higher frequencies will have less ripple and/or quicker response. Since in our application most signals do not change or do not change very often, we use a slower 8 Mhz clock. There is a lot of flexibility here. At 24 Mhz with a 3 byte per channel overhead the rate of change of one channel is 1 µsec so changing all channels is a 256 µsec minimum. In a parallel port system with a nominal 30K byte per second capacity, these numbers become 10K channels per second max or 100 µsec per channel. In some applications we have used block io software commands to output when all channels have to be written, but for the most part very few channels in our system change at once so that the actual data rate to the channels is fine at 10K or even less. In one instance where we needed more speed we reduced the data width and number of channels so that the standard parallel port worked. It is the ability to adapt the design that makes the PLDs and the FPGAs of value here by extending generic circuit designs without completely new boards for each design.

While we have not needed to change yet, we anticipate using fast parallel ports, speeding up the software, and using the higher speed PLD design discussed here for future implementations. In addition several remote microwire sites could be controlled from the central control location. The PLD speed also opens the possibility for future designs involving interfaces to coax or fiber. At the moment it seems more practical to let the PC handle this level of interface if it is required.

Summary

What started as a limited scope application has evolved into a larger structure which is being designed with the goal of minimal board changes components and maximum flexibility of implementation. The use of both PLDs and FPGAs increases this flexibility. In addition each stage of the design seems to open more possibilities for the customer as well as the designers.

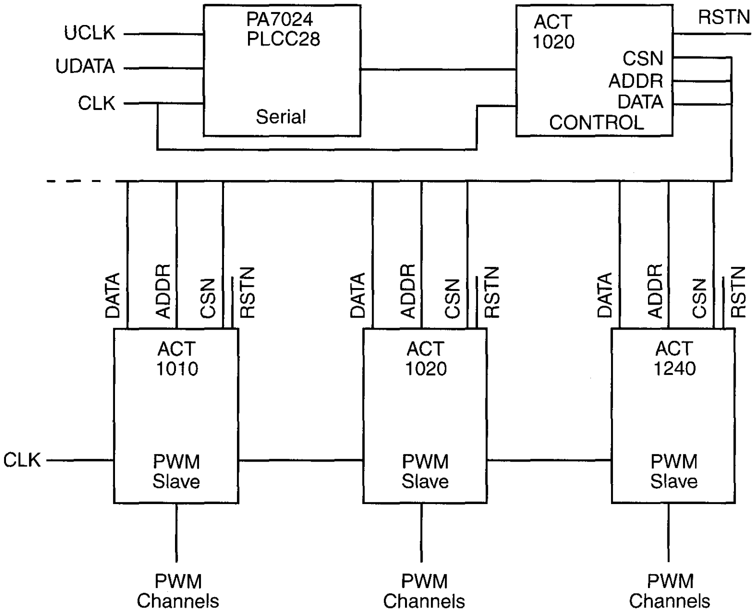


Figure 3 • *Microwire Diagram*

Using an FPGA on an S-Bus Card for High Speed Serial Data Interface

Steve Eigsti, President
Western Engineering & Satellite Technology

Specialized high speed data processing requires custom interface circuits to handle data rates which exceed capabilities of standard discrete logic parts. Using an FPGA realizes a savings of both power and circuit board space, and gives additional flexibility to accommodate combinations of serial or parallel interfaces with special handshaking signals.

Background

Processing image data from Earth observation satellites requires the ability to receive continuous streams of high speed serial data, Continuous data rates of 100 Mbps for 20 minutes are not uncommon. It is desired to store this incoming data in a computer for processing at a later time, Until recently, customized computer arrays or expensive data recorders were required to capture the input data. With the advent of inexpensive, high speed disk drives and powerful workstations, an inexpensive data processing system can be designed. An S-Bus adapter card which is a front end for such a data processing system using Actel's FPGA technology will be described.

System Description

A block diagram of the S-Bus adapter card is shown in Figure 1. Requirements for the adapter card include low power, small circuit board area and ability to receive continuous high speed serial data. The adapter card consists of an S-Bus controller to DMA received data to host computer disk storage, an Actel FPGA which receives high speed serial data and outputs 32 bit parallel data, and a 32 x 32 bit FIFO to buffer the input data during DMA operation.

Serial Input Data Reception

The high speed serial input data is converted to 32 bit parallel data using an Actel FPGA as shown in Figure 2. The design consists of a 32 bit serial-in-parallel-out-register (SIPO), a 32 bit output buffer register and a high speed 5 bit counter which counts the number of received bits and generates the clock enable for the output register. The FPGA also contains FIFO control logic and S-Bus control logic. The present design is implemented with a single data input channel. Additional data channels can be supported to increase the data rate while using reasonable baud rates.

Data Bit Counter

The 5 bit counter is shown in Figure 3. To obtain the highest possible performance, it is desirable to keep fanout on nets low and to use "look ahead" schemes to reduce the number of logic levels in the design. In the counter implementation, the FC2 signal goes high on the clock edge which drives Q0 and Q1 high. This technique reduces the width of the input logic driving F3. A similar technique is used for the terminal count indication (Tcn) which goes high on the clock edge which drives the counter output to 31. The terminal count is used to enable the clock on the 32 bit output register. Duplication of the terminal count signal allows the fanout on the clock enable net to be kept low for minimal delay.

Glue Logic

Interface signals for the S-Bus controller and FIFO (WR_FIFO, L_RDY, and L_HLDA) are generated in the Actel FPGA to eliminate the need for external circuitry.

Implementation Results

The serial to parallel data conversion is implemented in an Actel A1425 FPGA. This design used 28 percent of the device, Performance of the FPGA measured by the Actel Timer tool was 108 MHz using a standard speed device. The low utilization of the Actel device allows additional features to be added in future design revisions.

Conclusion

A high speed serial to parallel data converter was implemented in an Actel FPGA achieving high performance, low power and circuit board area savings. The flexibility of an FPGA in this application allows for quick design modifications to increase performance requirements or change functionality.

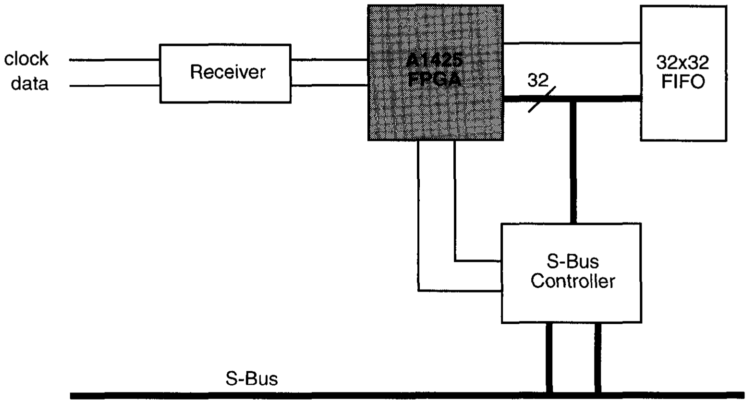


Figure 1 • S-Bus Adapter Card

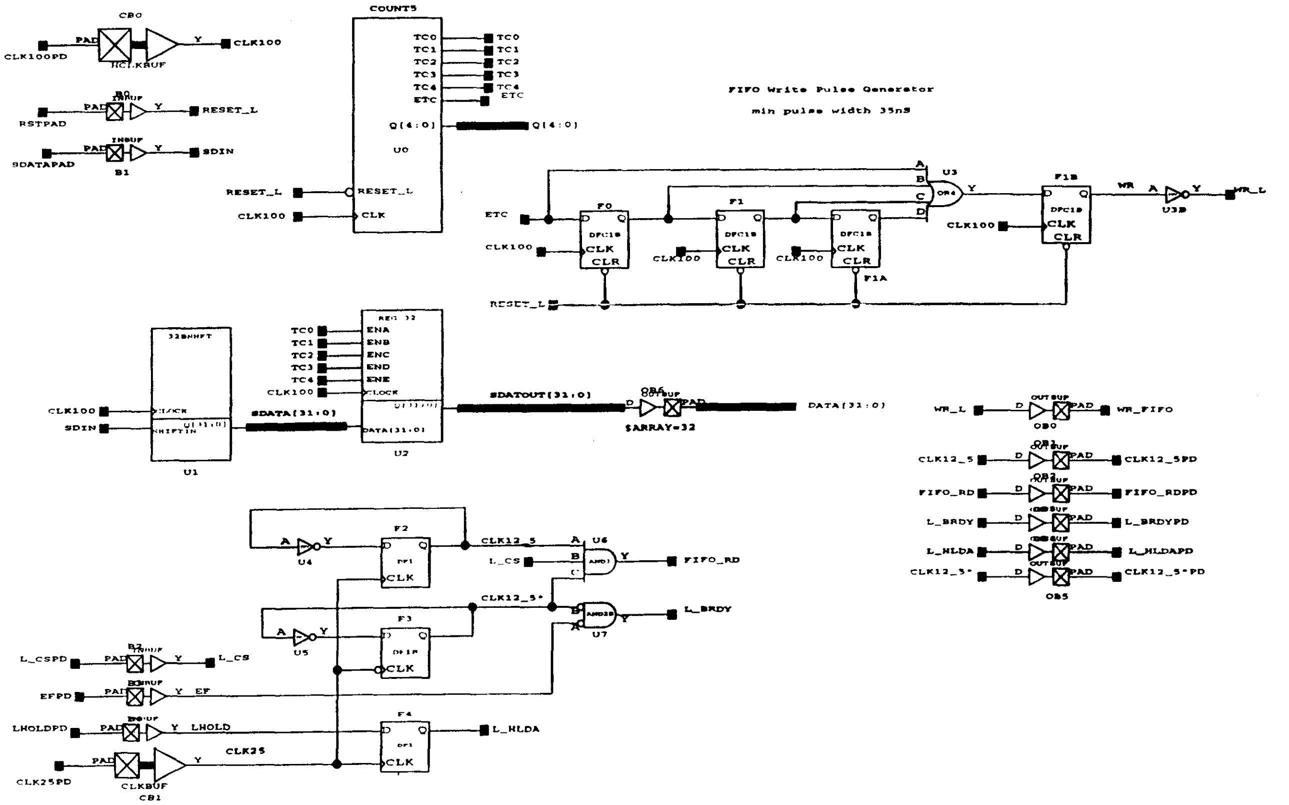


Figure 2 • Top Level Schematic

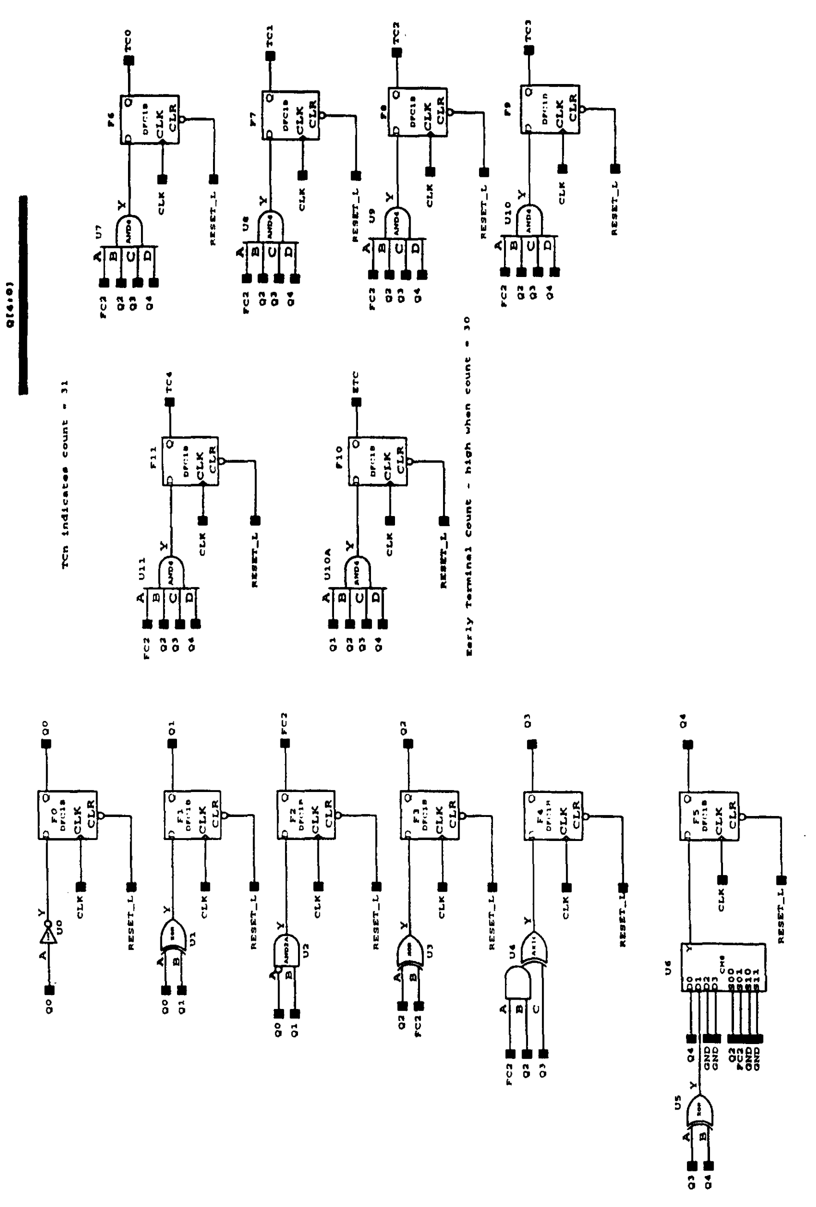


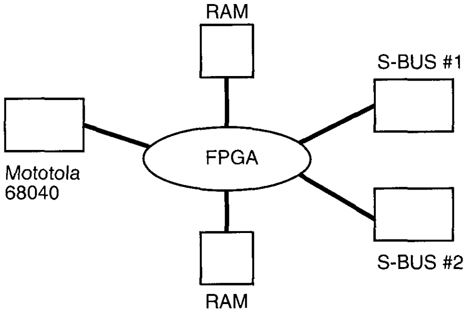
Figure 3 • 5 Bit Data Counter

Bus Translation Design Using FPGAs

Venkata Ramana Kalapatapu, Design Engineer
Sand Microelectronics, Inc.

Abstract

This paper discusses the use of a 6K gate FPGA to implement a design that controls and manages the communication between a Motorola 68040 bus, two SUN S-BUS devices, and two static RAMs at the rate of 25 MHz.



Introduction

The above diagram describes the configuration of the board with multi-processors and the role of the FPGA to control the communication between all the devices.

Each processor can be a master or a slave in a particular configuration while both RAMs have a slave role at all times. When the Motorola 68040 is the master, either of the SBUS devices or both can be a slave. When either of the SBUS devices is the master, only the 68040 can be the slave and the other SBUS device is idle.

The chip controlling the communication between the various devices on the board must determine where an incoming address has initiated, the destination device of choice, perform bus translation, and detect and correct any parity errors that occur during transmission.

