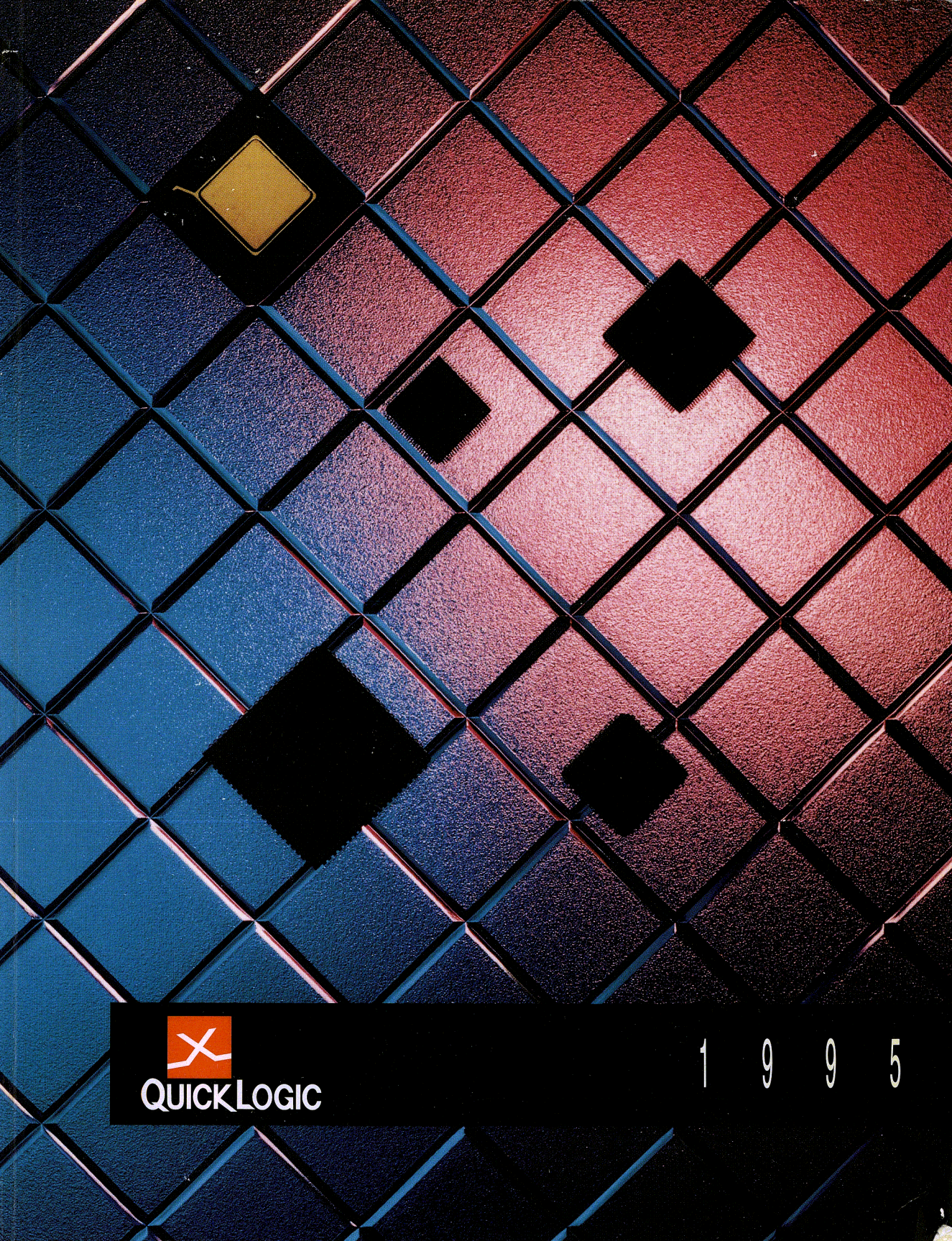
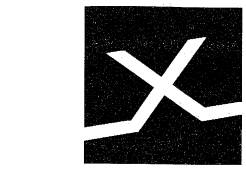

QUICKLOGIC
1995



QUICKLOGIC

REVOLUTIONIZING THE FPGA INDUSTRY

QuickLogic Corporation
2933 Bunker Hill Lane
Santa Clara, CA 95054

General Information (408) 987-2000
Applications Hotline (408) 987-2100
FAX: (408) 987-2012
EMAIL: info@qlogic.com

pASIC®, Vialink®, and QuickLogic® are registered trademarks and SpDE™ , *Wild*Cat™ and Quick*Works*™ are trademarks of QuickLogic Corporation

QuickLogic Corporation acknowledges the trademarks of other organizations for their respective products or services mentioned in this databook, including:

PAL® and Palasm® are registered trademarks of Advanced Micro Devices
Verilog® is a registered trademark and Concept™ is a trademark of Cadence Design Systems, Inc.
Data I/O® is a registered trademark and Synario™, ABEL™, and ECS™ are trademarks of Data I/O Corporation
PowerPC™ is a trademark of IBM Corporation
Pentium™ is a trademark of Intel Corporation
SmartModel® is a registered trademark of Logic Modeling Corporation
MSDOS® and Microsoft® are registered trademarks and Windows™ is a trademark of Microsoft Corporation
Motif® is a registered trademark of Open Software Foundation
Sun Workstation™ is a trademark of Sun Microsystems
Synopsys® is a registered trademark and Design Compiler™ is a trademark of Synopsys, Inc.
ViewLogic®, Workview®, ViewDraw®, and ViewSim® are registered trademarks of ViewLogic Systems, Inc.

Life Support Policy

QuickLogic's products are not authorized for use as critical components in life support devices or systems unless a specific written agreement pertaining to such intended use is executed between the manufacturer and the President of QuickLogic.

1. Life support devices or systems are devices which (a) are intended for surgical implant into the body or (b) support or sustain life and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or affect its safety or effectiveness.

QuickLogic does not assume any liability resulting from the application or use of any product described within. No circuit patent licenses are implied. QuickLogic Corporation reserves the right to make changes without notice to any product herein to improve reliability, function, or design.

Products mentioned in this databook are covered by one or more of the following US patents:
5,196,724 ; 5,122,685 ; 5,220,213 ; 5,243,226 ; 5,280,202 ; 5,362,676 ; 5,319,238 ; 5,302,546 ; 5,327,024 ; 5,293,133 ; 5,294,846
Additional patents pending.

QUICKLOGIC

QuickLogic Corporation
2933 Bunker Hill Lane
Santa Clara, CA 95054
Tel: 408-987-2000
Fax: 408-987-2012

QUICKLOGIC

Welcome to the 1995 edition of the QuickLogic Data Book.

New in this edition are:

1. 8,000 gate *Wild*Cat Series FPGAs featuring fully PCI 2.0 compliant input/output capability.
2. "L" series Low Power FPGA devices offering 3.3 Volt supply operation.
3. "-X" speed grade FPGAs for cost sensitive applications.

Also new is the 5.0 release of our SpDE design tools and the new Quick*Works* toolset which includes:

- Verilog® / VHDL Synthesis
- Context Sensitive HDL Editing
- Color Coded HDL Template Expansion
- Mixed-Mode Design Entry
- Enhanced Schematic Capture
- Comprehensive Verilog Simulation
- Interactive Cross-Probing Between Tools
- Improved Logic Optimization

New information includes a Thermal Management section and upgrades to the Power vs Operating Frequency and Packagings sections.

QuickLogic's technology, architecture and design tools are revolutionizing the FPGA industry by solving the problems of traditional FPGAs.

For a QUICK response fax us at (408) 987-2012 or call us at 1-800-842-FPGA (3742) or email us at info@qlogic.com for more information on QuickLogic products.

Thank you for your interest in QuickLogic and its products.

Table of Contents

General Information

FPGA Data Sheets

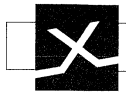
CAE Design Tools

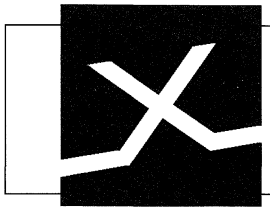
Application Notes

Quality, Reliability, and Packaging Information

Papers and Article Reprints

Sales Representatives and Distributors





Quick Reference Product Guide

pASIC DEVELOPMENT TOOLS

PART #	PRODUCT NAME	DESCRIPTION
PC		
QT-QWK-50-PC-A	QuickWorks Toolkit	Complete design kit for Windows-PC with programmer
QT-QTL-50-PC-A	QuickTools Toolkit	Includes QS-SPDE-50-PC, QS-UTIL-50-PC and programmr.
QT-SPDE-50-PC-A	SpDE Toolkit	Place & route with programmr. for users with third party tools
QS-QWK-50-PC	QuickWorks Software	Complete software package for Windows-PC
QS-QTL-50-PC	QuickTools Software	Includes QS-SPDE-50-PC & QS-UTIL-50-PC
QS-SPDE-50-PC	SpDE Software	Place & route for users with third party tools
QS-UTIL-50-PC	Software Utilities Package	3rd Party Simulation Interfaces, Logic Re-Optimizer
QT-DP-50-PC-A	Designer Programmer	Up to eight programmers may be ganged off one computer
QS-QWK-50-PC-EV	"Check-out Your Design" Eval Kit	Complete a full design! Includes 30 day license & support
QS-VL-PC	Viewlogic Technology Libraries	For Workview Pro; Plus, supports Draw, Sim & Synthesis
QS-INTG-PC	Intergraph Technology Libraries	For Intergraph design tools
SUN		
QS-QTL-50-SUN	QuickTools Software	Includes QS-SPDE-50-SUN & QS-UTIL-50-SUN
QS-SPDE-50-SUN	SpDE Software	Place and route software for SUN SparcStation
QS-UTIL-50-SUN	Software Utilities Package	3rd Party Simulation Interfaces, Logic Re-Optimizer
QS-VL-SUN	Viewlogic Technology Libraries	Support for Powerview including ViewSynthesis
QS-SYN-SUN	Synopsys Technology Libraries	Support for Design Compiler with VHDL and Verilog
QS-MEN-SUN	Mentor Technology Libraries	Support for Mentor Schematic Capture and Simulator
QS-CNC-SUN	Cadence Technology Libraries	Support for Concept schematic package
HP		
QS-QTL-50-HP	QuickTools Software	Includes QS-SPDE-50-HP & QS-UTIL-50-HP
QS-SPDE-50-HP	SpDE Software	Place and route software for HP Workstation
QS-UTIL-50-HP	Software Utilities Package	3rd Party Simulation Interfaces, Logic Re-Optimizer
QS-MEN-HP	Mentor Technology Libraries	Support for Mentor Schematic Capture and Simulator

Notes:
(1) All QT products include designer programmer unit, and (2) The "-A" suffix = 110V power, "-B" = 220V, "-C" = 240V(U.K).
(2) -50 version numbers subject to change as products are upgraded.
(3) Software listed above without a version number may have been revised. Please contact sales representative before ordering.

FPGA DEVICES

PART #	USABLE GATES	LOGIC CELLS	I/O PINS	HIGH-DRIVE/ CLOCK PINS	LOGIC CELL DELAY	CLOCK SKEW	PACKAGES
QL8x12	1000	96	56	8	< 2.0 nsec	< 0.5 nsec	44PLCC
							68PLCC
							68PGA
							100TQFP
QL12x16	2000	192	80	8	< 2.0 nsec	< 0.5 nsec	68PLCC
							84PLCC
							84CPGA
							100TQFP
							100CQFP
QL16x24	4000	384	114	8	< 2.0 nsec	< 0.5 nsec	84PLCC
							100TQFP
							100CQFP
							144TQFP
							144CPGA
							160CQFP
QL24x32	8000	768	180	8	< 2.0 nsec	< 0.5 nsec	144TQFP
							208PQFP

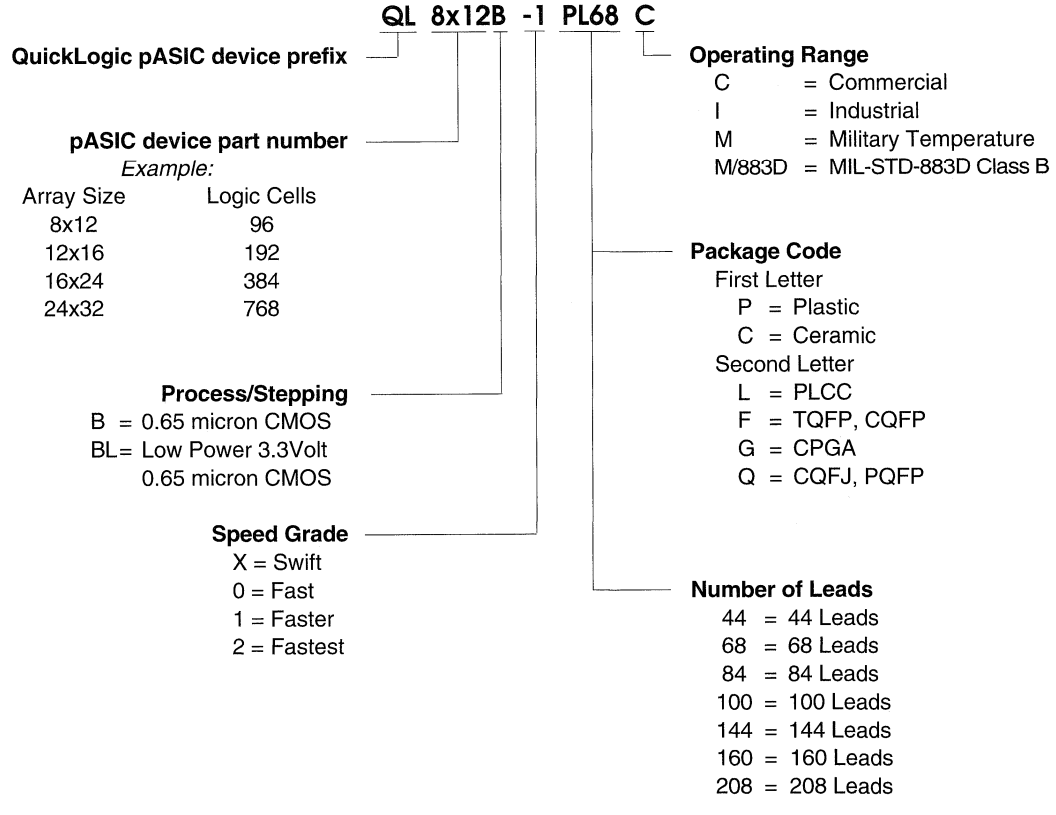
PROGRAMMING ADAPTERS

PART #	DEVICES SUPPORTED	DESCRIPTION
QP-PL44	QL8X12	Adapts 44PLCC To Program/Tester Unit
QP-PL68	QL8X12,QL12x16	Adapts 68PLCC To Program/Tester Unit
QP-CG68	QL8X12	Adapts 68CPGA To Program/Tester Unit
QP-CG84	QL12X16	Adapts 84CPGA To Program/Tester Unit
QP-PF100	QL8X12, QL12X16	Adapts 100TQFP To Program/Tester Unit
QP-CF100	QL12X16	Adapts 100CQFP To Program/Tester Unit
QP-PL4084	QL16X24	Adapts 84PLCC To Program/Tester Unit
QP-PF4100	QL16X24	Adapts 100TQFP To Program/Tester Unit
QP-CF4100	QL16X24	Adapts 100CQFP To Program/Tester Unit
QP-PF4144	QL16X24	Adapts 144TQFP To Program/Tester Unit
QP-CG4144	QL16X24	Adapts 144CPGA To Program/Tester Unit
QP-CF4160	QL16X24	Adapts 160CQFP To Program/Tester Unit
QP-PF8144	QL24X32	Adapts 144TQFP To Program/Tester Unit
QP-PQ8208	QL24X32	Adapts 208PQFP To Program/Tester Unit

pASIC 1 FAMILY
Ordering Information

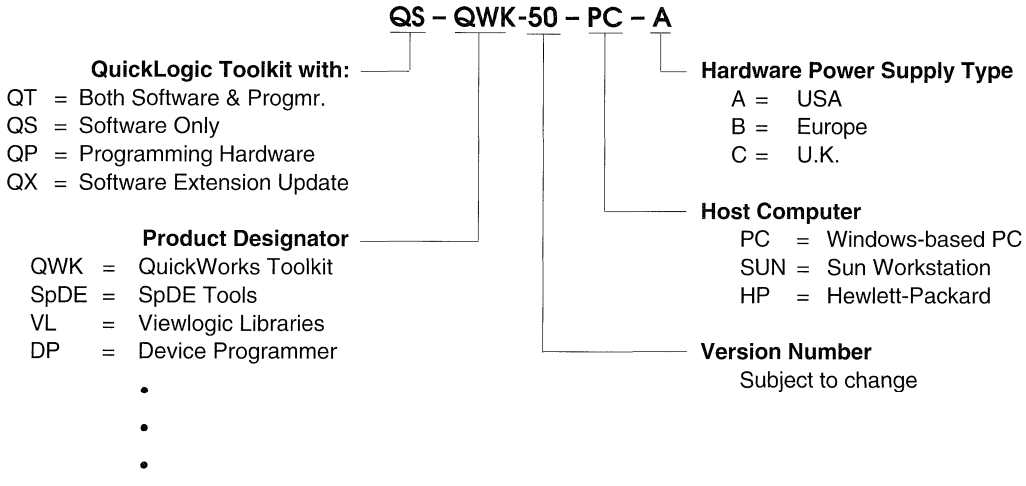
ORDERING INFORMATION

pASIC device ordering part numbers are composed as follows:



