XCELL

Issue 14
Third Quarter 1994

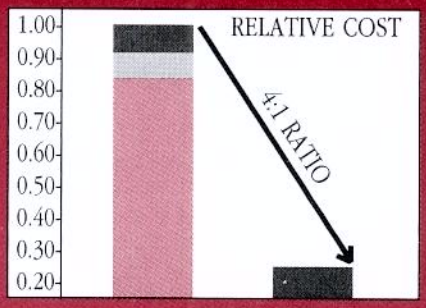
THE QUARTERLY JOURNAL FOR XILINX PROGRAMMABLE LOGIC USERS

XILINX®

The Programmable Logic Company℠

Inside This Issue:

GENERAL FEATURES



Lowering PLD Costs

Marketing vice president Chuck Fox examines the factors driving PLD price reductions...

See Page 3

PRODUCT INFORMATION

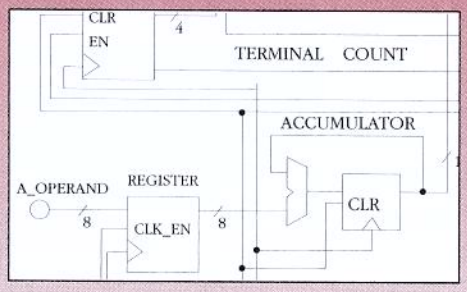
New Product:

The 5 ns EPLD

Introducing the first EPLDs with 5 ns pin-to-pin timing...

See Page 16

DESIGN TIPS & HINTS



Multipliers in FPGAs

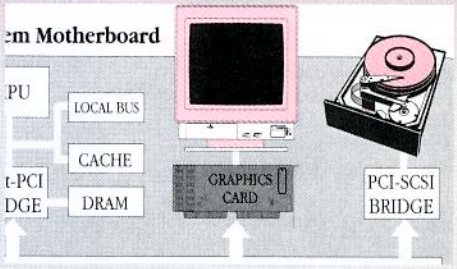
How to implement a fast integer multiplier in an FPGA...

See Page 17

Meeting PCI Compliance

PCI-compliant EPLDs can implement bus interfaces and bridges...

See Page 21



WANTED: User Feedback

By **BRADLY FAWCETT** ◆ Editor

The 31st annual Design Automation Conference (DAC) was held in San Diego in early June. Although I did not attend personally, Xilinx sponsored a booth and sent various research and development, applications engineering, marketing, and sales personnel to attend the show. I chatted with a few of them when they returned, and can report that some recurring themes of importance to programmable logic users and vendors were evident at the conference.

The market growth of programmable logic has attracted a lot of attention from CAE tool suppliers. Five years ago, the DAC exhibit floor was a mixture of CAE and custom silicon suppliers, with the emphasis almost entirely on gate array and custom cell design. Now, the tool vendors are still there, but you have to look harder to find the gate array manufacturers, and Xilinx wasn't the only programmable logic vendor present on the floor. More significantly, many of the CAE companies were emphasizing their programmable logic design tools, including major suppliers such as Mentor Graphics, Cadence Design Systems, and Synopsys. One observer noted that Xilinx support was being actively touted in more than 25 CAE vendors' exhibits.

> *"One observer noted that Xilinx support was being actively touted in more than 25 CAE vendors' exhibits."*

Our thanks to the many visitors who stopped by our booth at DAC. It was gratifying to see the high level of interest in our products, particularly the recently-released XC4025 FPGA.

Not surprisingly, one recurring theme both on the exhibit floor and in the technical program was the control of design flows and tool interoperability. The pendulum seems to have swung away from the large front-to-back-end tool sets that were pervasive in the 1980s and fueled the growth of companies like Mentor Graphics and Daisy. Instead, it has swung toward the collections of "point tools" offered by several suppliers.

Naturally, design engineers facing productivity and time-to-market pressures want to use the best tool available for each design task, but want those tools to work together in a smooth, easy-to-use development flow. Design flow managers from companies such as Data I/O, Mentor Graphics, and Intergraph were receiving a lot of attention on the exhibit floor.

Tool interoperability and design flows have been a subject of great interest to Xilinx since the introduction of our first interface to a third-party schematic editor in 1987. Since then, our design environment has always been "open," in that we rely on third-party CAE vendors to provide design entry and simulation tools, and assist them in establishing links to our architecture-specific "place-and-route" implementation programs. This process was formalized about five years ago with the formation of our Alliance Program, which currently boasts more than 30 member companies *(see the chart on page 12).*

Continued on page 8

XCELL

Please direct all inquiries, comments and submissions to:

Editor: Bradly Fawcett

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
Phone: 408-879-5097
FAX: 408-879-4676
E-Mail: brad.fawcett@xilinx.com

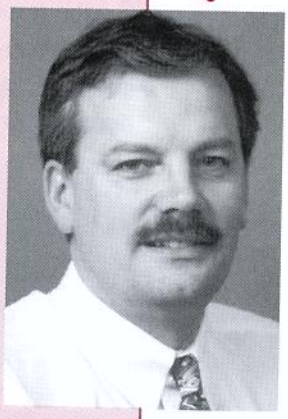
XILINX®

Lowering High-Density PLD Costs

by CHUCK FOX, Vice-President of Product Marketing

Industry experts project that the programmable logic market will grow another 30 to 35 percent in 1994. The reason is simple — all things being equal, designers would rather have the flexibility of programmable logic than the rigidity of mask-programmed logic in implementing their designs.

Of course, all things aren't equal. High-density PLDs are more expensive, less dense, and typically offer lower performance than mask-programmed gate arrays. PLD manufacturers such as Xilinx are aggressively narrowing these gaps in all three areas. However, of the three, we believe that *cost is the most important factor in expanding the use of high-density programmable logic.* By reducing programmable logic prices, the incentive for implementing designs in mask-programmed gate arrays is removed.



PLD price reductions are driven by three factors:

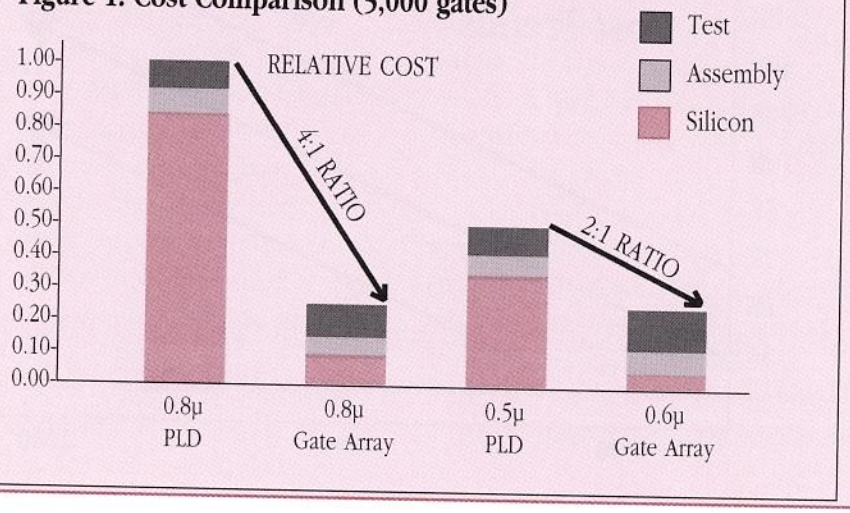
- New submicron wafer-process technologies.
- Higher volumes resulting in improved manufacturing efficiency.
- New device architectures.

New submicron wafer-process technologies reduce the cost of the silicon very quickly, allowing substantial price reductions to continue. Perhaps more significantly, *existing Xilinx PLD architectures will decrease in price faster than their equivalent-density mask-programmed counterparts.* The largest component of cost for large PLDs is the silicon (die size). This is not true of the mask-programmed device (Figure 1). For example, at 5,000 gates, the silicon cost of a PLD constitutes over 80 percent of the overall cost to produce. However, for a 5,000-gate mask-programmed gate array, the silicon cost is a very small part of the manufacturing

Continued on page 4

> *"Unlike mask-programmed devices, PLDs benefit from manufacturing economies-of-scale..."*

Figure 1: Cost Comparison (5,000 gates)



XILINX®

Continued from page 3

"Price and density points like these will change the way logic design is done by the end of the decade."

costs (less than 10 percent); the die size is small enough that the package/assembly and test costs dominate. Converting to 0.6 micron may cut both die sizes in half, but will have a much greater cost reduction effect on the programmable device. Thus, new submicron technology will narrow the cost gap with mask programmed devices. The effect is even more pronounced at higher gate densities.

Increasing production volumes also lower PLD costs. Unlike mask-programmed devices, PLDs benefit from manufacturing economies-of-scale because they are standard products. No factory customization is required. This allows fixed costs — largely associated with equipment for packaging and testing — to be amortized over an increasingly large volume of standard parts, reducing the per unit cost-to-manufacture.

The third significant factor driving PLD cost reduction is *new architecture development.* Xilinx is actively pursuing new product developments that will increase the number of programmable gates per area of silicon. These increases in "gate efficiency" will greatly accelerate FPGA price-performance.

The combined result of these three factors will be astounding. We can project what the future will bring by looking at what we've already done (Figure 2). In 1985, volume pricing for a 1,000-gate device was hundreds of dollars; today that same device costs a few dollars. At that time, the largest device we could provide was 1,000 gates. Today, for the same "hundreds of dollars" price, we can provide 25,000 gates. In 1990, a 10,000 gate device in ceramic packaging (XC4010) cost almost $1,000; today, the XC4010D device in PLCC84 (plastic) packaging is well under $100 in high volume.

So what's possible by the year 2000 (only 5+ years away):

- 20,000-gate FPGAs for < $5
- 100,000-gate FPGAs for < $50
- FPGAs with over 200,000 gates

Price and density points like these will change the way logic design is done by the end of the decade. With ample capacity in advanced submicron process technology and increasing R&D investment in new products, our users stand to benefit greatly. Take a look at today's prices, and ask for 1995 price projections. You might be surprised at what you find out! ◆

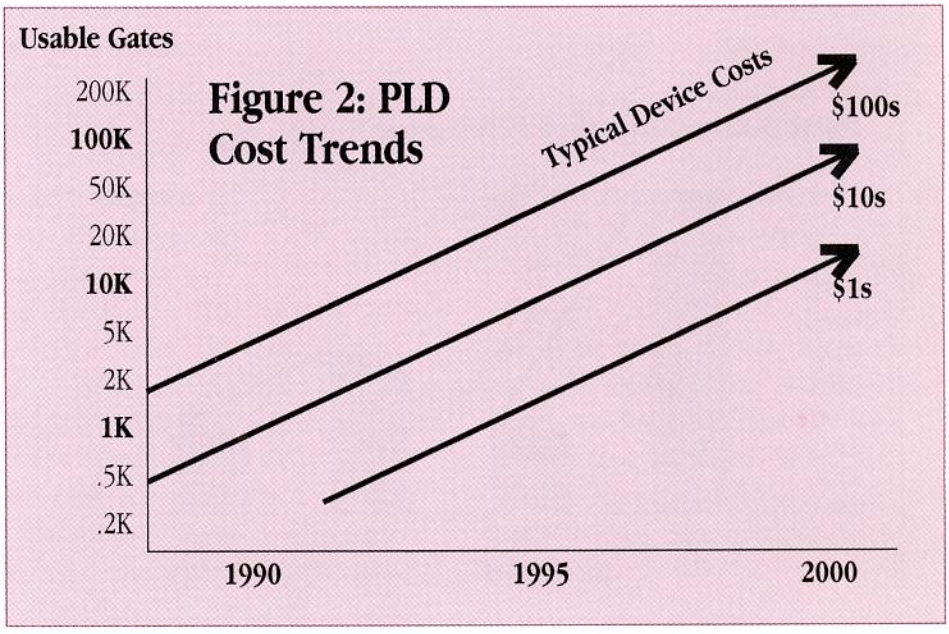


Figure 2: PLD Cost Trends

Silicon Graphics Counts on FPGA Synthesis

Silicon Graphics, Inc. (SGI) excels in the design and production of high performance, graphical workstations. Their advanced image-processing and visualization techniques have made them a market leader. The company is extremely innovative, and strives to stay on the cutting edge of technology. In order to remain a technology leader, SGI must design creative solutions and get products to market quickly. Xilinx has been able to aid SGI in meeting this goal during the development of the Sirius Video Card.

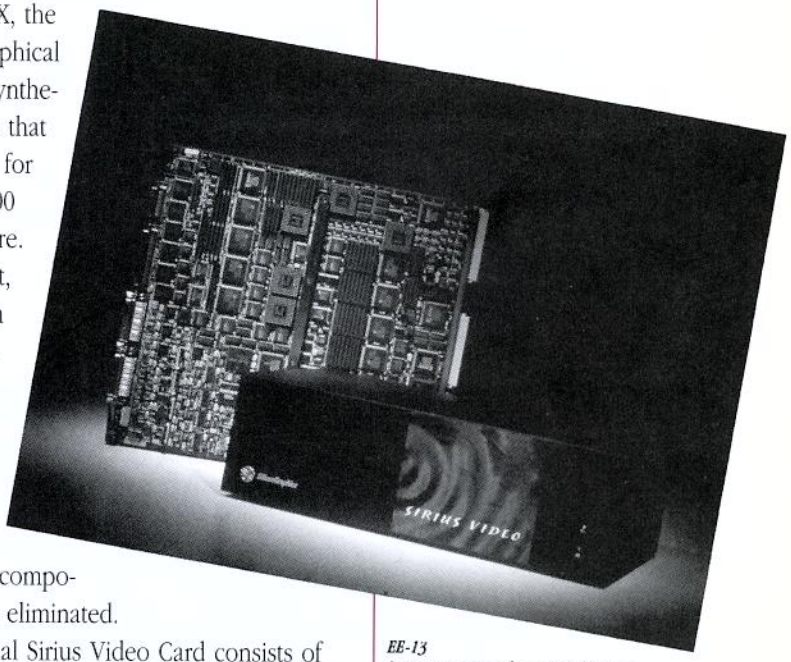
Recently, SGI's Advanced Graphics Division developed a high-end video card to produce broadcast-quality digital video effects. This card allows multiple video-production-related applications to coexist on a single hardware platform, replacing the need for a $200,000 cluster of video effect boxes. The card is a $25,000 option for the SGI Onyx Reality Engine, the company's flagship product.

A team of six SGI engineers was given eight months to develop the card, based on a previous design which consisted of more than 70 PALs. The PAL count was expected to rise to 200 once the additional required functions were added. Kirk Law, Video Hardware Manager, said, "We needed a technology with the integration benefits of an ASIC, but without the ASIC NRE costs and development time."

Previous positive experience with Xilinx FPGAs led the team members to use Xilinx XC4000 family devices. Thirteen new FPGA designs were required, ranging from the low-density XC4003 up to the high-density XC4010.

To meet the aggressive schedule, the design team relied on synthesis to design the FPGAs. The Verilog-HDL hardware description language and Synopsys' FPGA Compiler were used. Using synthesis allowed the engineers to spend a significant amount of time on simulation. This upfront work led to very clean designs. Board-level synthesis helped to identify many problems before system fabrication.

SGI's design team also took advantage of X-BLOX, the Xilinx graphical module synthesis system that optimizes for the XC4000 architecture. As a result, the design cycle progressed rapidly, and the need to debug low level components was eliminated.