Requirements

The choice of the device had to meet the following criteria:

- Data transfer rate of 25 MHz.
- High I/0 count (157 I/Os) with a high degree of flexibility for ease of board layout (pins had to be fixed early in the design and cannot be restricted to be of one type or another).
- Reliability and high level of signal integrity.
- Fast prototyping ability.
- Synthesizability (verilog code).
- Low cost.
- Ease of debugging, and ability to re-place and route without changing pin assignments.

FPGA Details

The device of choice for this design was Actel's A1460A-1 in a 208-pin PQFP package. The device had a core capacity of 6000 gates. The device has three high fanout clock networks for the core and one for the I/Os. Two of the core clock networks was used in this design. The HCLK hard wired high speed internal clock ran at 40 MHz. It clocked the control state machines, the address decoding, and the data transmission. Another clock network (CLKA) ran at 25 MHz. It controlled the parity error checking and correction. The fine granularity of this FPGA was highly suitable for synthesis and the closest in performance to a full ASIC.

The core of the chip was partitioned into two major blocks:

- Control logic and address decoding.
- Data Path.

The control logic section occupied the majority of the core. It included the functions of detecting the Master device, determining the slave device, controlling data transmission, and checking and correcting any parity errors that occurred during bus translation and transmission. The decoding logic took care of address decoding and of translation.

The Data Path block consisted of the bus translation function between the Motorola 68040 and the S-bus devices (both the S-bus and the Motorola bus are 32 bits wide). Also, the Data Path section included the Read/Write operations from any of the master devices to the RAMs and visa versa.

The S-Bus operates in a burst transfer mode while the Motorola bus is not capable of burst transfer mode operation. Thus a data transfer from the S-Bus device to the Motorola 68040 is a single burst cycle operation.

The RAMs are 16-bit wide. Thus a Read/Write operation involving any of the master devices and a RAM is done in two cycles, each cycle covering 16 bits of data.

Design Encoding and Simulation

The design was encoded using standard verilog VHDL. It was then simulated using a zero-delay verilog behavioral model simulator. The design was then synthesized using the Synopsys standard compiler. The gate level netlist was then simulated using Actel's macrocells verilog models library (unit delay models).

The design was then placed and routed using Actel's ALS software. The ALS timer was used to check the post place and route timing on the critical paths as well as overall chip performance.

In the encoding process some specific flip-flops were directly instantiated. All the I/Os were also directly instantiated. Our experience has been that the performance one gets from an FPGA using synthesis is highly dependent on proper coding that makes use of the particular architecture of that FPGA (mux based in the case of the A1460).

Post layout delays were back annotated to the verilog gate level netlist using Actel's SDF interface. The back annotated design was then simulated using the verilog-XL simulator. The design was simulated by the same set of input vectors for both the behavioral design and the back annotated gate level design producing the same result.

Sample Code

```
module SRamCtrl ( SC_NIO_CLK,
SC_MEZZ_RESET, SC_ICE2SRAM,
SC_ISP2SRAM, ........................,
SC_SRAM_UB_L, SC_SRAM_LB_L );
input [1:0] SC_NIO_SIZE;
input [2:0] SC_SBUS_ACK_L;
input [2:0] SC_SBUS_SIZE;
input SC_NIO_CLK, SC_MEZZ_RESET,
SC_ICE2SRAM, SC_ISP2SRAM, SC_INT_TS,
SC_AS_L,
  SC_NIO_RW, SC_SBUS_RW,
SC_NIO_ADR:
output SC_SRAMCYC_L, SC_CSDELAY,
SC_SRAM_CS_L, SC_SRAM_OE_L,
SC_SRAM_WE_L,
  SC_SRAM_UB_L, SC_SRAM_LB_L;
  wire n640, n641, n642, n630, n631, n632,
n633, n634, n635, n636, n637,
  n638, n639, n627, n628, n629,
IspSRamCyc, \*cell* 111/Z_O,
     \*cell*106/CONTROL1;
  DFP1 SC_SRAM_CS_L_reg (
.PRE(SC_MEZZ_RESET),
.CLK(SC_NIO_CLK), .D(n637),
    .Q(SC_SRAM_CS_L) );
  DFP1 SC_SRAM_WE_L_reg (
.PRE(SC_MEZZ_RESET),
.CLK(SC_NIO_CLK), .D(n640),
    .Q(SC_SRAM_WE_L) );
....................
....................
  DFP1 SC_SRAMCYC_L_reg (
.PRE(SC_MEZZ_RESET),
.CLK(SC_NIO_CLK), .D(n639),
    .Q(SC_SRAMCYC_L) );
endmodule;
```

Results

The design went through two iterations. In the first pass the core utilization was 98 percent and the design included JTAG testing. The results were satisfactory but a decision was made to get rid of the JTAG. The design was modified in a major manner to push the performance further. The pinout was fixed from the first round. Re-placing and re-routing the design went through without any problems at all. The final core utilization was 70 percent. The final design was backannotated and simulated at gate level. It was fully functional at a data transfer rate of 30 MHz.

Although a "–2" speed grade was available for this part allowing for another 10 percent speed improvement. The design was finalized in a A1460-1 in a 208-pin PQFP package.

After the design was verified a fuse file was generated in the ALS environment and Actel's activator was used to program prototypes.

Conclusions

Using FPGAs for bus translation applications proved to be feasible, easy, and reliable. FPGAs provided high I/0 counts needed for such an application. Also, sufficient core gates were available for control and arbitration logic. I/0 instantiation and writing code that makes use of the available architecture is instrumental in obtaining optimal results.

Design of an Active Noise Control System Using Combinations of DSP and FPGAs

Reza Hashemian, Senior Member IEEE
Associate Professor, Northern Illinois University

Field Programmable Gate Arrays (FPGAs) offer a quick and cost effective implementation for medium to large size digital designs that traditionally have been carried out mainly by ASIC and DSP implementation. However, there are still a number of processing-intensive applications that no single FPGA chip, developed today, can handle the entire design.

In this paper, an experimental design of an active noise control system is introduced that combines both DSP and FPGA in a task oriented structure. In this design, data received by the local microphone and the output signals to the secondary speakers in the noise field are handled by the FPGA. This include I/O buffers, data paths, memory access, FIFO, bit-wise manipulations (data shift and logical operations), and the control block (FSM). Data crunching such as high speed manipulations and additions for updating the coefficients in the Signal Processing Block is handled by the DSP. There are two major design criteria considered here: 1) split the task between the FPGA and the DSP in order to reduce the number of DSP instructions as much as possible, and 2) make the devices to work simultaneously and with minimum dependency.

Introduction

As an alternative to passive noise cancellation techniques and often in combination with them, active noise cancellation (ANC) offers an effective solution in certain applications. Although still in the development stage, ANC is receiving considerable attention for applications involving industrial apparatus, dynamic systems, and domestic appliances. In contrast to passive techniques, ANC systems are small, portable, adjustable to different environments, and less costly. An ANC system can be effective across the entire noise spectrum, but it is particularly appropriate at low frequencies of up to 300 Hz, where passive systems are less effective.

The success of an ANC system depends mainly on fulfilling two criteria: first, the anti-noise waveform must closely match the shape and frequency of the noise waveform; and second, the anti-noise wave must be precisely 180 degrees out of phase with respect to the original noise waveform, when reached to the target area. Failure to fulfill one or both of these criteria may cause the ANC system to generate a second acoustic noise rather than cancel the original one. These criteria apparently impose some restrictions on the canceling system and somewhat limit its application. First, for a highly effective canceling system the noise source must be nearly stationary in relation to the speaker emitting the anti-noise waveform. Second, the noise source should be located in close proximity to the ANC system and, for the best results, the target noise must be dominantly propagating in one direction. This suggests more of an effective "zone silencing" rather than an open area cancellation.

Acoustic delay is another important issue that must be dealt with in a noise canceling system. Physically there are always distances between the source, the anti-noise generator (speaker) and the residue noise detector (microphone). These physical distances provide noise propagation delays which in turn cause different phase shifts, depending on the relative location of objects. In a general case of a non-periodic noise, prediction techniques and adaptive systems are used to deal with the problem. In this case, an ANC system is dependent on its ability to predict noise from its memory of the past, or adapt the response to the incoming signal as closely as possible. In a periodic noise system, however, prediction is simply done by storing one or more cycles of noise. This makes the periodic noise cancelers much simpler and more effective. And, as a matter of fact, periodic noise is one of the most common and dominated source of noise in industrial and even domestic environment today.

Traditionally, two basic methods are used in ANC systems. In the first method, known as adaptive cancellation, noise is detected by one or more microphones. The system then adapts itself to generate anti-noise waveform which minimizes the residue (mic.) noise. Adaptive cancellation can be used for both periodic and non-periodic noise. However, when used with non-periodic noise the adaptive method is limited because it usually involves a prediction or an estimation process. Feed forwarding is often used to predict the noise before it reaches the target. As for the periodic noise, the adaptation and estimation is basically reduced to utilizing the past noise cycle for the generation of the present anti-noise waveform.

The second type of active noise cancellation system is based on the synthesis method. This involves sampling and storing one or more noise cycles and, based on this information received an actual anti-noise waveform is generated to suppress the noise. This method assumes that the current noise cycle will be close to, if not the same as, the past noise cycle. In other words, this method is more appropriate to a periodic noise environment, as mentioned earlier. Anti-noise waveform, in this case, is generated by associating a digital pulse train with the noise cycle and using this pulse train to synchronize the anti-noise waveform with the emitted noise. Typically, this pulse train can be derived from a non-acoustic source, such as an engine's tachometer or odometer.

Active Periodic Noise Cancellation

Periodic noise such as that emitted by industrial equipment and even domestic appliances is quite common in our highly industrialized society. A number of techniques are used in active periodic noise cancellation (APNC) systems. Some APNC systems address the sound cancellation needs of multi-dimensional environments such as a working factory. Others are more limited in scope and function as zone silencing systems. For example, a zone silencing system might be used to cancel the noise around the head of a driver in a vehicle.

Here, in this article, we use the synthesis method to generate the anti-noise waveform, in an APNC system. Contrary to the traditional approach, the signal processing is totally done in the time domain. This not only adds to more real and physical understanding of the processes it also saves time avoiding unnecessary switching domains between time and frequency. The second major change from the traditional approach is the separation of signal processing and signal flows, used in this method. We use both a DSP and a FPGA to share the processes needed in an APNC system, as will be discussed shortly.

In a shared processing environment where different processing types are being used simultaneously it is important to appropriately distinguish between different processes and data flows in order to place them in the right type of environment with the right timing. In our design case, the process sharing between a DSP and a FPGA is aimed at: i) reduction of both the memory requirement and the programming instruction for the DSP; and ii) leaving all the data flows and interfacing to be handled by the FPGA. One may be facing with several problems in this kind of task oriented environment. First, how the data is handled between the two units (the DSP and the FPGA)? Apparently the clocking for the two units can not be the same, because the response times are different. Then, an asynchronous communication link must be set up between the two units. The second problem is to separate the tasks in such a manner that optimization of both response time and hardware are fulfilled. In other words, one must keep both units operational in most of the time, and occupied with the most suitable operations for that unit.

Figure 1 conceptually shows how the system is constructed. A FPGA (TI's TPC12 Series) and a TMS320C2x DSP are the two major units used in our Noise Control System. As shown, the residue (error) noise is received by the FPGA after being digitized. The data is pre-processed (logical shift, inversion, and addition) by the FPGA and stored into a Data Buffer. The DSP then picks up the data from the Buffer and goes into further processing (multiplication, summation, and signal adaptation) and delivers the data back to the FPGA. Finally, the FPGA performs some post-processing (data shifting and sequencing) on the data and generates the anti-noise waveform in digital samples. As illustrated, the process requires "interrupt request" by the FPGA to send the data to the DSP, and "data ready" signal from the DSP to place it into the Buffer. This communication is naturally asynchronous because the clocking for the two units (CLK1 and CLK2) are different.

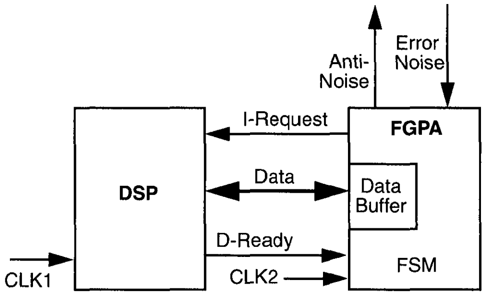


Figure 1 • *Acoustic Noise Cancellation with DSP and FPGA*

Figure 2 shows the overall structure of the APNC system. It consists of three basic parts: i) the noise channel, where the source produces the noise, the anti-noise speaker generates the opposing noise signal, and the microphone which picks up the residue noise. ii) The second part is the data flow section. This part consists of ADC and DAC (signal filtering is also included), clocking (pulse train), and signal pre-processing (coefficient μ and subtraction) operations. iii) The third part is the Anti-noise Generating Block producing the desired waveform for the noise cancellation. First, we discuss the noise channel. Consider x(t) and y(t) to be the noise and the anti-noise waveforms at the target location (microphone, in our case), respectively. The residue noise, e(t), picked up by the microphone is expressed by

$$e(t) = x(t) + y(t) \tag{1}$$

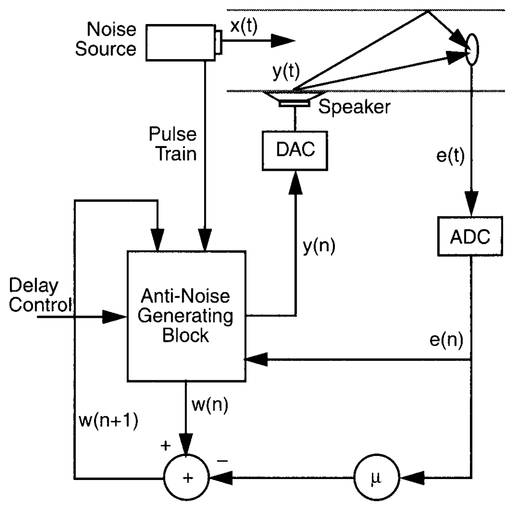


Figure 2 • *APNC System with Noise Channel*

This indicates that the residue noise is the algebraic sum of the original noise, x(t), and the anti-noise waveform, y(t), in the vicinity of the microphone. For the sake of simplicity, we assume that both the noise and the anti-noise waves propagate within a noise channel with limited number of reflections and with negligible distortion along the channel. After digitizing the signal and assuming that data conversions are lossless (except for the processing delays) we may write Eq. (1) in its digital form as

$$e_j(n) = x_j(n) + y_j(n) \qquad (2)$$

Where, n is the noise cycle count and j stands for the position of the sample within the cycle. No signal delay is assumed in Eq. (2). However, with acoustic delays and multiple reflections of the anti-noise waveform within the channel we can substitute Eq. (2) by Eq. (3).

$$e(n) = x(n) + \sum_i a_i \bullet y(n - v_i) \qquad (3)$$

Where, a_i and v_i are the attenuation factor and the acoustic delay of the ith reflection of the anti-noise within the channel, respectively. Note that the sampling indicator, j, is dropped for simplicity.