QUICKLOGIC

pASIC development tool ordering part numbers are composed as follows:



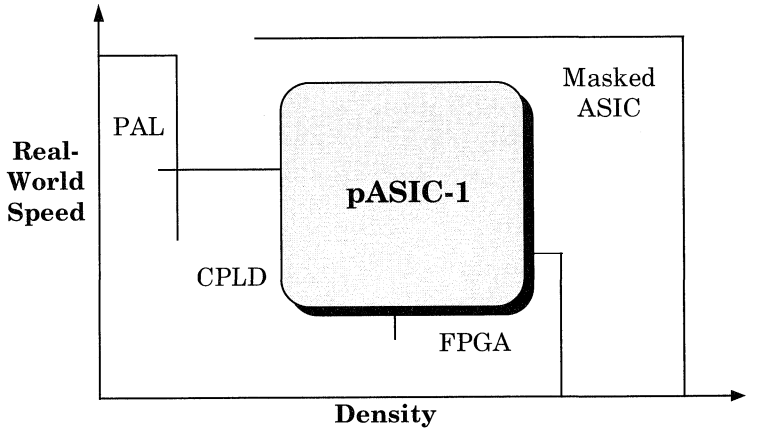
Very-High-Speed FPGAs Through Technology, Architecture, and Tools

QuickLogic Corporation provides very-high-speed programmable ASIC solutions for designers of high-performance systems who must get their products to market quickly.

The company was founded by the engineers who invented the PAL device and PALASM software. By offering the ability to create high-speed custom logic circuits with standard off-the-shelf products, their invention revolutionized the world of logic design. First-generation FPGAs extended these benefits to higher levels of density, but at much lower speeds. QuickLogic was formed to create a single solution combining the fast speed of PAL devices with the high density, low power and logic flexibility of FPGAs.

FIGURE 1 Comparative Speed/Density Chart

...FPGAs operating two to three times faster



QuickLogic FPGAs operate at two to three times the usable system speed of first-generation devices.

To achieve these goals, QuickLogic has created a CMOS antifuse technology, called the ViaLink™ element, which couples small size with high speed. It results in FPGAs operating two to three times faster than other technologies using the same process lithography. The small size also provides a technology migration path to 50,000 gates and beyond.

QUICKLOGIC

...can be used in applications with 33, 66 and 80 MHz microprocessors

QuickLogic's new *Wild*Cat series of devices in the pASIC 1™ Family of Very-High-Speed CMOS FPGAs, delivers new levels of speed and density in an easy-to-use product. High-density programmable devices can now be used in applications with 33, 66 and 80 MHz microprocessors and in designs with useful internal logic functions operating at over 150 MHz.

QuickLogic engineers achieve industry-leading performance by addressing the design task through the three aspects of Technology, Architecture, and Tools. In each area they combine a variety of techniques to yield the optimum solution.

Speed Through Technology

- ViaLink direct metal-to-metal antifuse
- Speed-critical links less than 50 ohms
- Standard high-speed 0.65 micron CMOS logic process

These features yield both the smallest physical programming element size and *fastest interconnect* speed of all programmable technologies.

Speed Through Architecture

- Up to 14-input wide gates
- Highly tuned, dedicated register in every logic cell
- Plentiful regular and orthogonal interconnect wiring resources

These features provide an architecture optimized for a wide range of high-speed control, data path and general-purpose logic integration applications.

Speed Through Tools

...100% automatic place and route

- Architecture-optimized place and route tools
- Precise timing simulation using actual wire lengths, fanout, and loading
- Rapid design iterations permit design optimization

Users achieve high speed and fast design implementation through 100% automatic place and route tools even on functions using up to 100% of the available logic cells.

SPEED THROUGH TECHNOLOGY

At the heart of every programmable device is an electronic switch for user configuration of on-chip logic functions. It may be an active device, such as an EPROM cell or an SRAM bit. Or a passive fuse or an antifuse element.

Two basic electrical characteristics limit the maximum useful operating speed which can be achieved with each approach.

- The capacitance, C, of the programmable element in the OFF (OPEN) state. This determines the capacitive loading effect of unprogrammed elements on metal interconnect wires.
- The resistance, R, of the programmable element in the ON (CLOSED) state. This determines the series resistance of the programmed element when interconnecting wires and logic functions.

The technology with the lowest values of R and C delivers the fastest raw speed to the circuit designer.

Other CMOS programmable ASIC devices using EPROM and SRAM programming elements typically have high resistance values, above 1000 ohms, plus large physical size. Dielectric antifuses improve on both these factors but still cannot yield the performance demanded by today's systems.

ViaLink Antifuses

QuickLogic created the ViaLink antifuse to provide a low-resistance, low-capacitance programmable connection directly from one metal layer to another. The ViaLink element is formed by depositing a very high resistance layer of programmable silicon into a via between the two metal layers of a standard high-volume CMOS gate array process.

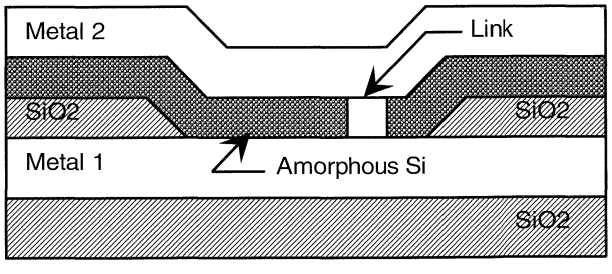


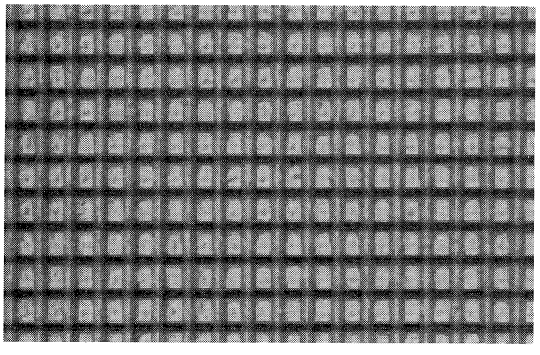
FIGURE 2 ViaLink Cross Section

The ViaLink element provides a direct metal-to-metal connection.

A programming voltage is applied to selected vias in an algorithm of pulses. This creates a direct metal-to-metal link by permanently converting the silicon to a low resistance state. Typical resistance is less than 50 ohms.

In a 0.65 micron process, the size of a ViaLink via is approximately one micron square, orders of magnitude smaller than active elements. This, coupled with the high dielectric constant of the via material, ensures that unprogrammed ViaLink devices exhibit low capacitive loading (less than 1fF). As the size of the programmed link is physically much smaller than the via, the technology can be scaled below 0.5 microns for future very-high-density applications.

FIGURE 3 Photomicrograph of an Array of ViaLink Elements



A ViaLink antifuse is located at the intersection of every horizontal and vertical wire. Array density is limited by the process lithography, not by the programmable element size.

EPLD AND FPGA TECHNOLOGY COMPARISONS

Programmable Element	SRAM	E/EEPROM	Dielectric Antifuse	ViaLink Antifuse
Typical ON Resistance R	~1000 ohm	~1000 ohm	~500 ohm	~50 ohm
Typical OFF Capacitance C	~50 fF	~15 fF	~5 fF	~1 fF
Physical Size	Very Large	Large	Medium	Small

The ViaLink element delivers the lowest ON resistance, lowest OFF capacitance and smallest physical size of any programmable technology. These characteristics open up a new spectrum of high-speed, high-density applications to programmable devices.

ViaLink Reliability

ViaLink elements have been subjected to accelerated stress testing in both programmed and unprogrammed states. Test results indicate that the ViaLink element has no measurable impact on the reliability of the underlying CMOS process. Product reliability levels in the system environment compare favorably with gate arrays and other FPGAs.

SPEED THROUGH ARCHITECTURE

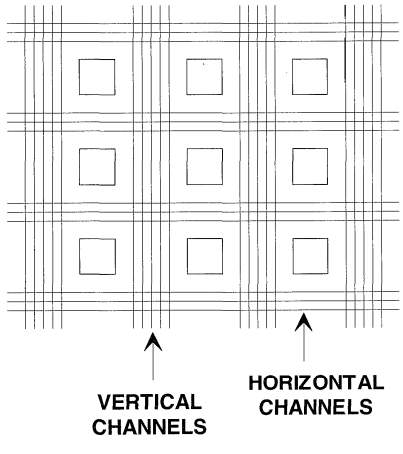
A wealth of interconnect resources, a performance-oriented logic cell, and optimized I/O circuitry maximize the fundamental speed advantage of the ViaLink technology.

Fast, Yet Regular Routing

Many FPGAs employ a variety of wiring types to limit the number of high-resistance connections in a net. For example, signals can be routed through a variety of short lines, long lines, switch boxes and other structures. These irregular routing resources cause unpredictable delay behavior and limit logic utilization.

The low impedance of the ViaLink antifuse allows a regular and orthogonal architecture. The pASIC structure features a matrix of logic cells interconnected by vertical and horizontal routing channels. A ViaLink element located at every wire intersection allows direct, metal-to-metal links between signal lines.

FIGURE 4
pASIC Architecture



A matrix of speed-optimized logic cells is set in a grid of vertical and horizontal wiring channels which can be selectively connected with metal-to-metal links.

According to users (*Computer Design*, 12/91, pg. 78), QuickLogic FPGAs are the closest to masked gate arrays both in predictability and in absolute levels of performance.

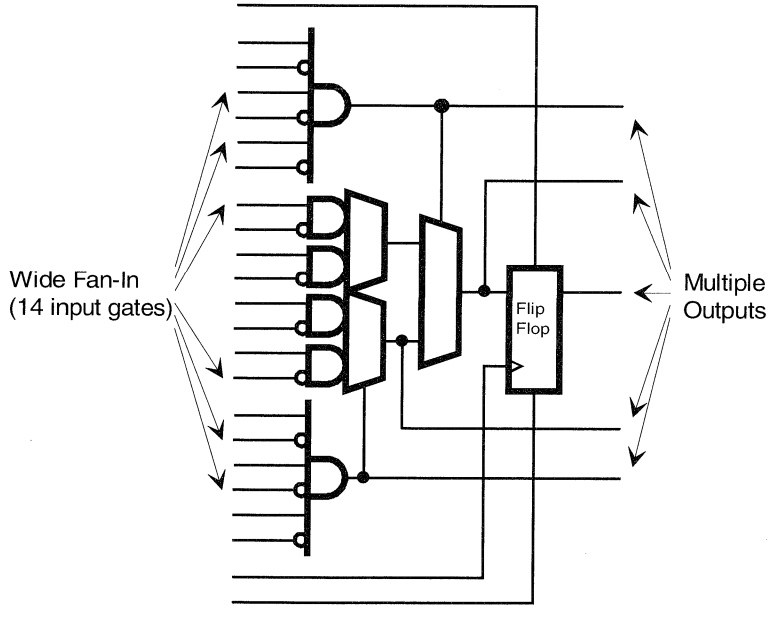
- Delays are "predictable no matter what the layout is."
- "Wire delays are shorter than logic delays."
- As net lengths increase, delays increase proportionally.
- Small design changes result only in small changes in speed.

All these characteristics allow rapid iteration towards the fastest possible solution.

High-Speed Logic Cell

The pASIC logic cell is optimized for high-speed applications. Gates up to 14 inputs wide, multiplexers, decoders and sum-of-products functions can be implemented in a single cell delay with worst case commercial t_{PD} of under 2.5 ns. Many logic functions therefore incur just half the propagation delay of traditional, narrow fan-in, FPGA logic cell structures.

FIGURE 5 pASIC 1 Logic Cell



The wide fan-in and multiple outputs of the pASIC logic cell permit high-speed logic functions and efficient use of resources.

...24-bit loadable counters operate at over 125 MHz

Each cell includes a dedicated, highly-tuned register element. Simple Johnson counters run at 214 MHz. 24-bit loadable counters, capable of doing useful work, require one logic cell delay and operate at over 150 MHz.

Fast sum-of-product state machines require just two cell delays. Complex arithmetic functions using up to three levels of logic run faster than 90 MHz.

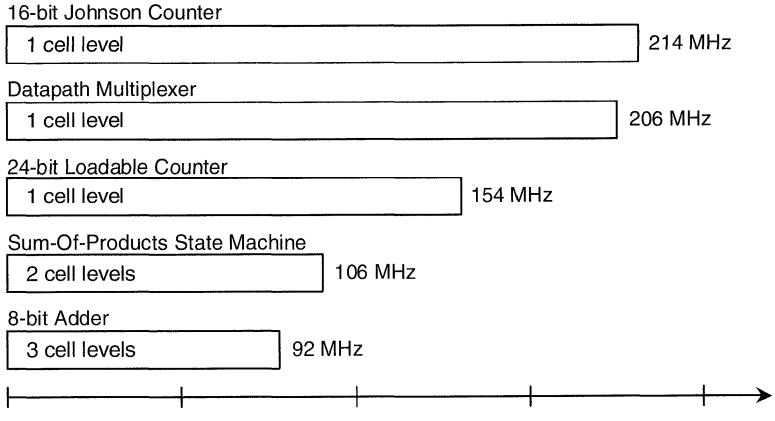


FIGURE 6 Real World Operating Speeds of 50 to 125 MHz

Typical examples of speed versus logic function complexity.

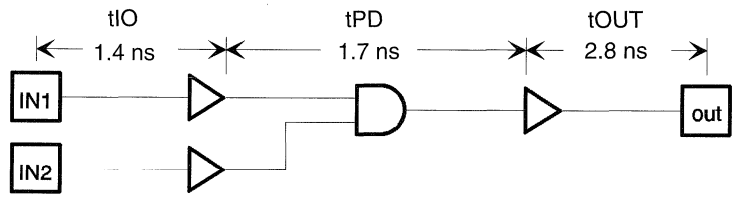
Fast I/O

These impressive on-chip specs are matched by fast I/O performance. Worst case commercial input pad to output pad delay through a 14-input AND gate, or a 4-input MUX, for example, is less than 10 ns.

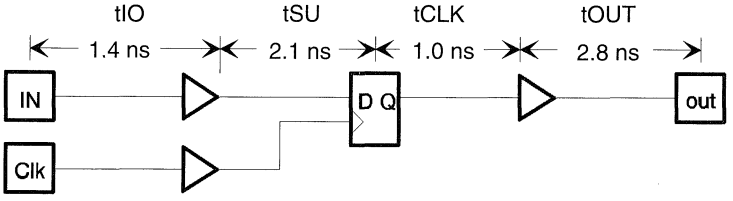
Fast I/O circuits allow logic functions in separate packages to operate together at over 75 MHz. This speed is achieved while preserving low output switching noise levels. Switching of up to 48 outputs simultaneously has been demonstrated with less than 1 volt of ground bounce.