EE-13
Image courtesy Silicon Graphics, Inc.

The final Sirius Video Card consists of 18 FPGAs. The largest is a VME Master Bus Controller, managing a 64-bit DMA interface. This design uses an XC4010 and supports data rates up to 50 Mbits per second. The fastest circuits within the system are state machines running at 50 MHz. These are implemented in XC4006 and XC4010 devices.

Xilinx FPGAs, in conjunction with synthesis, allowed six SGI engineers to complete the Sirius Video card prototype in six months — two months ahead of schedule. The resulting product has helped SGI retain their competitive edge in the graphical workstation market. ◆

New Product Literature

Learn about the newest Xilinx products and services through our extensive library of product literature. The most recent pieces are listed below. To order please contact your local Xilinx sales representative. ◆

For a complete list, please contact your sales representative or see XCELL Issue #10.

TITLE	DESCRIPTION	PART NUMBER
EPLDs		
XC7336 Datasheet	Technical Data	#0010190-02
A Zero-Wait-State Synchronous DRAM Controller for the Pentium Processor	Technical applications note	#0010217-01
Designing Flexible PCI Interfaces with Xilinx EPLDs	Technical applications note	#0010216-01
Designing with the XC7336 and XC7318	Technical applications note	#0010218-01
PROMs		
XC17128D/XC17256D Serial PROM Datasheet	Technical Data	#0010212-01
MILITARY		
XC4000 Hi-Reliability Datasheet	Technical Data	#0010209-01
HARDWIRE LCA		
1994 HardWire Databook	Technical Data	#0010164-02
DEVELOPMENT SYSTEMS		
Development Systems Reference Guide	Pricing & ordering info on Xilinx development system software and documentation	#0010161-03
Development Systems Quick Reference Card	One-page overview of development system pricing & ordering information	#0010104-04
Base Development System Overview	Features & benefits	#0010154-02
X-BLOX Overview	Features & benefits	#0010107-04
Designing Xilinx with Mentor Graphics	Features & benefits	#0010067-04
Designing Xilinx with OrCAD	Features & benefits	#0010135-03
Designing Xilinx with Synopsys	Features & benefits	#0010145-03
Designing Xilinx with Viewlogic	Features & benefits	#0010166-02
CORPORATE		
1994 Databook (Second Edition, August 1994)	Technical data on FPGAs, EPLDs, PROMs & development systems. Includes app notes.	#0401253-01
The Total High Performance Logic Solution	Technical information on high-performance XC3100A, XC4000 & XC7000 families. Includes customer testimonials.	#0010194-01

UPCOMING EVENTS

Xilinx will be participating in several conferences and workshops this autumn. Look for Xilinx technical papers and/or product exhibits at these upcoming industry forums. For further information about any of these conferences, please contact Kathleen Pizzo (Tel: 408-879-5377 FAX: 408-879-4780). ◆

Sept. 7-9
Fourth International Workshop on Field Programmable Logic and Applications (FPL '94)
Prague, Czech Republic

Sept. 19 - 23
EuroDAC/EuroVHDL
Grenoble, France

Sept. 19 - 23
IEEE ASIC Conference and Exhibit (ASIC '94)
Rochester, New York

Sept. 27 - 29
Wescon/IDEA Electronic Conference and Exhibition
Anaheim, California

Oct. 10 - 12
International Conference on Computer Design (ICCD '94)
Cambridge, Massachusetts

Oct. 25 - 26
Silicon Design Show
London, England

Oct. 26 - Nov. 4
The Electronic Design Automation & Test Conference and Exhibition
Taipei (Oct. 26 - 28),
Seoul (Oct. 31 - Nov. 2),
Beijing (Nov. 4 - 5)

Nov. 2 - 6
Frontiers in Education
San Jose, California

Nov. 8 - 12
Electronica
Munich, Germany

For further information about any of these conferences, contact Kathleen Pizzo (Tel: 408-879-5377 Fax: 408-879-4780).

FINANCIAL REPORT

Xilinx Named Best Financially Managed 'Fabless' Chip Company For 4th Year in a Row

For a record fourth consecutive year, market research firm In-Stat Inc. named Xilinx the Best Financially Managed IC Company. The coveted Kachina Award was presented to Xilinx at In-Stat's annual Semiconductor Forum in May.

The Kachina Award is a statue carved by the Hopi Indians of Arizona. According to Hopi legend, the Kachina is a warrior who faces great difficulties and shows great strength.

Xilinx was chosen over all other "fabless" companies, including Adaptec, Altera, Cirrus Logic, and Lattice Semiconductor.

The companies are ranked based on criteria such as operating income, net income, return on investment, inventory turnover, and sales per employee. A separate set of awards is presented to IC companies that have their own fabrication facilities.

This marked the first time that any company has won the award four years in a row.

"To win our fourth Kachina award is particularly satisfying considering recent predictions about shortages of submicron foundry capacity for fabless suppliers, " noted Xilinx President Bernie Vonderschmitt. "We have no delinquency issues and remain in an excellent position for submicron availability."

Xilinx sales revenues for the first fiscal quarter (ending July 2, 1994) were $75.2 million, an increase of 38 percent over the same quarter last year. Domestic revenue was down slightly from the previous quarter, due to aggressive price reductions (especially for the XC4000 FPGA family) and inventory adjustments by several networking companies. However, international revenues increased by $4.6 million over the March-ending quarter, accounting for 32 percent of total revenues. ◆

Continued from page 2

Several of the new features of the recent XACT 5.0 release targeted design-flow issues, such as making timing constraints easier to enter, improving the back-annotation of timing results to support simulation, and the development of XSimMake, an automated script supporting the generation of simulation netlists. Look for further improvements, including the redesign of our own flow manager (XDM) in the next few months. *(See the preview of the next XACT release on page 22.)*

With increasing productivity on every engineer's mind, logic synthesis tools were of great interest to DAC attendees. Xilinx has established an offshoot to the Alliance Program, called the Synthesis Syndicate, specifically to work with synthesis-tool vendors on improving programmable logic support. In addition, we are co-developing synthesis tools with market-leader Synopsys.

Graphical-HDL tools, programs that generate VHDL from a block diagram entered with graphical symbols, are starting to appear on the market, in an attempt to overcome the verbosity of textual high-level languages and provide an easier bridge to synthesis tools for non-English-speaking engineers. Xilinx is proud to be on the forefront of this trend with the release of the X-BLOX library three years ago and a lead role in the development of the LPM (Library of Parameterized Modules) concept.

The Xilinx booth at DAC.

Windows for the PC is rapidly gaining popularity as a development platform. PC-based Verilog and VHDL simulators are migrating to this platform. The XACT 5.0 release was a step in the right direction, as the Xilinx implementation tools became Windows compatible; the next release will be a further step in improving design productivity, including tighter integration with the Windows environment. However, we don't feel that Windows will replace UNIX, which will remain the primary operating system for workstation environments, and we intend to continue to support both environments in our development system offerings.

Another interesting trend tied to increasing designer productivity is the growing popularity of FPGA-based emulation systems for verifying large digital designs. This technology fills an important gap in digital system design, providing a viable validation mechanism when the complexity of the system makes software simulation too difficult and time-consuming. Since all of today's commercially-available emulators are based on Xilinx FPGAs, this is a trend we're glad to see and a market we will continue to aggressively support!

Perhaps the most significant trend that we observed at DAC has more to do with the behavior of the users than the tools — CAE tool users are becoming more open about telling their CAE tool suppliers what they want in addition to demanding good price-performance from their design tools. This is certainly a trend that we encourage; we welcome your feedback, comments, and suggestions about our products. In fact, we'd like to hear more from you; your inputs are key to product refinement.

Some of you will be receiving surveys from Xilinx in October or November;

Continued

"Some of you will be receiving surveys from Xilinx in October or November; please set aside a few minutes, answer the questions, and return your survey to us promptly."

Continued from previous page

please set aside a few minutes, answer the questions, and return your survey to us promptly.

There are a lot of other ways of getting your comments to us — talk to our sales representatives and distributors (believe me — they are not shy about passing along your comments to us), contact your local Field Applications Engineers, call the Application Engineers on the hotline, or write, call, fax, or e-mail us here at Xilinx headquarters. If you'd like to send your comments to me, I'll gladly pass them along to the appropriate R&D and marketing managers.

It's important that you provide us with the background information that led to your comment or suggestion — the size of your designs, your level of experience with the tools, your designs' requirements (speed, power, cost, etc.), whether you use PLDs for prototyping or production, and so on.

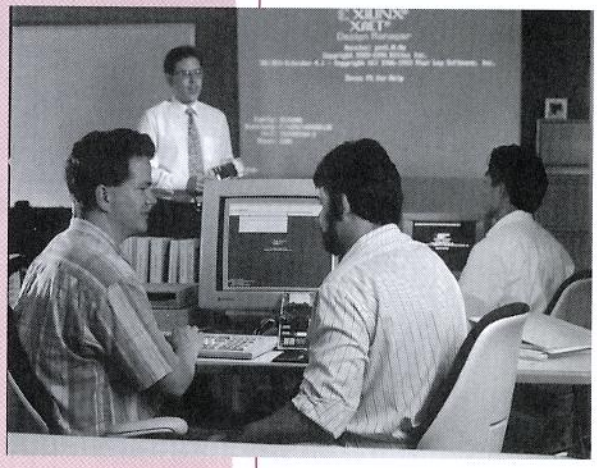
Be a vocal user and tell us what you want and need. We're hoping to hear from you soon! ◆

Advanced Training Sessions Scheduled

The Advanced Training classes long offered at the Xilinx headquarters and often used as the basis of customer presentations by Field Application Engineers are now being scheduled at other locations throughout the United States. The classes are open to all Xilinx customers at no charge. Contact your local sales office for more information, or call the Xilinx Training Registrar at (408) 879-5090.

Advanced Training differs from the regular Training Class in that students must have prior experience with the Xilinx software. Instead of using labs to teach how to run the software, Advanced Training focuses on design techniques to optimize density and performance. The latest features of the software are described in detail, with demonstrations of many of the advanced implementation options.

Advanced Training is another link in the technical support offered to Xilinx customers, in addition to local Field Applications Engineers, Hotline support, seminars, and technical literature. If Advanced Training is not scheduled in your area, and you are interested in more detail on how to use Xilinx products, please contact your local sales office or Field Applications Engineer. ◆



COMPONENT AVAILABILITY CHART - AUGUST 1994

PINS	TYPE	CODE	XC2064	XC2018	XC2064L	XC2018L	XC3020	XC3030	XC3042	XC3064	XC3090	XC3020A	XC3030A	XC3042A	XC3064A	XC3090A	XC3020L	XC3030L	XC3042L	XC3064L	XC3090L	XC3120	XC3130	XC3142	XC3164	XC3190	XC3195	XC4003	XC4005	XC4006	XC4008	XC4010	XC4010D	XC4013	XC4025	XC4002A	XC4003A	XC4004A	XC4005A	XC4003H	XC4005H	XCZ236	XCZ272	XCZ236A	XCZ272A	XCZ318	XCZ336	XCZ354	XCZ372	XCZ3108
44	PLASTIC LCC	PC44	◆	◆				◆				◆										◆																		◆		◆		◆	◆	◆				
	PLASTIC QFP	PQ44																																										N	N					
	CERAMIC LCC	WC44																																								◆		◆		◆		◆			
48	PLASTIC DIP	PD48	◆																																															
64	PLASTIC VQFP	VQ64		◆	◆	◆						◆					◆																																		
68	PLASTIC LCC	PC68	◆	◆	◆		◆	◆				◆	◆									◆	◆																				◆		◆			◆	◆			
	CERAMIC LCC	WC68																																								◆		◆			◆	◆			
	CERAMIC PGA	PG68	◆																																															
84	PLASTIC LCC	PC84		◆		◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆	◆		◆	◆	◆	◆			◆		◆			◆	◆			
	CERAMIC LCC	WC84																																								◆		◆			◆	◆			
	CERAMIC PGA	PG84		◆			◆	◆	◆			◆	◆	◆								◆	◆	◆																			◆		◆			◆				
100	CERAMIC QFP	CQ100				◆		◆																																											
	PLASTIC PQFP	PQ100				◆	◆	◆				◆	◆	◆								◆	◆	◆				◆									◆	◆												N	N
	PLASTIC TQFP	TQ100		◆			◆	◆														◆	◆																												
	PLASTIC VQFP	VQ100			◆							◆	◆				◆	◆				N	N															◆	◆											
	TOP BRZ. CQFP	CB100				◆		◆			◆		◆									◆	◆															◆													
120	CERAMIC PGA	PG120																									◆									◆	◆	◆												
132	PLASTIC PGA	PP132						◆	◆				◆	◆									◆	◆																												
	CERAMIC PGA	PG132						◆	◆				◆	◆									◆	◆																												
144	PLASTIC TQFP	TQ144											◆	◆				◆	◆				◆	◆																◆	◆										
	CERAMIC PGA	PG144																																																◆
156	CERAMIC PGA	PG156																										◆	◆											◆											
160	PLASTIC PQFP	PQ160							◆	◆				◆	◆									◆	◆	◆		◆	◆	◆	◆	◆	◆					◆	◆											◆
164	CERAMIC QFP	CQ164								◆																																								
	TOP BRZ. CQFP	CB164								◆					◆										◆			◆																						
175	PLASTIC PGA	PP175								◆					◆										◆	◆																								
	CERAMIC PGA	PG175								◆					◆										◆	◆																								
176	PLASTIC TQFP	TQ176													◆					◆					◆																									
191	CERAMIC PGA	PG191																														◆	◆								◆								
196	TOP BRZ. CQFP	CB196																															◆																
208	PLASTIC PQFP	PQ208								◆					◆										◆	◆						◆	◆	N	N		N				◆	◆								
	METAL MQFP	MQ208																															◆	◆		◆													
223	CERAMIC PGA	PG223																								◆										◆						◆								
225	PLASTIC BGA	BG225																															◆			◆													◆
240	PLASTIC PQFP	PQ240																																		N						N								
	METAL MQFP	MQ240																																		◆	◆					◆								
299	CERAMIC PGA	PG299																																	N															