The residue noise leaving the channel is multiplied by a converging coefficient $0.0 < \mu < 1.0$ and $\mu \bullet e(n)$ is used to update the anti-noise samples stored in the FIFO registers, as shown in Eq. (4).

$$w(n + M) = w(n) - \mu \bullet e(n) \qquad (4)$$

Where, M is an integer 1 or more. The multiplication $\mu \bullet e(n)$ is simply performed by an appropriate right shift applied to e(n). For values of μ=l, 0.5, 0.25, or 0.125 we apply 0, 1, 2, or 3 right shifts, respectively. Shifted data ($\mu \bullet e(n)$) is then subtracted from a previous noise sample stored in the FIFO, inside the FPGA, and the result is directed back into the FIFO which contains M•N registers in sequence; where, N indicates the number of samples per noise cycle.

The next step is to synthesize the anti-noise waveform from the samples stored in the FIFO. This is done by the Signal Processing Block (SPB), shown in Figure 3. The operation works as follows: A set of noise samples stored in the FIFO are accessed by the SPB through a multi-bus line v_0, v_1, ... Each of these samples, taken from different locations (registers) within the FIFO, represents a sample of noise with a particular delay (v_i) attached to it. It is the SPB to use one or more of the samples, scale and add them together to produce a sample of the anti-noise waveform to best suppress the noise. The performance is simply measured by residue noise, e(n), which is also used to further adapt and modify the anti-noise waveform, as given in Eq. (4). The anti-noise is then generated using Eq. (5).

$$y(n) = \sum_i b_i \bullet w(n + v_i) \qquad (5)$$

Where, b_i is a multiplier constant assigned by the SPB. While the FIFO resides in the FPGA the SPB is handled by the DSP chip, as shown in Figure 1. To better understand the process in the SPB we consider the following cases.

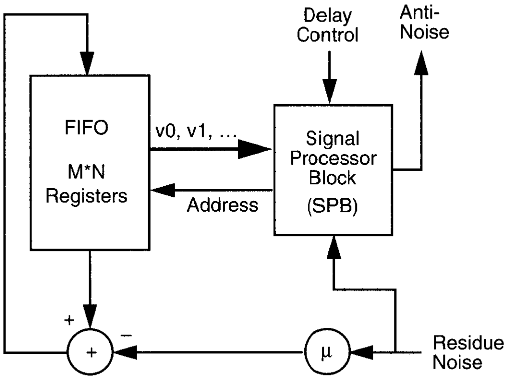


Figure 3 • *Anti-noise Generator Block*

Case 1

We ignore any wall reflections in this case and assume a direct acoustic path from the speaker to the microphone, as shown in Figure 4. We assume an acoustic delay of v_0 (including data processing delays, as well) in this case. Now, for $a_0 = b_0 = 1.0$, and M=1 the noise equations (3), (4), and (5) are reduced to:

$$e(n) = x(n) + y(n - v_0)$$
$$w(n+l) = w(n) + \mu \bullet e(n) \quad (6)$$
$$y(n) = w(n+v_0)$$

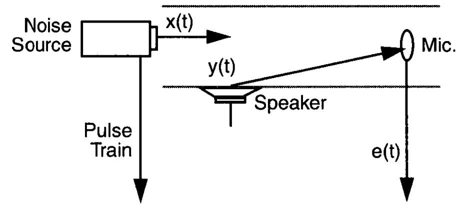


Figure 4 • Channel Signal Path for Case 1

From Eq. (6) it is easy to calculate the next cycle residue noise, e(n+1), which is given by expression

$$e(n + 1) = (1 - \mu)\bullet e(n) \quad (7)$$

It is evident from Eq. (7) that for any value of $0.0 < \mu < 1.0$ the residue noise will progressively vanish as long as the periodicity of the noise is not grossly violated.

Case 2

In this case we include all single reflections from the walls of the noise channel, as shown in Figure 5. We assume a constant attenuation factor a_l for the reflected anti-noise signal. The modified noise equations for this case become as

$$e(n) = x(n) + y(n - v_0) + a_l \bullet y(n - v_l)$$
$$w(n + l) = w(n) - \mu \bullet e(n) \quad (8)$$
$$y(n) = w(n+v_0) + b_l \bullet w(n+v_l)$$

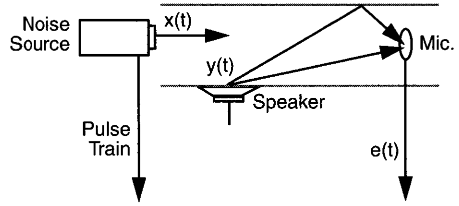


Figure 5 • Channel Reflection for Case 2

Where, v_l is the acoustic delay for the reflected waveform, and v_0 is still is the acoustic delay for the direct path. Assuming $v = v_l - v_0$ as the "difference delay" between the reflected path and the direct path, we can calculate the next cycle residue noise as

$$e(n + 1) = e(n) - \mu(1 + a_l b_l)e(n)$$
$$-\mu[b_l e(n + v) + a_l e(n - v)] \quad (9)$$

It is evident from Eq. (9) that, with a right selection of the convergence factor μ, and the fact that a_l and b_l are much smaller than 1.0, the residue noise e(n) theoretically converges to zero. However, channel distortion, higher reflections, and other factors may prevent e(n) to vanish completely.

Our analysis may further continue for multiple wall reflections, although, in practical sense, this may only have minor effect on the overall performance of the system.

Shared Processing with DSP and FPGA Combined

In our design we limit our case to Case 2, i.e., direct acoustic path between the anti-noise speaker and the target area (microphone) with single hit wall reflections included, as shown in Figure 5.

There are three parameters b_l, v_0, and v_l to take care of by the SPB which basically resides in the DSP. Here is how it works: the SPB searches for the best location in the FIFO to pick up the sampled data. The location is specified by a 3-bit control signal (or address) and a set of 8 to 1 MUXs deliver the data to the SPB. The data is then multiplied by a coefficient b_l and is prepared for summation and generation of the anti-noise sample waveform.

The search process is split into two steps. In the first step we neglect any reflection and assume $a_l = b_l = 0.0$, and then search for v_0. In this search the SPB tries for the best location inside the FIFO, as discussed earlier, and receives $w(n+v_0)$ from the FIFO. With this information the SPB can generate the anti-noise waveform for Case 1, direct path, as given in Eq. (6). In the second step, with v_0 being fixed (unless the "Delay Control" line is high indicating that the target has relatively moved), the SPB looks for the best values for the parameters b_l and v_l. This is done by measuring the residue noise after the first step and trying to select and modify the best values for b_l and v_l such that the residue noise is minimized.

Hardware Implementation

We have used an ACT 2 (TI's TPC1240 84-Pin PLCC) FPGA for the hardware implementation. This type of FPGA has the advantage of providing a large degree of functionality in a relatively small gate area, and being multiplexer based makes it suitable for our proposed architecture. Some of the design criteria of the chip are as follows:

- Data is represented by 8-bit words.

- The number of samples per cycle is N = 12, and M = 1 and expandable to 3.
- An external pulse train is used to generate the internal clock.
- In addition to the input and output ports eight different locations along the FIFO registers are also accessed for signal delay purposes. The selection of the location is done through multiplexing.
- For the noise fundamental frequency about 300 Hz the system is capable of adjusting delays from zero to 10 mSecs.
- The system is built with a test mode feature. In its test mode the actual noise is picked up by the microphone. The anti-noise is then added to the noise internally, very much similar to that happens in the noise channel, and the residue noise in subsequently measured for the system performance. While in this mode acoustic delays could also be added to the operation.
- A control block (FSM) is included in the design to provide clocking and all other control signals necessary for the operation. This block also controls the communication between the FPGA and the DSP.

Figure 6 shows a block diagram of the FPGA design in its simplest form. The control signals and the communication links to the DSP are not present. As mentioned earlier, the design is using an ACT 2 (TI's TPC1240 84-Pin PLCC) chip with 4,000 usable gate modules.

Conclusions

In recent years, governmental bodies, industrial concerns and even increasing numbers of individual consumers have realized that noise is a pollutant of our industrialized society and that noise pollution can be just as harmful, if not more so, to our psychological and physiological well being as other pollutants that are commonly classified as toxic. Noise is just as toxic as chemicals and other forms of pollution.

Both active and passive noise control methods will be used in the years ahead to combat noise pollution. Several automobile manufacturers have already implemented active noise control systems in their products and manufacturers of domestic appliances are studying the most cost effective way of actively canceling the noise produced by motors in dish and clothes washers, and other household appliances. Many of the commercial airlines are asking the aerospace industry to incorporate active noise cancellation systems to decrease the noise created by jet engines, In the workplace, enlightened manufacturing concerns are considering active and passive noise control techniques to reduce the noise levels in factories and in the operating enclosures of large equipment such as bulldozers and heavy cranes.

Field Programmable Gate Arrays (FPGAs) in combine with DSPs will play an important role in the further development and implementation of active noise control systems. The ease with which FPGAs can be used to prototype active noise cancellation systems will facilitate more effective experimentation which will bring actual active noise cancellation products to market faster.

References:

1. E. Ziegler Jr. "Selective Active Cancellation System for Repetitive Phenomena." United States Patent No. 4,878,188, Oct. 31, 1989.
2. N.R. Shanbhag, and K.K. Paraki. "A Pipelined LSM Adaptive Filter Architecture." proc. Asilomar Conf. on Sig., Syst. and Comput., Pacific Grove, CA, Nov. 1991.
3. S.D. Brown, R.J. Fransis, J. Rose, and Z.G. Vranesic, "Field Programmable Gate Arrays", KLUWER Aca Pub, 1992.
4. R. Hashemian, K. Golla, S.M. Kuo, and A. Joshi, "Design and Construction of an Active Periodic Noise Canceling System Using FPGAs," 36th Midwest Symposium on Circuits and Systems, Detroit, MI, 1993.
5. S.J. Elliott and P.A. Nelson, "Active Noise Control," IEEE Signal Processing Magazine, October 1993.

Test and Reliability Reports

Section 7: Test and Reliability Reports

Testing and Programming Actel Field Programmable Gate Arrays (FPGAs)

Testing has long been a struggle for users of masked gate arrays. To avoid board-level, system-level, or even possible field failures, the system designer must expend great effort in developing test vectors for gate array designs. Even after the vectors are developed, fault coverage for typical designs may be only about 70 percent, with about 95 percent coverage being the best possible. With a 70 percent fault coverage, typical masked gate array designs are likely to have 2 to 5 percent defective devices[1].

In general, field programmable logic devices have allowed users to avoid the need to develop test vectors. These devices allow tests to be performed by the semiconductor vendor prior to programming. However, most one-time programmable logic devices have not yet achieved the functional quality levels of other semiconductor devices, because they don't allow the chip manufacturer to access and test all internal gates. Early one-time programmable devices had poor test coverage, and users were often disappointed to see functional failure rates of more than 10 percent on parts that had passed programming. Over time, on-chip test circuits and testing techniques have greatly improved, and now one-time programmable devices have functional defect rates in the range of 0.1 to 1 percent[2]. Although this failure rate is low for individual chips, putting 10 such chips on a single board can still mean a board failure rate of 5 to 10 percent.

The Actel FPGA Product Family

There are five Actel FPGA product families. The ACT 1 family offers 1200 (A1010) and 2000 (A1020) gate products. The ACT 2 family consists of three products—A1225A, A1240A, and A1280A—with 2500, 4000, and 8000 gate array equivalent gates, respectively. This family offers improved performance and number of I/Os compared with ACT 1 products. The 1200XL products (A1225XL, A1240XL, and A1280XL) are functionally equivalent to the ACT 2 products but offer even better performance. The 3200DX family currently consists of two products with 6500 and 14,000 gates (3265DX and 32140DX) but will soon expand to include products up to 40,000 gates. The ACT 3 family contains five products (A1415A, A1425A, A1440A, A1460A, and A14100A) with 1500, 2500, 4000, 6000, and 10,000 equivalent gates, respectively. The ACT 3 products offer the highest performance of the five families.

Testability of Actel FPGAs

Although Actel's FPGA families use a one-time programmable technology, the device's unique architecture permits a degree of testability comparable to reprogrammable devices. Special test modes allow functional testing of unprogrammed devices at essentially 100 percent fault coverage. This testability is independent of the large number of equivalent gates in the A1010 (1200 gates) through the 32140DX (14,000 gates). To show how this is accomplished, we will first review the architecture of the Actel FPGAs and describe how they are programmed.

Architecture

The basic building block of all Actel FPGAs is the logic module. Each logic module is programmable and capable of implementing all two-input logic functions, most three-input functions, and many other functions up to eight inputs. With an architecture similar to a channeled gate array, logic modules are organized in rows and columns across the chip (Figure 1). Adjacent to each row of logic modules are routing channels. Horizontal routing channels are shown in the figure, but vertical channels also run through the logic modules. These are used to configure a logic module and connect inputs and outputs of logic modules to implement a design. Surrounding the array of logic modules and routing channels are I/O buffers and test circuits.

Within the routing channels are programmable antifuse (PLICE) elements. The antifuse is normally open and is programmed to form an electrical connection between routing elements. An antifuse that connects a horizontal routing track to a vertical track is called a cross-antifuse. An example of a logic module interconnection (or a net) is shown in Figure 2. Here the output from Module 3 is connected to a horizontal routing track by programming a cross-antifuse. Another cross-antifuse is programmed to connect an input to Module 4. In a similar manner, the output of Module 3 is connected to the input of Module 2. Notice that not all horizontal tracks are continuous across the chip. Often, tracks are broken into a series of smaller tracks called segments. Segments are useful because it is often desirable to connect logic modules that are close to each other, and a full horizontal track would waste routing resources and slow down circuit performance. Sometimes, however, it is necessary to connect two segments to form a longer segment.