FIGURE 7
Fast I/O Pads

QL16x24B Nominal I/O Delays



Input Delay + Combinatorial Delay + Output Delay = 5.9 ns

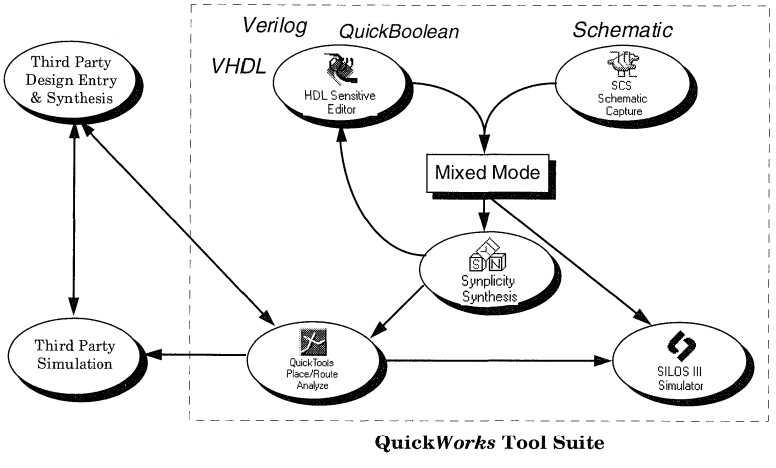


Input Delay + Reg Setup + Clock to Output + Output Delay = 7.3 ns

SPEED THROUGH TOOLS

Fast device operation and rapid design execution is achieved by combining the best third-party CAE tools with device-specific place and route and delay modeling software.

FIGURE 8
The pASIC Toolkit



QuickLogic tools communicate with third-party software through an object-oriented design database.

Design Entry: Third Party & QL Integrated

Designs can be created using general-purpose schematic, language entry and simulation packages on PC, Sun and HP workstation platforms. These tools interface to the QuickLogic Seamless pASIC Design Environment (SpDE — pronounced "Speedy") for optimization place and route, delay modcling, and other tasks best served by architecture specific code.

VHDL, Verilog, state machine, and Boolean (ABEL, CUPL, MINC, PALASM, etc.) language entry are available through several tool and synthesis package vendors (Data I/O, Viewlogic, Synopsys, Exemplar, Cadence, Intergraph).

Quick*Works*

A low-cost, fully integrated design solution operating under Microsoft Windows on a PC is all that is needed to complete a design. The Quick*Works* tookit includes Verilog/VHDL Synthesis, Context Sensitive HDL Editing, Color Coded HDL Template Expansion, Mixed-Mode Design Entry, Enhanced Schematic Capture, Comprehensive Verilog Simulation plus QuickLogic's SpDE software. The package can also include a device programmer.

Optimized Place and Route

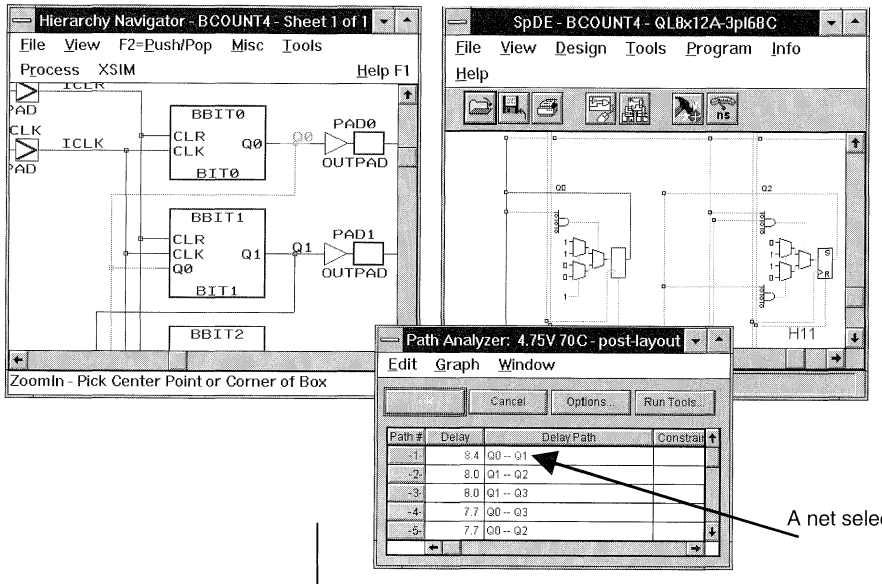
The regular gate array-like architecture, combined with abundant routing resources, enables 100% automatic placement and routing with high utilization. Users report completing designs using 100% of the logic cells, even with a large number of fixed pin locations.

A physical viewer shows how the automatic tools fit logic functions into the silicon resources, right down to specific routing wires utilized.

A path analyzer provides timing for selected interconnect nets in the device without having to run the simulator. It also allows entry of specific timing requirements, by path, for SpDE to use in timing-driven placement to meet your timing requirements.

Cross probing between applications allows for easy analysis of designs and fast debugging of problems. Click on a net in the schematic and it is instantly highlighted in the physical layout. Or vice versa. Similar interactive links exist to the simulator, and path analyzer.

FIGURE 9
Windows
Screen Shot



A net selected in the path analyzer is instantly highlighted in the schematic and the physical viewer, and vice versa.

Precise Timing

Asymptotic Waveform Evaluation, AWE, techniques generate precise timing data, using actual circuit wire lengths, fanout and loading. This enhances speed by providing accurate delay models for back annotation into the simulator. It also identifies the most heavily loaded nets for selective programming to less than 50 ohms.

Test Vectors and Programming

Scan path circuitry built into the pASIC flip-flops allows automatic generation of test vectors for post-programming functional testing.

Programmers attach to the PC via the RS-232 port. No special-purpose programming cards occupy valuable expansion slots.

pASIC 1 FAMILY PRODUCTS

Family Features

The pASIC 1 Family serves general-purpose, high-speed logic integration tasks in arithmetic, control, data path and RISC and CISC microprocessor support applications. Four basic devices cover a broad range of speed, package, density, and I/O options.

Device part numbers describe the organization of logic cells. For example, the QL8x12B features 96 logic cells in an eight by twelve matrix. A fully utilized QL8x12B accommodates 800 to 1200 gates, for an average of ten gates per cell. QuickLogic describes this as a 1000 usable-gate FPGA. It is equivalent in capacity to many, so called "3000 to 4000 gate" EPLD and LCA devices.

As each logic cell contains a dedicated register function plus combinatorial logic sufficient to create two latches, the QL8x12B can contain up to 288 storage elements.

Part Number	Logic Cells	Storage Elements	Max I/O Cells	Dedicated Inputs	Package Pins	Usable Gates	CPLD/LCA Gates
QL8x12B	96	288	56	8	44, 68, 100	1000	3000
QL12x16B	192	576	80	8	68, 84, 100	2000	6000
QL16x24B	384	1152	114	8	84, 100, 144, 160	4000	12000
QL24X32B	768	2304	172	8	144, 208	8000	24000

Product Features

PLCC, TQFP, CPGA, CQFP and other package styles are available. Multiple speed selections for each device allow the user to trade-off performance and cost.

For worst-case commercial:

- Input buffer plus logic cell plus output buffer delay of under 10 ns.
- Multiple-chip operating frequencies over 75 MHz.
- Logic function delays, up to two levels deep, in under 2.5 ns.
- Useful counter speeds up to 125 MHz.
- Low output switching noise. Less than 1 volt of ground bounce with 48 outputs switching simultaneously.
- Worst-case clock skew of less than 1 ns.
- Input hysteresis provides high reliability in noisy operating environments.
- Low CMOS power consumption: typically 2 mA standby current.

Future pASIC Families

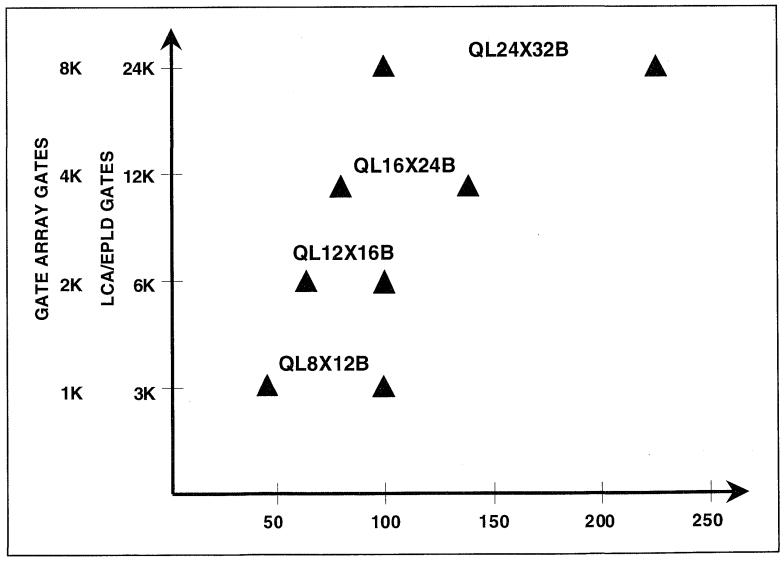
Future pASIC products will use the small size of the ViaLink element to offer FPGA families with densities and I/O counts comparable to today's masked gate arrays. ViaLink technology will also be used to extend the flexibility and ease of use of user-configurable logic to new programmable ASIC architectures.

pASIC 1 FAMILY
ViaLink Technology
Very-High-Speed CMOS FPGAs

FAMILY HIGHLIGHTS

- **Very High Speed** – ViaLink metal-to-metal, programmable-via anti-fuse technology ensures useful internal logic function speeds at over 100 MHz, and logic cell delays of under 2 ns.
- **High Usable Density** – Up to 8,000 "gate array" gates, equivalent to 24,000 EPLD or LCA Gates.
- **Low Power/ 3.3 Volt Supply**– Stand-by current typically 2 mAwith 5.0 Volt supply and less than 1 mA with 3.3 Volt supply.
- **Flexible FPGA Architecture** – The pASIC 1 logic cell supports efficient, high-speed arithmetic, counter, data path, state machine and random logic applications with up to 14-input wide gates.
- **Low-Cost, Easy-to-Use Design Tools** – Designs entered and simulated using third-party CAE tools. Fast, fully automatic place and route on PC and workstation platforms.

FIGURE 1
pASIC 1 Family



QUICKLOGIC

FAMILY SUMMARY

The pASIC 1 Family of very-high-speed CMOS user-programmable ASIC (pASIC) devices is based on the first FPGA technology to combine high speed, high density and low power in a single architecture.

All pASIC 1 Family devices are based on an array of highly flexible logic cells which have been optimized for efficient implementation of high-speed arithmetic, counter, data path, state machine, random and glue logic functions. Logic cells are configured and interconnected by rows and columns of routing metal and ViaLink metal-to-metal programmable-via interconnect elements.

ViaLink technology provides a nonvolatile, permanently programmed custom logic function capable of operating at counter speeds of over 150 MHz. Internal logic cell nominal worst case delays are under 2 ns and total input to output combinatorial logic delays are under 8 ns. This permits high-density programmable devices to be used with today's fastest CISC and RISC microprocessors, while consuming a fraction of the power and board area of PAL, GAL and discrete logic solutions.

pASIC 1 Family devices range in density from 1000 "gate array" gates (3,000 EPLD/LCA gates) in a 44-pin package, to 8,000 (24,000) gates in high-pin-count packages. All devices share a common architecture and CAE design software to allow easy transfer of designs from one product to another. The small size of the ViaLink programming element ensures a technology migration path to devices of 20,000 gates and above.

Designs are entered into the pASIC 1 Family of devices on PC or workstation platforms using third-party, general-purpose design-entry and simulation CAE packages, together with QuickLogic device-specific place and route and programming software, called SpDE tools. Sufficient on-chip routing channels are provided to allow fully automatic place and route of designs using up to 100% of the available logic cells.

All the necessary hardware and software, required to complete a design, from entering a schematic to programming a device are included in pASIC Toolkits available from QuickLogic. The Quick*Works* Toolkit includes Verilog®/ VHDL Synthesis, Context Sensitive HDL Editing, Color Coded HDL Template Expansion, Mixed-Mode Design Entry, Enhanced Schematic Capture, Comprehensive Verilog Simulation plus QuickLogic's SpDE software. All applications run on the PC under the Microsoft Windows 3.1 & NT graphical user interface to ensure a highly productive and easy-to-use design environment. An open interface (QDIF) allows many other third-party tools (Exemplar, Viewlogic, etc.) to be used with QuickLogic software on both PC and workstation platforms.

VIALINK PROGRAMMING ELEMENT

Programmable devices implement customer-defined logic functions by interconnecting user-configurable logic cells through a variety of semiconductor switching elements. The maximum speed of operation is determined by the effective impedance of the switch in both programmed, ON, and unprogrammed, OFF, states.

In pASIC 1 devices the switch is called a ViaLink element. The ViaLink element is an antifuse formed in a via between the two layers of metal of a standard CMOS gate array process. The direct metal-to-metal link created as a result of programming achieves a connection with resistance values below 50 ohms. This is less than 5 percent of the resistance of an EPROM or SRAM switch and 10 percent of that of a dielectric antifuse. The capacitance of an unprogrammed ViaLink site is also lower than these alternative approaches. The resulting low RC time constant provides speeds two to three times faster than older generation technologies.

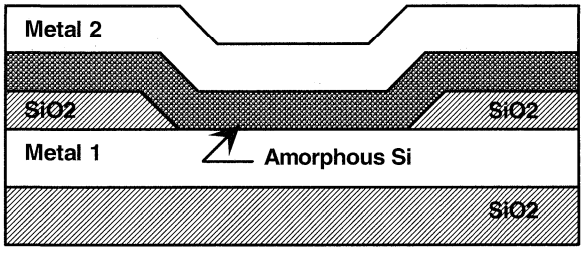


FIGURE 2a Unprogrammed ViaLink Element

Figure 2a shows an unprogrammed ViaLink site. In a custom metal masked ASIC, such as a gate array, the top and bottom layers of metal make direct contact through the via. In a ViaLink programmable ASIC device the two layers of metal are initially separated by an insulating silicon layer with resistance in excess of 1 gigaohm.

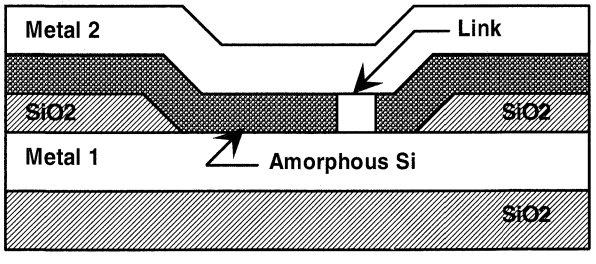
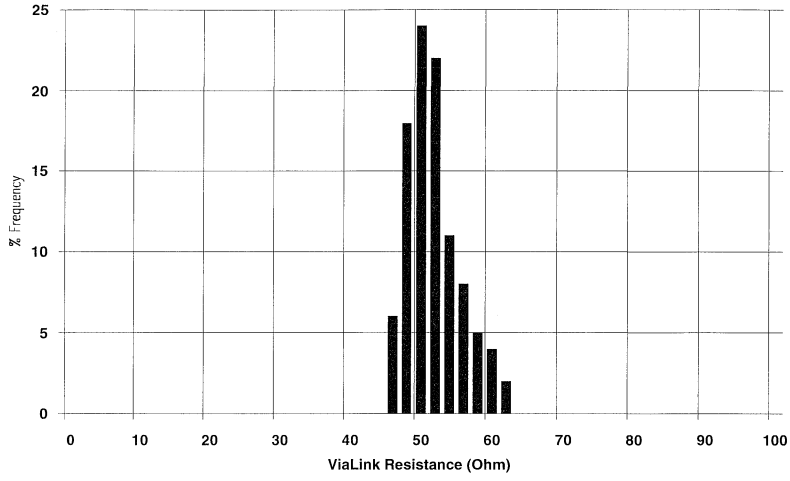


FIGURE 2b Programmed ViaLink Element

A programming voltage pulse applied across the via forms a bidirectional conductive link connecting the top and bottom metal layers, Figure 2b. The tight distribution of link resistance is shown in Figure 3.