◆ = Product currently shipping or planned N = New since last XCELL

XILINX RELEASED SOFTWARE STATUS - AUGUST 1994

Product Category	Product Description	Product Function	Xilinx Part Number	Previous Ver. Rel.	Current Version By Platform PC1 *6.2*	SN2 *4.1.x*	AP1 *10.4*	HP7 *9.01*	Last Update
XILINX INDIVIDUAL PRODUCTS									
CORE FPGA	XC2,3,4K Support	Core Implementation	DS-502-xxx	1.42	5.00	5.00	1.42	5.00	07/94
CORE EPLD	XC7K Support	Core Implementation	DS-550-xxx	4.10	5.00	5.00		5.00	07/94
Mentor[1]	V7.00	I/F and Libraries	DS-343-xxx	4.10			4.20		07/94
Mentor[1]	V8.2_5	I/F and Libraries	DS-344-xxx	1.10		5.00	1.10	5.00	07/94
OrCad[2]		I/F and Libraries	DS-35-xxx	4.23	5.00				07/94
Synopsys[1]		I/F and Libraries	DS-401-xxx	2.00		3.01	3.01	3.01	09/93
Viewlogic[2]	ViewDraw	I/F and Libraries	DS-390-xxx	4.15	5.00				07/94
Viewlogic[2]	ViewSim	I/F and Libraries	DS-290-xxx	4.15	5.00				07/94
Viewlogic[2]		I/F and Libraries	DS-391-xxx	4.15	5.00	5.1		5.1	07/94
XABEL[2]		Entry, Sim, Lib, Opt.	DS-371-xxx	4.30	5.00	5.00			7/94
X-BLOX[1]		Module Generation & Opt.	DS-380-xxx	1.04	5.00	5.00	1.04	5.00	7/94
XILINX PACKAGES									
Mentor 8	Standard		DS-MN8-STD-xxx	1.10		5.00	1.10	5.00	07/94
OrCad	Base		DS-OR-BAS-xxx	5.0	5.01				07/94
OrCad	Standard		DS-OR-STD-xxx	1.10	5.00				07/94
Synopsys	Standard		DS-SY-STD-xxx	1.01		1.10	1.10	1.10	09/93
Viewlogic	Base		DS-VL-BAS-xxx	5.0	5.01				07/94
Viewlogic	Standard		DS-VL-STD-xxx	1.20	5.00	5.00		5.00	07/94
Viewlogic/S	Base		DS-VLS-BAS-xxx	5.0	5.01				07/94
Viewlogic/S	Standard		DS-VLS-STD-xxx	1.20	5.00				07/94
Viewlogic/S	Extended[3]		DS-VLS-EXT-xxx	1.20	5.00				07/94
XILINX HARDWARE									
Device Pgmr.	Prom. Pgmr.		HW-112 (XPP)	3.31	5.00	5.00			06/94
Device Pgmr.	EPLD/Prom. Pgmr.		HW-120 (Prolink)	3.14	5.00				06/94
THIRD PARTY PRODUCTION SOFTWARE VERSIONS									
Cadence	Composer	Schematic Entry	N/A	4.2.2		4.30		4.30	N/A
Cadence	Verilog	Simulation	N/A	1.7BP		2.0.5		2.0.5	N/A
Cadence (Valid)	Concept	Schematic Entry	N/A	1.60		1.7		1.7	N/A
Cadence (Valid)	Rapidsim	Simulation	N/A	3.0BP		4.10		4.10	N/A
Mentor	NETED	Schematic Entry	N/A				7.XX		N/A
Mentor	QuickSim	Simulation	N/A				7.XX		N/A
Mentor	Design Architect	Schematic Entry	N/A	8.2		8.2_5	8.2_5	8.2_5	N/A
Mentor	QuickSim II	Simulation	N/A	8.2		8.2_5	8.2_5	8.2_5	N/A
OrCad	SDT 386+	Schematic Entry	N/A		1.10				N/A
OrCad	VST 386+	Simulation	N/A		1.10				N/A
Synopsys	FPGA/Design Comp.	Synthesis	N/A	3.0c		3.1	3.1	3.1	N/A
Viewlogic	ViewDraw	Schematic Entry	N/A		4.1.3a	5.1		5.1	N/A
Viewlogic	ViewSim	Simulation	N/A		4.1.3a	5.1		5.1	N/A
Data I/O	ABEL Compiler	Entry and Simulation	N/A		5.0	5.0			N/A
Data I/O	Synario	Entry and Simulation	N/A		1.0				N/A

NOTE: [1]FPGA Only [2]FPGA and EPLD [3]Includes ViewSynthesis v2.3.1

ALLIANCE PROGRAM - COMPANIES & PRODUCTS - AUGUST 1994

Company	Product Name	Version	Function	Vendor Interface Name	FPGA Support	EPLD Support	X-BLOX Support
Accel (Omation)	Tango Schema III	1.4 3.34	Schematic Entry Schematic Entry	SCH2XNF Schema Xilinx Interface	✓ ✓		
Acugen			Automatic Test Generation		✓		
ALDEC	Susie	6.12	Simulation	SusieXNF	✓		
Altium	P-CAD	6.0	Schematic Entry	PC-Xilinx	2k,3k		
Cadence (Valid)	Concept Rapidsim Composer Verilog	1.7 4.10 4.3 2.0.5	Schematic Entry Simulation Schematic Entry Simulation	Xilinx Front End Xilinx Front End Xilinx Front End Xilinx Front End	✓ ✓ ✓ ✓		✓ ✓
Capilano	DesignWorks	3.1	Schematic Entry/Simulation	XDK-1	✓		
Compass	Asic Navigator QSim X-Syn		Schematic Entry Simulation Synthesis	Xilinx Design Kit	3k,4k		 ✓
CV (Prime)	Design Entry	2.0	Schematic Entry	Xilinx Kit	✓		
Data I/O	ABEL Synario	5.0 1.0	Synthesis Schematic Entry	Xilinx Fitter Xilinx Fitter	✓ ✓	✓ ✓	
EPS	SIMETRI	2.0	Simulation	XNF2SIM	✓	✓	
Exemplar Logic	CORE	1.2	Synthesis	FS-001	✓		✓
Flynn Systems	FS-ATG		Automatic Test Generation	FS-High Density	✓		
GenRad	System Hilo	4.3	Simulation	Xilinx Tool Kit	✓		
IKOS	2800/2900 Voyager	5.02 1.2	Simulation Simulation	Xilinx Tool Kit Xilinx Tool Kit	✓ ✓		
Intergraph	ACE Plus AdvanSIM 1076 VeriBest Design Systems Synovation PLDSyn	12.0 12.0 12.1 12.0 12.0	Schematic Entry Simulation Schematic Entry/Simulation Synthesis Schematic Entry/ Synthesis/Simulation	AdvanSIM & Veribest Xilinx FPGA Design Kit AdvanSIM " " " VeriBest Design Kit SynLibs	 ✓ ✓ ✓ ✓ ✓	 Q4 Q4 Q4 Q4 ✓	 ✓ ✓ ✓
ISDATA	LOG/iC	3.4	Synthesis	XNF-PP	✓		
Logic Modeling	Smart Model Library LM1200		Simulation Models Hardware Modeler	(In Library) Xilinx Logic Module	✓ ✓	✓	
Logical Devices	CUPL		Synthesis	Xilinx Fitter			
Mentor Graphics	QuickSim II Design Architect Autologic	8.2_5 8.2_5 8.2	Simulation Schematic Entry Synthesis	Call Xilinx Call Xilinx Xilinx Synthesis Library	✓ ✓ ✓	✓ ✓	✓ ✓
MINC	PLDesigner-XL	3.0	Synthesis	Xilinx Design Module	✓	✓	
Minelec	Ulticap	1.32	Schematic Entry	Xilinx Interface	2k,3K		
Nishimura	G-DRAW G-LOG	5.0 4.03	Schematic Entry Simulation	GDL2XNF XNF2GDL	✓ ✓		
OrCAD	SDT 386+ VST 386+ PLD 386+	1.1 1.1 2.0	Schematic Entry Simulation Synthesis	Call Xilinx Call Xilinx	✓ ✓ ✓	✓ ✓	✓ ✓
Phase Three Logic	CapFast	2.2	Schematic Entry	SCH2XNF	✓		
Protel		2.0	Schematic Entry		✓	Q3	
Quad Design (Viewlogic Division)	Motive	3.4 Plus	Timing Analysis	XNF2MTV	✓		
Simucad	Silos III	92.115	Simulation	Included	✓		
Sophia Systems	Vanguard	5.31	Schematic Entry	Xilinx I/F Kit	✓	Q4	✓
Synopsys	FPGA Compiler Design Compiler	3.1 3.1	Synthesis Synthesis	Call Xilinx Call Xilinx	3K,4K 3K,4K		
Teradyne	Lasar	6	Simulation	Xilinx I/F Kit	✓		
Topdown Design	V-BAK	1.0	XNF to VHDL translator	XNF interface	?		
Viewlogic	ViewDraw ViewSim ViewSynthesis	4.1.3a 4.1.3a 2.3	Schematic Entry Simulation Synthesis	Call Xilinx Call Xilinx Call Xilinx	✓ ✓ ✓	✓ ✓ ✓	✓ ✓ ✓

ALLIANCE PROGRAM - PLATFORMS & CONTACTS

Company	Contact Name	Platform				Phone Number
		PC	Sun	Apollo	HP	
Accel - Tango	Nancy Eastman	✓				619-554-1000
Accel - Schema	Steve Halbrook	✓				214-231-5167
Acugen	Peter de Bruyn Kops					603-881-8821
ALDEC	Gregor Sowinski	✓				805-499-6867
Altium	Ray Turner	✓	✓			408-534-4148
Capilano	Chris Dewhurst					604-522-6200
Compass	Mahendra Jain		✓	✓	✓	408-434-7950
CV (Prime)	Kevin O'Leary		✓			617-275-1800
EPS	Michael Massa	✓	✓			617-487-9959
Exemplar Logic	Michiel Ligthart	✓	✓	✓		510-849-0937
Flynn Systems	Mike Jingozian					603-891-1111
GenRad	Anne Crow		✓		✓	011-44-329-82-2240
IKOS	Larry Melling		✓	✓	✓	408-255-4567
Intergraph	Vince Mazur		✓			303-581-2301
ISDATA	Peter Bauer	✓	✓	✓	✓	011-49-0721-751087
Logic Modeling *(Synopsis Division)*	Laura Horsey	✓	✓	✓	✓	503-531-2271
Logical Devices	Joleen Rasmussen					305-428-6868
MINC	Lynne Dolan	✓	✓		✓	719-590-1155
Minelec						011-32-02-4603175
Nishimura	Robert Bartels	✓	✓		✓	415-398-1669
Omation *(See Accel)*						
Phase Three Logic		✓	✓		✓	503-645-0313
Protel	Matthew Schwaiger	✓	✓			408-243-8143
Quad Design *(Viewlogic Division)*	Vern Potter	✓	✓	✓	✓	805-988-8250
Simucad	John Williamson	✓				510-487-9700
Sophia Systems	Terry Wilfley	✓	✓	Q3		415-813-4762
Teradyne	Phil Maukalis	✓	✓			617-422-3677

Programming Solutions Update

The recently-established Xilinx Programming Solutions group is responsible for coordinating the one-time programming of Xilinx devices. The Serial Configuration PROMs, as well as the XC7200 and XC7300 EPLD families require programming support (as opposed to the in-system configuration of the SRAM-based FPGA families).

In addition to supporting the Xilinx HW112 PROM Programmer and the HW120 EPLD/PROM Programmer, the Programming Solutions group works with more than 30 third-party device programming vendors, providing information on programming algorithms and technical support. The Programming Solutions group also works with users who may be having difficulties programming Xilinx devices.

The following tables contain detailed listings of programming support currently available for Xilinx SPROM and XC7200 EPLD products. For each device programmer, the number shown in the Xilinx product column is the earliest software/firmware revision that provided support for that device. Where appropriate, part numbers for adapters are listed under the package type. *Support for Xilinx devices is continuously being added to these vendors' product lines, so if a desired combination is not listed here, please contact the programmer vendor or your local Xilinx sales representative.* ◆

PROGRAMMER SUPPORT FOR XILINX XC7300 EPLDS — AUGUST 1994

Vendor	Model	7318	7336	7354	7372	73108	73144	PC44	PC68	PC84	PQ44	PQ100	PQ160	PG184	BG225
Advin Systems	Pilot-U84	P 8/94	P 8/94	P 8/94	P 3Q94	P 8/94	P4Q94	USA-84	USA-84	USA-84		AM-XC100Q	AM-XC160Q		
B&C Microsystems	Proteus-UPXX	3.7x	3.7x	3.7x	P 9/94	3.7x	P 4Q94	X	X	X					
BP Microsystems	BP 1200	S 8/94	S 8/94	S 8/94	P 9/94	S 8/94	P 4Q94	SM44P	SM68P or SM84UP	SM84P or SM84UP					
Data I/O	2900	S 8/94	S 8/94	S 8/94	P 9/94	S 8/94	P 4Q94	PPI-0243	PPI-XXXX	PPI-0208					
	3900	S 8/94	S 8/94	S 8/94	P 9/94	S 8/94	P 4Q94	3900-PLCC	3900-PLCC	3900-PLCC					
	UniSite	S 8/94	S 8/94	S 8/94	P 9/94	S 8/94	P 4Q94	USBASE-PLCC	USBASE-PLCC	USBASE-PLCC					
	AutoSite	S 8/94	S 8/94	S 8/94	P 9/94	S 8/94	P 4Q94	PLCC-44-1	PLCC-68-1	PLCC-84-1					
Deus Ex Machinex	XPGM	V1.00	V1.00	V1.00	P 8/94	V1.00	P 4Q94	5	6	7					
Elan	6000 APS	P 8/94	P 8/94	P 8/94	P 3Q94	P 8/94	P 4Q94	X							
Hi-Lo System Research	ALL-03A	P 10/94	P 10/94	P 10/94	P 10/94	P 9/94	P 4Q94								
	ALL-07	S 8/94	S 8/94	V3.02	P 4Q94	P 9/94	P 4Q94	PAC-PLCC44	PAC-PLCC68						
ICE Technology Ltd.	Micromaster 1000	S 8/94	S 8/94	S 8/94	S 9/94	S 8/94	P 4Q94	AD-7336-PLCC-44	AD-7354-PLCC-68	AD-73108-PLCC-84					
	Speedmaster 1000	S 8/94	S 8/94	S 8/94	S 9/94	S 8/94	P 4Q94	AD-7336-PLCC-44	AD-7354-PLCC-68	AD-73108-PLCC-84					
	Micromaster LV	S 8/94	S 8/94	S 8/94	S 9/94	S 8/94	P 4Q94	AD-7336-PLCC-44	AD-7354-PLCC-68	AD-73108-PLCC-84					
	LV40 Portable	S 8/94	S 8/94	S 8/94	S 9/94	S 8/94	P 4Q94	AD-7336-PLCC-44	AD-7354-PLCC-68	AD-73108-PLCC-84					
	Speedmaster LV	S 8/94	S 8/94	S 8/94	S 9/94	S 8/94	P 4Q94	AD-7336-PLCC-44	AD-7354-PLCC-68	AD-73108-PLCC-84					
Logical Devices	Allpro-88	P 8/94	P 8/94	P 8/94	P 9/94			X	X	X					
	Allpro-88XR	P 8/94	P 8/94	P 8/94	P 9/94			X	X	X					
	XPRO-1	1.01	1.01	1.01	1.01	1.01		MODXP1-5444L	MODXP1-5468L	MODXP1-108L			MODXP1-160Q	MODXP1-184G	MODXP1-108B
SMS	Expert	S 8/94	S 8/94	S 8/94	P 10/94	P 9/94	P 4Q94	TOP1	TOP1	TOP1					
System General	Turpro-1	S 8/94	S 8/94	P 8/94	P 9/94	P 9/94	P 4Q94	X							
Tribal Microsystems		S 8/94	S 8/94	V3.02	P 4Q94	P 9/94	P 4Q94	X	PAC-PLCC68						
Xeltek	Superpro	2.1	2.1	2.1	P 9/94	P 8/94	P 4Q94	XXC7354-44PL/40D	XXC7354-68PL/40D						
	Superpro II	2.1	2.1	2.1	P 9/94	P 8/94	P 4Q94	XXC7354-44PL/40D	XXC7354-68PL/40D						
Xilinx	HW-120	V3.21	V3.21	V3.21	V3.21	V3.21	P4Q94	HW126-PC44	HW126-PC68	HW126-PC84	S 8/94	S 8/94	HW126-PQ160	HW126-PG184	HW126-BG225

C = Currently Supported (no version number) X = No adapter required P = In Production S = Shipping

NOTE: XC7200A programmer support available from most third-party vendors. Contact specific vendor for latest version.