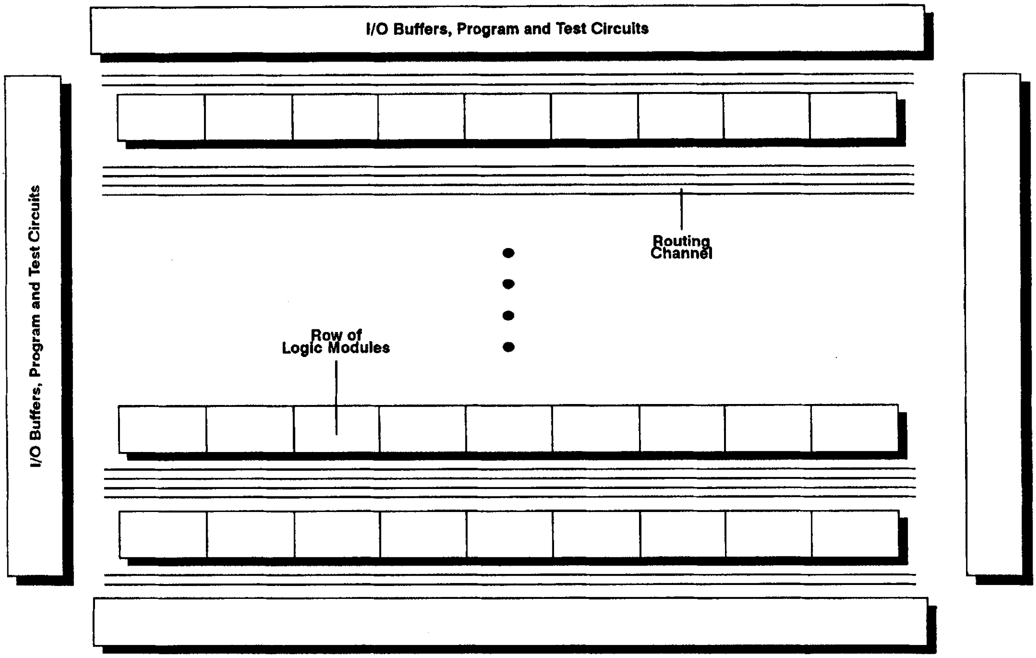

I/O Buffers, Program and Test Circuits
I/O Buffers, Program and Test Circuits
Routing Channel
Row of Logic Modules

Figure 1 • *Logic Module Architecture*

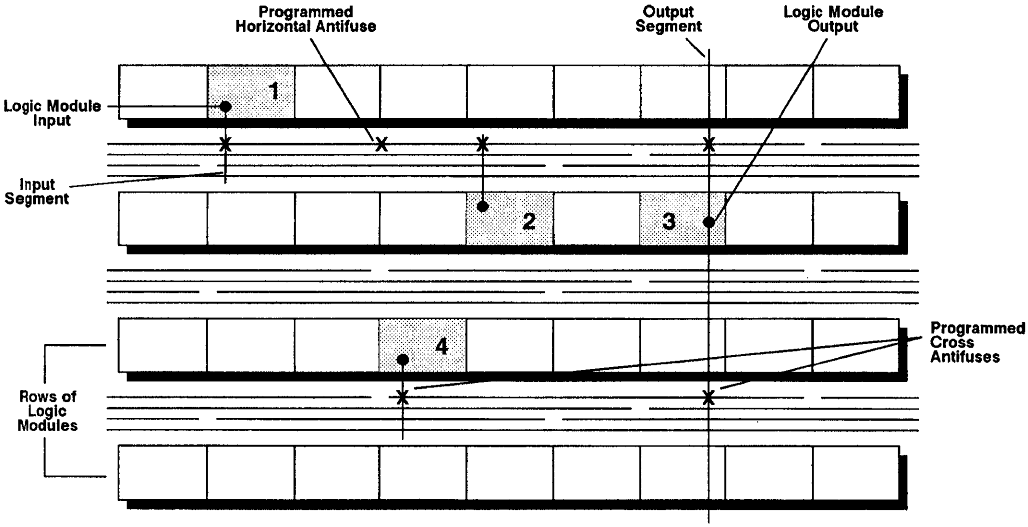


Figure 2 • *Logic Module Interconnection*

This can be done by programming a special type of antifuse referred to as a *horizontal antifuse*. As an example, the output of Module 3 is also connected to the input of Module 1 by programming two cross-antifuses and one horizontal antifuse. Vertical antifuses are used to connect two vertical segments (not shown).

A more detailed example of the Actel FPGA architecture is shown in Figure 3. Six logic modules (two rows, three columns) are shown. Between the two rows are six horizontal tracks. Down each column are five vertical tracks. Note that the products actually have 25 to 36 (or more) horizontal and 13 to 15 (or more) vertical tracks. The circles at the intersections of vertical and horizontal tracks represent cross-antifuses. There are also circles at certain points on the horizontal tracks; these are horizontal antifuses. No vertical antifuses are shown. Notice the transistors that connect both horizontal and vertical tracks. By turning on selected transistors, various horizontal or vertical tracks can be connected even though an antifuse has not been programmed. This ability to connect tracks in unprogrammed devices is used extensively during antifuse programming and is one of the key elements responsible for the excellent testability of the Actel FPGAs.

Logic configuration of modules is interesting because there are no dedicated antifuses in the module to accomplish this. Instead, the inputs (and outputs) of logic modules extend into the cross-antifuse array. Each logic module has eight to ten inputs and one output. By programming appropriate antifuses, an input can be connected to a dedicated horizontal ground line, a Vcc line, or a horizontal routing track. The logic module implements a particular logic function by tying appropriate unused inputs to ground or Vcc.

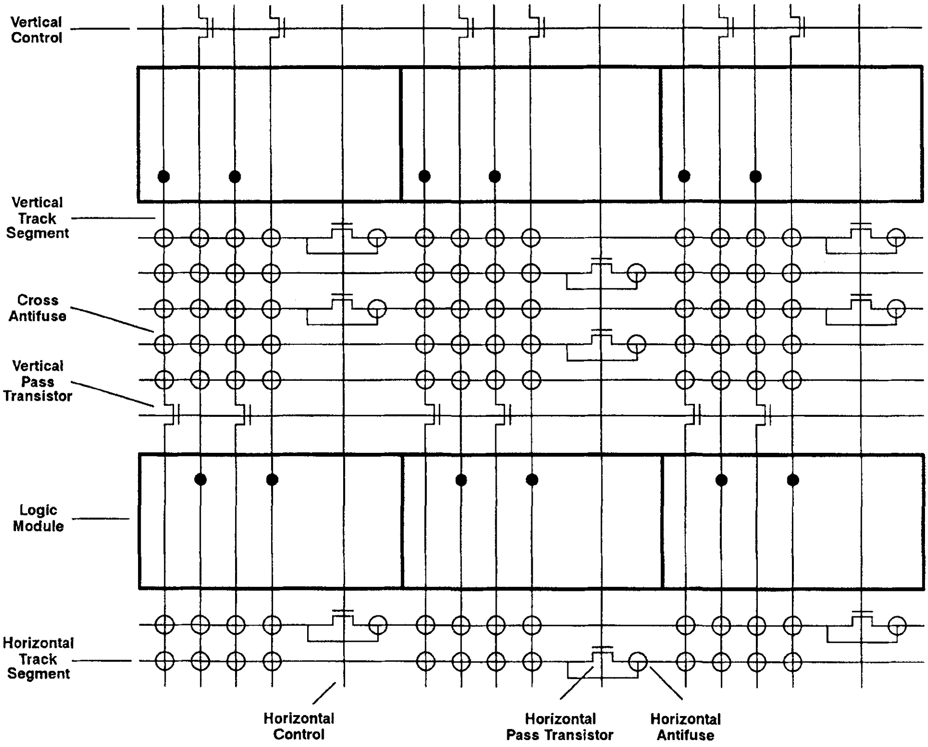


Figure 3 • Programmable Interconnect

Programming

The following discussions about programming and testing modes are specific to the ACT 1 family of FPGAs. However, basic concepts also apply to all other antifuse FPGA families.

An antifuse is programmed by applying a sufficiently high voltage across it. This voltage is referred to as *Vpp*. To access an antifuse deep inside the chip, it is necessary to create electrical paths from Vpp and ground to the antifuse. This is done by turning on the appropriate horizontal and vertical pass transistors. (In normal chip operation, these transistors are always off.) The transistors are turned on by applying Vpp to their gates. In Figure 4, we see an example of programming a typical cross-antifuse. Vpp is applied to a vertical track at the top of the chip, and ground is applied to a horizontal track on the right side. The design of the A1010/A1020 actually allows Vpp or ground to be applied from the top, bottom, left, or right, as is most appropriate to access a particular antifuse. Notice that Vpp is also applied to the gates of the horizontal and vertical pass transistors on the tracks accessing the cross-antifuse. The circled cross-antifuse now has Vpp applied to it on one side and ground on the other. This voltage breaks down the antifuse's dielectric and creates an electrical connection between the horizontal and vertical routing tracks.

There is one other important consideration when programming an antifuse. Notice that the cross-antifuses in the same vertical track as the antifuse to be programmed also have Vpp applied to them on one side. This is true until the track is broken by a vertical pass transistor, below it, that is turned off. However, the potential on the other side of the antifuses is not being driven. Should this potential be at ground, the other cross-antifuses on the vertical segment could be accidentally programmed.

The same logic applies to other antifuses on the same horizontal track. Here, one side of the antifuse is being driven to ground, and if the other side were at Vpp, extra antifuses could be programmed. This problem is solved by first applying what is referred to as a *precharge cycle*. During the precharge cycle, all horizontal and vertical tracks are charged to Vpp/2. As a result, there is no voltage across the

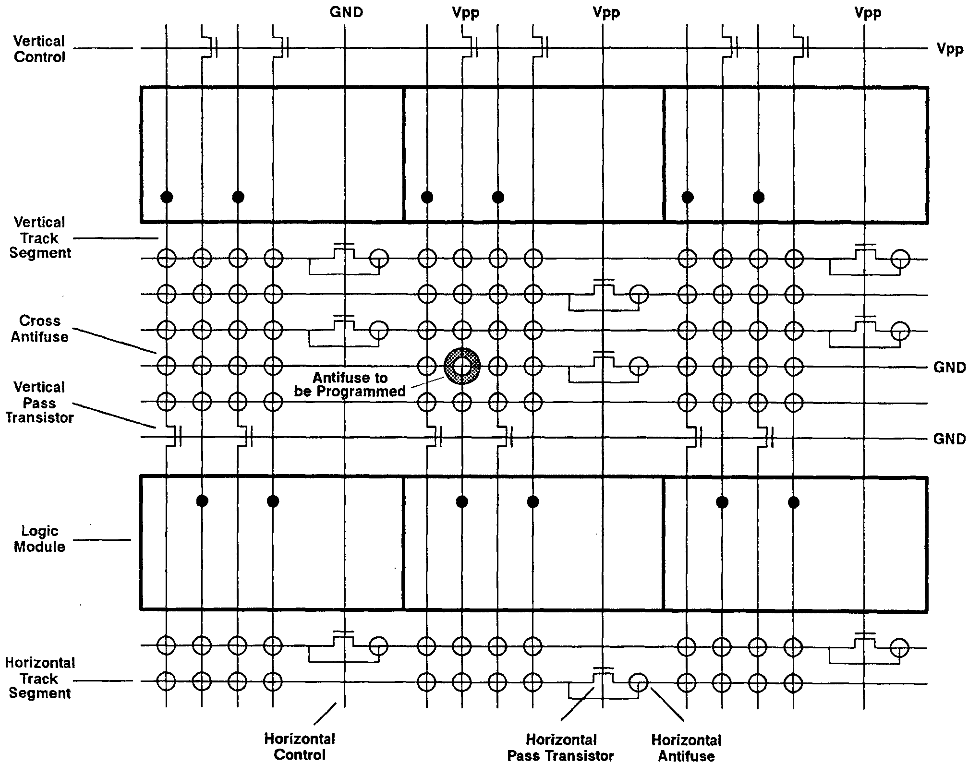


Figure 4 • *Programmable Interconnect*

antifuses. The appropriate vertical track is then driven to Vpp, and a horizontal track to ground (Figure 5). At this point, other antifuses on the vertical track have a potential of Vpp/2 across them (Vpp on one side and Vpp/2 on the other). This Vpp/2 voltage is not sufficient to program the antifuses. Other antifuses on the same horizontal track also have Vpp/2 across them (Vpp/2 on one side and ground on the other). Most other antifuses in the chip still have Vpp/2 on both sides and will not be programmed.

Programming Algorithm

In concept, the Actel FPGAs are programmed in a manner very similar to many other programmable logic devices, and similar to memories such as EPROMs. The programming algorithm consists of the following steps:

1. An addressing sequence to select the antifuse to be programmed
2. A programming sequence whereby Vpp is applied in pulses until the antifuse is programmed
3. A soak or "overprogram" step to ensure uniform, low antifuse resistance
4. A verify step to make sure the antifuse was properly programmed

Unlike a memory in which an antifuse is addressed by applying a parallel address, the FPGAs are addressed in a serial manner by using the special DCLK (Data Clock) and SDI (Serial Data In) pins. There is a large shift register that travels around the periphery of the chip. Bits in this shift register can be used to drive tracks to ground, Vcc, Vpp, or float. It is also possible to sense the level on the track (high or low) and to load this information into the shift register. By shifting in the correct address, any antifuse can be selected for programming. The shift register also plays a key role in testing the chip. This will be discussed later.

The programming sequence starts with the precharge pulse whereby Vpp/2 is applied to the Vpp pin. This is followed by a programming pulse that applies Vpp to the pin. Following the

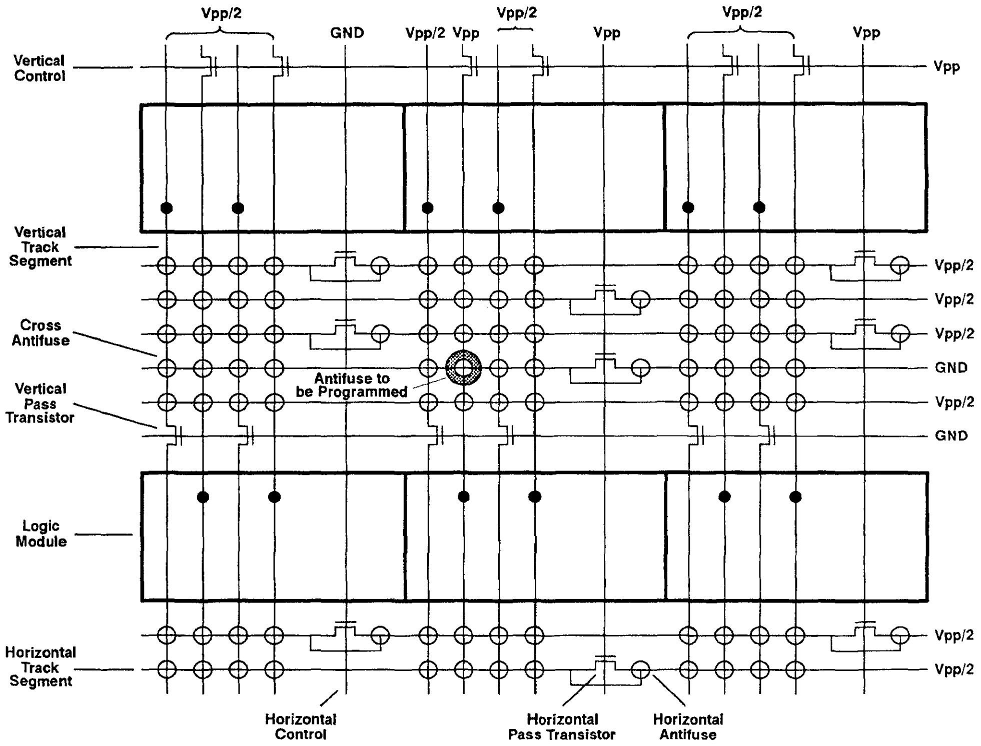


Figure 5 • Programmable Interconnect

program pulse, the voltage on the Vpp pin is returned to a nominal value (about 6 V). See Table 1 for a typical Vpp waveform. The precharge/program pulse sequence is repeated until either the selected antifuse programs or a maximum number of pulses is exceeded (in which case the antifuse is considered unprogrammable and the device is rejected).