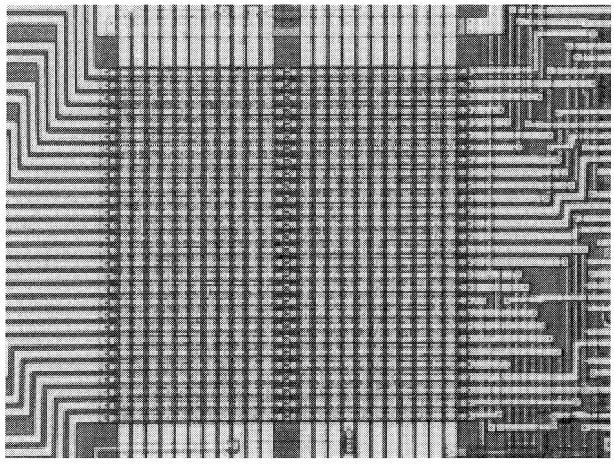
FIGURE 3
Distribution of Programmed Link Resistance



STANDARD CMOS PROCESS

QuickLogic pASIC 1 devices are the first FPGA devices to be fabricated on a conventional high-volume CMOS gate array process. The base technology is a 0.65 micron, n-well CMOS technology with a single polysilicon layer and two layers of metal interconnect. The only deviation from the standard process flow occurs when the ViaLink module is inserted between the metal deposition steps.

FIGURE 4
An Array of ViaLink Elements



As the size of a ViaLink via is identical to that of a standard metal interconnect via, programmable elements can be packed very densely. The microphotograph of Figure 4 shows an array of ViaLink elements. The density is limited only by the minimum dimensions of the metal-line to metal-line pitch. Migration of processes to submicron geometries will allow the development of pASIC 1 devices with tens of thousands of usable gates.

pASIC FAMILY ARCHITECTURE

The pASIC 1 device architecture consists of an array of user-configurable logic building blocks, called logic cells, set in a grid of metal wiring channels similar to those of a gate array. Figure 5 shows a section of a pASIC 1 device containing internal logic cells, input/output cells and dual-layer vertical and horizontal metal routing channels. Through ViaLink elements located at the wire intersections, the output of any cell may be programmed to connect to the input of any other cell.

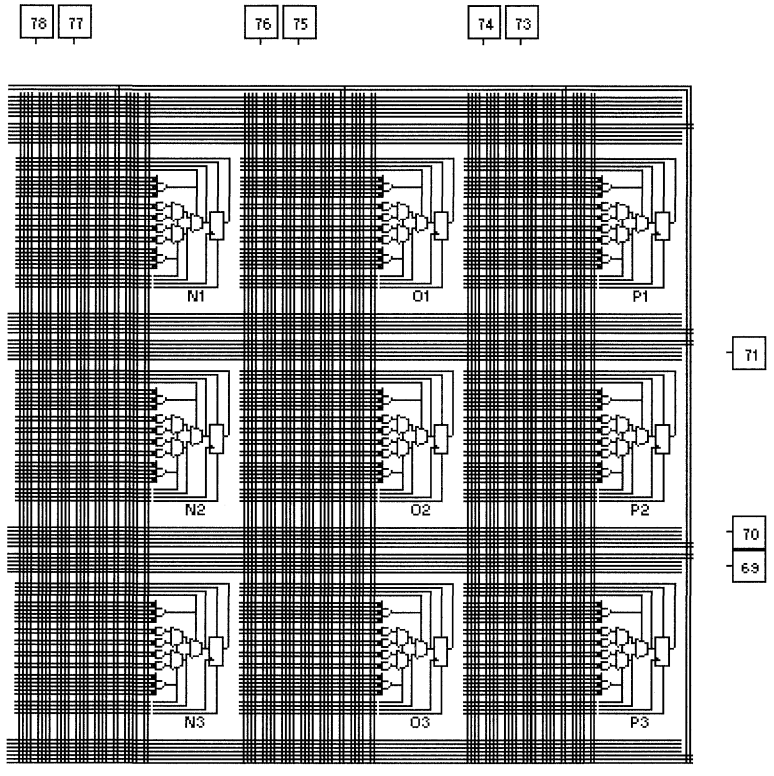


FIGURE 5
A Matrix of Logic Cells and Wiring Channels

This regular and orthogonal interconnect makes the pASIC 1 architecture similar in structure and performance to a metal masked gate array. It also makes system operating speed far less sensitive to partitioning and placement decisions, as minor revisions to a logic design result only in small changes in performance.

Adequate wiring resources permit 100% automatic placement and routing of designs using up to 100% of the logic cells. This has been demonstrated on designs that include a high percentage of fixed pin placements.

ORGANIZATION

The pASIC 1 Family of very-high-speed FPGAs contains devices covering a wide spectrum of I/O and density requirements. The four members range from 1,000 gates in a 44-lead package to 8000 gates in a 208-lead package and are shown in Figure 6.

Device part numbers are derived from the organization of internal logic cells. For example, in Figure 6, the QL8x12B contains 96 logic cells in an 8-by-12 matrix. The single lines between logic cells represent channels containing up to twenty-two wires. Each of the internal logic cells has the logic capacity of up to 30 "gate array gates." As a typical application will use 10 to 12 gates from each logic cell, the QL8x12B is described as a 1000-usable-gate device. Based on the "available gate" gate counting approach of some programmable logic vendors, it would be called a 3000-gate part. The QL8x12B is available in a 68-lead package with 56 bidirectional I/O pins and 8 dedicated input/high drive clock pins and a 44-lead package with 32 I/O pins.

The key features of these four pASIC 1 devices are as follows:

Device Cells	Logic Cells	Max I/O Inputs	Dedicated Pins	Package Pins	Usable Gates	EPLD/LCA Gates
QL8x12B	96	64	8	44,68,100	1000	3000
QL12x16B	192	88	8	68,84,100	2000	6000
QL16x24B	384	122	8	84,100,144,160	4000	12000
QL24x32B	768	180	8	144, 208	8000	24000

See individual product data sheets for specific information on each device.

FIGURE 6a
pASIC 1 Family Members

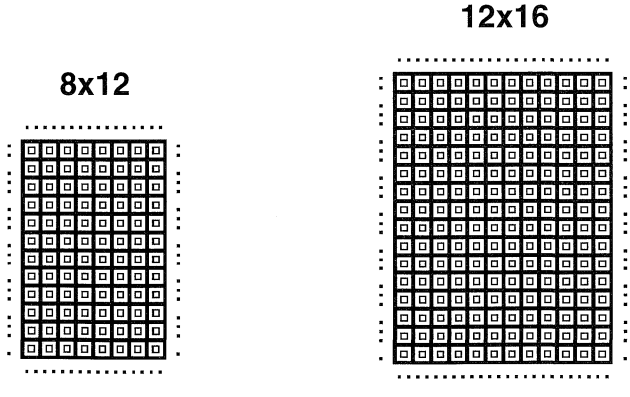
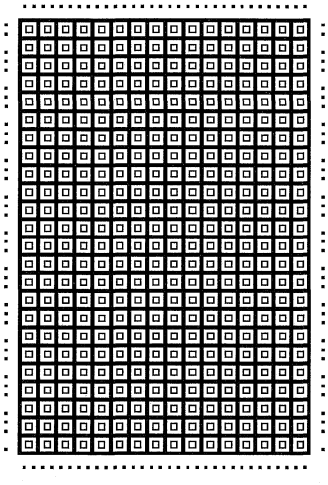


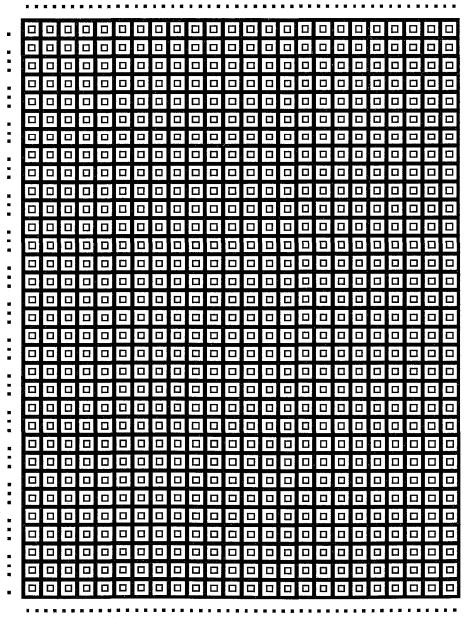
FIGURE 6b
pASIC 1 Family Members

■ = I/O / High-drive Input / Clock Cells
□ = Logic Cells
\+ = Interconnect Wiring Channels

16x24



24x32



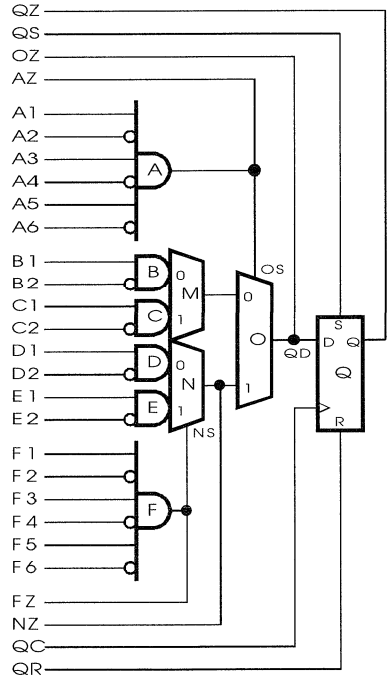
pASIC 1 INTERNAL LOGIC CELL

The pASIC 1 internal logic cell, shown in Figure 7, is a general-purpose building block that can implement most TTL and gate array macro library functions. It has been optimized to maintain the inherent speed advantage of the ViaLink technology while ensuring maximum logic flexibility.

The logic cell consists of two 6-input AND gates, four 2-input AND gates, three 2-to-1 multiplexers and a D flip-flop. In addition to the dedicated flip-flop, logic gates in each cell can be configured to provide two latches. As noted above, each cell represents approximately 30 gate-equivalents of logic capability. Multiple outputs from the logic cell allow the automatic place and route software to pack unrelated logic functions into a single cell to maximize silicon utilzation.

The pASIC 1 logic cell is unique among FPGA architectures in that it offers up to 14-input-wide gating functions. This allows many logic functions to be accomplished in a single cell delay that require two or more delays with other architectures. It can implement all possible Boolean transfer functions of up to three variables as well as many functions of up to 14 variables.

FIGURE 7
pASIC 1 Internal Logic Cell



Glitch-free switching of the multiplexer is ensured as the internal capacitance of the circuit maintains enough charge to hold the output in a steady state during input transitions. The multiplexer output feeds the D-type flip-flop which can also be configured to provide J-K, S-R, or T-type functions as well as count with carry-in. Two independent SET and RESET inputs can be used to asynchronously control the output condition. The combination of wide gating capability and a built-in sequential function makes the pASIC 1 logic cell particularly well suited to the design of high-speed state machines, shift registers, encoders, decoders, arbitration and arithmetic logic, as well as a wide variety of counters.

The function of a logic cell is determined by the logic levels applied to the inputs of the AND gates. ViaLink sites located on signal wires tied to the gate inputs perform the dual role of configuring the logic function of a cell and establishing connections between cells.

The pASIC 1 macro library contains more than 400 of the most frequently used logic functions optimized to fit the logic cell architecture. A detailed understanding of the logic cell is therefore not necessary to design successfully with pASIC 1 devices. CAE tools will automatically translate a conventional logic schematic into a device and provide excellent performance and utilization.

INPUT AND OUTPUT CELLS

Three types of input and output structures are provided on pASIC 1 devices to configure buffering functions at the external pads. They are the Bidirectional Input/Output (I/O) cell, the Dedicated Input (I) cell and the Clock Input cell (I/CLK).

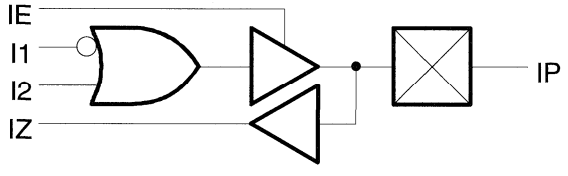
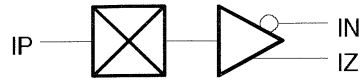


FIGURE 8 Bidirectional I/O Cell

The bidirectional I/O cell, shown in Figure 8, consists of a 2-input OR gate connected to a pin buffer driver. The buffer output is controlled by a three-state enable line to allow the pad to also act as an input. The output may be configured as active HIGH, active LOW, or as an open drain inverting buffer.

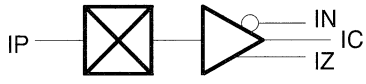
The output buffers (IOL/IOH of 8 mA) are designed to ensure quiet switching characteristics while maintaining high speed. Measured results show up to 48 outputs switching simultaneously into a 10 pF load with less than ±1 volt of output switching noise.

FIGURE10a
Dedicated Input
High-Drive Cell



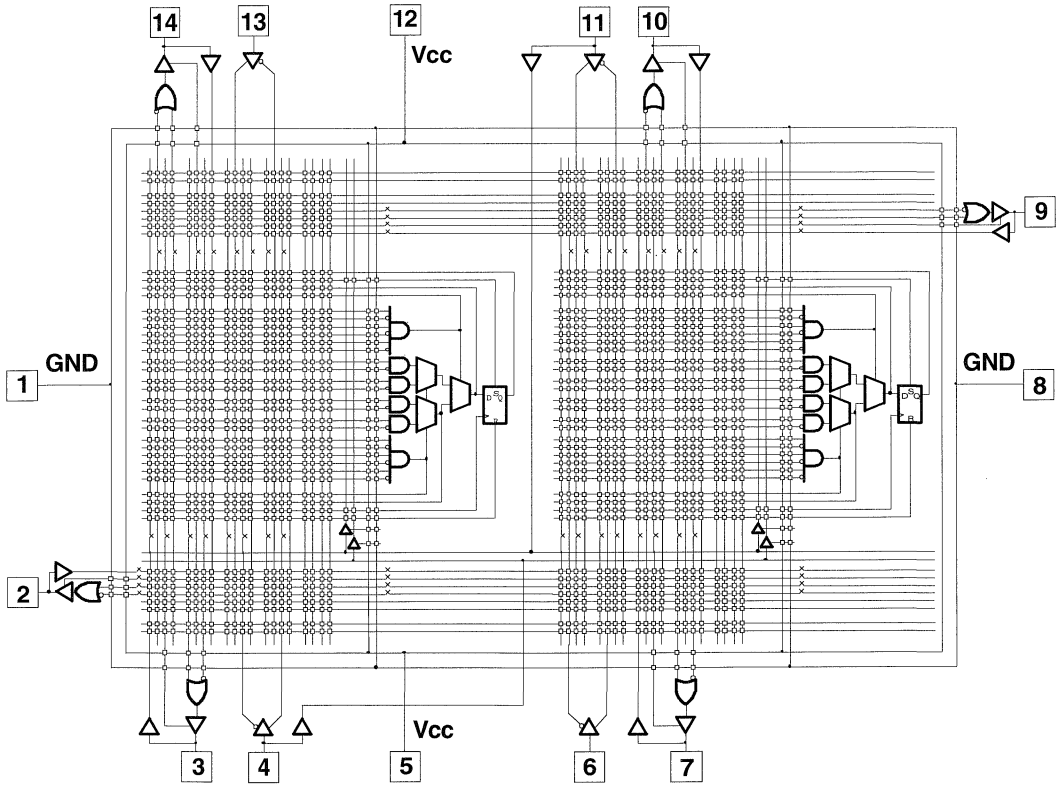
The Dedicated Input I cell, Figure 10a, conveys true and complement signals from the input pads into the array of logic cells. As these pads have nearly twice the current drive capability of the I/O pads, they are useful for distributing high fanout signals across the device. The Clock Input I/CLK cell (Figure 10b) drives a low-skew, fanout-independent clock tree that can connect to the clock, set, or reset inputs of the flip-flop. The QL12x16B device, for example, has 80 I/O cells, 6 I cells, and 2 I/CLK cells.

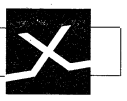
FIGURE10b
Clock Input Cell



Multiple logic cells can be programmed to form a complex logic function by interconnection through the routing channels. To describe the organization of these routing channels, a hypothetical 14-pin function consisting of two logic cells is shown in Figure 11. This device contains the same architectural features as the members of the pASIC 1 family.