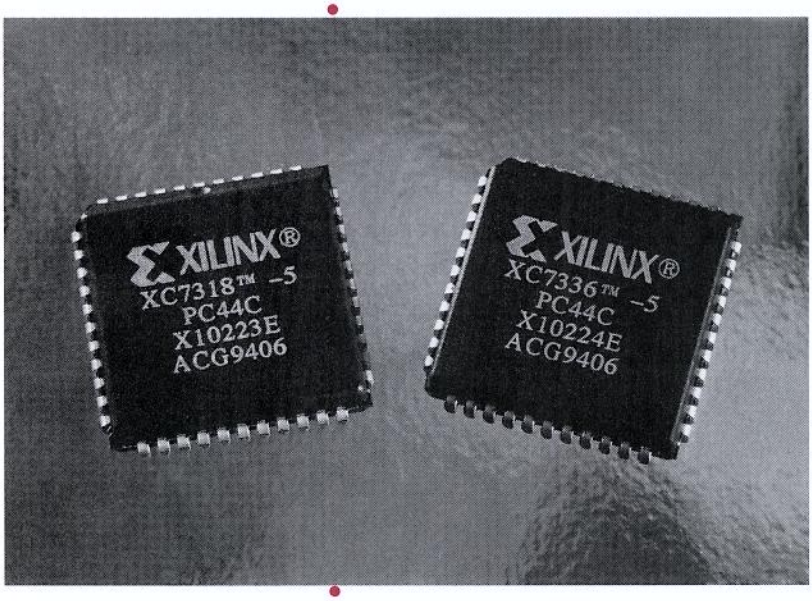
PROGRAMMER SUPPORT FOR XILINX XC1700 SERIAL PROMS — AUGUST 1994

VENDOR	MODEL	1736A	1765	17128	17xxD	17xxL	DIP8	PC20	SO8
Advin Systems	PILOT-U24	10.53	10.53	10.76C	10.71	10.77	X	PX-20	SO-8
	PILOT-U28	10.53	10.53	10.76C	10.71	10.77	X	PX20	SO-8
	PILOT-U32	10.53	10.53	10.76C	10.71	10.77	X	PX20	SO-8
	PILOT-U40	10.53	10.53	10.76C	10.71	10.77	X	PX20	SO-8
	PILOT-U84	10.53	10.53	10.76C	10.71	10.77	X	PX20	SO-8
	PILOT-142	10.73	10.73	10.76C	10.73	10.77	AM-1736	PX20	SO-8
	PILOT-143	10.73	10.73	10.76C	10.73	10.77	AM-1736	PX20	SO-8
	PILOT-144	10.73	10.73	10.76C	10.73	10.77	AM-1736	PX20	SO-8
	PILOT-145	10.73	10.73	10.76C	10.73	10.77	AM-1736	PX20	SO-8
B&C Microsystems	Proteus-UP40	V3.4e	V3.4e	V3.4e	V3.5f	V3.7f	X	AMUPLC84	
BP Microsystems	CP-1128	C	C	V2.17*	V2.21c*	V2.34	FH28A	FH28A + 3rd Party	FH28A + 3rd Party
	EP-1140	C	C	V2.17	V2.21c	V2.34	FH40A	FH40A + 3rd Party	FH40A + 3rd Party
	BP-1200	C	C	V2.17	V2.21c	V2.34	SM48D	SM20P or SM84UP	3rd Party
DATA I/O	UniSite/Site40/48	V3.0	V4.0	V4.1	V4.1	V4.6	X	USBASE-PLCC	USBASE-SOIC
	UniSite/ChipSite	V3.4	V4.0	V4.1	V4.1	V4.6	X	USBASE-PLCC	USBASE-SOIC
	UniSite/PinSite	V3.4	V4.0	V4.1	V4.1	V4.6	X	USBASE-PLCC	USBASE-SOIC
	2900	V1.5	V2.1	V2.2	V2.2	V3.4	X	2900-PLCC	2900-SOIC
	3900	V1.0	V1.5	V1.6	V1.6	V2.4	0101	3900-PLCC	3900-SOIC
	AutoSite	V1.1	V1.5	V1.6	V1.6	V2.4	DIP-300-1	PLCC-20-2	
	UniPak 2B	V23	V24				351B120D		
	ChipLab	V1.1	V1.0	V1.0	V1.0	V1.1	X		AS-08-08-015-3
Elan Digital Systems	3000-145	C	C	C			A116		
	5000-145	C	C	C			A116		
	6000 APS	K2.01	K2.01	K2.02	K2.01	K2.10	X	uPDi84	
Hi-Lo System Research	All-03A	V3.30	V3.30	V3.30	V3.30	V3.30	X	CNV-PLCC-XC1736	CVN-SOP-NDIP16
	All-07	V3.30	V3.30	V3.30	V3.30	V3.30	PAC-DIP40	PAC-PLCC44	
ICE Technology Limited	Micromaster 1000/1000E	V1.1	V1.5	V2.23	V2.21		X	AD-1736/65-20	
	Speedmaster 1000/1000E	V1.1	V1.5	V2.23	V2.21		X	AD-1736/65-20	
Link Com. Gphcs.	CLK-3100	C	C	V4.1	V4.13		X1736	PLCC-17XX	
Logical Devices	ALLPRO-88	2.2	2.2	2.3	V2.3		X		OPTSOI-080
	ALLPRO-88XR	1.1	1.0	2.3	V1.3		X	OPTSOI-080	
	CHIPMASTER 3000	2.0	2.0	2.1	2.0		X	OPTPLC-208	OPTSOI-080
	XPRO-1	1.01	1.01	1.01	1.01	1.01	MODXLN-173	MODXLN-173	MODXLN-173
MQP Electronics	S2610			C	C		X		
Micro Pross	ROM 5000 B	C	C	V1.70	V1.70		Mu 40		
	ROM 3000 U	C	C	V3.60	V3.60				
Red Square	IQ-180	C	C		V8.2				
	IQ-280	C	C		V8.2				
	Uniwriter 40	C	C		V8.2				
	Chipmaster 5000	C	C		V8.2				
Retnel Systems	ZAP-A-PAL	C	C		V3.8J		Module #36		
SMS	Expert	B/93	B/93	A/94	A/94	A/94	TOP40DIP	TOP1PLC or	
	Optima	B/93	B/93	A/94	A/94	A/94	"	TOP3PLC/TOP3PLC	
	Multisyte				A/94		"	"	
	Plus48	B/93	B/93	A/94	A/94	A/94			
	Sprint Plus	B/93	B/93	A/94					
Stag	Eclipse			4.4	2.2		EPU48D	EPU84P	
	Quasar	10.76C	10.76C	10.76C	10.76C	10.76C	X	AMPLCC20	
SUNRISE	T-10	X	X		V3.13		X		
System General	TURPRO-1	C	C	V2.14	V2.01	V2.12	DIP-Adapter	P20-Adapter	
	Turpro-1 F\X	C	C	V2.14	V2.01	V2.12	DIP-Adapter	P20-Adapter	
	APRO	C	C	V2.14	V2.01	V2.12	X	X	
Tribal Microsystems	TUP-300	C	C	V3.31	V3.31	V3.37C	X	CNV-PLCC-XC1736	P Q494
	TUP-400	C	C	V3.31	V3.31	V3.37C	X	"	P Q494
	FLEX-700	C	C	V3.31	V3.31	V3.37C	X	"	P Q494
Xeltek	SuperPRO	1.5B	1.5B	1.7C	1.7D	1.8	X*	20-PL/8-D-ZL-XC1736	16SO15/D6-ZL
	SuperPRO II	1.5B	1.5B	1.7C	1.7D	1.8	X*	20-PL/8-D-ZL-XC1736	16SO15/D6-ZL
XILINX	HW112	C	C	V3.11	V3.31	V3.31	X	HW-112-PC20	HW-112-SO8
	HW120	V5.00	V5.00	V5.00	V5.00	V5.00	HW-120-PRM	HW-120-PRM	HW-120-PRM

C = Currently Supported (no version number) X = No adapter required

Xilinx Unveils World's First 5 ns EPLDs

Xilinx is proud to introduce the highest performing Complex PLDs in the industry: the 5 ns XC7318 and XC7336 EPLDs. This level of performance was previously available only from low-density, limited-function PAL devices. The



new Xilinx ultra-high speed EPLDs are perfect for providing support for the latest generation of fast microprocessors such as the Pentium, PowerPC and Alpha.

The introduction of Xilinx EPLDs with 5 ns pin-to-pin speeds forces a shift in the programmable logic market — *there is no longer any need to resort to low-density PALs for speed-critical designs.* Multiple high-speed PALs and GALs can be integrated into the XC7318 and XC7336 without compromising performance. The resulting higher integration levels also lead to reductions in design complexity and cost; fewer parts translates into lower system power consumption, fewer programming patterns, lower inventory requirements, increased system reliability and reduced board space.

The XC7336 and XC7318 can support the logic requirements of today's fastest systems through the use of the XACT EPLD Version 5.0 software tools. XACT 5.0 EPLD provides push-button conversion of PAL designs, and interfaces with all of the industry's most popular schematic and synthesis design tools, as well as many popular PAL-specific design packages such as PALASM, ABEL and CUPL. Furthermore, design fitting and the use of special XC7000 architectural features are done automatically by the new tools. XACT EPLD 5.0 is so automatic that the Xilinx EPLD architecture can remain completely transparent to the user, and the design can still be completed with ease.

The XC7000 EPLDs and supporting software provide an extremely effective solution to design integration needs. Designs that are easily implemented in the XC7336-5 and XC7318-5 devices would require competitive offerings that are up to 60% larger. Development tools for the XC7318 and XC7336 include SMARTswitch™, an optimizer that automatically utilizes an extra level of high fan-in logic embedded in the device's Universal Interconnect Matrix.

The XC7318-5 and XC7336-5 are available now in 44-pin PLCC and space-saving 44-pin PQFP packages. Programming support is available using the HW120 programmer for prototyping, with Xilinx offering programming services directly from the factory for large-volume requirements. Xilinx also supports an extensive network of third-party programmer platforms. (See page 14) *Please contact your Xilinx sales representative for more information.* ◆

XILINX®

Quad Flat Packs for EPLDs

Xilinx is expanding the packaging options for the XC7300 EPLD family by introducing quad flat packs, including PQ44 versions of the XC7318 and XC7336 and PQ100 versions of the XC7372 and XC73108. The new packages will be available for these products in all speed grades.

The PQ44 and PQ100 have significantly smaller dimensions than their PLCC counterparts. The PQ44 is 43 percent smaller in terms of surface area than the PC44; the PQ100 is 53 percent smaller than the PC84. This smaller area helps to conserve board real estate, a vital concern when form factor is a significant issue.

The PQ100 offers another specific advantage: the increased number of pins allows the XC7372 and XC73108 to be more completely bonded out than with the PC84 option. This increases the number of user-accessible I/Os compared to the PC84 package.

Engineering samples are available now for both package options, with production scheduled for the end of September. Programming support will be available on the HW120 (both algorithms and adapters) in August. The PQ44 and PQ100 package files for the XEPLD 5.0 development tools are available on the Xilinx Bulletin Board Service. *Please contact EPLD Marketing or your local sales representative for more information.* ◆

"The PQ44 and PQ100 have significantly smaller dimensions than their PLCC counterparts."

Xilinx Introduces New FPGA Technology

In July, Xilinx released initial information about a new FPGA technology currently in development. Named MicroVia® technology, it has three main elements: three layer metal technology, a metal-to-metal antifuse, and a sea-of-gates architecture.

The combination of these three elements has the potential to make MicroVia-based devices very low cost. The routing and programmable elements can be constructed "above" the logic in the upper layers of metal. This means that the FPGA can have extensive routing, a small die size, and high gate use.

The MicroVia technology will be the third process — after SRAM and EPROM — adopted by Xilinx for the manufacture of programmable logic devices. We believe that multiple processes and device architectures are required for the fastest improvement in the capabilities of today's PLDs. We don't believe that any one process is "best" — rather there are tradeoffs associated with each process and architecture. MicroVia-based products will complement our current and future products made in other technologies. Our strategy is to invest in multiple technologies.

As stated by Bernie Vonderschmitt, Xilinx President, "The FPGA products that will result from the MicroVia technology, along with cost reductions in our current SRAM- and EPROM-based offerings, clearly underscore our commitment to an ongoing program of continually driving down costs through technology and process improvements."

Specific product details will be available when the products are in production, which is projected for early 1995. ◆

"Design Once"

The product development and time-to-market advantages of using FPGAs has been well understood for some time. More recently, users have been appreciating the Xilinx commitment to continual cost reduction of FPGA components through improvements in IC architectures and processes.

Further dramatic cost reductions, with no additional engineering load, are possible through conversion to HardWire products. HardWire Logic Cell Arrays (LCAs) are mask-programmed versions of the XC2000, XC3000, XC3100, and XC4000 family FPGA devices. In the HardWire components, the FPGA's memory cells and logic that they control have been replaced by metal connections. Thus, a HardWire LCA is a semicustom device manufactured to provide a specific functionality, yet is completely compatible with the FPGA it replaces.

"...the conversion process is completely transparent to the user, seamless and risk free..."

Xilinx uses the term "Design Once" to describe the key HardWire advantage — the conversion process is completely transparent to the user, seamless and risk free, without the engineering burden associated with gate array redesign.

Over the last year, users moving into volume production have migrated their FPGA designs to HardWire products in record numbers. Xilinx leading-edge programmable IC solutions combined with HardWire cost reduction provide high-volume users with a cost effective manufacturing solution as well as a highly efficient design and development methodology.

The proprietary HardWire conversion process is turn-key — automatically mapped, verified and guaranteed by Xilinx based upon the user's verified FPGA design. No remapping to other cell libraries, redesign of logic, translation, vector generation or simulation is required of the user. Using patented test logic structures and Automatic Test Pattern Generation programs (ATPG), Xilinx guarantees 100% fault coverage without the need for test development or vector generation by the user. Also, all unique functions and features of the FPGA (e.g., power-on reset, configuration modes, JTAG and RAM/ROM) are supported. Prototype devices are delivered in as little as 4 to 6 weeks in the qualified production package and process.

The "Design-Once" advantage is made possible by Xilinx DesignLock Technology, an automatic conversion process which preserves all characteristics and features of the original FPGA design. Once proven in production, the working FPGA design is, itself, the primary design verification vehicle. User simulation of the converted HardWire design is not required because the design has been effectively emulated in hardware by the FPGA.

DesignLock Technology uses the .LCA file generated by the designer in developing the FPGA as the verified physical data base from which the HardWire device is personalized. The relative physical LCA placement and routing of the FPGA, and therefore relative timing, is precisely maintained in the HardWire device. The conversion process is self-verifying and all device features are preserved.

The resulting HardWire die size is

Continued

greatly reduced due to the elimination of all programmable elements. Overall routing delays are reduced as well, while relative timing is maintained. Prior to beginning the HardWire conversion, Xilinx uses proprietary software to identify potential design hazards, such as race conditions, asynchronous signals, delay lines and one-shots, which may be exacerbated by the reduced routing delays. Such potential hazards are analyzed, reviewed with the designer and corrected as necessary.