Confirmation that an antifuse has been programmed is determined by monitoring the current on the Vpp pin. This current is very low (typically < 10 µa) until an antifuse is programmed. Once an antifuse is programmed, an electrical connection is made between Vpp and ground, in which case currents in the range of 3 to 15 mA may be observed on Vpp. Once this current is observed, the antifuse is considered programmed and enters the soak or "overprogram" cycle. Here, extra pulses are applied to the antifuse to achieve minimum antifuse resistance. Figure 6 shows the Vpp waveform for ACT 1 devices.

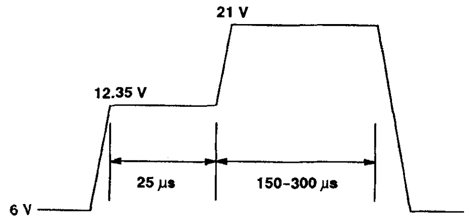


Figure 6 • *Vpp Waveform*

Table 1 • *Vpp Waveform*

ACT 1 Programming Algorithm		Current Parameters
V Program	=	21 V
V Precharge	=	12.35 V
V Verify	=	6.0 V
t Program	=	150–300 µs
t Precharge	=	25 µs
I Threshold	=	–2.5 mA (to detect programmed antifuse)
I Max	=	15 mA (clamp current)
# Soak	=	30–800 pulses
Maxpulses	=	60,000

Test Modes of Actel FPGAs

The unique architecture of Actel FPGAs allows outstanding testability of unprogrammed devices at the factory. Details of the various test modes are as follows:

- The shift register circling the periphery of the chip can be both downloaded and uploaded. This allows the use of various test patterns to ensure that the shift register is fully functional.
- All vertical and horizontal tracks can be tested for continuity and shorts. There are several ways to implement these tests. One way of doing continuity testing is to precharge the array, turn on all vertical or horizontal pass transistors on a track, drive the track low from one side of the chip, and read a low on the other side. Shorts can be detected by driving every other track low after precharge and reading back on the other side. Note that these tests also confirm that the vertical and horizontal pass transistors will turn on.
- It is important for programming to make sure that all tracks can hold the precharge level. By charging a track, floating it, and waiting a predetermined amount of time, the track can be read back and confirmed to be still high.
- Leakage of vertical and horizontal pass transistors can be tested by driving one side of a track to a voltage via the Vpp pin and grounding the other side. All pass transistors except the one being tested are turned on. If excess current is detected on the Vpp pin, the pass transistor is considered defective.
- There are one or two dedicated clock buffers that travel across all horizontal channels. These buffers can be tested by driving with the clock pin and reading for the proper levels at the sides of the array.
- There are two special pins referred to as *Probe A* and *Probe B* (Actionprobes). By entering a test mode, the shift register can be made to address the internal output of any logic module. This output is then directed to one of two dedicated vertical tracks, which in turn can be observed externally on the Probe A or Probe B pin. This ability to observe internal signals (even on unprogrammed parts) allows Actel to perform a large number of functional tests. The first such test is the input buffer test. Input buffers on all I/O pins can be tested for functionality by driving at the input pad and reading the internal I/O output node through the probe pins.
- Test modes exist to drive all output buffers low, high, or tristate. This allows testing of Vol, Voh, Iol, Ioh, and leakage on all I/Os.

- One of the key tests is the ability to test functionally all internal logic modules. By turning on various vertical pass transistors and driving from the top or bottom of the chip, any of the eight to ten module inputs can be forced to a high or low. The output of the module can then be read through the Actionprobe pins. The logic module test allows 100 percent fault coverage of each module. In addition, the architecture allows modules to be tested in parallel for reduced test time.
- Actel FPGAs have one or two dedicated columns on the chip that are transparent to the user and used by the factory for speed selection. These columns are referred to as the *Binning Circuit*. Modules in the columns are connected to each other by programming antifuses. The speed of the completed test circuit can then be tested. The Binning Circuit allows the separation of units into different speed categories. It also allows the speed distribution within each category to be minimized.
- There are several tests to confirm that the programming circuitry is working. The first such test is a basic junction stress/leakage test. The program mode is enabled and Vpp voltage plus a guard band is applied to the Vpp pin. All vertical and horizontal tracks are driven to Vpp; thus, no voltage is applied across the antifuses. The Ipp current is then measured. If it exceeds its normal value, the device is rejected.
- There is a test to ensure that all antifuses are not programmed. This is referred to as the *antifuse shorts test* (or *blank test*). The array is precharged, and then the vertical tracks are driven to ground. The horizontal tracks are then read to confirm that they are still high. (A programmed or leaky antifuse would drive a horizontal track low.) The test is repeated by driving horizontal tracks low and reading vertical tracks.
- The functionality of the programming circuitry can be verified by programming various extra antifuses, on the chip, that are transparent to the user. Some of these antifuses were already described earlier when the Binning Circuit was discussed. Actel FPGAs also have a Silicon Signature. In the ACT 1 family, the Silicon Signature consists of four words of data, each word 23 bits in length. The first word is hardwired (no antifuses) and contains a manufacturer ID number as well as a device ID number. These numbers can be read by a programmer, and the proper programming algorithm can be automatically selected. The other words contain antifuses and are programmable. Actel is currently using bits in these words to store information such as the chip's run number and wafer number. Thus, each Actel FPGA has traceability down to the wafer level. By programming this information, the functionality of the programming circuitry is also tested. Actel software also allows the user to program a design ID and check sum into the Silicon Signature. By later reading this back, the user can verify that the chip is correctly programmed to a given design.
- The most important antifuse test is the stress test. When this test is enabled, a voltage applied to the Vpp pin can be applied across all antifuses on the chip. (The other side is grounded.) The voltage applied is the precharge voltage plus a significant guard band. After the voltage is applied, the antifuse shorts test is again used to make sure no antifuses have been programmed. The antifuse stress test is effective at catching antifuse defects. Because the reliability of the antifuse is much more voltage dependent than it is temperature dependent, this test is also an effective antifuse infant mortality screen. See the "Actel Device Reliability Report" for details.

Burn-In of Actel FPGAs

As mentioned earlier, Actel has found that antifuse infant mortality failures can be effectively screened out during electrical testing, and it is thus unnecessary to do any kind of burn-in for standard commercial production units to screen out antifuse infant mortality failures. However, burn-in is still an effective screen for standard CMOS infant mortality failure mechanisms, and it is required for all military 883D products. MIL-883D Method 1005 allows several types of burn-in screens. These can be divided into two categories: *steady state* (*static*) and *dynamic*. Static burn-in applies DC voltage levels to the pins of the device under test. The device may or may not be powered up. Dynamic burn-in applies AC signals to device inputs with the unit powered up. These signals are selected so that the device receives internal and external stresses similar to those it may see in a typical application.

Static burn-in is by far the simplest to implement. By choosing appropriate biasing conditions and load resistors, it is possible to design a single burn-in circuit that can be used for both unprogrammed and programmed devices. It would not matter what pattern is programmed into the device. Static burn-in can be an effective screen for some types of failure modes, particularly those that may happen at device inputs or outputs (such as screening for mobile ionic contamination). It is not, however, very effective at stressing internal device circuits. Many internal nodes may be biased at ground without receiving any voltage or current stress. Signal lines will not toggle, and it may not be possible to screen failure modes such as metal electromigration.

A properly designed dynamic burn-in can effectively stress inputs, outputs, and internal circuits. However, dynamic burn-in of ASIC products can be very expensive because customer-specific burn-in circuits and burn-in boards must be designed and built to properly stress each design implemented in the ASIC. This results in large NRE costs and long lead times to design and build these boards. From the standpoint of burn-in, a programmed FPGA is essentially the same as a mask-programmed ASIC, and it would require similar custom burn-in circuits to do a dynamic burn-in. However, Actel has been able to use the testability features of its FPGA products to allow effective dynamic burn-in of unprogrammed devices. This dynamic burn-in allows users to stress circuits in a way that static burn-in would be unable to duplicate.

During burn-in of unprogrammed units, test commands are serially shifted into each device by using the SDI pin and clocked by using the DCLK pin. There are three test modes shifted into each device. The first test stresses each cross-antifuse with a voltage of Vpp – 2V. (Vpp is normally set at 7.5–11V so that each antifuse gets 5.5–9V across it.) This voltage is applied to all vertical tracks while the horizontal tracks are grounded. Once enabled, the stress mode is held for 10 ms.

The second test mode is identical to the first except that the horizontal tracks are driven to Vpp – 2V while the vertical tracks are grounded. Note that both of these modes are similar to the antifuse stress test described earlier (although the stress voltage is lower during burn-in). Not only do these tests stress the antifuses, but they also toggle all routing tracks in the chip to Vpp – 2V and ground. All input and output tracks to the logic modules are also toggled.

The third test drives several I/O pins on the chip to a low state. Prior to this, they are at high impedance state and held at Vcc through pull-up resistors. This test confirms that the burn-in is being properly implemented by looking at these I/O pins to see if they display the proper waveform. It also passes current through each I/O as it toggles low.

Although the chip is unprogrammed, these tests allow users to apply stresses to the inputs, outputs, and internal nodes that are similar to what a programmed device may see in normal operation. Once burn-in is completed, post-burn-in testing, as specified by MIL-883D, is performed (including PDA) to ensure that fully compliant devices are shipped to the customers.

Conclusion

The description of the Actel FPGA architecture and the numerous test modes attest to the outstanding testability of these devices. All internal logic gates can be tested without programming antifuses other than the few for the Binning Circuit and Silicon Signature. Because Actel FPGAs are one-time programmable, the only item that is not fully tested at the factory is the programmability of all the individual antifuses. However, this is done on the programmer while the units are being programmed. Being able to test all internal gates allows Actel to achieve functional yields superior to other one-time programmable devices and equivalent to reprogrammable parts.

References:

1. Henshaw, "User Requirements for Fault Coverage," Wescon Proceedings, 1990, p. 179.
2. AMD PAL Device Data Book, 1988, p. 3-106.

Actel Device Reliability Report

Actel's field programmable gate arrays (FPGAs) are currently available in five product families—ACT 1, ACT 2, 1200XL, 3200DX, and ACT 3. The ACT 1 family consists of the A1010 and A1020, which are 1200- and 2000-gate FPGAs, respectively. The ACT 2 family includes the A1225, A1240, and A1280 FPGAs, which offer 2500, 4000, and 8000 gates, respectively. The 1200XL products (A1225XL, A1240XL, and A1280XL) are compatible with ACT 2 but offer greatly improved performance. The A3265DX and A32140DX are the first two products of the 3200DX family. They have 6500 and 14,000 gates, respectively, although the product family will soon extend to 40,000 gates. The ACT 3 family consists of products ranging from 1500 to 10,000 gates (A1415, A1425, A1440, A1460, and A14100).

The programming element for all Actel FPGAs is an Actel-invented PLICE (Programmable Low-Impedance Circuit Element) antifuse. An antifuse is a device that is normally open and in which an electrical connection is established by applying programming voltage. Although Actel FPGAs are one-time programmable devices, their unique architecture includes complete functional testability.

With time, improvements in processing technology have allowed all products to evolve to tighter design rules (reduced line widths and spacing), smaller die size, improved performance, and reduced cost. ACT 1 products were originally manufactured using 2 μm design rules (A1010/A1020) and are no longer in production. Later they were linearly shrunk in size by 20 percent to 1.2 μm (A1010A/A1020A). Currently, the family is being manufactured using aggressive 1.0 μm design rules (A1010B/A1020B). ACT 2 products were first introduced with 1.2 μm design rules (A1225/A1240/A1280) and, as a result of a 20 percent linear shrink, are now available in 1.0 μm versions (A1225A/A1240A/A1280A). The 1200XL and 3200DX families offer 0.6 μm design rules. ACT 3 products were originally manufactured with a 0.8 μm process and are now available with 0.6 μm processing. In all cases, the technology is a standard double-metal, twin-well, CMOS process in which three additional masking steps have been added to implement the PLICE antifuse. The main process parameters are shown in Table 1. Because Actel FPGAs are manufactured with a conventional CMOS process, normal CMOS failure modes are observed. However, the addition of the antifuse involves another structure that could affect the device's reliability.

The PLICE Antifuse

The antifuse is a vertical, two-terminal structure. It consists of a polysilicon layer on top, N+ doped silicon on the bottom, and an ONO (oxide-nitride-oxide) dielectric layer in-between. The size of the antifuse is 1.4 μm2, 1.0 μm2, 0.8 μm2, and 0.6μm2 for the 1.2 μm, 1.0 μm, 0.8 μm, and 0.6 μm processes, respectively. This small size, along with a low programmed on-resistance (typically 150–200 ohms in speed sensitive paths) makes the PLICE antifuse an attractive alternative to EPROM, EEPROM, or RAM as a programming element in a large programmable gate array. In the unprogrammed state, the resistance of the antifuse is more than 100 megohms. Actel FPGAs may contain anywhere from 112,000 antifuses (A1010) to 1,250,000 (32140DX). However, typical applications that use 85 percent of the available gates require you to program only 2 to 3 percent of the available antifuses.

To quantify the reliability of the antifuse, Actel has completed numerous studies. These lead to the conclusion that the time-to-failure of the antifuse is substantially more than 40 years under normal operating conditions and that the combined contribution of all antifuses to the gate array product's hard failure rate is less than 10 FITs (failures-in-time, or 0.001% failures per 1000 hours). Detailed reliability reports specifically addressing the antifuse can be obtained from your local Actel sales representative.

Actel FPGA Product Reliability

To date, product reliability has been evaluated on 18 Actel products (1.2 μm process and smaller): 1200-gate FPGAs (A1010A, A1010B); 2000-gate FPGAs (A1020A, A1020B); 2500-gate FPGAs (A1225, A1225A, A1225XL, A1425, A1425A); 4000-gate FPGAs (A1240, A1240A, A1440A); a 6000-gate FPGA (A1460A); a 6500-gate FPGA (A3265DX); 8000-gate FPGAs (A1280, A1280A, A1280XL); and a 10,000-gate FPGA (A14100A). Reliability tests were conducted on units assembled in many different package types, as listed in Tables 4 through 9. Package characteristics for Actel FPGAs are shown in Table 2.