FIGURE 11
pASIC 1 Device
Features





Four types of signal wires are employed; Segmented Wires, Express Wires, Clock Wires and Quad Wires (QL16x24B and QL24x32B only). Segmented wires are predominantly used for local connections and have a ViaLink element, known as a Cross Link (denoted by the open box symbol) at every crossover point. They may also be connected to the segmented wires of cells above and below through ViaLink elements, called Pass Links (denoted by the X symbol). Express lines are similar to segmented wires except that they are not divided by pass links.

Dedicated Clock wires are lightly loaded with only three links per cell to distribute high-speed clock edges to the flip-flop CLK, SET and RESET pins. Express wires may also be used to deliver clock signals into the multiplexer region of the cell for combinatorial gating. Quad wires are similar to segmented wires in that they are employed for local interconnect, but instead of having Pass Links above and below each cell, they have Pass Links every fourth logic cell. The automatic place and route software allocates signals to the appropriate wires to ensure the optimum speed/density combination.

Horizontal wiring channels, called rows, provide connections via cross links to other columns of logic cells and to the periphery of the chip. Appropriate programming of ViaLink elements allows electrical connection to be made from any logic cell output to the input of any other logic or I/O cell. Ample wires are provided in the channels to permit automatic place and route of many designs using up to 100% of the device logic cells. Designs can be completed automatically even with a high percentage of fixed user placement of internal cells and pin locations.

RELIABILITY

The pASIC 1 Family is based on a 0.65 micron high-volume CMOS fabrication process with the ViaLink programmable-via antifuse technology inserted between the metal deposition steps. Devices from this base CMOS process have been qualified to meet the requirements of MIL-STD-883D, Revision B.

The ViaLink element exists in one of two states: a highly resistive unprogrammed, OFF, state and the low impedance, conductive, ON, state. It is connected between the output of one logic cell and the inputs of other logic cells directly or through other links. No D.C. current flows through either a programmed or an unprogrammed link during operation as a logic device. An unprogrammed link sees a worst case voltage equal to VCC biased across its terminals. A programmed link carries A.C. current caused by charging and discharging of device and interconnect capacitances during switching.

Studies of test structures and complete pASIC 1 devices have shown that an unprogrammed link under VCC bias remains in the unprogrammed state over time. Similar tests on programmed links under current bias exhibit the same stability. These tests indicate that the long term reliability of the combined CMOS and ViaLink structure is similar to that of the base gate array process. For further details see the pASIC 1 Family Reliability Report.

QL8X12B
*Wild*Cat 1000
Very-High-Speed 1K (3K) Gate CMOS FPGA

Rev A

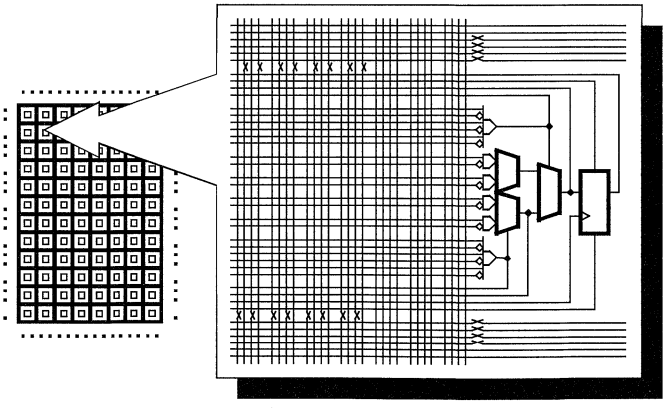
pASIC HIGHLIGHTS

...1000 usable gates, 64 I/O pins

- **Very High Speed** – ViaLink™ metal-to-metal programmable–via antifuse technology, allows counter speeds over 150 MHz and logic cell delays of under 2 ns.
- **High Usable Density** – An 8-by-12 array of 96 logic cells provides 3,000 total available gates, with 1000 typically usable "gate array" gates in 44-pin and 68-pin PLCC, 68-pin CPGA, and 100-pin TQFP packages.
- **Low-Power, High-Output Drive** – Standby current typically 2 mA. A 16-bit counter operating at 100 MHz consumes less than 50 mA. Minimum IOL of 12 mA and IOH of 8 mA
- **Low-Cost, Easy-to-Use Design Tools** – Designs entered and simulated using QuickLogic's new Quick*Works*™ development environment, or with third-party CAE tools including Viewlogic, Synopsys, Mentor, Cadence and Intergraph. Fast, fully automatic place and route on PC and workstation platforms using QuickLogic software.

QL8x12B Block Diagram

96 Logic Cells



QUICKLOGIC

▪ = Up to 56 prog. I/O cells, 6 Input high-drive cells, 2 Input/Clk (high-drive) cells

PRODUCT SUMMARY

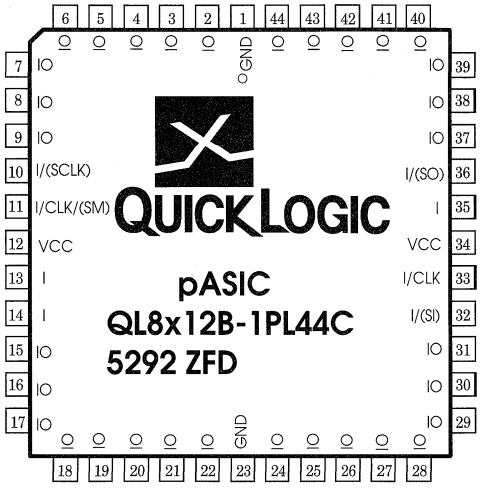
The QL8x12B is a member of the pASIC 1 Family of very-high-speed CMOS user-programmable ASIC devices. The 96 logic cell field-programmable gate array (FPGA) offers 1,000 usable "gate array" gates (equivalent to 3,000 gate claims of some vendors) of high-performance general-purpose logic in 44-pin and 68-pin PLCC packages, 68-pin CPGA, and 100-pin TQFP packages.

Low-impedance, metal-to-metal, ViaLink interconnect technology provides nonvolatile custom logic capable of operating above 150 MHz. Logic cell delays under 2 ns, combined with input delays of under 1.5 ns and output delays under 3 ns, permit high-density programmable devices to be used with today's fastest CISC and RISC microprocessors and DSPs.

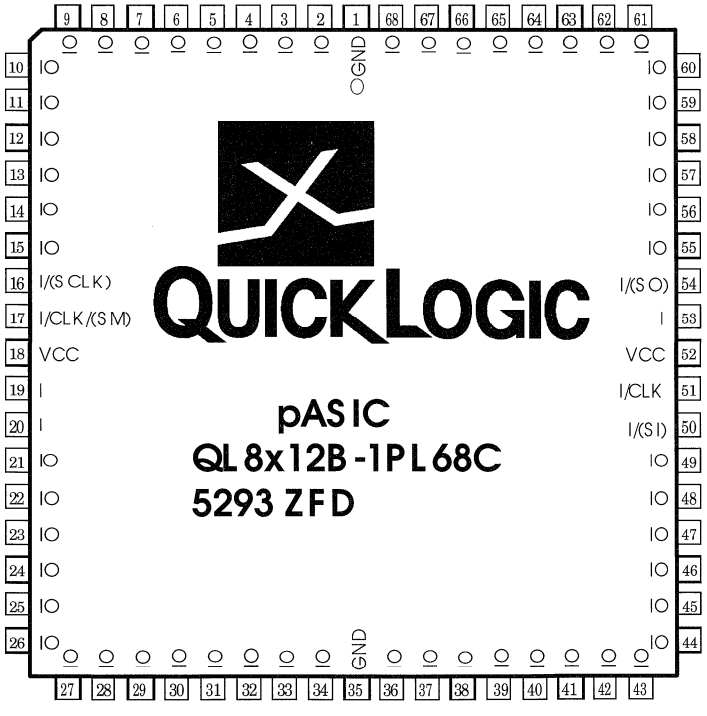
Designs are entered into the QL8x12B using a pASIC Toolkit which combines third-party design entry and simulation tools with QuickLogic's SpDE device-specific place & route and programming software. Ample on-chip routing channels are provided to allow fast, fully automatic place and route of designs using up to 100% of the logic and I/O cells, while maintaining fixed pin-outs.

FEATURES

- Total of 64 I/O pins
  - 56 Bidirectional Input/Output pins
  - 6 Dedicated Input/High-Drive pins
  - 2 Clock/Dedicated input pins with fanout-independent, low-skew clock networks
- Input + logic cell + output delays under 6 ns
- Chip-to-chip operating frequencies up to 110 MHz
- Internal state machine frequencies up to 150 MHz
- Clock skew < 0.5 ns
- Input hysteresis provides high noise immunity
- Built-in scan path permits 100% factory testing of logic and I/O cells and functional testing with Automatic Test Vector Generation (ATVG) software after programming
- Available packages are 44- and 68-pin PLCC, 68-pin CPGA, and a 100-pin TQFP
- 68-pin PLCC compatible with QL12x16B
- 100-pin TQFP compatible with QL12x16B and QL16x24B
- 0.65µ CMOS process with ViaLink programming technology



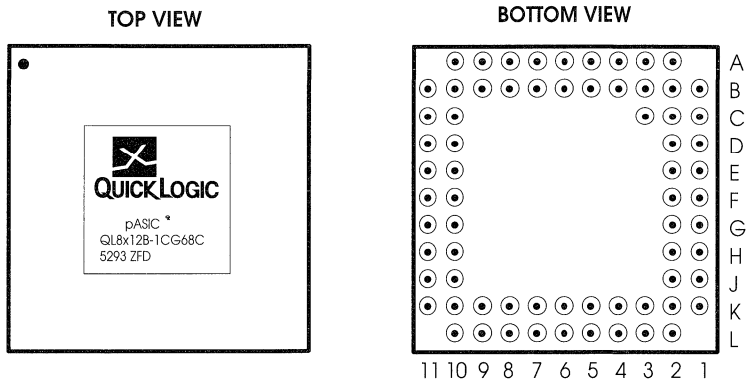
Pinout Diagram 44-pin PLCC



Pinout Diagram 68-pin PLCC

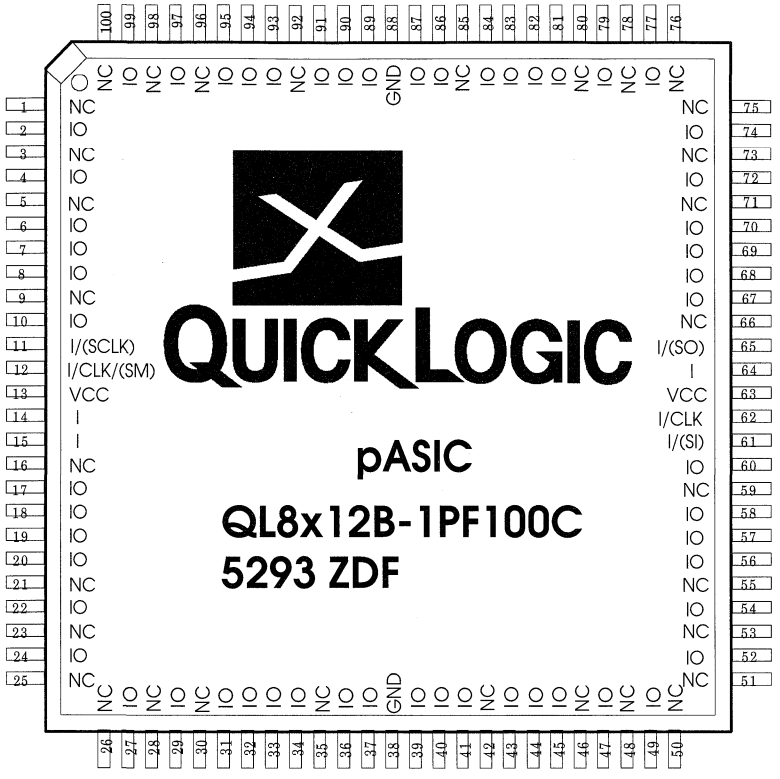
Pins identified I/SCLK, SM, SO and SI are used during scan path testing operation.

Pinout Diagram 68-pin CPGA



CPGA 68 Function/Connector Pin Table

PIN	FUNC	PIN	FUNC	PIN	FUNC	PIN	FUNC
B10	IO	B2	IO	K2	IO	K10	IO
A10	IO	B1	IO	L2	IO	K11	IO
B9	IO	C2	IO	K3	IO	J10	IO
A9	IO	C1	IO	L3	IO	J11	IO
B8	IO	D2	IO	K4	IO	H10	IO
A8	IO	D1	IO	L4	IO	H11	IO
B7	I/(SCLK)	E2	IO	K5	I/(SI)	G10	IO
A7	I/CLK/(SM)	E1	IO	L5	I/CLK	G11	IO
B6	VCC	F2	GND	K6	VCC	F10	GND
A6	I	F1	IO	L6	I	F11	IO
B5	I	G2	IO	K7	I/(SO)	E10	IO
A5	IO	G1	IO	L7	IO	E11	IO
B4	IO	H2	IO	K8	IO	D10	IO
A4	IO	H1	IO	L8	IO	D11	IO
B3	IO	J2	IO	K9	IO	C10	IO
A3	IO	J1	IO	L9	IO	C11	IO
A2	IO	K1	IO	L10	IO	B11	IO



Pinout Diagram
100-pin TQFP

ABSOLUTE MAXIMUM RATINGS

Supply Voltage ..... –0.5 to 7.0V
Input Voltage ..... –0.5 to VCC +0.5V
ESD Pad Protection ..... ±2000V
DC Input Current ..... ±20 mA
Latch-up Immunity ..... ±200 mA

Storage Temperature
Ceramic ..... –65°C to + 150°C
Plastic ..... –40°C to + 125°C
Lead Temperature ..... 300°C

OPERATING RANGE

Symbol	Parameter		Military		Industrial		Commercial		Unit
			Min	Max	Min	Max	Min	Max	
VCC	Supply Voltage		4.5	5.5	4.5	5.5	4.75	5.25	V
TA	Ambient Temperature		-55		-40	85	0	70	°C
TC	Case Temperature			125					°C
K	Delay Factor	-X Speed Grade	0.39	3.00	0.4	2.75	0.46	2.55	
		-0 Speed Grade	0.39	1.82	0.4	1.67	0.46	1.55	
		-1 Speed Grade	0.39	1.56	0.4	1.43	0.46	1.33	
		-2 Speed Grade			0.4	1.35	0.46	1.25	

DC CHARACTERISTICS over operating range

Symbol	Parameter	Conditions	Min	Max	Unit
VIH	Input HIGH Voltage		2.0		V
VIL	Input LOW Voltage			0.8	V
VOH	Output HIGH Voltage	IOH = -4 mA	3.7		V
		IOH = -8 mA	2.4		V
		IOH = -10 µA	VCC-0.1		V
VOL	Output LOW Voltage	IOL = 12 mA*		0.4	V
		IOL = 10 µA		0.1	V
II	Input Leakage Current	VI = VCC or GND	-10	10	µA
IOZ	3-State Output Leakage Current	VI = VCC or GND	-10	10	µA
CI	Input Capacitance [1]			10	pF
IOS	Output Short Circuit Current [2]	VO = GND	-10	-80	mA
		VO = VCC	30	140	mA
ICC	Supply Current [3]	VI, VIO = VCC or GND		10	mA

*IOL = 12 mA for commercial range only. IOL = 8 mA for the industrial and military ranges.

Notes:

[1] Capacitance is sample tested only. CI = 20 pF max on I/(SI).

[2] Only one output at a time. Duration should not exceed 30 seconds.

[3] For AC conditions use the formula described in the Data Book, Section 5 — Power vs Operating Frequency.

[4] Stated timing for worst case Propagation Delay over process variation at VCC = 5.0V and TA = 25°C. Multiply by the appropriate Delay Factor, K, for speed grade, voltage and temperature settings as specified in the Operating Range.

[5] These limits are derived from a representative selection of the slowest paths through the pASIC logic cell ***including net delays***. Worst case delay values for specific paths should be determined from timing analysis of your particular design.