Leading-edge system design innovations are increasing at a furious rate. In order to compete effectively and meet market windows, designers must reduce development time while making the most efficient use of limited engineering resources. Also, new and evolving system standards require adaptive and agile design methodologies. Today, digital designers are increasingly looking at ASIC solutions in terms of total cost of ownership — development efficiency, performance and density, as well as component cost. The high growth rate of programmable logic speaks for its efficiency as a total development and production solution. This, combined with HardWire cost reduction, is providing Xilinx users with optimum product solutions. See for yourself and Design Once! ◆

Xilinx Announces Third Price Reduction of 1994

Xilinx continues to drive down the cost of FPGAs with design improvements, new process technology and the addition of lower cost packages. Beyond the reductions announced in January and April, prices have been reduced again in July for the XC4000H family, the XC3000A family and all XC4000 devices in 208- and 240-pin plastic quad flat packs. These price reductions are part of an aggressive effort to make Xilinx FPGAs a cost-effective solution for high-volume applications.

The XC4000H family is a high I/O version of XC4000. With almost twice the number of IOBs and pins as the XC4000, these products already offered the lowest price per pin of any FPGA. In July, the prices for these devices were reduced by up to 22 percent. For I/O intensive applications, the XC4000H clearly offers a tremendous value.

Xilinx has also reduced prices on the XC4008, XC4010, and XC4013 devices in plastic quad flat packs by up to 10 percent in July. These packages have the same pin-outs and dimensions as the higher-priced metal quad flat packs. Using the PQFP as a drop-in replacement for a MQFP can combine the inherent cost savings of a plastic package with the new price reductions for an immediate cost savings of up to 25 percent.

Prices for the XC3000A family also were lowered in July. Through design improvements, the XC3000A not only offers higher performance and improved routing resources, but now also a lower cost, making it the most cost-effective solution in the XC3000 series. The XC3000A devices are upward-compatible with the corresponding XC3000 series FPGAs and can be used as drop-in replacements.

Contact your local Xilinx distributor or sales representative for specific pricing information. ◆

Designing EPLDs Using Sy

Designing complex systems with Xilinx EPLDs keeps getting easier and easier. The new Xilinx EPLD Synopsys Interface, scheduled for release in the fourth quarter, allows designers to use either the Synopsys Design Compiler or FPGA Compiler to create efficient, high-performance EPLD designs quickly.

Now, as with Xilinx FPGAs, technology-independent, high-level design languages such as VHDL and Verilog-HDL can be used to create EPLD designs without understanding the details of EPLD architecture. However, advanced EPLD users have the option of precisely controlling design implementation from within the HDL design in order to fine-tune mapping efficiency and performance.

Comprehensive design verification is provided from beginning to end, including both functional and timing simulation, via the Synopsys VSS simulator. Xilinx software also provides a detailed timing analysis along with thorough reports of device resource utilization and physical mapping.

All EPLD-specific optimization is performed by the Xilinx DS-550 EPLD software, achieving the best possible implementation for any target EPLD. Furthermore, since design implementation takes only a few minutes, design iterations are based on actual timing and resource data rather than estimates from the synthesizer.

"...technology-independent, high-level design languages such as VHDL and Verilog-HDL can be used to create EPLD designs without understanding the details of EPLD architecture."

The following examples illustrate how easy it is to express common logic functions using generic VHDL and how to explicitly control EPLD resources.

Flip-Flops and Latches

Registers can be inferred in EPLD designs just like in any other Synopsys design. For example, the following behavioral VHDL process implements a D-type flip-flop with asynchronous clear and synchronous clock-enable:

```
process (CLEAR, CLOCK)
begin
   if (CLEAR = '1') then
      Q <= '0';
   elsif (CLOCK'event and CLOCK='1') then
      if (CE = '1') then
         Q <= D;
      end if;
   end if;
end process;
```

Simple registered functions are placed into either macrocells or input pads at the discretion of the XEPLD software, which optimizes resource allocation in the device. To explicitly assign a register to an input pad, instantiate the IFD component from the XC7000 library as follows:

```
U1: IFD port map (d=>DATA_INPUT, c=>CLOCK,
q=>REG_DATA);
```

The XEPLD register optimization feature can be disabled by specifying the NO_IFD attribute. Then, all registers in the design, except IFD instances, will be placed into macrocells.

I/O Ports

Unless otherwise specified, the Synopsys compiler automatically infers ordinary input buffers, output buffers, and I/O buffers for all top-level I/O ports.

For example, the compiler infers the generic OBUFE buffer to implement the following behavioral description of a three-state output:

```
OUTPUT1 <= SIGNAL1 when (OE = '1')
else 'Z';
```

XILINX®

opsys Logic Synthesis

The XC7000 component library also includes special-purpose I/O buffer cells that allow the explicit instantiation of EPLD-specific I/O functions. For example, a high-speed clock input can be assigned to the global FastCLK pin by instantiating the BUFG component.

Some of the inputs to an EPLD can be routed directly to Function Blocks using high-speed FastInput paths that bypass the Universal Interconnect Matrix (UIM™). To designate FastInput signals, instantiate the "F" attribute cell in the source design and connect it to the intended FastInput signal. For example:

```
port (STROBE : in std_logic; ...
  — in the design entity —
U1: F port map (STROBE);
  — in the architecture body —
```

Similar XC7000 library attribute cells can be used to control the automatic assignment by XEPLD of global clocks, FOE signals, input-pad registers, UIM AND-gates, and Fast Function Blocks.

High-Performance Counters

The XC7000 library supports the DesignWare increment and decrement operators, allowing counter functions to be described behaviorally as in any other Synopsys design. Full-featured counters of any size are implemented automatically and efficiently by XEPLD to operate at the EPLD's maximum clock frequency. The following example shows a 24-bit loadable counter with synchronous reset and count-enable controls described behaviorally in generic VHDL:

```
signal COUNT: std_logic_vector (23 downto 0);
  -- in architecture header —
process (CLOCK)
begin
  if (CLOCK'event and CLOCK = '1') then
    if (RESET = '1') then
      COUNT <= "000000000000000000000000";
    elsif (LOAD = '1') then
      COUNT <= DATA;
    elsif (COUNT_ENAB = '1') then
      COUNT <= COUNT + 1;
    end if;
  end if;
end process;
```

If programmed into an XC7354-10 EPLD, the above counter can be switched at random between increment, hold, load, and reset operations on any clock cycle while being clocked at up to 76 MHz.

Arithmetic Functions

The XC7000 library also supports the DesignWare addition (+), subtraction (-) and magnitude compare (<, <=, >=, >) operators, allowing generic behavioral VHDL descriptions of these arithmetic functions for any size operands. Each of these functions benefits from the high-speed arithmetic carry chain of the EPLD architecture. For example, the following behavioral VHDL description can be used to express a 16-bit adder-subtracter:

```
signal A, B, SUM : std_logic_vector (15 downto 0);
  — in arch. header —
process (A, B, SUBT)
begin
  if (SUBT = '1') then
    SUM <= A - B;
  else
    SUM <= A + B;
  end if;
end process;
```

For creating registered arithmetic functions, such as accumulators or pipelined adder/subtracters with register control logic, the XC7000 library contains EPLD-specific macros that produce the most efficient implementations. The ACC component (Adder/Subtracter/Accumulator) or the ADSUR component (Adder/Subtracter with registered output) can be instantiated, and each is scalable for any size operand.

For example, to connect the outputs of the above 16-bit adder-subtracter into a register with clock-enable and reset, replace the process shown above with the following ADSUR component instance. This ensures that each bit of arithmetic and register logic remain together in the same EPLD macrocell:

```
U1: ADSUR generic map (width=>16)
port map (a=>A, b=>B, q=>SUM,
sub=>SUBT, c=>CLOCK, ce=>CLK_EN,
r=>RESET);
```

If programmed into an XC7354-10 EPLD, the above registered adder can be clocked at up to 48 MHz.

The Xilinx EPLD Synopsys Interface allows design at the conceptual level without concern for device details. Hence, users are free to concentrate on the functionality of their designs and to create working devices quickly with minimal effort. Full-simulation capabilities and comprehensive reports to verify your design process are provided. In addition, the XC7000 library and the Xilinx software allow users to exercise precise control over all specialized device features to create the most efficient, high-performance designs.

The EPLD Synopsys Interface will be included with the next release of the DS-401 Xilinx-Synopsys Interface (XSI) software.

Please contact Xilinx EPLD product marketing for more information. ♦

Previewing the Next Major XACT Software Release

The next software release will take another major step to improve design productivity when implementing EPLDs and FPGAs.

Interactive applications are being redesigned with the focus on ease-of-use. These programs will run as standard Windows applications on the PC and as Motif applications on Workstations. All interactive programs will also offer hyper-text-based on-line help.

"Interactive applications are being redesigned with the focus on ease-of-use. These programs will run as standard Windows applications on the PC and as Motif applications on Workstations. All interactive programs will also offer hyper-text-based on-line help."

For FPGA users:

FPGA users can expect to see a new a Design Manager replace the current XDM tool. This new product will help users manage their design revisions, control the processing of their designs and launch other Xilinx design tools. It will contain a graphical design flow manager, easy-to-use option selection, a report browser, and on-line help. Other interactive tools such as XDelay, XChecker, and MakePROM are being redesigned to enhance their functionality and ease-of-use.

Designers using higher-density FPGAs will have access to the new XACT-Floorplanner, an interactive design layout and constraints generation tool. The Floorplanner will be especially useful for bus-oriented or highly-structured designs.

For EPLD users:

The new Design Manager will also support Xilinx EPLDs with improved timing-driven design optimization. In addition, this release is expected to include automatic multi-chip partitioning software and automatic device selection within the EPLD family of devices.

The current target for production release is the 2nd quarter of 1995. Limited Beta copies may be available in the 1st quarter. The features and functions may vary (increase or decrease) depending on development schedules and quality checks. Xilinx will keep users updated on the current status and feature set of this next release via *XCELL*, customer presentations and mailers to in-warranty customers. ◆

Patch Required for XC3000A, XC3000L, XC3100A

An XACT 5.0 patch with new BSD files has recently been mailed to all XACT 5.0 sites. After installing this patch, all XC3000A, XC3100A and XC3000L designs done with XACT 5.0 must rerun Makebits. No rerouting or resimulation is necessary. Designs done with ES-3KA software or XC3000/XC3100 designs are not affected. ◆

Automatic Software Makes PAL to EPLD Conversion Easier

The type of design implemented in a PAL® typically consists of wide fan-in, two-level logic (AND-OR) functions. Since EPLDs contain these types of logic structures and have the deterministic timing and high performance of PALs, it's easy to see why many designers are turning to these devices for PAL consolidation. However, there are some subtle (and perhaps not-so-subtle) differences between the architectures of PALs and EPLDs which raise concerns about whether a PAL design will operate the same way when designed into an EPLD.

During a typical PAL-to-EPLD conversion process, a designer might unwittingly change the logical functionality and performance of a known working design. These problems can be avoided by using Xilinx EPLD hardware and the DS-550 EPLD implementation software because they work together to maintain the correct logic functionality of the design.

Consider a design originally implemented with popular industry-standard 22V10 PALs. The registers on the 22V10 may be synchronously preset and asynchronously reset. Since the inversion of output signals occurs after the register, signals declared as active Low actually behave at the device pin as if they were synchronously reset and asynchronously set.

Because inversion control occurs before the register in the Xilinx EPLD, the DS-550 automatically changes the asynchronous reset functions to an asynchronous preset in cases where the output was declared active Low in the original design.

The registers on the Xilinx EPLD don't have explicit synchronous presets. However, the DS-550 software automatically programs the macrocell to emulate this function by adding a preset product term to the original equation. As a result, a design created originally for the 22V10 will function exactly the same in a Xilinx EPLD.

PALCONVT

The DS-550 software also contains a PAL conversion utility, PALCONVT, that speeds up the conversion process. PALCONVT generates a top-level design file that defines each signal in the design as an EPLD input pin, output pin, or node. It automatically inserts statements in the file that tell the fitter which PAL equation files are used in the design.

PALCONVT automatically corrects signal inversion inconsistencies between different PAL files. To establish proper interconnectivity, signal names in the PAL file pin lists must match and be declared with the same polarity. Depending on the PAL compiler used, the signal polarities may not match. Rather than require the designer to edit the PAL equation files, PALCONVT automatically resolves the polarity inconsistencies in the pin lists while maintaining the correct logical functionality of the original design.

After verifying that the signal declarations are correct, a designer can use the DS-550 fitter to process the design. The fitter reads the top-level design file and, using the PALCONVT-generated pin declarations to define the EPLD's I/O, maps the logic equations from the included PAL files into an EPLD.

The DS-550's powerful PAL conversion features allow designers to convert multiple PAL designs into one Xilinx EPLD easily. In addition to speeding up the conversion process, the software automatically programs the Xilinx EPLD to emulate 22V10-specific functionality, thus eliminating design inconsistencies. ◆

Improved Synopsys Libraries To Be Availabile in October

Synopsys users targeting Xilinx FPGA devices will soon be able to take advantage of the recent development work in the area of synthesis libraries. Version 3.1 of the Xilinx/Synopsys Interface (DS-401) will be available in October. These libraries incorporate a number of improvements.

1. All cell names and pin names have been converted to XACT 5.0 Unified Library standards.
2. The design flow, including netlist translation and processing through XACT, has been updated to allow simple operation with XACT 5.0. For example, a utility program is now included that helps the user set up the Synopsys setup file with the required library information.
3. Libraries for all popular XC3000, XC3000A, XC3000L, XC3100, XC3100A, XC4000, XC4000A, and XC4000H devices are now included.
4. Wire load models have been re-engineered to more accurately reflect actual device delays. Hundreds of test cases were run on the full range of device types and speed grades, and the results were analyzed statistically to produce the new models. Using FPGA Compiler, estimated circuit delays are usually within ±10% of post-layout results.
5. Input/output cells have been updated to allow more comprehensive pad insertion by the Synopsys tools. Using features in Synopsys version 3.1, the compiler can now automatically insert a wide range of I/O pads including input, output, bi-directional, input registers, and output registers. In addition, separate I/O libraries are now available for the XC4000, XC4000A, and XC4000H families that reflect the differences in the I/O structures for those devices.

Users of the new Xilinx/Synopsys Interface v3.1 will achieve a higher quality of result with less effort and a higher degree of predictability. Many of the improvements are available to all Synopsys synthesis users; however, the high degree of coupling between FPGA Compiler and XACT 5.0 makes that combination the best choice for performance-oriented designs or where a close match between synthesis results and post-layout results is important. ◆

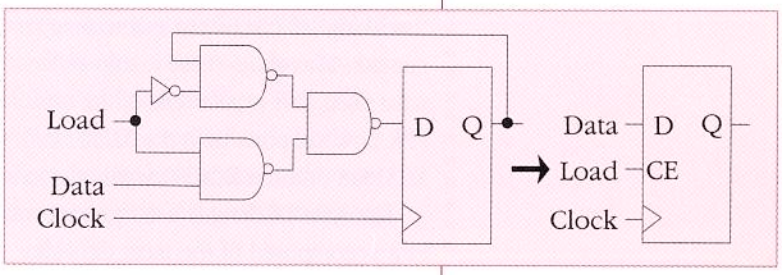
"Users of the new Xilinx/Synopsys Interface v3.1 will achieve a higher quality of result with less effort and a higher degree of predictability."