High Temperature Operating Life (HTOL)

The intent of an HTOL test is to operate a device dynamically (device is powered up with I/Os and internal nodes toggling to simulate actual system use) at a high temperature (usually

Table 1 • ACT 1 and ACT 2 Process Description

Dimensions	1.2 µm Process		1.0 µm Process		0.8 µm Process		0.6 µm Process	
	Width (µm)	Space (µm)	Width (µm)	Space (µm)	Width (µm)	Space (µm)	Width (µm)	Space (µm)
N +	3.2	1.6	2.3	1.2	1.8	1.4	1.8	1.4
p +	3.2	1.6	2.3	1.2	1.8	1.4	1.8	1.4
Cell PolySilicon	2.1	2.4	1.5	0.9	1.2	1.2	1.2	1.2
Gate PolySilicon	1.6	1.6	1.2	1.2	1.0	1.2	1.0	1.2
Metal I	3.2	1.6	2.2	1.2	1.6	1.2	1.6	1.2
Metal II	3.8	1.8	2.1	1.6	1.6	1.4	1.6	1.4
Contact	1.2 x 1.2	1.8	1.0 x 1.0	1.0	1.0 x 1.0	1.2	1.0 x 1.0	1.2
Via	1.3 x 1.3	1.9	1.0 x 1.0	1.0	1.0 x 1.0	1.2	1.0 x 1.0	1.2
Thickness	**1.2 µm Process**		**1.0 µm Process**		**0.8 µm Process**		**0.6 µm Process**	
Normal Gate Oxide	25 nm		20 nm		18 nm		18 nm	
High-Voltage Gate Oxide	40 nm		32.5 nm		35 nm		35 nm	
Cell PolySilicon	450 nm		400 nm		400 nm		400 nm	
Gate PolySilicon	450 nm		400 nm		400 nm		400 nm	
Metal I	900 nm		720 nm		750 nm		750 nm	
Metal II	1000 nm		1050 nm		1050 nm		1050 nm	
Passivation	1100 nm		1100 nm		1100 nm		1100 nm	
Compositions								
Metal I	Al–SI (1%)–Cu (0.5%)		Ti/TiN/Ai–Si (1%)–Cu (0.5%)		Ti/TiN/Ai–Si (1%) –Cu (0.5%)		Ti/TiN/Ai–Si (1%) –Cu (0.5%)	
Metal II	Al–SI (1%)–Cu (0.5%)		Ti/Al–Si (1%)–Cu (0.5%)		Ti/Al–Si (1%)–Cu (0.5%)		Ti/Al–Si (1%)–Cu (0.5%)	
Passivation	300 nm SiO_2, 800 nm SiN		300 nm SiO_2, 800 nm SiN		300 nm SiO_2, 800 nm SiN		300 nm SiO_2, 800 nm SiN	

Note: *Process parameters can vary. For more specific details, contact your local Actel sales representative.*

125°C or 150°C) and extrapolate the failure rate to typical operating conditions. This test is defined by Military Standard-883 in the Group C Quality Conformance Tests. The Arrhenius equation is used to do the extrapolation:

$$R = R0 * \exp (Ea/kT)$$

where R is the failure rate, R0 is a constant for a particular process, T is the absolute temperature in degrees Kelvin, k is Boltzmann's constant ($8.62 \text{ X } 10^{-5}$ eV/°K), and Ea is the activation energy for the process in electron volts.

To determine the acceleration factor for a given failure mode at temperature T2 as compared with temperature T1, we derive from the Arrhenius equation:

$$A(T1, T2) = \exp[(Ea/k) * \{(1/T1)-(1/T2)\}]$$

To use the Arrhenius equation, we need to know the activation energy of the failure mode. Table 3 gives the activation energies of general semiconductor failure modes.

For evaluating the Actel FPGA products, we program an actual design application into most devices (some units are burned in unprogrammed) and perform a dynamic burn-in by toggling the clock pins at 1 MHz or higher. The designs selected use 85 to 97 percent of the available logic modules and 85 to 94 percent of the I/Os. Outputs are loaded with 1.2–2.8K ohm resistors to Vcc, which results in up to 4 mA of sink current as each I/O toggles low. Under these conditions, each unit typically draws a minimum of 100 mA during dynamic burn-in. Most of this current comes from the output loading, and about 5 mA is from the device supply current. For a 125°C burn-in, this results in junction temperatures of at least 150°C for plastic packages and 145°C for ceramic packages (depending on package type). Because junction temperatures can vary a great deal due to package, product, frequency, design, voltage, and other factors, we use ambient temperature to calculate failure rates. This is worst case, because ambient temperature is always lower than junction temperature, and there is thus less acceleration when extrapolating device hours at lower temperatures. Most burn-in was done at 5.75 V or 6.0 V (for voltage acceleration of the antifuse) and 125°C or 150°C.

As mentioned previously, some units are burned in unprogrammed. To accomplish this, we use a special burn-in circuit that allows us to take advantage of the product's test features to shift serially in commands to the chip during burn-in. All internal routing tracks are toggled between Vss and Vcc. When vertical tracks are at Vcc, horizontal tracks are held at Vss, and vice versa. Thus, all antifuses that can connect vertical and horizontal tracks receive a full Vcc stress in both directions. These toggling vertical tracks connect to logic module inputs and outputs when a part is

Table 2 • Actel FPGA Package Characteristics

Characteristics	PLCC	PQFP
Molding Compound	Sumitomo 6300 H	Sumitomo 6300 H
Filler Material	Fused silicon 70% by weight	Fused silicon 70% by weight
Lead Frame Material	Copper (Olin 150 or equiv.)	Copper (Olin 150 or equiv.)
Lead Plating Composition	Solder, 300–800 micro inches (µin)	Solder, 300–800 micro inches (µin)
Die Attach Material	Silver Epoxy	Silver Epoxy
Flame Retardance	UL-94, V-0	UL-94, V-0
Bond Wire	Gold, 1.3 mil diameter	Gold, 1.3 mil diameter
Bond Attach Method	Thermosonic	Thermosonic
Characteristics	**JQCC**	**PGA/CQFP**
Body Material	Ceramic	Alumina
Lid Material	Ceramic	Kovar, 50 µin gold plated
Sealant	Glass	Au/Sn Solder
Lead Frame Material	Alloy 42 (40% Nickel, 60% Iron)	N/A
Bond Wire	99% Aluminum, 1% Si, 1.25 mil diameter	99% Aluminum, 1% Si, 1.25 mil dia
Bond Attach Method	Ultrasonic	Ultrasonic
Lead Finish	Solder dip, 200 µin min	A42 or Kovar
Die Attach Material	Silver loaded glass	Silver loaded glass

Thermal Resistance (°C/Watt)

Package	44 PLCC	44 JQCC	68 PLCC	68 JQCC	80 VQFP	84 PLCC	84 JQCC	84 PGA	84 CQFP
θ_{JC}	15	8	13	8	12	12	8	8	5
θ_{JA} (still air)	52	38	45	35	68	44	34	33	40
Package	**100 PQFP**	**100 PGA**	**132 PGA**	**144 PQFP**	**160 PQFP**	**172 CQFP**	**176 PGA**	**207 PGA**	**208 PQFP**
θ_{JC}	13	5	5	15	15	8	8	8	15
θ_{JA} (still air)	55	35	30	35	33	25	23	22	33

Table 3 • CMOS Failure Modes

Failure Mechanism	Activation Energy
Ionic Contamination	1.0 eV
Oxide Defects	0.3 eV
Hot Carrier Trapping in Oxide (Short Channels)	–0.06 eV
Silicon Defects	0.5 eV
Aluminum-Silicon-Copper Electromigration	0.6 eV
Contact Electromigration	0.9 eV
Electrolytic Corrosion	0.54 eV

programmed. Finally, a command is sent to the chip to toggle some external I/O pins between Vss and Vcc. This special dynamic burn-in circuit is the same one used by Actel to screen unprogrammed products to MIL-STD-883 requirements. Since virtually all antifuses receive a full Vcc stress, this screen is much more effective at catching unprogrammed antifuse infant mortality failures than is burning in programmed devices in which only a fraction of the antifuses are stressed.

A summary of the HTOL data collected by Actel is shown in Table 4. A failure is defined as any device that shows a functional failure, exceeds data sheet DC limits, or exhibits AC speed drift. Among the parts tested, no speed drift, faster or slower, was observed within the accuracy of the test setup. Failure rates at 55°C, 70°C, and 90°C were extrapolated by using the Arrhenius equation, and general activation energies of 0.6 eV and 0.9 eV. Poisson statistics were used to derive a calculated failure rate with a 60 percent confidence level. Using Poisson statistics is valid for a failure rate that is low and a failure mode that occurs randomly with time. At 55°C, the calculated failure rate with 60 percent confidence level (0.6 eV) was found to be 19 FITs, or 0.0019 percent failures per 1000 hours. This number was derived from nearly nine million device hours (125°C) of data. There were seven total failures out of 7704 tested units representing 134 different wafer lots. There were no infant mortality failures, which would normally occur in the first 80 hours of burn-in. Six of the seven failures observed were due to common CMOS failure modes (gate oxide failure, silicon defects, or open via). Only one unit failed due to the antifuse. This unit was burned-in in the unprogrammed state to stress all antifuses. It was stressed at 6 V, 150°C. It passed at 168 hours and failed at 650 hours because an antifuse became programmed. By passing at 168 hours, the unit received a total stress well in excess of 100 years of operation at 5.5 V, 125°C. With only one antifuse-related failure in nine million device hours at 125°C, we can derive that this represents a product antifuse failure rate of significantly less than 10 FITs at 5.5 V.

Unbiased Pressure Pot Test

This test is used to qualify products in plastic packages. Units are placed into an autoclave (pressure pot) and exposed to a saturated steam atmosphere at 121°C and 15 psi. Problems with bonding, molding compounds, or wafer passivation can cause metal corrosion to occur in this atmosphere. The existence of metal corrosion is detected during a full electrical test of the device following exposure in the autoclave.

Table 4 • High Temperature Operating Life (HTOL) Summary

Product	Process	Total Units	Total Wafer Runs	Device Hours at 125°C (0.6 eV)	Total Failures	Equiv. Dev. Hrs. at 55°C (0.6 eV)	Equiv. Dev. Hrs. at 55°C (0.6 eV)	Equiv. Dev. Hrs. at 55°C (0.6 eV)	Packages used for HTOL
1010A	1.2 um	618	13	8.20E+05	1	3.43E+07	1.36E+07	4.43E+06	68 PLCC
1010B	1.0 um	612	6	5.84E+05	1	2.44E+07	9.65E+06	3.15E+06	68 PLCC, 100 PQFP
1020A	1.2 um	1720	29	2.61E+06	3	1.09E+08	4.32E+07	1.41E+075	84 PLCC, 100 PQFP, 84 JLCC, 84 PGA
1020B	1.0 um	1128	19	1.07E+06	1	4.46E+07	1.76E+07	0.76E+06	84 PLCC, 100 PQFP, 80 VQFP, 84 PGA
1225	1.2 um	208	1	2.07E+05	0	8.66E+06	3.42E+06	1.12E+06	100 PQFP, 100 PGA
1225A	1.0 um	80	1	8.00E+04	0	3.35E+06	1.32E+06	4.32E+05	100 PQFP
1225XL	0.6 um	26	1	2.60E+04	0	1.09E+06	4.30E+05	1.40E+05	100 PQFP
1240	1.2 um	485	7	6.11E+05	0	2.55E+07	1.01E+07	3.30E+06	132 PGA, 144 PQFP, 84 PLCC
1240A	1.0 um	210	4	2.40E+05	0	1.00E+07	3.97E+06	1.30E+06	144 PQFP, 84 PLCC
1240XL	0.6 um	52	2	5.20E+04	0	2.18E+06	8.60E+05	2.81E+05	144 PQFP
1280	1.2 um	586	13	7.03E+05	1	2.94E+07	1.16E+07	3.80E+06	176 PGA, 160 PQFP
1280A	1.0 um	639	13	5.21E+05	0	2.18E+07	8.62E+06	2.82E+06	160 PQFP, 176 PGA, 84 PLCC
1280XL	0.6 um	127	5	2.03E+05	0	8.49E+06	3.36E+06	1.10E+06	160 PQFP
3265DX	0.6 um	78	1	7.80E+04	0	3.26E+06	1.29E+06	4.21E+05	160 PQFP
1425	0.8 um	275	3	2.75E+05	0	1.15E+07	4.55E+06	1.49E+06	133 PGA, 84 PLCC
1425A	0.8 um	260	6	2.60E+05	0	1.09E+07	4.30E+06	1.40E+06	133 PGA
1440A	0.8 um	158	2	1.58E+05	0	6.61E+06	2.61E+06	8.54E+05	100 VQFP
1460A	0.8 um	367	6	3.98E+05	0	1.67E+07	6.59E+06	2.15E+06	207 PGA, 208 PQFP
14100A	0.8 um	76	2	7.60E+04	0	3.18E+06	1.26E+06	4.11E+05	208 RQFP
Totals:		**7704**	**134**	**8.97E+06**	**7**	**3.75E+08**	**1.48E+-8**	**4.85E+07**	

Overall FITs:

Ambient Temperature	Activation Energy	Observed	60% Confidence
55°C	0.6 eV	19	22
70°C	0.6 eV	47	57
90°C	0.6 eV	144	173
55°C	0.9 eV	3	3
70°C	0.9 eV	11	13
90°C	0.9 eV	57	68

A total of 1453 units from 39 wafer runs were evaluated. Read points were taken at 96, 168, 240, and 336 hours. There were a total of two failures (Table 5). Both units were A1010As from the same wafer lot and assembly lot. The units failed at 336 hours because of metal corrosion at the bond pads, but both units had passed at 168 hours.

Table 5* • *121°C, 15 PSI Steam Pressure Pot (Unbiased Autoclave)

				Number of Failures (Hours)			
Product	Run Number	Package	Number of Units	96	168	240	336
1010A	TI15	68 PLCC	77	0	0	—	0
1010A	TI24	68 PLCC	129	0	0	—	0
1010A	TI1104	68 PLCC	77	0	—	0	0
	TI1243						
	TI1263						
	TI1297						
1010A	E01-1	68 PLCC	30	0	0	—	0
1010A	E02-1	68 PLCC	30	0	0	—	0
1010A	E03-1	68 PLCC	30	0	0	—	2
1010B	TI2072857	68 PLCC	45	0	0	0	
	TI2072858						
	TI2072860						
1010B	U1G-01	68 PLCC	40	—	0	—	—
1020A	TI1800	84 PLCC	77	0	—	0	—
	TI1859						
	TI2156						
1020A	E-01	84-PLCC	26	0	0	—	0
	E-02	84-PLCC	28	0	0	—	0
	E-03	84-PLCC	26	0	0	—	0
1020A	ADK29	84-PLCC	27	—	0	—	
	ADC21X	84-PLCC	27	—	0	—	
	ADA72X	84-PLCC	27	—	0	—	
1020A	JF-207	100 PQFP	80	—	0	—	—
1020A	S-1702A	84-PLCC	25	—	0	—	0
	S-1702B		26	—	0	—	0
	S-1702C		25	—	0	—	0
1020B	JJ14-17	84-PLCC	81	—	0	—	
1020B	U1P-01	84-PLCC	40	—	0	—	
1225	UJ-01	100 PQFP	80	—	0	—	0
1240	TI10301	144 PQFP	79	—	0	—	0
1240	UI-03	84-PLCC	79	—	0	—	0
1240A	E-02	144 PQFP	25	—	0	—	
1240A	E-03	144 PQFP	30	—	0	—	
1240A	E-04	84 PLCC	25	—	0	—	
1280	JH-14	160 PQFP	80	—	0	—	0
1280	ADC18X	160 PQFP	82	—	0	—	

FAILURE ANALYSIS (AUTOCLAVE):

Product	Run Number	Hours	Cause
1010A	E03-1	336	Continuity due to corroded pads
1010A	E03-1	336	Continuity due to corroded pads

Biased Moisture Life Test (85/85)

In this test, the units are placed into a chamber at a temperature of 85°C and a relative humidity of 85 percent. A voltage of 5.5 V is applied to every other device pin while other pins are grounded. The same voltage is applied to Vcc while Vss is grounded. This test effectively detects die-related and plastic-package-related problems.