AC CHARACTERISTICS at VCC = 5V, TA = 25°C (K = 1.00)

Logic Cell

Symbol	Parameter	Propagation Delays (ns) [4] Fanout				
		1	2	3	4	8
tPD	Combinatorial Delay [5]	1.7	2.1	2.6	3.0	4.8
tSU	Setup Time [5]	2.1	2.1	2.1	2.1	2.1
tH	Hold Time	0.0	0.0	0.0	0.0	0.0
tCLK	Clock to Q Delay	1.0	1.5	1.9	2.3	4.2
tCWHI	Clock High Time	2.0	2.0	2.0	2.0	2.0
tCWLO	Clock Low Time	2.0	2.0	2.0	2.0	2.0
tSET	Set Delay	1.7	2.1	2.6	3.0	4.8
tRESET	Reset Delay	1.5	1.8	2.2	2.5	3.9
tSW	Set Width	1.9	1.9	1.9	1.9	1.9
tRW	Reset Width	1.8	1.8	1.8	1.8	1.8

Input Cells

Symbol	Parameter	Propagation Delays (ns) [4]					
		1	2	3	4	6	8
tIN	High Drive Input Delay [6]	2.1	2.2	2.3	2.4	2.6	2.9
tINI	High Drive Input, Inverting Delay [6]	2.1	2.2	2.3	2.5	2.8	3.1
tIO	Input Delay (bidirectional pad)	1.4	1.8	2.2	2.6	3.4	4.2
tGCK	Clock Buffer Delay [7]	2.7	2.7	2.8	2.9	3.0	
tGCKHI	Clock Buffer Min High [7]	2.0	2.0	2.0	2.0	2.0	
tGCKLO	Clock Buffer Min Low [7]	2.0	2.0	2.0	2.0	2.0	

Output Cell

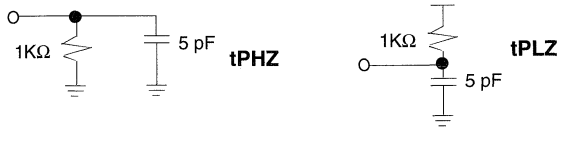
Symbol	Parameter	Propagation Delays (ns) [4] Output Load Capacitance (pF)				
		30	50	75	100	150
tOUTLH	Output Delay Low to High	2.7	3.4	4.2	5.0	6.7
tOUTHL	Output Delay High to Low	2.8	3.7	4.7	5.6	7.6
tPZH	Output Delay Tri-state to High	4.0	4.9	6.1	7.3	9.7
tPZL	Output Delay Tri-state to Low	3.6	4.2	5.0	5.8	7.3
tPHZ	Output Delay High to Tri-state [8]	2.9				
tPLZ	Output Delay Low to Tri-state [8]	3.3				

Notes:

[6] See High Drive Buffer Table for more information.

[7] Clock buffer fanout refers to the maximum number of flip flops per half column. The number of half columns used does not affect clock buffer delay.

[8] The following loads are used for tPXZ:



High Drive Buffer

Symbol	Parameter	Clock Drivers Wired Together	Propagation Delays (ns) [4]				
			Fanout				
			12	24	48	72	96
tIN	High Drive Input Delay	1	4.0	4.9			
		2		3.5	5.0		
		3			4.0	4.8	5.6
		4				4.1	4.8
tINI	High Drive Input, Inverting Delay	1	4.2	5.1			
		2		3.7	5.2		
		3			4.2	5.0	5.8
		4				4.3	5.0

AC Performance

Propagation delays depend on routing, fanout, load capacitance, supply voltage, junction temperature, and process variation. The AC Characteristics are a design guide to provide initial timing estimates at nominal conditions. Worst case estimates are obtained when nominal propagation delays are multiplied by the appropriate Delay Factor, K, as specified in the Delay Factor table (Operating Range). The effects of voltage and temperature variation are illustrated in the graphs on page 2-55, K Factor versus Voltage and Temperature. The SpDE Toolkit incorporates data sheet AC Characteristics into the QDIF database for pre-place-and-route timing analysis. The SpDE Delay Modeler extracts specific timing parameters for precise path analysis or simulation results following place and route. For definition of timing parameters, see page 2-56, Timing Waveforms.

ORDERING INFORMATION

QL 8x12B -1 PL68 C

- QL — QuickLogic pASIC device prefix
- 8x12B — pASIC device part number for 8 by 12 array of 96 logic cells, B = 0.65 micron CMOS
- -1 — Speed Grade
- PL68 — Package Code
  - PL44 = 44-pin PLCC
  - PL68 = 68-pin PLCC
  - CG68 = 68-pin CPGA
  - PF100 = 100-pin TQFP
- C — Operating Range
  - C = Commercial
  - I = Industrial
  - M = Military
  - M/883D = MIL-STD-883D Class B

QL12x16B
*Wild*Cat 2000
Very-High-Speed 2K (6K) Gate CMOS FPGA

Rev A

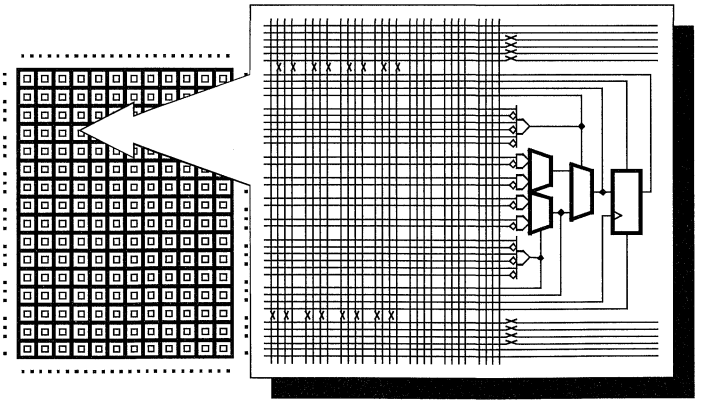
pASIC HIGHLIGHTS

...2000 usable gates, 88 I/O pins

- **Very High Speed** – ViaLink™ metal-to-metal programmable–via antifuse technology, allows counter speeds over 150 MHz and logic cell delays of under 2 ns.
- **High Usable Density** – A 12-by-16 array of 192 logic cells provides 6,000 total available gates, with 2000 typically usable "gate array" gates in 68-pin and 84-pin PLCC, 84-pin CPGA, 100-pin CQFP, and 100-pin TQFP packages.
- **Low-Power, High-Output Drive** – Standby current typically 2 mA. A 16-bit counter operating at 100 MHz consumes less than 50 mA. Minimum IOL of 12 mA and IOH of 8 mA
- **Low-Cost, Easy-to-Use Design Tools** – Designs entered and simulated using QuickLogic's new Quick*Works*™ development environment, or with third-party CAE tools including Viewlogic, Synopsys, Mentor, Cadence and Intergraph. Fast, fully automatic place and route on PC and workstation platforms using QuickLogic software.

QL12x16B Block Diagram

192 Logic Cells



QUICKLOGIC

▪ = Up to 80 prog. I/O cells, 6 Input high-drive cells, 2 Input/Clk (high-drive) cells

PRODUCT SUMMARY

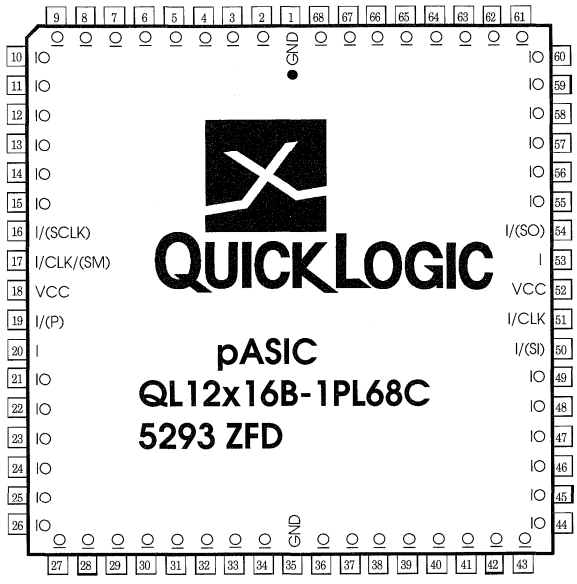
The QL12x16B is a member of the pASIC 1 Family of very-high-speed CMOS user-programmable ASIC devices. The 192 logic cell field-programmable gate array (FPGA) offers 2,000 usable "gate array" gates (equivalent to 6,000 gate claims of some vendors) of high-performance general-purpose logic in 68-pin and 84-pin PLCC packages, 84-pin CPGA, 100-pin CQFP, and 100-pin TQFP packages.

Low-impedance, metal-to-metal, ViaLink interconnect technology provides nonvolatile custom logic capable of operating above 150 MHz. Logic cell delays under 2 ns, combined with input delays of under 1.5 ns and output delays under 3 ns, permit high-density programmable devices to be used with today's fastest CISC and RISC microprocessors and DSPs.

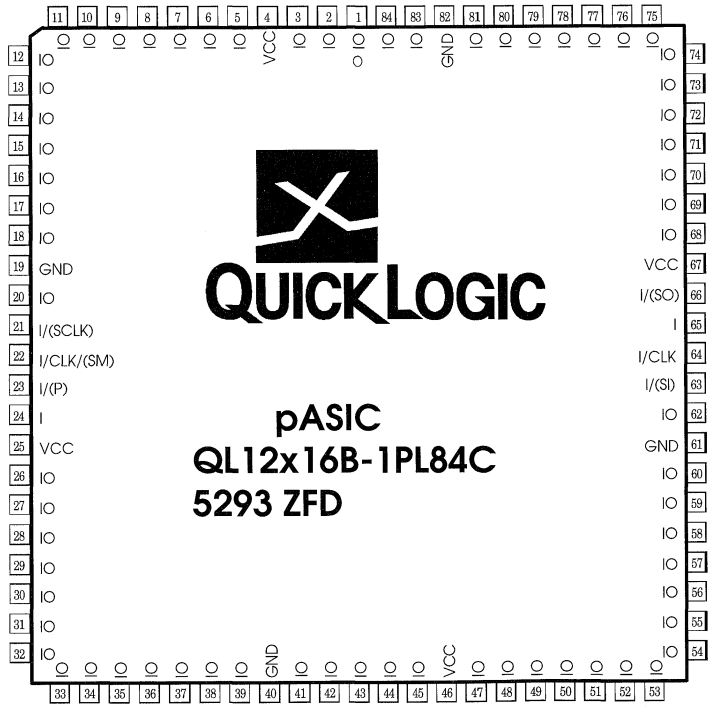
Designs are entered into the QL12x16B using a pASIC Toolkit which combines third-party design entry and simulation tools with QuickLogic's SpDE device-specific place & route and programming software. Ample on-chip routing channels are provided to allow fast, fully automatic place and route of designs using up to 100% of the logic and I/O cells, while maintaining fixed pin-outs.

FEATURES

- Total of 88 I/O pins
  - 80 Bidirectional Input/Output pins
  - 6 Dedicated Input/High-Drive pins
  - 2 Clock/Dedicated input pins with fanout-independent, low-skew clock networks
- Input + logic cell + output delays under 6 ns
- Chip-to-chip operating frequencies up to 110 MHz
- Internal state machine frequencies up to 150 MHz
- Clock skew < 0.5 ns
- Input hysteresis provides high noise immunity
- Built-in scan path permits 100% factory testing of logic and I/O cells and functional testing with Automatic Test Vector Generation (ATVG) software after programming
- Available packages are 68- and 84-pin PLCC, 84-pin CPGA, 100-pin CQFP, and a 100-pin TQFP
- 68-pin PLCC compatible with QL8x12B
- 84-pin PLCC compatible with QL16X24B
- 100-pin TQFP compatible with QL8x12B and QL16x24B
- 0.65µ CMOS process with ViaLink programming technology



Pinout Diagram 68-pin PLCC



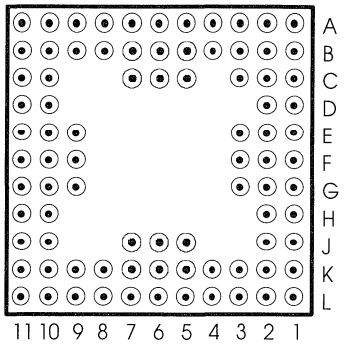
Pinout Diagram 84-pin PLCC

Pins identified I/SCLK, SM, SO and SI are used during scan path testing operation.

Pinout Diagram 84-pin CPGA

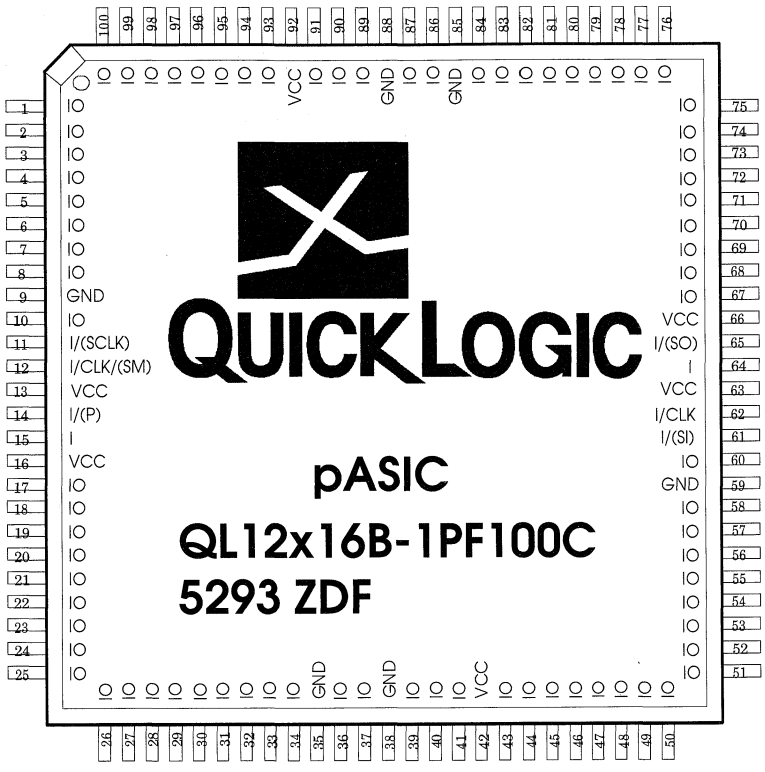
TOP VIEW

BOTTOM VIEW



CPGA 84 Function/Connector Pin Table

PIN	FUNC	PIN	FUNC	PIN	FUNC	PIN	FUNC
B10	IO	B2	IO	K2	IO	K10	IO
B9	IO	C2	IO	K3	IO	J10	IO
A10	IO	B1	IO	L2	IO	K11	IO
A9	IO	C1	IO	L3	IO	J11	IO
B8	IO	D2	IO	K4	IO	H10	IO
A8	IO	D1	IO	L4	IO	H11	IO
A7	IO	E1	IO	L5	IO	G11	IO
C7	GND	E3	GND	J5	GND	G9	GND
A6	IO	E2	IO	L6	IO	G10	IO
B7	I/(SCLK)	F1	IO	K5	I/(SI)	F11	IO
C6	I/CLK/(SM)	F2	IO	J6	I/CLK	F10	IO
B6	I(P)	F3	IO	K6	I	F9	IO
B5	I	G1	IO	K7	I/(SO)	E11	IO
C5	VCC	G3	VCC	J7	VCC	E9	VCC
A5	IO	G2	IO	L7	IO	E10	IO
A4	IO	H1	IO	L8	IO	D11	IO
B4	IO	H2	IO	K8	IO	D10	IO
A3	IO	J1	IO	L9	IO	C11	IO
A2	IO	K1	IO	L10	IO	B11	IO
B3	IO	J2	IO	K9	IO	C10	IO
A1	IO	L1	IO	L11	IO	A11	IO



Pinout Diagram
100-pin TQFP

ABSOLUTE MAXIMUM RATINGS

Supply Voltage ..... –0.5 to 7.0V
Input Voltage ..... –0.5 to VCC +0.5V
ESD Pad Protection ..... ±2000V
DC Input Current ..... ±20 mA
Latch-up Immunity ..... ±200 mA

Storage Temperature
Ceramic ..... –65°C to + 150°C
Plastic ..... –40°C to + 125°C
Lead Temperature ..... 300°C

OPERATING RANGE

Symbol	Parameter		Military		Industrial		Commercial		Unit
			Min	Max	Min	Max	Min	Max	
VCC	Supply Voltage		4.5	5.5	4.5	5.5	4.75	5.25	V
TA	Ambient Temperature		-55		-40	85	0	70	°C
TC	Case Temperature			125					°C
K	Delay Factor	-X Speed Grade	0.39	3.00	0.4	2.75	0.46	2.55	
		-0 Speed Grade	0.39	1.82	0.4	1.67	0.46	1.55	
		-1 Speed Grade	0.39	1.56	0.4	1.43	0.46	1.33	
		-2 Speed Grade			0.4	1.35	0.46	1.25	

DC CHARACTERISTICS over operating range

Symbol	Parameter	Conditions	Min	Max	Unit
VIH	Input HIGH Voltage		2.0		V
VIL	Input LOW Voltage			0.8	V
VOH	Output HIGH Voltage	IOH = -4 mA	3.7		V
		IOH = -8 mA	2.4		V
		IOH = -10 µA	VCC-0.1		V
VOL	Output LOW Voltage	IOL = 12 mA*		0.4	V
		IOL = 10 µA		0.1	V
II	Input Leakage Current	VI = VCC or GND	-10	10	µA
IOZ	3-State Output Leakage Current	VI = VCC or GND	-10	10	µA
CI	Input Capacitance [1]			10	pF
IOS	Output Short Circuit Current [2]	VO = GND	-10	-80	mA
		VO = VCC	30	140	mA
ICC	Supply Current [3]	VI, VIO = VCC or GND		10	mA

*IOL = 12 mA for commercial range only. IOL = 8 mA for the industrial and military ranges.