New Features Added to ViewSynthesis

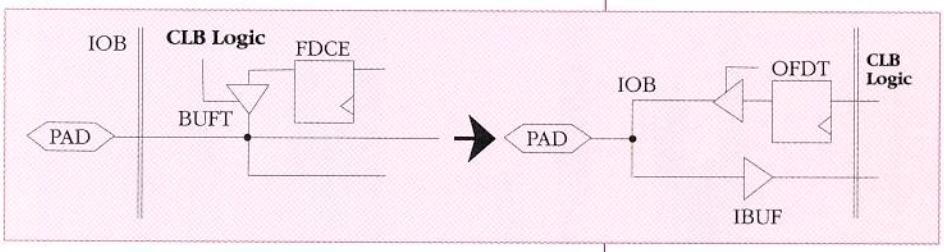
A new, improved version of the ViewSynthesis logic synthesis compiler is available. ViewSynthesis is included in the Xilinx XACT 5.0 Viewlogic Stand-Alone Extended package or can be purchased from Viewlogic. The ViewSynthesis v2.3.1 release contains many new features that will increase device performance and utilization when synthesizing to Xilinx FPGAs.

New features added to ViewSynthesis v2.3.1 for Xilinx FPGA devices include:

- **Synthesis to the Clock Enable Pin**
  ViewSynthesis v2.3.1 will take advantage of the dedicated clock enable pin for the flip-flops inside the Configurable Logic Blocks (CLBs) automatically for both XC3000 and XC4000 FPGAs. No additional work or attributes are required. ViewSynthesis will synthesize to the clock enable pin whenever it finds a multiplexer preceding the data pin of a register.



- **Synthesis to logic in IOBs (including latches, flip-flops, and 3-state buffers)**
- **Automatic I/O insertion**
  A program called IOPAD is included with ViewSynthesis v2.3.1. This program will insert the I/O and use the logic in the Input/Output Block (IOB) automatically as in the example below. The logic resources used in the IOB include the input latch (INLAT), the input and output flip-flops (OUTFF, INFF), and the 3-state buffer (OBUFT, OUTFFT). Schematics are no longer required to instantiate the I/O. The IOPAD program will instantiate the I/O without the need for any additional VHDL or specify special attributes.
- **Global signal and Global Clock Buffer specification**
  This feature provides for the easy specification of signals to be used as global signals or connected to global clock buffers.

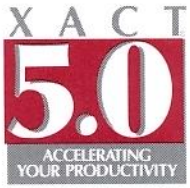


New synthesis libraries are not required for ViewSynthesis v2.3.1. However, ViewSynthesis v2.3.1 also includes library support for the Xilinx Unified Libraries. As with previous versions of ViewSynthesis, the libraries include support for all XC2000, XC3000 and XC4000 FPGA devices.

Viewlogic is not stopping here. The next version of ViewSynthesis, v3.0, promises more Xilinx-specific features, such as support for XACT-Performance™, XC7000 EPLDs, CLB mapping for XC3000 and XC4000 FPGAs, Xilinx optimization and finite-state machines. This version should be available in the fourth quarter of 1994.

ViewSynthesis v2.3.1 is being sent as an update to all Xilinx users under current maintenance for the DS-VLS-EXT-PC1 package, and, of course, is included in new purchases of this product. Viewlogic has this version available by request for all of their customers under maintenance. ◆

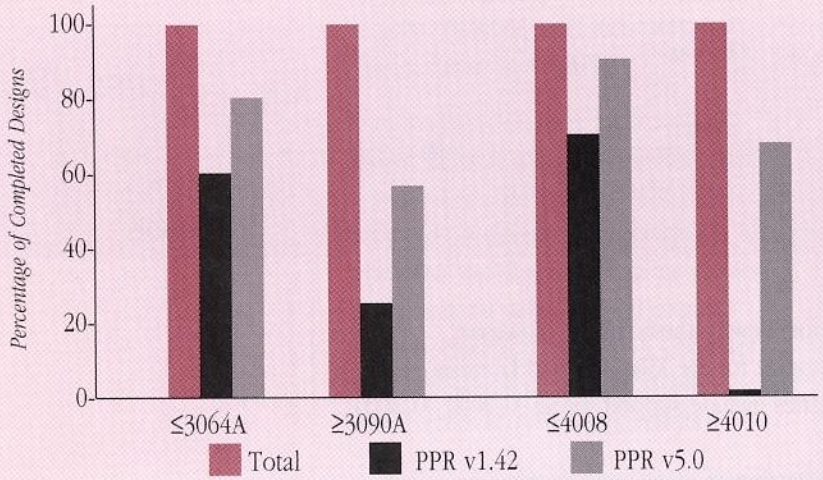
XACT 5.0 Delivers "Place & Route" Improvements



XACT 5.0 is delivering industry-leading completion rates and performance levels for FPGA placement and routing. Using a benchmark suite of more than 250 difficult-to-route designs, comparisons were made against the previous version of the Xilinx place and route tools, PPR v1.42 (XC4000) and APR (XC3000). Overall results are shown in the accompanying chart. These comparisons were made using default parameters, without any "tweaking" of placement and routing parameter. Therefore, these results reflect what users can expect "out of the box." PPR now supports both the XC3000A/XC3100A and the XC4000 family devices.

The suite of designs used to benchmark PPR is composed of the most difficult designs received from users plus test cases developed here at Xilinx. Many are synthetic cases designed to stress the tools and are not intended to fully route. The Xilinx R&D group refers to this test suite as the "Designs from Hell." Actual "real-life" users will experience much higher completion rates than reflected by this data.

Improved First Time Completion Rates (PPR v1.42 *vs.* v5.0)



These positive results are very design-dependent. A few user designs still will not complete automatically the first time. For those designs, adjustment of the PPR parameters controlling placement and routing effort levels may help. *(See Volume II of the XACT Reference Guide.)* If a satisfactory implementation is not achieved with the default parameters, set the effort parameters to different settings at substantially different points on the scale. However, maximum effort in placement or routing doesn't always generate successful completion, and will increase PPR run times. Again, this is highly dependent on the structure of the design. Although not a preferred method, the interactive design editor provides another option should the automatic tools fail to complete the design.

Significant device performance improvements have been achieved by PPR v5.0 as well. An average of 20% performance improvements is being seen on XC3000A and XC4000 series devices. The test suite was a subset of the "Designs from Hell" suite. A 20% improvement allows the use of a slower speed grade, resulting in a significant decrease in component costs. Again, results are design-dependent.

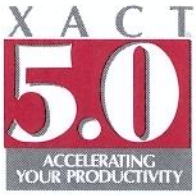
XACT-Performance, the timing-driven placement and routing feature within PPR, is now available for XC3000A/XC3100A devices as well as the XC4000 family. A new XACT-Performance "Group" feature aids users in achieving the desired device performance by allowing the tools to focus on critical timing areas of the design.

Xilinx is continuing to make algorithmic improvements that increase the capabilities of the automated implementation tools. As these improvements become available, they will be incorporated in subsequent releases. ◆

Controlling PPR Run-Time

With version 5.0, the latest update of XACT, designs have seen better chip performance and faster PPR run times overall. However, a few rare designs have actually taken longer to process using PPR's automatic defaults.

The PPR program was designed to handle a wide range of situations automatically, from a half-full 2000-gate part to a 100-percent-full 13,000-gate part. Recognizing that every situation is unique, we also have made it easy to adjust the PPR defaults to better accommodate individual needs. Benchmarks and tests are being performed continuously to determine the optimal set of defaults.



Speeding Up PPR

For XC4000 designs, we have discovered that a run time reduction of 20 to 30 percent, with a minimal impact on design performance can be achieved by the following method:

1. Using a text editor, create a file called XACTINIT.DAT in your design directory.
2. Add the line: *permute_size = 4*

To achieve an additional five to seven percent decrease in PPR run time, you can also add the following lines to that local XACTINIT.DAT file:

```
/ppr/improve_routing = false
/ppr/router_effort = not_in_use
```

If your XC3000A, XC3100A or XC4000 design does not have strict performance and routability requirements, you can dramatically reduce run-time with the following command line options:

```
PPR <design_name> placer_effort=1
router_effort=1
```

However, these two changes (especially the last one) are NOT advisable for general designs. These settings may be useful for early exploration. PPR will complete much sooner, but is likely to produce an implementation with lower performance than if the default options are used.

On the other end of the spectrum, if you have an XC3000A, XC3100A or XC4000 design that has trouble routing or achieving the desired performance, use the following PPR command line options:

```
PPR <design_name> placer_effort=5
router_effort=4
```

This forces PPR to work harder and produce a better result, but at the expense of longer run times. ◆

"The PPR program was designed to handle a wide range of situations automatically, from a half-full 2000-gate part to a 100-percent-full 13,000-gate part. Recognizing that every situation is unique, we also have made it easy to adjust the PPR defaults to better accommodate individual needs."

Fast Integer Multipliers

This application example was prepared by Ken Chapman, a Xilinx Field Applications Engineer based in England. An abbreviated version appeared in EDN Magazine's Design Ideas column in March, 1993, and was recently chosen the overall 1993 Design Idea winner. Congratulations, Ken!

Digital multipliers are needed in many system applications, including digital filters, correlators, and neural networks. These multipliers typically are required to handle operands of up to 16 bits, and need to provide results in less than 50 ns (20 MHz systems). Figure 1 is a common module found in many signal processing applications. (For example, with the output 'y' applied to an activation function, this would be a simple neuron.)

The increasing density of programmable logic devices has led to complete systems and sub-systems being implemented in one FPGA device. However, digital multipliers generally are considered too slow when implemented in FPGAs, or too large to make effective use of a programmable part. As an alternative, dedicated multiplier devices are connected to the main system, often resulting in a performance degradation caused by the delays inherent in getting the data to and from the multiplier device. (Often, these data movements are time-multiplexed to reduce the large I/O requirement.) Thus, techniques for implementing a compact and fast digital multiplier in a programmable part are needed.

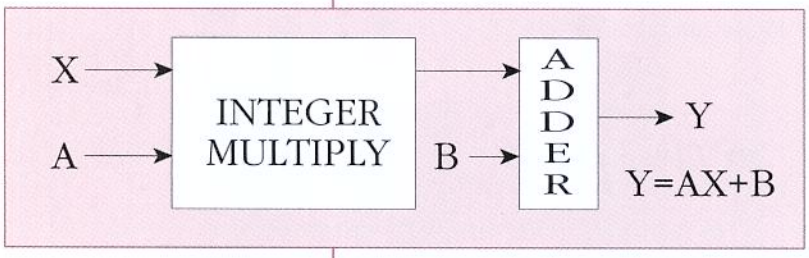


Figure 1

Implementation Techniques

There are three main implementations of digital multipliers:

1) **Shift and Add** — One operand is shifted to the left by one bit each cycle and applied to an accumulator when the corresponding bit in the second operand is high.
2) **Look-up table** — The operands are applied as addresses to a pre-programmed memory that outputs the result.
3) **Logical tree** — Each of the resultant bits are a logic function of the relevant bits of each operand.

Using a 16-bit x 16-bit multiplier as an example, let's consider each technique.

The **shift and add** implementation is compact but very slow. The result is obtained after 16 clock cycles and the accumulator must be 32-bits wide. The accumulator will determine the maximum clock rate dependent on the carry logic chain. This implementation also precludes any new operands from being applied until the calculation has been completed, read, and cleared.

The speed of the **look-up table** solution depends on the speed of the memory used, but rapidly becomes unwieldy as operand size increases. This 16x16 example requires a 4,294,967,296 x 32-bit memory! Small multipliers work well this way, such as a 4-bit x 4-bit multiply implemented in a byte-wide ROM.

Some very complex implementations of **logical trees** have been developed employing product sharing, and are to be found in many dedicated arithmetic devices. The gate count is high and can be considered a reduced version of the ROM table. The logic involved tends to have

high fan-in requirements (up to 32 inputs).

The XC4000 FPGA architecture includes the types of logic resources needed for digital multipliers:

- High-density devices are available (currently, up to 25,000 gates), providing ample density for a multiplier and substantial additional logic.
- Fast, dedicated carry logic circuitry provides for compact and high-speed simple arithmetic functions such as addition.
- The XC4000 architecture includes the ability to configure custom blocks of RAM and ROM using the look-up-table-based function generators.

Having all these features available means that any of the described techniques may be implemented.

Fast and Compact Multipliers

If the multiplier needs to be both fast and compact, the choice of multiplier implementations can be difficult.

The compact **shift and add** technique may be too slow for many applications, but is easily implemented in the XC4000 architecture. The accumulator can make excellent use of the fast carry logic, with a reasonably low cycle time (even for a result of 32 bits).

A **look-up table** built from a custom ROM block is ideal for small multipliers. Since any configurable logic block (CLB) can be configured as either two 16 x 1 memories or a single 32 x 1 memory, it is easy to see how small look-up tables can be implemented. Unfortunately, the larger memory requirements of larger operands can use up the available CLB resources rapidly. Operands with more than four bits soon become difficult to handle.

Logic trees can be implemented, but again, as the operand width increases, the fan-in requirement of the logic soon exceeds the nine inputs available to a CLB. As a result, functions have to be split. This prevents a compact design, and impacts performance due to the multiple block delays between the operands and the result. However, the XC4000 device architecture does offer an easy way to pipeline any design, using the two registers in each CLB. Once again, operands of more than four bits become an escalating problem to handle.

"The real strengths of the XC4000 FPGA architecture in this application are its ability to handle small look-up tables, provide logic functions of less than nine inputs, and form fast adders of any size."

The real strengths of the XC4000 FPGA architecture in this application are its ability to handle small look-up tables, provide logic functions of less than nine inputs, and form fast adders of any size. Therefore, the ideal solution may be to use a hybrid technique tailored to these properties: using small look-up tables for partial products and combining the results by addition.

Anatomy of a Hybrid Multiplier

Partial products are formed by splitting the first operand into sections and multiplying each section by all of the second operand. These products contain all the information about the multiplication that must then be combined by addition, while at the same time restoring the bit weighting of the sections from the first operand.

Figure 2 is a block diagram of the implementation of an 8 x 8 multiplier that multiplies the two operands called 'X_OPERAND' and 'A_OPERAND'. (X-BLOX modules could be used to enter this design easily in a block diagram format.) The first operand is split into two nibbles. Both nibbles produce a 12-bit partial product after multiplication by the second operand. These products are applied to a

Continued on the next page

Fast Integer Multipliers Using FPGAs

Continued from the previous page

16-bit combining adder to form the final 16-bit result.

To restore the weighting factor of the two nibbles, the 12-bit partial products are expanded into 16-bit operands. The product from the low order nibble, with a weighting of one, is converted to 16 bits by padding four additional MSBs with the value of zero. The product from the high order nibble, with a weighting of 16, is effectively shifted by four bits using four additional LSBs, again with value of zero. The resulting 16-bit operands form the inputs to the adder.

It is not possible for the adder to overflow because it is known that an integer multiply of two operands with 'n' and 'm' bits (respectively) will generate an 'n+m' bit result. In this 8-bit x 8-bit example, the maximum result occurs when multiplying FF x FF = FE01 (hex).