As shown in Table 6, a total of 906 units have been stressed. There have been no failures.

Table 6 • 85°C/85% Humidity with DC Alternate Pin Bias of 0 V to 5.5 V

				Number of Failures (Hours)			
Product	Run Number	Package	Number of Units	168	500	1000	2000
1010A	E01-1	68 PLCC	79	—	0		
1010A	E02-2	68 PLCC		—	0		
1010A	E03-1	68 PLCC		—	0		
1010A	TI1104	68 PLCC	80	0	0	0	—
	TI1243						
	TI1263						
	TI1297						
1010B	2072857	68 PLCC	201	0	0	0	—
	2072858						
	2072860						—
1020A	E-01	84 PLCC	64	—	0	0	
	E-02	84 PLCC					
	E-03	84 PLCC					
1020A	ADK29	84 PLCC	27	—	0	0	
	ADC21X	84 PLCC	27	—	0	0	
	ADA72X	84 PLCC	27	—	0	0	
1020A	E14	84 PLCC	24	—	0	0	
	E15	84 PLCC	29	—	0	0	
	E17	84 PLCC	32	—	0	0	
1020A	UP-06	100 PQFP	77	—	0	0	
1020B	JJ-14	84 PLCC	27	—	—	0	
	JJ-15	84 PLCC	27	—	—	0	
	JJ-17	84 PLCC	27	—	—	0	
1240	TI3256	144 PQFP	78	—	0	0	
1280	UH-04	160 PQFP	80	—	0	0	

Highly Accelerated Stress Test (HAST) at 131°C/85% Humidity

As in the 85/85 test, units receive an alternate pin bias (5.5 V and 0 V) but are exposed to a higher pressure and a higher temperature environment. Fifty hours of HAST is generally considered to be equivalent to 1000 hours of 85/85. HAST testing has gained wide industry acceptance, and Actel is currently using HAST testing in place of both 85/85 and autoclave in most qualifications. As summarized in Table 7, 1899 units from 51 wafer runs have been tested with only one failure, which occurred at 100 hours.

Temperature Cycling/Thermal Shock

These tests check for package integrity by cycling units through temperature extremes. Data was taken for cycles of 0°C to 125°C, –40°C to 125°C, and –65°C to 150°C. Both programmed and unprogrammed units were placed on temperature cycle. As shown in Table 8, of 4760 units tested to date, there has been only one failure. We have also performed –65°C to 150°C thermal shock tests on 272 units (Table 9) and have seen no failures.

Table 7 • Biased Humidity (HAST) 131°C/85% Humidity

			Number of	Number of Failures (Hours)					
Product	Run Number	Package	Units	50	100	200	240	300	400
1020A	TI1130 TI1139 TI1210	84 PLCC	77	—	0	0	0	—	
1020A	U1P05	100 PQFP	45	—	0				
1020A	U1P41HM	100 PQFP	81	—	0				
1020A	U1P-209B	84-PLCC	15	—	0				
1020B	EBFJ001	84-PLCC	44	—	0				
	EBFI004	84-PLCC	36	—	0				
1020B	U9P01	84-PLCC	29	—	0				
	U9P021A	84-PLCC	50	—	0				
1225	UJ-03	100 PQFP	80	0	0				
1225XL	ACP02187.1	100 PQFP	17	—	0				
1240	UI-03	84-PLCC	80	0	0				
1240A	E-04	84-PLCC	77	0	—				
1240A	E-02,03	144 PQFP	53	0	—				
1240A	U1I26	144 PQFP	80	—	0				
1240XL	ACP01117.1 ACN51939.1	144 PQFP	31	—	0				
1280	ADC20X	160 PQFP	80	0					
1280A	U1H-01	160 PQFP	40	0	—				
1280A	U1H-02	160 PQFP	40	0	—				
'280A	EBFJ002	160 PQFP	30	0	—				
1280A	EBFJ003	160 PQFP	30	0	—				
1280A	EBFJ004	160 PQFP	20	0	—				
1280A	U1H-18	160 PQFP	80	—	0				
1280A	EWAJ003,4	160 PQFP	79	—	0				
1280A	UIH235/6	160 PQFP	65	—	1				
1280XL	24381610	160 PQFP	11	—	0				
	24464430	160 PQFP	18	—	0				
	24442620	160 PQFP	17	—	0				
1280XL	ACP19329.1	160 PQFP	26	—	0				
	ACP212072.2	160 PQFP	25	—	0				
3265DX	ACP163684	160 PQFP	40	—	0				
1425	JK8,9,10	84-PLCC	81	—	0				
1425A	ACN32804 ACN30805 ACN33807	100 PQFP	80	—	0				
1425A	UCJ01,2,3	100 PQFP	80	—	0				
1440A	JN05	100 VQFP	45	—	0				
1460A	JL04A	208 PQFP	80	—	0				
1460A	WB24279010	208 PQFP	47	—	0				
14100A	24239130	208 RQFP	14	—	0				
	UCLO1	208 RQFP	31	—	0				

Failure Analysis (HAST):

Product	Run Number	Hours	Cause
1280A	UIH235/6	100	Unable to perform failure analysis; unit damaged during decapsulation

Table 8 • Temperature Cycle

				Number of Failures (Cycles)			
Product	Run Number	Package	Number of Units	100	500	1000	2000
0°C to 125°C Cycles							
1010A	TI15	68 PLCC	125	0	—	0	—
1010A	TI24	68 PLCC	176	0	—	0	—
1010A	TI1104 TI1243 TI1263 TI1297	68 PLCC	129	0	0	0	0
1020A	TI1800 TI1859 TI2156	84 PLCC	129	0	0	0	0
–40°C to 125°C Cycles							
1010A	TI1104 TI1243 TI1263 TI1297	68 PLCC	129	0	0	0	0
1020A	TI1800 TI1859 TI2156	84 PLCC	129	0	0	0	0
–65°C to 150°C Cycles							
1010A	TI1104 TI1243 TI1263 TI1297	68 PLCC	129	0	0	0	0
1010B	TI2072857 TI2072858 TI2072860	68 PLCC	201	0	0	0	
1010B	U1G-01,02	68 PLCC	40	—	—	0	
1020A	TI1800 TI1859 TI2156	84 PLCC	129	0	0	0	0
1020A	JF-71	100 PQFP	129	—	—	0	
1020A	E01 E02 E03	84-PLCC	85	—	—	0	
1020A	S-1702A,B,C	84-PLCC	144	0	—	0	
1020B	JJ14-17	84-PLCC	81	—	—	0	
1020B	U1P-01,02	84-PLCC	40	—	—	0	
1020B	EBFJ001	84-PLCC	80	—	—	0	
1020B	U1P41HM	100 PQFP	80	—	—	0	
1020B	EWAI003	84-PLCC	80	—	—	0	
1020B	U1P-209B	84-PLCC	15	—	—	0	
1020B	U1P05	100 PQFP	80	—	—	0	
1020B	U9P021A	84-PLCC	55	—	—	0	
	U9P01		23	—	—	0	

Table 8 • Temperature Cycle (continued)

				Number of Failures (Cycles)			
Product	Run Number	Package	Number of Units	100	500	1000	2000
–65°C to 150°C Cycles							
1225	UJ-01	100 PGA	80	0	—	0	
1225	UJ-01	100 PQFP	80	—	—	0	
1225XL	ACP02187.1	100 PQFP	18	—	0	0	
1240	TI1053932	132 PGA	15	0	0	0	
	TI1045571		20				
	TI1053933		42				
1240	TI 3256	144 PQFP	80	—	0	0	
1240	UI-01	132 PGA	50	0	—	0	
1240	UI-02	132 PGA	50	0	—	0	
1240	UI-03	84 PLCC	80	—	—	0	
1240A	E-02	144 PQFP	25	—	—	0	
1240A	E-03	144 PQFP	28	—	—	0	
1240A	E-04	84 PLCC	77	—	—	0	
1240A	U1I-26	144 PQFP	80	—	—	0	
1240XL	ACP01117.1	144 PQFP	30	—	0	0	
	ACN51939.1						
1280	TI1136307	176 PGA	71	0	—	0	
	TI1143650	176 PGA	5	0	—	0	
	TI1143649	176 PGA	1	0	—	0	
1280	UH-01	176 PGA	25	0	—	0	
1280	UH-02	176 PGA	26	0	—	0	
1280	UH-04	160 PQFP	34	0	—	0	
1280	JH-14	160 PQFP	48	0	—	0	
1280	ADE16X	160 PQFP	27	—	—	0	
1280	ADC18X	160 PQFP	30	—	—	0	
1280	ADC20X	160 PQFP	27	—	—	0	
1280A	U1H-01	160 PQFP	40	—	—	0	
1280A	U1H-02	160 PQFP	40	—	—	0	
1280A	U1H-18	160 PQFP	80	—	—	0	
1280A	EWAF003	160 PQFP	75	—	—	0	
1280A	EWAJ03,4	160 PQFP	80	—	—	0	
1280A	U1H-235/6	160 PQFP	30	—	—	0	
1280A	U1H262	176 TQFP	79	—	0	0	
1280XL	25026540	176 TQFP	17	—	0	0	
1280XL	24484820	160 PQFP	31	—	1	1	
	24464430						
1280XL	24485920	84 PLCC	24	0			
	24484810	84 PLCC	24	0			
1280XL	24524350	160 PQFP	24	0			
	25026550	160 PQFP	24	0			
1280XL	24381610	160 PQFP	25	0			
	24464430						
	24442620						
1280XL	ACP19329.1	160 PQFP	26	0	0	0	
	ACP212072.2	160 PQFP	25	0	0	0	
3265DX	ACP163684	160 PQFP	45	—	—	0	

Table 8 • Temperature Cycle (continued)

				Number of Failures (Cycles)			
Product	Run Number	Package	Number of Units	100	500	1000	2000
–65°C to 150°C Cycles							
1425	JK8,9,10	133 PGA	81	—	—	0	
1425	JK8,9,10	84 PLCC	83	—	—	0	
1425A	UCJ01,2,3	100 PQFP	80	—	—	0	
1425A	ACN32804 ACN30805 ACN33807	100 PQFP	80	—	—	0	
1440A	JN-02	160 PQFP	80	—	—	0	
1440A	JN-05	100 VQFP	80	—	—	1	
1440A	51940.1	100 VQFP	45	—	—	0	
1460A	JL-01	208 PQFP	80	—	—	0	
1460A	JL-01	207 PGA	80	—	—	0	
1460A	PC435091	207 PGA	80	—	—	0	
1460A	PC435092	207 PGA					
1460A	PC435093	207 PGA					
14100A	24239130	208 RQFP	14	—	—	0	
	UCLO1	208 RQFP	31	—	—	0	

Failure Analysis (Temperature Cycle):

Product	Run Number	Cycles	Cause
1440A	JN-05	1000	In failure analysis

Table 9 • Thermal Shock –65°C to 150°C

				Number of Failures (Cycles)				
Product	Run Number	Package	Number of Units	100	200	500	1000	2000
1010A	TI1104 TI1243 TI1263 TI1297	68 PLCC	77	0	—	—	0	0
1010B	TI2072857 TI2072858 TI2072860	68 PLCC	45	0	0	—	0	
1020A	TI1800 TI1859 TI2156	84 PLCC	77	0	—	—	0	0
1240	TI1149935		43	—	0			
1280	UH-01,02	176PGA	30	0	—			

Other Tests

A variety of other tests are performed on Actel devices. These include electrostatic discharge, latch-up, and radiation hardness tests.

Electrostatic Discharge (ESD)

All Actel products contain static electricity protection circuitry and are tested for sensitivity to static electricity by using the human body model as described in MIL-STD-883D (100 pf discharged through 1.5 Kohms). Three positive and three negative pulses are discharged into each pin tested at each voltage level. For inputs and I/Os, these six pulses are applied with three different grounding conditions: Vss only grounded, Vcc only grounded, and all other I/Os grounded. Thus, each pin receives a total of 18 pulses for each test voltage. After pulsing, the units are then tested on a VLSI tester. Leakage currents are measured at 0 V and 5.5 V. Any pin showing more than 1 µA of leakage is considered to be a failure. To date, all Actel products pass ESD testing at 1500 V or higher. Newer 1.0µm and 0.8µm products exceed 2000V on all I/Os. For further information about specific products and packages, please contact Actel.

Latch-Up

Latch-up is a well known cause of failure in CMOS circuits. Parasitic bipolar transistors are created by the P-channel transistors, the N-channel transistors, the N-well, and the P-substrate. These transistors are connected in a manner that effectively creates an SCR. If a voltage on an external pin forwards bias to the substrate, the parasitic SCR can be latched to the on state, thereby creating a low-impedance path between Vcc and ground. A large amount of current then flows through this path. This current can, at best, temporarily make the device nonfunctional and, at worst, cause permanent damage.

There are several techniques used by CMOS designers to reduce the chance of latch-up. One of the most common techniques is to use guard rings to isolate P-channel and N-channel transistors. The disadvantage of this method is that it requires additional silicon die area. Another method is to use a substrate bias generator. Creating a negative substrate bias means that an input must go even more negative to cause latch-up. A third technique is to use EPI wafers to achieve low substrate resistance, which lowers the chance of triggering latch-up. Actel uses both guard ring and EPI wafer techniques for all FPGA devices.

The latch-up test method is defined by JEDEC Standard No. 7. Each I/O pin on a tested device is forward biased in both directions (to Vss and Vcc) by forcing negative and positive currents ranging from ±50 mA to ±400 mA in 50 mA increments. Following each stress, the device Icc current is measured. If the current exceeds the data sheet limit of 10 mA, the unit is rejected. The device is also functionally tested.

Six units from three different wafer lots are tested to qualify each Actel product. Testing is done at room temperature as well as at a worst-case temperature of 125°C. All device I/Os and power supplies are tested. To date, all products pass a minimum of 250 mA.