Notes:

[1] Capacitance is sample tested only. CI = 40 pF max on I/(SI) and I/(P).

[2] Only one output at a time. Duration should not exceed 30 seconds.

[3] For AC conditions use the formula described in the Data Book, Section 5 — Power vs Operating Frequency.

[4] Stated timing for worst case Propagation Delay over process variation at VCC = 5.0V and TA = 25°C. Multiply by the appropriate Delay Factor, K, for speed grade, voltage and temperature settings as specified in the Operating Range.

[5] These limits are derived from a representative selection of the slowest paths through the pASIC logic cell ***including net delays***. Worst case delay values for specific paths should be determined from timing analysis of your particular design.

AC CHARACTERISTICS at VCC = 5V, TA = 25°C (K = 1.00)

Logic Cell

Symbol	Parameter	Propagation Delays (ns) [4] Fanout				
		1	2	3	4	8
tPD	Combinatorial Delay [5]	1.7	2.2	2.6	3.2	5.2
tSU	Setup Time [5]	2.1	2.1	2.1	2.1	2.1
tH	Hold Time	0.0	0.0	0.0	0.0	0.0
tCLK	Clock to Q Delay	1.0	1.5	1.9	2.5	4.6
tCWHI	Clock High Time	2.0	2.0	2.0	2.0	2.0
tCWLO	Clock Low Time	2.0	2.0	2.0	2.0	2.0
tSET	Set Delay	1.7	2.1	2.6	3.2	5.2
tRESET	Reset Delay	1.5	1.9	2.2	2.7	4.3
tSW	Set Width	1.9	1.9	1.9	1.9	1.9
tRW	Reset Width	1.8	1.8	1.8	1.8	1.8

Input Cells

Symbol	Parameter	Propagation Delays (ns) [4]					
		1	2	3	4	6	8
tIN	High Drive Input Delay [6]	2.4	2.5	2.6	2.7	3.0	3.3
tINI	High Drive Input, Inverting Delay [6]	2.5	2.6	2.7	2.8	3.1	3.4
tIO	Input Delay (bidirectional pad)	1.4	1.9	2.2	2.8	3.7	4.6
tGCK	Clock Buffer Delay [7]	2.7	2.8	2.8	2.9	2.9	3.0
tGCKHI	Clock Buffer Min High [7]	2.0	2.0	2.0	2.0	2.0	2.0
tGCKLO	Clock Buffer Min Low [7]	2.0	2.0	2.0	2.0	2.0	2.0

Output Cell

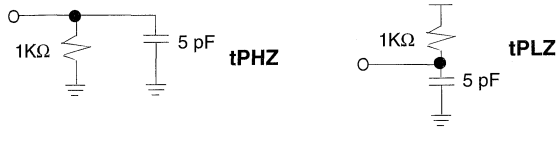
Symbol	Parameter	Propagation Delays (ns) [4] Output Load Capacitance (pF)				
		30	50	75	100	150
tOUTLH	Output Delay Low to High	2.7	3.4	4.2	5.0	6.7
tOUTHL	Output Delay High to Low	2.8	3.7	4.7	5.6	7.6
tPZH	Output Delay Tri-state to High	4.0	4.9	6.1	7.3	9.7
tPZL	Output Delay Tri-state to Low	3.6	4.2	5.0	5.8	7.3
tPHZ	Output Delay High to Tri-state [8]	2.9				
tPLZ	Output Delay Low to Tri-state [8]	3.3				

Notes:

[6] See High Drive Buffer Table for more information.

[7] Clock buffer fanout refers to the maximum number of flip flops per half column. The number of half columns used does not affect clock buffer delay.

[8] The following loads are used for tPXZ:



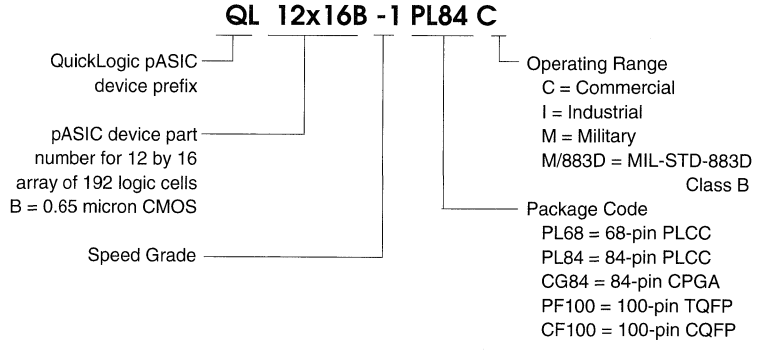
High Drive Buffer

Symbol	Parameter	# High Drives Wired Together	Propagation Delays (ns) [4] Fanout				
			12	24	48	72	96
tIN	High Drive Input Delay	1	4.5	5.4			
		2		3.9	5.6		
		3			4.5	5.3	6.3
		4				4.6	5.3
tINI	High Drive Input, Inverting Delay	1	4.7	5.6			
		2		4.0	5.8		
		3			4.6	5.5	6.4
		4				4.8	5.5

AC Performance

Propagation delays depend on routing, fanout, load capacitance, supply voltage, junction temperature, and process variation. The AC Characteristics are a design guide to provide initial timing estimates at nominal conditions. Worst case estimates are obtained when nominal propagation delays are multiplied by the appropriate Delay Factor, K, as specified in the Delay Factor table (Operating Range). The effects of voltage and temperature variation are illustrated in the graphs on page 2-55, K Factor versus Voltage and Temperature. The SpDE Toolkit incorporates data sheet AC Characteristics into the QDIF database for pre-place-and-route timing analysis. The SpDE Delay Modeler extracts specific timing parameters for precise path analysis or simulation results following place and route. For definition of timing parameters, see page 2-56, Timing Waveforms.

ORDERING INFORMATION



QL16x24B
*Wild*Cat 4000
Very-High-Speed 4K (12K) Gate CMOS FPGA

Rev A

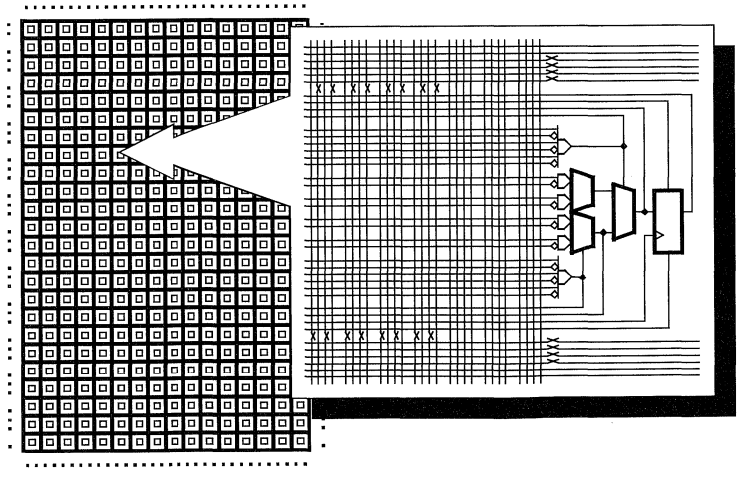
pASIC HIGHLIGHTS

...4000 usable gates, 122 I/O pins

- **Very High Speed** – ViaLink™ metal-to-metal programmable–via antifuse technology, allows counter speeds over 150 MHz and logic cell delays of under 2 ns.
- **High Usable Density** – A 16-by-24 array of 384 logic cells provides 12,000 total available gates, with 4000 typically usable "gate array" gates in 84-pin PLCC and 100-pin and 144-pin TQFP, 144-pin CPGA, 100-pin and 160-pin CQFP packages.
- **Low-Power, High-Output Drive** – Standby current typically 2 mA. A 16-bit counter operating at 100 MHz consumes less than 50 mA. Minimum IOL of 12 mA and IOH of 8 mA
- **Low-Cost, Easy-to-Use Design Tools** – Designs entered and simulated using QuickLogic's new Quick*Works*™ development environment, or with third-party CAE tools including Viewlogic, Synopsys, Mentor, Cadence and Intergraph. Fast, fully automatic place and route on PC and workstation platforms using QuickLogic software.

QL16x24B Block Diagram

384 Logic Cells



▪ = Up to 114 prog. I/O cells, 6 Input high-drive cells, 2 Input/Clk (high-drive) cells

QUICKLOGIC

PRODUCT SUMMARY

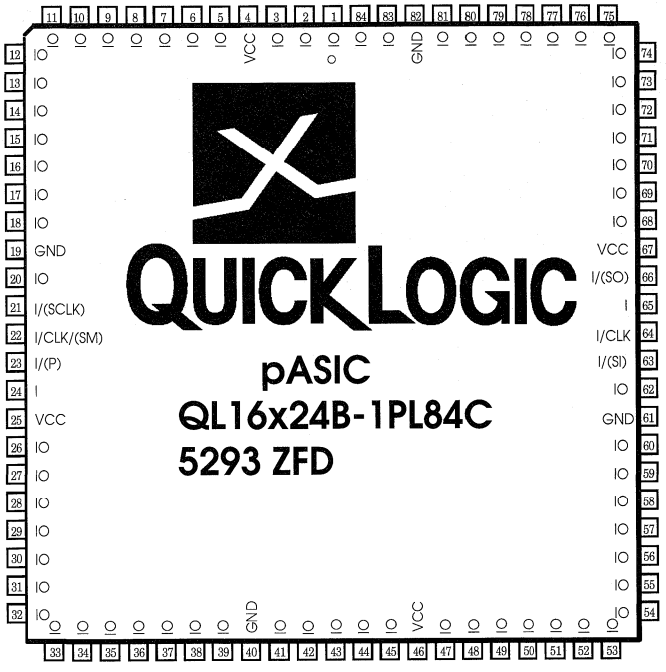
The QL16x24B is a member of the pASIC 1 Family of very-high-speed CMOS user-programmable ASIC devices. The 384 logic cell field-programmable gate array (FPGA) offers 4,000 usable "gate array" gates (equivalent to 12,000 gate claims of some vendors) of high-performance general-purpose logic in 84-pin PLCC, and 100-pin and 144-pin TQFP, 144-pin CPGA, 100-pin and 160-pin CQFP packages.

Low-impedance, metal-to-metal, ViaLink interconnect technology provides nonvolatile custom logic capable of operating above 150 MHz. Logic cell delays under 2 ns, combined with input delays of under 1.5 ns and output delays under 3 ns, permit high-density programmable devices to be used with today's fastest CISC and RISC microprocessors and DSPs.

Designs are entered into the QL16x24B using a pASIC Toolkit which combines third-party design entry and simulation tools with QuickLogic's SpDE device-specific place & route and programming software. Ample on-chip routing channels are provided to allow fast, fully automatic place and route of designs using up to 100% of the logic and I/O cells, while maintaining fixed pin-outs.

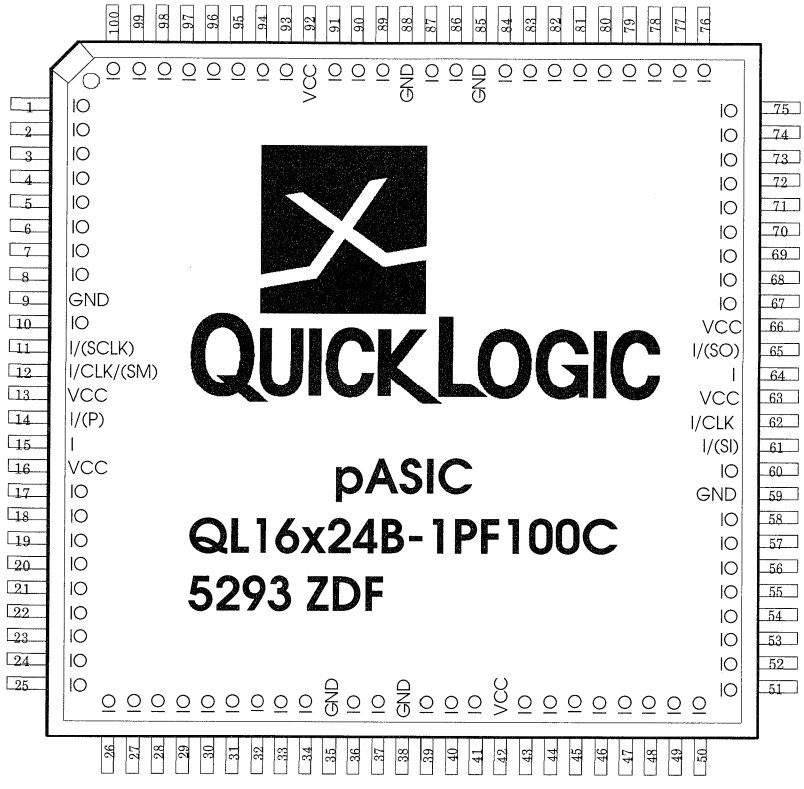
FEATURES

- Total of 122 I/O pins
  - 114 Bidirectional Input/Output pins
  - 6 Dedicated Input/High-Drive pins
  - 2 Clock/Dedicated input pins with fanout-independent, low-skew clock networks
- Input + logic cell + output delays under 6 ns
- Chip-to-chip operating frequencies up to 110 MHz
- Internal state machine frequencies up to 150 MHz
- Clock skew < 0.5 ns
- Input hysteresis provides high noise immunity
- Built-in scan path permits 100% factory testing of logic and I/O cells and functional testing with Automatic Test Vector Generation (ATVG) software after programming
- Packages: 84-pin PLCC, 100- and 144-pin TQFP, 144 CPGA, 100- and 160-pin CQFP
- 84-pin PLCC compatible with QL12x16B
- 100-pin TQFP compatible with QL8x12B and QL12x16B
- 144-pin TQFP compatible with QL24x32B
- 0.65µ CMOS process with ViaLink programming technology