To create a compact, fast and easy-to-implement design, make the look-up tables for the partial products as small as possible. The simple 16 x 1 memory elements are combined to expand the width of the memory to that required for the partial products. In this example, 16 x 12 memories are synthesized using just six CLBs, each containing two 16 x 1 tables.

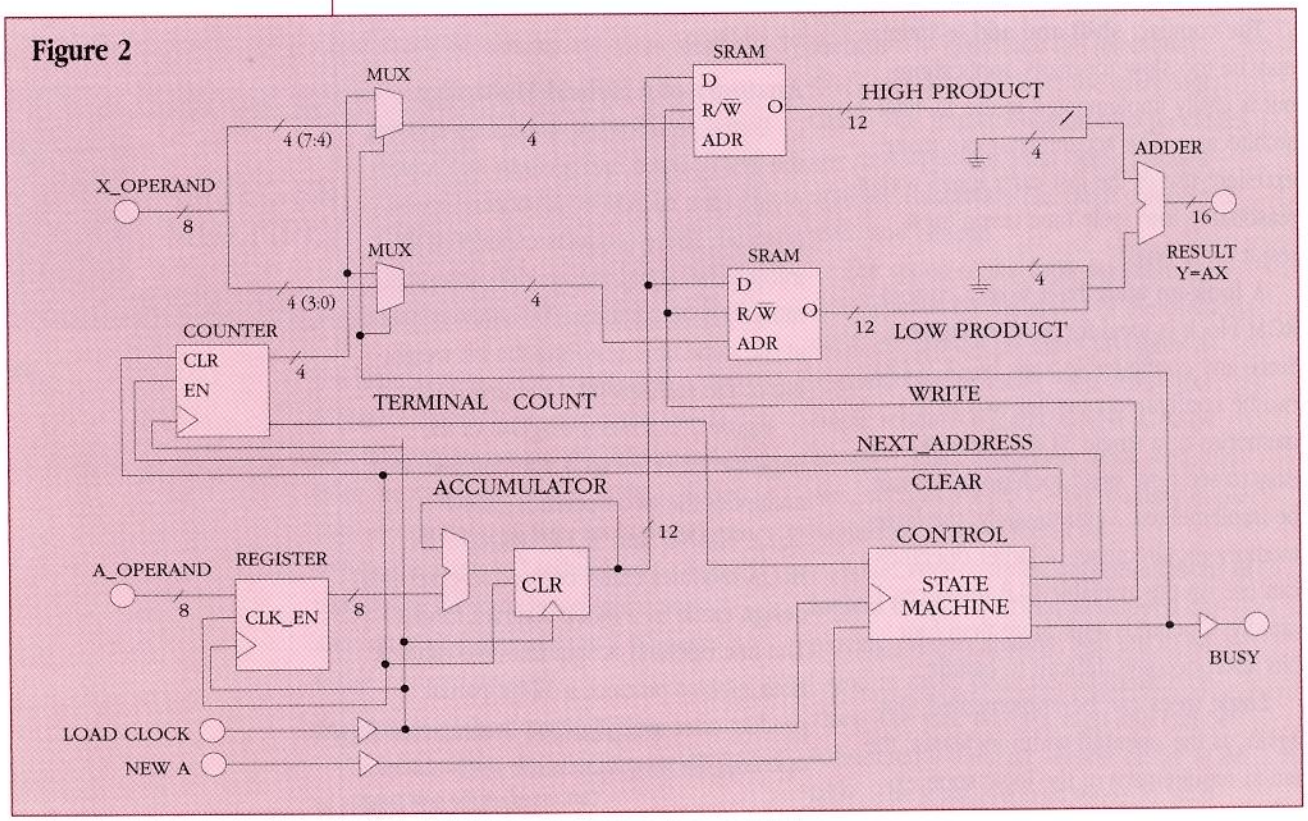
A memory with 16 addresses has 4 address lines, meaning that only the nibble section from the X_OPERAND can be applied. The need to connect the A_OPERAND has been totally eliminated in this design by ensuring that only those partial products that can be obtained by a single value of A_OPERAND are available at any one time. In other words, at any given time, the X_OPERAND can only be multiplied by a constant, as determined by the contents of the look-up tables.

By preloading the look-up tables with only those partial products that can be obtained from the present value of A_OPERAND, the size of the look-up table is greatly reduced. These partial products in the look-up table are, quite simply:

0, 1 x A_OPERAND, 2 x A_OPERAND, ... , 15 x A_OPERAND

These 16 partial products must be recalculated and reloaded for each new value of A_OPERAND. Building the look-

Figure 2



up tables from RAM means that the 16 partial products relating to a given value of A_OPERAND can be modified. This is performed using an accumulator, address counter and small state machine that accesses each location of the memory and stores a new product. Both tables are modified to contain the same data, so each nibble of the X_OPERAND is multiplied by the same value.

Clearly, the multiplier can not be used for the period during which the table contents are being modified, and results cannot be obtained quickly when changing the value of A_OPERAND. In contrast, changes in the X_OPERAND will give a result with minimal delay, and the data path can easily be pipelined to optimize performance in a clocked system.

Most applications can tolerate the time spent modifying the look-up tables for a new value of the A_OPERAND. Consider the module of Figure 1. In many cases, only the ability to modify 'A' is required in order to tune a system. Another example is video processing, where 'A' is changed only at the end of each page, during the screen fly-back.

For some applications, 'A' will be fixed from the design concept and throughout the life of the system. (Computer simulations may provide the optimum values for multiplicands and offsets in the system.) In these cases, the look-up tables can be implemented as ROMs in the XC4000 with the partial products predefined, thereby eliminating the overhead logic needed to program the tables and increasing performance by removing the address multiplexers from the data path. The reconfigurable nature of XC4000 devices would still permit different contents to be loaded into the look-up tables (representing different values for A_OPERAND) by using multiple device configuration bitstreams.

The technique expands easily by units of four bits on the X_OPERAND, where each additional nibble connects to a separate look-up table. The size and contents of each table are determined by the second operand (A_OPERAND). Further levels of combining adders are required for more than two look-up tables. These will impact performance, but their effect can be minimized with a pipelined design (registers at the table outputs and each adder). Negative values in two's-complement format can be converted using simple 'invert and add 1' pipeline stages where necessary, handling the sign bits separately.

For example, a 16 x 16 multiplier of this type uses four tables to generate 20-bit partial products corresponding to each nibble. Two 24-bit combining adders are used to form partial products for the high and low order bytes of the input data. A final 32-bit adder combines these products, restoring the weighting of 256 to the high order product (a shift of 8 bits).

Performance

The figures shown in Table 1 are worst case for an XC4000-5 device. The designs were each processed using the Xilinx automatic tools without user intervention.

The combinatorial speed is taken as a device pin-to-pin delay, including 10 nanoseconds of I/O delays. Since the multiplier is generally imbedded in the system, these time values may be deducted in estimating in-system performance.

Pipelined speed is determined by the speed of the slowest element, which is the largest adder in the system.

Style	8 x 8-bit RAM Look-up	8 x 8-bit ROM Look-up	16 x 16-bit RAM Look-up	16 x 16-bit ROM Look-up
CLB Count	39	22	117	84
Combinatorial Delay	56 ns	46 ns	96 ns	88 ns
Pipelined Performance	39.9 MHz	41.3 MHz	25 MHz	25.5 MHz

Table 1: Performance of hybrid multipliers implemented in an XC4000-5 FPGA

High-Performance EPLD Design Techniques

Twenty years ago, CMOS logic technology was undemanding. It had high noise immunity, low power, and relatively slow edges during logic-state transitions. However, in the never-ending quest for speed and as an unavoidable result of today's smaller device geometries, CMOS devices have lost some of this 'user-friendliness'.

With high-performance devices like the XC7318 and XC7336 EPLDs, output edge rates can be faster than TTL. Excessive noise due to "ground bounce" must be avoided, and transmission-line effects on printed circuit board (PCB) traces must be considered. These problems can lead to poor signal quality and noise immunity, reducing system reliability.

The XC7318 and XC7336 EPLDs feature 5 ns pin-to-pin delays and perform at clock rates of up to 167 MHz in many applications. To add to the performance of systems requiring this speed, these parts also provide 24 mA high current drivers, eliminating the need for additional buffering.

Many printed circuit board (PCB) design techniques can be used to minimize problems associated with fast IC switching rates. These techniques have been explained in detail in a Xilinx application note, *Designing with the XC7336/XC7318*, and are summarized below. (Of course, many of these same techniques can be used on high-speed FPGA designs.)

"However, in the never-ending quest for speed and as an unavoidable result of today's smaller device geometries, CMOS devices have lost some of this 'user-friendliness'."

Design Techniques

1. Use wide spacing between fast signal lines — particularly clocks — to minimize crosstalk.
2. Avoid using sockets for PCB connections. Direct soldering minimizes inductance and reduces ground rise.
3. Locate the XC7318 and XC7336 EPLDs electrically near chips they drive or are driven by.
4. Connect all device GND pins to the PCB ground.
5. Unused inputs must be tied to ground, except $\overline{MR}$ (master reset), which must be tied High.
6. Use multi-layer PC boards with separate power and ground planes.
7. Decouple the chip V_{CC} with 0.1 µF capacitors directly at each chip V_{CC} pin to the nearest electrical ground.
8. Decouple the PCB with 0.1 µF ceramic and 100 µF electrolytic filter capacitors.
9. Limit the number of outputs that switch simultaneously to minimize ground bounce. Skew signals that simultaneously switch by a few nanoseconds. One simple method is to assign subgroups of the signals to one 3-state enabled driver group and a second subgroup to a slightly delayed version of the enable signal. A similar approach can be taken with clocks. ◆

Meeting PCI Compliance with XC7300 EPLDs

The Peripheral Component Interconnect (PCI) bus is becoming a fundamental building block in many high-performance PCs and workstations. Virtually all major computer and add-on adapter card manufacturers have released or announced products that support PCI. PCI is a fast, complex bus with unique electrical specifications. Operating at 33 MHz, the PCI bus is capable of transferring 32-bit data at a rate of 132 Mbytes/second, and is ideally suited for high-bandwidth graphics, disk storage and multi-media applications.

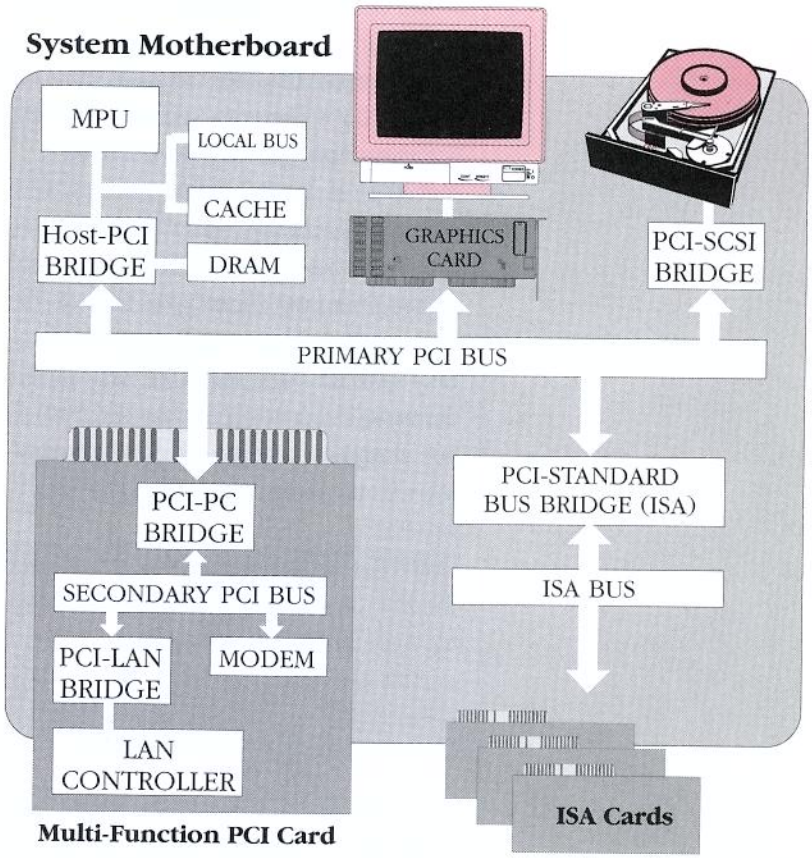
PCI equipment manufacturers face the challenge of locating PCI compliant devices. Xilinx now provides a broad range of programmable logic devices that help designers develop compliant, high-performance PCI solutions: *the XC7300 EPLD family*. These devices are fully-compliant with the PCI Local Bus specification, and can be used to implement the interface between an add-on adapter card (graphics, SCSI, ATM, etc.) and the PCI bus, also known as a PCI bridge. The PCI specification defines a multi-level bus structure of primary and secondary PCI buses, all of which include a PCI bridge.

In addition, the PCI specification outlines a 5 V to 3.3 V transition path to provide for a "quick and easy" transition from 5 V to 3.3 V component technology. The dual-voltage I/O drivers of the XC7300 EPLD family enable designers to develop a single, "universal" add-on adapter card that can operate in either a 5 V or 3.3 V PCI signaling environment.

As a member of the PCI Special Interest Group (PCI-SIG), Xilinx is dedicated to aiding designers in the development of PCI systems. Additional information that will assist users in understanding the PCI bus and the use of Xilinx devices in PCI applications is now available:

- ***The Peripheral Component Interconnect Bus*** — An X-Note overviewing the PCI Local Bus architecture, its electrical specifications, and recommended design solutions using Xilinx FPGAs and EPLDs.
- ***Designing Flexible PCI Interfaces with Xilinx EPLDs*** — An application note containing PCI compliant electrical specifications and design files for an example PCI bridge interface.

For further information regarding PCI-compliant Xilinx devices, contact your local sales representative, or Xilinx Product Marketing (Tel: 800-255-7778 or 408-879-5083 E-Mail: pci@xilinx.com). ◆



Handling Large PROM Files

The XPP utility program supports the programming of Xilinx serial PROM devices in the HW112 programmer. XPP supports the programming of multiple device types in a single session, allowing long configuration patterns for daisy-chained FPGAs to be programmed into multiple SPROM devices.

"If your daisy chain configurations are within the size that XPP can load, you can use the XPP software to split a PROM file..."

However, the current release of XPP can only load files of a limited size. For example, if you are programming XC4000 devices, the maximum file size that XPP can load is:

- a daisy chain of four XC4005 bit files
- a daisy chain of two XC4010 bit files
- a single XC4013 bit file

If your daisy chain configurations are within the size that XPP can load, you can use the XPP software to split a PROM file across multiple XC17xxx devices. This procedure is documented in the *XACT Hardware and Peripherals Guide*, in the XPP/Serial Configuration PROM Programmer section. Just enter a list of SPROM device part numbers and XXP automatically splits the file up to fit the SPROMs in the order listed; it is recommended that you always concatenate the SPROMs starting with the largest and ending with the smallest.

If your daisy chains are too long for the XPP program, the PROM programming file must be cut into smaller units.