Radiation Hardness

A programmed antifuse makes a connection between an upper layer of polysilicon and an N+ diffusion on the bottom. This connection is very similar to a *buried contact* used in some MOS processes. Many other programmable logic products rely on a stored charge to make their connections (for example, RAM, EPROM, or EEPROM). This stored charge can be susceptible to degradation from radiation exposure. The Actel antifuse makes a hard contact and does not rely on a stored charge. As a result, one would expect the Actel products to have superior radiation tolerance compared with products that use a stored charge.

Actel does not perform any radiation testing on devices, however, several customers and independent contractors have performed tests and shared their data. Although the results depend on device type and process technology, Actel devices have shown the capability to withstand total dose and single event upset effects. For detailed information on radiation results, please contact your local Actel sales representative.

Summary

The data presented in this report establishes the excellent reliability of Actel FPGAs. Both Actel models and actual device testing show that the antifuse is highly reliable and that the combined contribution of all antifuses to the gate array product's hard failure rate is less than 10 FITs, or 0.001 percent failures per 1000 hours.

References:

1. E. Hamdy, et al., "Dielectric Based Antifuse for Logic and Memory ICs," IEDM paper, pp. 786–789, 1988.
2. S. Chiang, et al., "Oxide-Nitride-Oxide Antifuse Reliability," Proc. Int. Rel. Phys. Symp., 1990.
3. J. Lee, I. Chen, and C. Hu, "Modeling and Characterization of Gate Oxide Reliability," IEEE Trans. of Elec. Dev., December 1988.

Technical Support Services

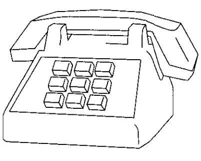
Section 8: Technical Support Services

Technical Support Services

Actel Technical Support

Actel's application engineers have developed many ways to meet your needs. Our services provide technical assistance to all of our customers worldwide. Customers can obtain technical assistance by means of the Technical Support Hotline, the Bulletin Board Service (BBS), Customer Training, International e-mail, and our recently improved fax-back system, Action Facts and Fax Broadcast.

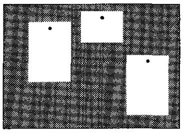
Customer Applications Center



The Technical Support Hotline provides technical information on Actel hardware and software products. Questions regarding software authorization, availability, and pricing are handled through Actel's customer service. All calls are answered by our Technical Message Center. The center retrieves information such as the caller's name, the company name, the phone number, and the caller's question, and then issues a case number. The center then forwards the information to a queue where the application engineers receive the data and return the customer call. Our goal is to return all customer inquiries within one hour. The hotline's phone hours are from 7:00 a.m. to 5:00 p.m. PST. In addition to answering the Customer Applications Center calls, our applications engineers develop other ways of assisting our customers such as writing application notes and user guides, creating design examples, and evaluating software.

The Technical Support Hotline number is **800-262-1060.**

Electronic Bulletin Board Service (BBS)



Actel currently offers information access and transfer via a 24-hour worldwide bulletin board. Customers can download information such as new soft macros and software bug fixes. In addition, our applications engineers use the BBS to help solve design problems. Sometimes a customer's issue cannot be solved by the technical hotline. In these cases, our applications engineers may request the customer upload their design to the BBS. All files uploaded to the Actel BBS are automatically stored in a private directory. This way the applications engineers can examine the design first hand and make the appropriate diagnosis.

The telephone numbers for the BBS is **408-739-6397** and **408-738-5717** with a baud rate of 9600 and 14400 respectively. To connect to the BBS by modem, the following equipment and configuration are required:

- Baud rate of 9600 or 14400
- Data format: 8 data bits, 1 stop bit, no parity

The following file transfer protocols are supported:

- Xmodem
- Ymodem
- Zmodem
- ASCII

First-time callers need to establish an account. Once connected to the BBS, the caller is then prompted for his/her name, company, phone number and so on. After the account is established, the customer calls the Technical Support Hotline at 800-262-1060 and requests file transfer class. Our Technical Message Center verifies the name and company and then upgrades the account's security level.

Customer Training

Actel offers an introductory two-part, three-day course covering all aspects of designing an Actel FPGA.

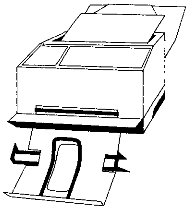
The first day and a half starts with the architecture similarities and differences of Actel's FPGA device families: ACT 1, ACT 2, 1200XL, ACT 3, and 3200DX. It then walks the student through the entire FPGA design process, starting with a hierarchical Viewlogic schematic (including ACTgen blocks) and ending with a programmed Actel FPGA.

The second day and a half is an introduction to VHDL and the ACTmap VHDL synthesis tool.

The course offers extensive hands-on labs, supported by lecture notes and a full set of Actel documentation.

Classes are offered monthly with a minimum of three people per session. Customers can register by calling the Customer Application Center and requesting training class registration information or via the Actel web page.

Action Facts



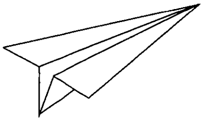
The Action Facts system is a 24-hour fax-back service. Customers can obtain a list of current software bugs and workarounds, application notes, design hints, package pinouts, and much more information. Simply call the toll free number and request that a catalogue be faxed to your office. Review the document and call the number again to request up to four documents per call. The Action Facts Catalogue is frequently updated.

The Action Facts phone number is **800-262-1062**.

Fax Broadcast

Actel has a new support mechanism that allows our customers to receive automatic information regarding latest software bugs and workarounds, new application notes, new products, and much more. This information, if you choose to participate, will be automatically faxed to your fax machine when it comes available. When you call the Technical Support Hotline, be sure to tell them you want to be a member of Fax Broadcast.

International E-mail Address



Actel has recently introduced a new method for customers to communicate with our applications engineers. Customers can e-mail their technical questions to our e-mail address and receive answers back either by e-mail, fax, or phone call. In addition, customers can quickly e-mail their design files to receive assistance. The e-mail account is monitored several times throughout the day, and our goal for responding is two working days after receiving the message.

Technical support's e-mail address is **tech@actel.com**.

The Actel Homepage

Actel has one of the most innovative web sites in the industry located at **http://www.actel.com**. With constant additions and enhancements, the Actel Homepage is the best place to find the latest information to evaluate and use Actel products. Here is some of the information available on our Homepage:

- All current device data sheets
- Device selector guides
- Application examples and design techniques (including design files)
- Links to Actel CAE partners' Homepages
- Software product release notes
- Actel training schedule and registration
- Current Actel representative and distributor contacts
- Actel employment opportunities
- Designer Digests containing the latest software issues
- Numerous software User's Guides
- Technology backgrounders
- The latest reliability reports
- Access to the technical support hotline
- Actel press releases
- Frequently asked questions
- The Actel corporate directory
- Literature and multimedia demo request form
- University program
- High-level design resource page

Many new features are certain to have been added since the printing of this Data Book, so if you are currently designing with Actel or just browsing, come visit the Actel Homepage at http://www.actel.com. And while you're there, be sure to sign our guestbook!

Glossary

Section 9: Glossary

Glossary of Terms

ACTgen Function-generation tool that automatically builds configurable structured macros including counters, shift registers, arithmetics, decoders, and large multiplexers. The user can configure size and optional controls for each macro. The ACTgen macros are usable in both schematic and synthesis environments.

Actionprobes A combination hardware and software function whereby internal nodes in the device can be connected to external I/O pins (PRA and PRB) during normal device operation. The nodes of interest are selected by a serial address scanned into the device using the SDI and DCLK pins.

Activator Actel nomenclature for a programmer.

ACTmap Synthesis and optimization utility available with all Actel development systems. ACTmap can synthesize VHDL or PALASM source files into Actel specific netlists, which can then be mapped into a device.

Antifuse A programmable element consisting of a two-terminal device that exhibits high impedance in the unprogrammed state and low impedance in the programmed state. Programming is accomplished by applying a high voltage across the antifuse's terminals, causing a dielectric breakdown.

BGA Abbreviation for a *ball grid array* package.

Block compiler A software tool that can be used to estimate cost and performance on complete or partial designs.

C-Module A module containing only a combinatorial function. It is characteristic of the Integrator and Accelerator families.

CQFP Abbreviation for *ceramic quad flat pack* package.

Debugger Software tool providing an interface to the Actionprobes and control for the Activator in a dynamic test mode.

Designer Series Actel design environment that can import netlists from third-party CAE tools and convert the information to silicon. Supporting functions in Designer Series include pin and performance definition capability, timing verification, and program file generation.

DirectTime Layout Timing-driven layout function within Designer Series. An editor in Designer Series supports delay constraint definition, which drives the DirectTime Layout engine.

Global clock A global network that can drive all flip-flops in a device. This network can alternatively be used for other functions such as power on reset in some devices.

Hard array clock (HCLK) A global clock that is hardwired to the clock inputs of all array flip-flops. HCLKs exhibit higher performance and lower skew capabilities than do other available global resources.

Hard macro Macro that maps directly into a single or dual logic module and whose timing is fixed.

Logic module The basic logic building block in Actel device architectures.

MPGA Abbreviation for *masked programmable gate array*. For high volumes, MPGAs can provide lower device cost.

ONO High-impedance dielectric layer of the antifuse (oxide-nitride-oxide). The dielectric normally exhibits very high impedance but can be broken down by applying high voltages to its terminals.

PGA Abbreviation for *pin grid array* package.

PLCC Abbreviation for *plastic leaded chip carrier* package.

PLICE Abbreviation for Actel's ONO antifuse (programmable low-impedance circuit element).

PQFP Abbreviation for *plastic quad flat pack* package.

Probe pins Four pins on all Actel devices (SDI, DCLK, PRA, and PRB). These pins are used to control the Actionprobe function of the device.

Quadrant clock A high-drive and low-skew resource that provides access only to a portion of the device. As a result, special restrictions limit the number of flip-flops that can be attached to these clocks. Although referred to as a clock, these resources can also be used to drive high-fanout combinatorial functions.

Routed clock (RCLK) A flexible, medium-performance, medium-skew global resource.

S-module A module in the Accelerator and Integrator families that contains a C-module element and a flip-flop.

Segmentation A patented capability in Actel devices providing wires of varying lengths to interconnect logic modules. The advantage of segmentation is that fewer programmable elements need to be used, thus reducing overall circuit impedance and improving predictability.

Soft macro A function composed of many hard macros. The function is considered soft because performance can vary from one instance to the next.

System speed A measure of the potential performance of a device, using complex functions.

Toggle rate The frequency at which an internal flip-flop can toggle, given no output loading. This parameter typically implies nothing about the actual achievable performance of a device.

TQFP Abbreviation for *thin quad flat pack* package.

Usable gates A measure of the amount of function achievable in a device when related to two-input NAND gates. For FPGAs, the number of usable gates is a function of programmable logic block efficiency and the percentage of usable logic blocks.

VQFP Abbreviation for *very thin quad flat pack* package.

Notes

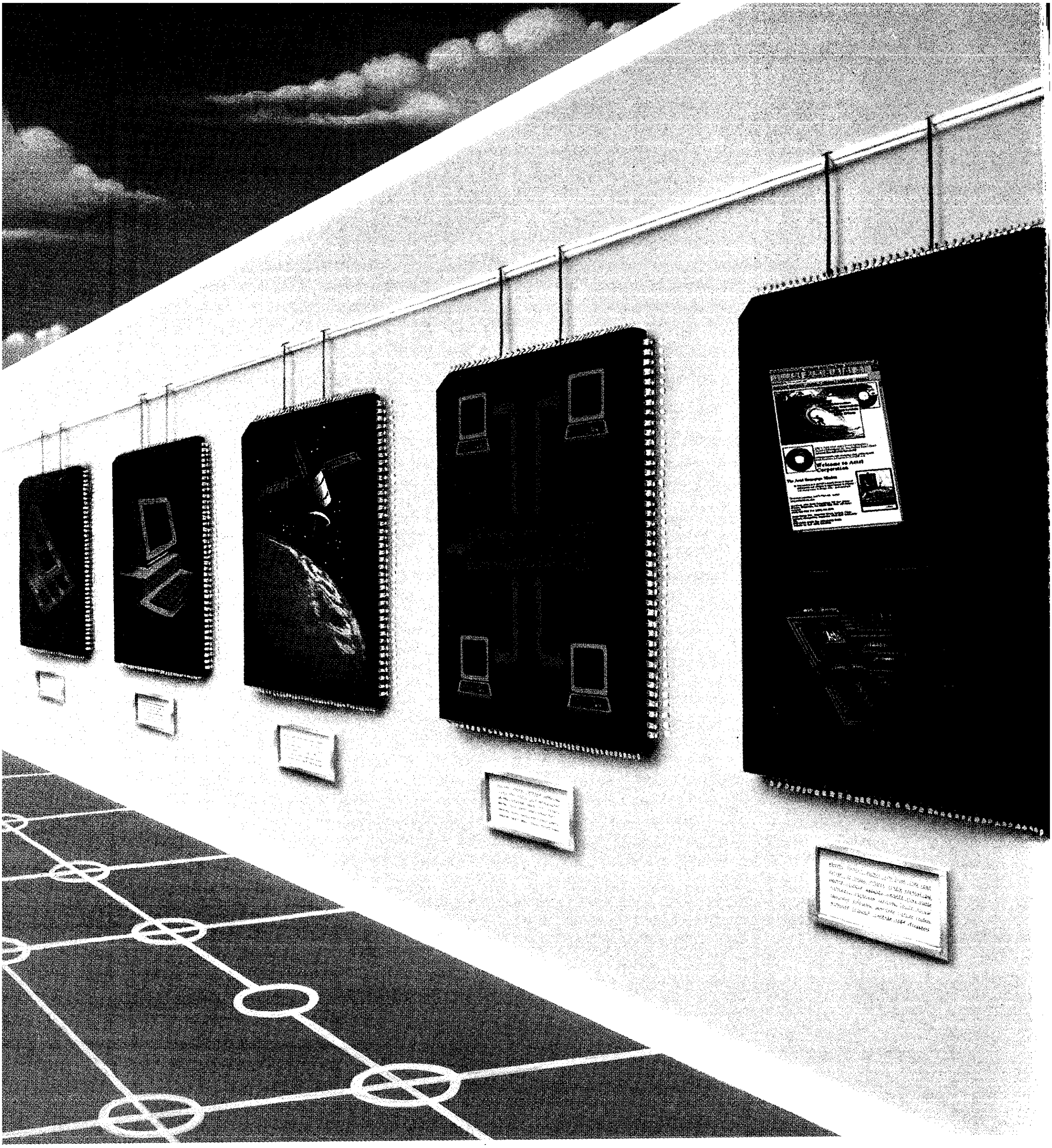
Notes

Notes

Notes

Notes

Notes



Actel Corporation
955 East Arques Avenue
Sunnyvale, CA 94086-4533
Tel: (408) 739-1010
Fax: (408) 739-1540

5172041-0

Actel Europe Ltd.
Daneshill House
Lutyens Close
Basingstoke, Hampshire RG24 8AG
England
Tel: (44) (1256) 29209
Fax: (44) (1256) 55420

Actel

Take it to a higher level.

http://www.actel.com