This can be accomplished with a utility called PSPLIT available on the Xilinx Bulletin Board System (BBS), as follows:

1. Call the XILINX BBS at 408-559-9327. If this is your first time calling into the BBS, fill in the login information, hang up and than call the Technical Hotline (800-255-7778) and request a change to your BBS access permissions to allow the downloading of files. (If you are unable to access the BBS, please contact the Technical Hotline or your local Xilinx Field Application Engineer to obtain a copy of PSPLIT.)
2. Download *psplit.zip*. This program will partition a large PROM file into smaller PROM files.
3. Unzip this file in the XACT directory. This will produce two files: *psplit.exe* and a *psplit.doc*. Please read the *psplit.doc* text file for directions on the use of PSPLIT.
4. Run "psplit" on the MAKEPROM file that contains your design file.

The PSPLIT utility accepts two input parameters: the original PROM file name and the size in kilobytes of the desired partition. Multiple executions of PSPLIT may be needed for a mix of SPROM sizes.

For example, suppose you had a PROM file named *<design>.EXO* that you wanted to partition into an XC17128D and XC1765D SPROM. The XC17128D holds 128 Kbits, or 16 Kbytes. Thus, the first command line would be

```
psplit <design>.EXO 16
```

psplit program — your prom file name — 128 Kbits = 16 Kbytes

This will create 2 files: *<design>1.EXO* and *<design>2.EXO*. Both of these files would be set up for a 16 Kbyte SPROM. In this case, the second file is intended for a 65 Kbit (8 Kbyte) SPROM, so PSPLIT is run again on that file:

```
psplit <design>2.EXO 8
```

If more than one file is created by this command, the file was too big to fit into a single XC1765D.

One peculiarity about using PSPLIT arises when dealing with the XC1736

See LARGE PROM FILES, next page

Mixed FPGA Daisy Chains

Xilinx FPGAs can be configured in a common daisy-chain structure. The lead device generates CCLK pulses and feeds serial configuration information into the next downstream device, which in turn feeds data into the next downstream device, and so on. There is no limit to the number of devices in a daisy chain. XC2000, XC3000 and XC4000 devices can be mixed freely with only one constraint: the lead device must be a member of the highest-order family used in the chain. The reason is shown and explained on pages 2-29 and 2-30 of the *1994 Xilinx Data Book.*

In a daisy-chain, all CCLK pins are interconnected, and DOUT of any upstream device feeds the DIN input of its downstream neighbor. Those are the basic connections. For control purposes, it is advisable to interconnect all the slave $\overline{\text{INIT}}$ pins (the XC2000 does not have this pin) and connect them to the INIT pin of the lead XC4000 device or the RESET input of the lead XC3000 device.

Interconnected $\overline{\text{INIT}}$ pins prevent the master from starting the configuration process until all slaves are ready. This is always assured at power-up because the master uses four times as many internal clocks for the power-up as any slave. However, during the reconfiguring process, master and slave devices consume the same number of clocks to clear a frame. A fast master might be ready before a slow slave, so interconnecting INITs solves this problem.

The D/$\overline{\text{P}}$ and $\overline{\text{RESET}}$ pins (XC2000, XC3000) and the XC4000 PROGRAM pins can be used in different ways, depending on the designer's preferences regarding reconfiguration, pin use, and the need for a global RESET input.

"There is no limit to the number of devices in a daisy chain. XC2000, XC3000 and XC4000 devices can be mixed freely with only one constraint: the lead device must be a member of the highest-order family used in the chain."

If there is no need for a global logic RESET input, it is best to permanently ground the XC2000/3000 D/$\overline{\text{P}}$ pin so that the $\overline{\text{RESET}}$ input functions as the Reconfigure input, and is connected to all XC4000 PROGRAM inputs.

If there is a need for a global logic $\overline{\text{RESET}}$ input to reset all flip-flops in the user logic without causing reconfiguration, external logic must combine $\overline{\text{RESET}}$ and D/$\overline{\text{P}}$ in such a way that pulling Low $\overline{\text{RESET}}$ does not affect D/$\overline{\text{P}}$, but pulling Low D/$\overline{\text{P}}$ also pulls down $\overline{\text{RESET}}$. Figure 22 on page 2-126 in the *1994 Data Book* shows such an implementation. ◆

Large PROM Files

Continued from previous page

SPROM device. Since its capacity in bits is not an even number of kilobytes, PSPLIT will accept the text string "xc1736" as its second input parameter in place of the number of kilobytes, as in:

```
psplit <design>.EXO xc1736
```

Once PSPLIT has been used to generate separate PROM files for the multiple SPROMs to be used in a daisy chain, use XPP to program the individual SPROMs ◆

XC3100A Breaks Through the 300-MHz Barrier

(for a presettable divide-by-n counter)

The newest XC3100A-2 devices can implement a presettable divide-by-N counter at a maximum input frequency of 325 MHz. This means that any phase-locked-loop design with a frequency of up to 325 MHz can be built in one XC3100A part and that the VCO frequency can be adjusted in increments of the reference frequency.

"This design minimizes power consumption. Only a very small part of the circuit operates at the high input frequency."

The design uses a pulse-skipping technique, a variation on the popular two-modulus prescaler method used in most high-frequency PLL designs.

This design minimizes power consumption. Only a very small part of the circuit operates at the high input frequency. A 255-MHz predecessor to this design was demonstrated at the Electronica show in November 1992 and was described in *XCELL #8.*

This example demonstrates the high performance of the XC3100 family devices. 325-MHz operation must be considered the "high water mark" for FPGA design. Typical performance is usually limited to a 60 to 95 MHz system clock range, but small, well-defined subsystems can operate much faster, as demonstrated by this 325-MHz PLL design. ◆

Inexpensive Voltage Sensor and RESET Delay Circuit

Dallas Semiconductor offers an inexpensive 3-terminal device that generates an active Low $\overline{\text{RESET}}$ output (while V_{CC} is below 4.0 V), and keeps that $\overline{\text{RESET}}$ signal Low for an additional 250 to 450 ms after V_{CC} exceeds 4.0 V. The device consumes only 50 µA when $\overline{\text{RESET}}$ is High, and up to 1 mA when $\overline{\text{RESET}}$ is Low (there is an internal pull-up resistor of 5 kΩ nominal). When V_{CC} drops below 4.0 V, $\overline{\text{RESET}}$ goes Low within 100 ns.

The device also monitors its own $\overline{\text{RESET}}$ output and keeps it Low for 250 to 450 ms after an external signal has stopped pulling this pin Low. This can be used for contact debouncing.

The device comes in a 3-lead TO-92 or 4-lead SOT-223 package. The part numbers are DS1233-15 for detecting 5 V -15%, and DS1233A-15 for detecting 3.3 V -15%.

Please contact Dallas Semiconductor (tel: 214-450-0448) or their sales representatives for price and availability information. ◆

A Guide to XACT 5.0 Software Documentation

Xilinx XACT development system software includes a broad spectrum of design tools for FPGAs and EPLDs. The documentation supporting these tools is fairly extensive and is organized into five categories: *installation notes, user/design guides, reference guides, library guides, CAE interface guides,* and *release notes*. These categories are outlined in the table, with a brief description of the content of each type of manual.

Type	Content
Installation	Instructions covering standard installation and custom options, including a walk-through of a typical installation.
User/Design	Overview and general information about design entry, implementation, and verification, including design and configuration hints, design flow, specific examples, and tutorials.
Reference	Capability and function, including details about syntax, commands, options, variables, and arguments, as well as warning and error message explanations where applicable.
Libraries	Details about all design elements provided in Xilinx libraries, from primitives to macros, including symbols, functional descriptions, truth tables, and schematic representations specific to each device architecture. A functional selection guide identifies all design elements available in the architecture.
CAE Interface	Design-tool-specific information about using Xilinx software within supported CAE environments, including extensive tutorials.
Release	Descriptions of the product, identifying the applicable software versions and the devices supported, and including workarounds to any known problems.

The XACT 5.0 release is supported by sixteen manuals. The document set included with the software updates is customized for the specific software package involved. In addition, Xilinx distributes manuals from Viewlogic and Data I/O for the appropriate packages. The complete set is described in the table at right. In addition to the manuals, there are several reference cards, and a release document specific to each software module and package. ◆

Manual	Part Number
XACT Installation Guide	0401195
XACT User Guide	0401128
XACT Reference Guide, Vol I, Design Entry	0401129
XACT Reference Guide, Vol II, Design Implementation	0401130
XACT Reference Guide, Vol III, Simulation	0401131
XACT Hardware and Peripherals Guide	0401132
XACT Libraries Guide	0401098
XEPLD Design Guide	0401191
XEPLD Reference Guide	0401203
X-BLOX User Guide	0401189
Xilinx ABEL User Guide	0401136
Xilinx Synopsys Interface User Guide	0401042
Viewlogic Interface User Guide	0401133
OrCAD Interface User Guide	0401134
Mentor Graphics V8 Interface User Guide	0401135
XACT Global Index	0401137
Viewlogic Workview Series I, Volume 1	0401044
Viewlogic Workview Series I, Volume 2	0401045
Viewlogic Workview Series I, Volume 3	0401046
Xilinx-ABEL Design Reference Manual	0401221

Reference Card	Part Number
XEPLD Software	0401202
XEPLD Hardware	0401201
X-BLOX Symbol	0401196
Viewlogic Quick Reference	0401200
OrCAD Quick Reference	0401197
Mentor Graphics Quick Reference	0401198
Mentor Graphics Help	0401199

Additional copies of Xilinx manuals can be ordered through your local sales office or sales representative.

General

Q. I can't run XKEY.EXE to configure my programmable key.

A. XKEY is used to configure the new programmable key. The version included with the XACT 5.0 release requires that a co-processor be present on the PC. This means that 386s and 486SXs cannot run XKEY. The workaround is to download XKEYEMUL.ZIP from the BBS.

Q. I can't install the XACT 5.0 PC software from a CD-ROM attached to my SUN workstation.

A. By default, the install program checks to see which platform you're running on, and only lets you install software for that platform. For example, this means that you cannot normally install PC software from a SUN CD-ROM drive. The workaround is as follows:

1) In a place on the network that is accessible by the PC, create a CD image directory called "cd_image", and a subdirectory named "pc1":
   mkdir /<path>/cd_image
   mkdir /<path>/cd_image/pc1
2) Copy the top-level files from the CD into the CD image directory:
   cp -p /cdrom/* /<path>/cd_image
3) Use the cd_copy program to copy the contents of the pc1 directory on the CD-ROM:
   /cdrom/sn2/install/cd_copy. -r /cdrom/pc1 / <path>/cd_image/pc1
4) On the target PC, mount the workstation drive.
5) From the PC, go to the mounted drive and run the INSTALL.EXE program.

EPLD

Q. How are pins assigned in an EPLD design?

A. There are two ways to assign pins before running the fitter:

1. In a PLD file, insert the keyword PIN followed by a list of numbers at the end of the pin declaration (for example: INPUTPIN sig1 sig2 sig3 PIN 11 12 13). This can be done only in top-level files (not include or schematic PLD files).
2. On a schematic, use the LOC=<pin#> attribute on PAD symbols.

Q. How do I freeze my pinout for design iterations?

A. After running the fitter once, run PINSAVE. This creates a file *<design>.vmf*. This file can be edited to change the pinout of the design. Use the -f (pin freeze) option during subsequent runs of fiteqn or fitnet to use the information in the .vmf file.

Q. During simulation, all flip-flops are stuck at 0 or the preload value.

A. There is a global signal in the simulation netlist that affects flip-flops. PRLD is an active-High signal that forces all flip-flops to their preload values. Make sure that PRLD is forced High to initialize the flip-flops at the start of simulation, then forced Low to release the flip-flops.

FPGA

Q. When I lock down some of my TBUFs, PPR gives an error message. This worked OK with the pre-XACT 5.0 software.

A. TBUF bus-lines cannot be partially constrained in PPR 5.0.0. Currently, you must constrain all TBUFs on a bus, or constrain none of them. We suggest that you try removing all of the TBUF constraints. Much work was put into PPR relating to TBUF placement and you should see much better results, even without constraints.

Mentor Graphics (HP)

Q. When PLD_DMGR is invoked, a warning message is generated and no icons appear.

A. Some HP systems do not have the necessary fonts installed for use with XACT 5.0. The fix is to perform mkfontdir on $LCA/mgc. This will compile the HP fonts and create the necessary fonts.dir.

Q. I can't simulate RAMs or ROMs correctly.

A. The RAM and ROM macros included in the Mentor Graphics Interface for the HP platform will not simulate properly. An update is available from Xilinx Technical Support to fix this problem. NOTE: This problem only occurs in the HP version of the interface. The Sun version works correctly.

ViewLogic

Q. Sometimes Workview does not recognize my old Xilinx key when the new programmable key is attached.

A. A possible workaround is to set SECDLY to 64000 in the WORKVIEW.INI file. This has fixed some, but not all, of the problems. Other workarounds are being investigated. Please contact the Hot-Line if you are still experiencing problems.

Q. I'm trying to simulate an old XC4000 design with XACT 5.0, but VSM complains about the STARTUP symbol.

A. If VSM 4.1.3 (included with XACT 5.0) is run on a design created with pre-unified libraries, VSM may report:

```
ER Error - Could not find WIR file startup.1
```

The solution is to add the shm4000 library (included with XACT 5.0) to the VIEWDRAW.INI file. This will allow VSM to find the files that it needs.

Q. During EPLD simulation, why do so many internal nodes have a "?" value?

A. During logic collapsing, most internal signal names are lost. A list of signals that will be visible during simulation can be found in the *<design>.lgx* file.

OrCAD

Q. After installing the XACT 5 unified libraries, the following error occurs when invoking OrCad SDT 4.xx:

```
library header is incorrect for c:\xact\xc4000
```

A. The unified libraries shipped with XACT 5 are in the OrCad 386+ 32 bit format. These libraries are not compatible with OrCad 4. The unified libraries are NOT available for OrCad 4. You may continue using OrCad 4 with the pre-XACT 5 libraries. However, you will not be able to take advantage of the new features offered with the unified libraries. An upgrade to OrCad 386+ is recommended. ◆

FAX RESPONSE FORM

Corporate Headquarters
Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
Tel: 408-559-7778
Fax: 408-559-7114

Europe
Xilinx, Ltd.
Suite 1B, Cobb House
Oyster Lane
Byfleet
Surrey KT147DU
United Kingdom
Tel: 44-932-349401
Fax: 44-932-349499

Japan
Xilinx, KK
Daini-Nagaoka Bldg. 2F
2-8-5, Natchobori,
Chuo-ku, Tokyo 104
Japan
Tel: 81-33-2879191
Fax: 81-33-2979189

Asia Pacific
Xilinx Asia Pacific
Unit 2308-2319 Tower 1
Metroplaza
Hing Fong Road
Kwai Fong, N.T.
Hong Kong
Tel: 852-410-2740
Fax: 852-418-1600

FAX in Your Comments and Suggestions

To: Brad Fawcett, *XCELL* Editor **Xilinx Inc. FAX**: 408-879-4676

From: _____ Date: _____

❑ **Please add my name to the XCELL mailing list.**

NAME _____

COMPANY _____

ADDRESS _____

CITY/STATE/ZIP _____

PHONE _____

❑ **I'm interested in having my company's design featured in a future edition of XCELL as a Customer Feature.**

Comments and Suggestions: _____

Please use this form to FAX in your comments and suggestions, or to request an addition to the XCELL mailing list. Please feel free to make copies of this form for your colleagues.

XILINX®

2100 Logic Drive
San Jose, CA 95124-3450

U.S. Postage
PAID
First Class
Permit No. 2196
San Jose, CA