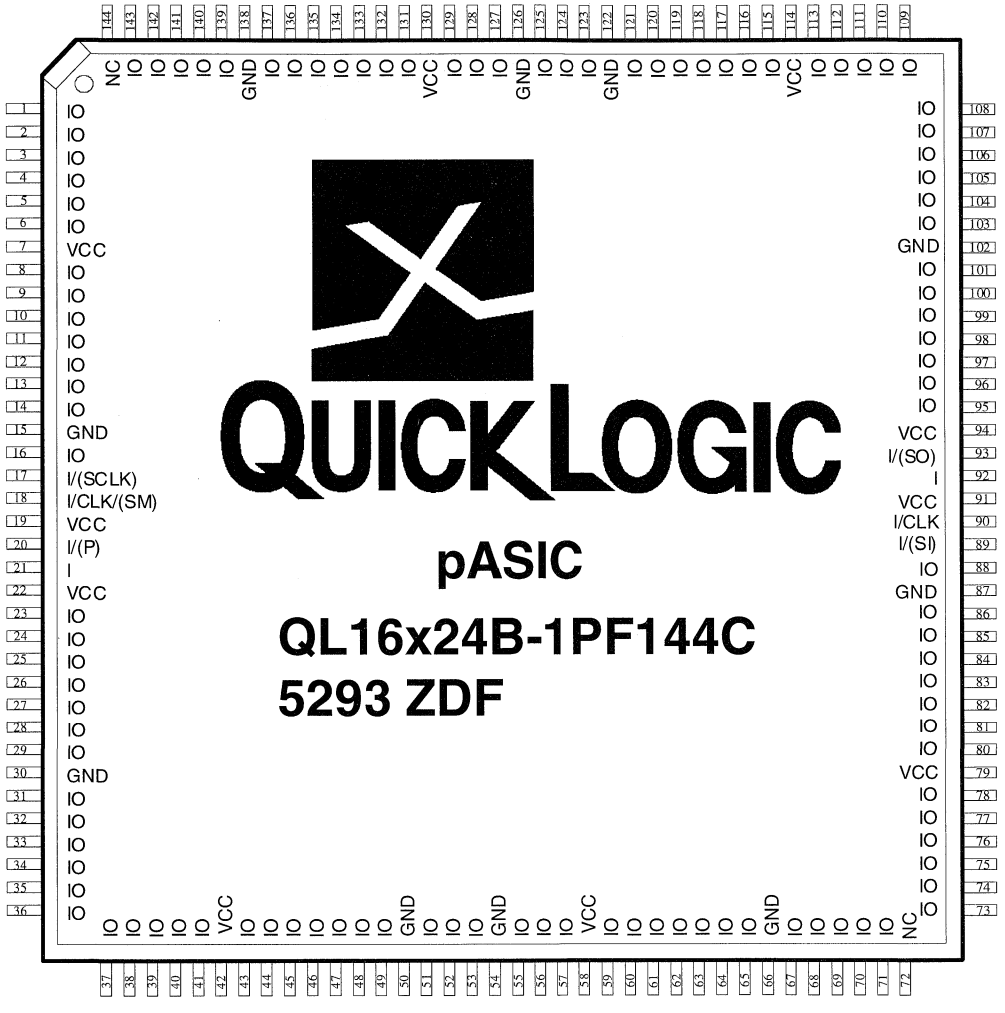


Pinout Diagram 84-pin PLCC

Pinout Diagram
100-pin TQFP



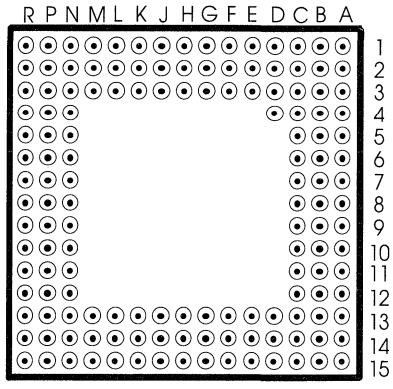
Pinout Diagram
144-pin TQFP



Pinout Diagram
144-pin CPGA

TOP VIEW

BOTTOM VIEW



CPGA 144 Function/Connector Table

PIN	FUNC	PIN	FUNC	PIN	FUNC	PIN	FUNC
A2	IO	B15	IO	R14	IO	P1	IO
B3	IO	C14	IO	P13	IO	N2	IO
C4	IO	D13	IO	N12	IO	M3	IO
A3	IO	C15	IO	R13	IO	N1	IO
B4	IO	D14	IO	P12	IO	M2	IO
A4	IO	E13	VCC	R12	IO	L3	VCC
C3	VCC	D15	IO	N13	VCC	M1	IO
B5	IO	E14	IO	P11	IO	L2	IO
A5	IO	E15	IO	R11	IO	L1	IO
C6	IO	F13	IO	N10	IO	K3	IO
B6	IO	F14	IO	P10	IO	K2	IO
A6	IO	F15	IO	R10	IO	K1	IO
A7	IO	G15	IO	R9	IO	J1	IO
B7	IO	C13	GND	P9	IO	N3	GND
C5	GND	G14	IO	N11	GND	J2	IO
A8	IO	H15	IO	R8	IO	H1	IO
B8	I/(SCLK)	H14	IO	P8	I/(SI)	H2	IO
C8	I/CLK/(SM)	G13	GND	N8	I/CLK	J3	GND
C7	VCC	H13	IO	N9	VCC	H3	IO
A9	I/(P)	J15	IO	R7	I	G1	IO
B9	I	J14	IO	P7	I/(SO)	G2	IO
C11	VCC	J13	VCC	N5	VCC	G3	VCC
A10	IO	K15	IO	R6	IO	F1	IO
A11	IO	L15	IO	R5	IO	E1	IO
B10	IO	K14	IO	P6	IO	F2	IO
A12	IO	M15	IO	R4	IO	D1	IO
B11	IO	L14	IO	P5	IO	E2	IO
C10	IO	K13	IO	N6	IO	F3	IO
A13	IO	N15	IO	R3	IO	C1	IO
C9	GND	L13	GND	N7	GND	E3	GND
B12	IO	M14	IO	P4	IO	D2	IO
A14	IO	P15	IO	R2	IO	B1	IO
B13	IO	N14	IO	P3	IO	C2	IO
C12	IO	M13	IO	N4	IO	D3	IO
A15	IO	R15	IO	R1	IO	A1	IO
B14	IO	P14	nc	P2	IO	B2	nc

ABSOLUTE MAXIMUM RATINGS

Supply Voltage ..... –0.5 to 7.0V
Input Voltage ..... –0.5 to VCC +0.5V
ESD Pad Protection ..... ±2000V
DC Input Current ..... ±20 mA
Latch-up Immunity ..... ±200 mA

Storage Temperature
Ceramic ..... –65°C to + 150°C
Plastic ..... –40°C to + 125°C
Lead Temperature ..... 300°C

OPERATING RANGE

Symbol	Parameter		Military		Industrial		Commercial		Unit
			Min	Max	Min	Max	Min	Max	
VCC	Supply Voltage		4.5	5.5	4.5	5.5	4.75	5.25	V
TA	Ambient Temperature		-55		-40	85	0	70	°C
TC	Case Temperature			125					°C
K	Delay Factor	-X Speed Grade	0.39	3.00	0.4	2.75	0.46	2.55	
		-0 Speed Grade	0.39	1.82	0.4	1.67	0.46	1.55	
		-1 Speed Grade	0.39	1.56	0.4	1.43	0.46	1.33	
		-2 Speed Grade			0.4	1.35	0.46	1.25	

DC CHARACTERISTICS over operating range

Symbol	Parameter	Conditions	Min	Max	Unit
VIH	Input HIGH Voltage		2.0		V
VIL	Input LOW Voltage			0.8	V
VOH	Output HIGH Voltage	IOH = -4 mA	3.7		V
		IOH = -8 mA	2.4		V
		IOH = -10 µA	VCC-0.1		V
VOL	Output LOW Voltage	IOL = 12 mA*		0.4	V
		IOL = 10 µA		0.1	V
II	Input Leakage Current	VI = VCC or GND	-10	10	µA
IOZ	3-State Output Leakage Current	VI = VCC or GND	-10	10	µA
CI	Input Capacitance [1]			10	pF
IOS	Output Short Circuit Current [2]	VO = GND	-10	-80	mA
		VO = VCC	30	140	mA
ICC	Supply Current [3]	VI, VIO = VCC or GND		10	mA

*IOL = 12 mA for commercial range only. IOL = 8 mA for the industrial and military ranges.

Notes:

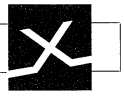
[1] Capacitance is sample tested only. CI = 45 pF max on I/(SI) and I/(P).

[2] Only one output at a time. Duration should not exceed 30 seconds.

[3] For AC conditions use the formula described in the Data Book, Section 5 — Power vs Operating Frequency.

[4] Stated timing for worst case Propagation Delay over process variation at VCC = 5.0V and TA = 25°C. Multiply by the appropriate Delay Factor, K, for speed grade, voltage and temperature settings as specified in the Operating Range.

[5] These limits are derived from a representative selection of the slowest paths through the pASIC logic cell ***including net delays***. Worst case delay values for specific paths should be determined from timing analysis of your particular design.



AC CHARACTERISTICS at VCC = 5V, TA = 25°C (K = 1.00)

Logic Cell

Symbol	Parameter	Propagation Delays (ns) [4] Fanout				
		1	2	3	4	8
tPD	Combinatorial Delay [5]	1.7	2.2	2.6	3.2	5.3
tSU	Setup Time [5]	2.1	2.1	2.1	2.1	2.1
tH	Hold Time	0.0	0.0	0.0	0.0	0.0
tCLK	Clock to Q Delay	1.0	1.5	1.9	2.6	4.7
tCWHI	Clock High Time	2.0	2.0	2.0	2.0	2.0
tCWLO	Clock Low Time	2.0	2.0	2.0	2.0	2.0
tSET	Set Delay	1.7	2.2	2.6	3.2	5.3
tRESET	Reset Delay	1.5	1.9	2.2	2.7	4.4
tSW	Set Width	1.9	1.9	1.9	1.9	1.9
tRW	Reset Width	1.8	1.8	1.8	1.8	1.8

Input Cells

Symbol	Parameter	Propagation Delays (ns) [4]					
		1	2	3	4	8	12
tIN	High Drive Input Delay [6]	2.8	2.9	3.0	3.1	4.0	5.3
tINI	High Drive Input, Inverting Delay [6]	3.0	3.1	3.2	3.3	4.1	5.7
tIO	Input Delay (bidirectional pad)	1.4	1.9	2.2	2.9	4.7	6.5
tGCK	Clock Buffer Delay [7]	2.7	2.8	2.9	3.0	3.1	3.3
tGCKHI	Clock Buffer Min High [7]	2.0	2.0	2.0	2.0	2.0	2.0
tGCKLO	Clock Buffer Min Low [7]	2.0	2.0	2.0	2.0	2.0	2.0

Output Cell

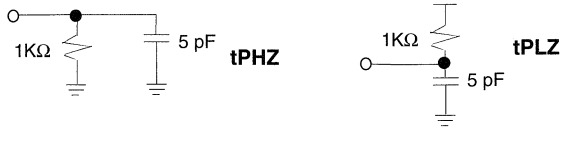
Symbol	Parameter	Propagation Delays (ns) [4] Output Load Capacitance (pF)				
		30	50	75	100	150
tOUTLH	Output Delay Low to High	2.7	3.4	4.2	5.0	6.7
tOUTHL	Output Delay High to Low	2.8	3.7	4.7	5.6	7.6
tPZH	Output Delay Tri-state to High	4.0	4.9	6.1	7.3	9.7
tPZL	Output Delay Tri-state to Low	3.6	4.2	5.0	5.8	7.3
tPHZ	Output Delay High to Tri-state [8]	2.9				
tPLZ	Output Delay Low to Tri-state [8]	3.3				

Notes:

[6] See High Drive Buffer Table for more information.

[7] Clock buffer fanout refers to the maximum number of flip flops per half column. The number of half columns used does not affect clock buffer delay.

[8] The following loads are used for tPXZ:



High Drive Buffer

Symbol	Parameter	# High Drives Wired Together	Propagation Delays (ns) [4]				
			Fanout				
			12	24	48	72	96
tIN	High Drive Input Delay	1	5.3	6.7			
		2		4.5	6.6		
		3			5.3	6.2	7.2
		4				5.4	6.2
tINI	High Drive Input, Inverting Delay	1	5.7	7.2			
		2		4.6	6.8		
		3			5.5	6.4	7.4
		4				5.6	6.4

AC Performance

Propagation delays depend on routing, fanout, load capacitance, supply voltage, junction temperature, and process variation. The AC Characteristics are a design guide to provide initial timing estimates at nominal conditions. Worst case estimates are obtained when nominal propagation delays are multiplied by the appropriate Delay Factor, K, as specified in the Delay Factor table (Operating Range). The effects of voltage and temperature variation are illustrated in the graphs on page 2-55, K Factor versus Voltage and Temperature. The SpDE Toolkit incorporates data sheet AC Characteristics into the QDIF database for pre-place-and-route timing analysis. The SpDE Delay Modeler extracts specific timing parameters for precise path analysis or simulation results following place and route. For definition of timing parameters, see page 2-56, Timing Waveforms.

ORDERING INFORMATION

QL 16x24B -1 PF144 C

- QL — QuickLogic pASIC device prefix
- 16x24B — pASIC device part number for 16 by 24 array of 384 logic cells B = 0.65 micron CMOS
- -1 — Speed Grade
- PF144 — Package Code
  - PL84 = 84-pin PLCC
  - PF100 = 100-pin TQFP
  - CF100 = 100-pin CQFP
  - PF144 = 144-pin TQFP
  - CG144 = 144-pin CPGA
  - CF160 = 160-pin CQFP
- C — Operating Range
  - C = Commercial
  - I = Industrial
  - M = Military
  - M/883D = MIL-STD-883D Class B

QL24x32B
*Wild*Cat 8000
Very-High-Speed 8K (24K) Gate CMOS FPGA

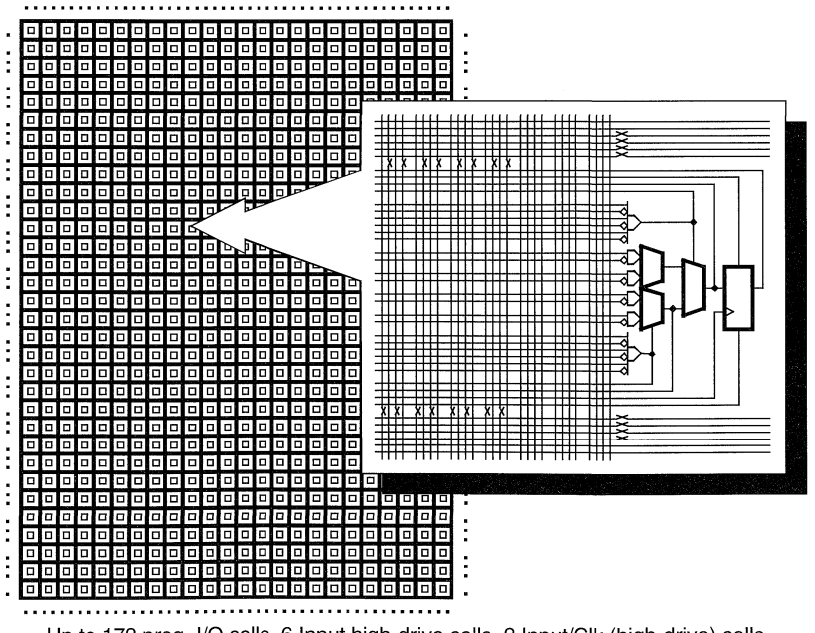
pASIC HIGHLIGHTS

...8000 usable gates, 180 I/O pins

- **Very High Speed** – ViaLink™ metal-to-metal programmable–via antifuse technology, allows counter speeds over 150 MHz and logic cell delays of under 2 ns.
- **High Usable Density** – A 24-by-32 array of 768 logic cells provides 24,000 total available gates, with 8000 typically usable "gate array" gates in 144-pin TQFP packages, and 208-pin metric PQFP packages.
- **PCI-Output Drive** – Fully PCI 2.0 compliant input/output capability. (including drive current)
- **Low-Cost, Easy-to-Use Design Tools** – Designs entered and simulated using QuickLogic's new Quick*Works*™ development environment, or with third-party CAE tools including Viewlogic, Synopsys, Mentor, Cadence and Intergraph. Fast, fully automatic place and route on PC and workstation platforms using QuickLogic software.

QL24x32B Block Diagram

768 Logic Cells



■ = Up to 172 prog. I/O cells, 6 Input high-drive cells, 2 Input/Clk (high-drive) cells

QUICKLOGIC

PRODUCT SUMMARY

The QL24x32B is a member of the pASIC 1 Family of very-high-speed CMOS user-programmable ASIC devices. The 768 logic cell field-programmable gate array (FPGA) offers 8,000 usable "gate array" gates (equivalent to 24,000 gate claims of some vendors) of high-performance general-purpose logic in 144-pin TQFP and 208-pin PQFP packages.

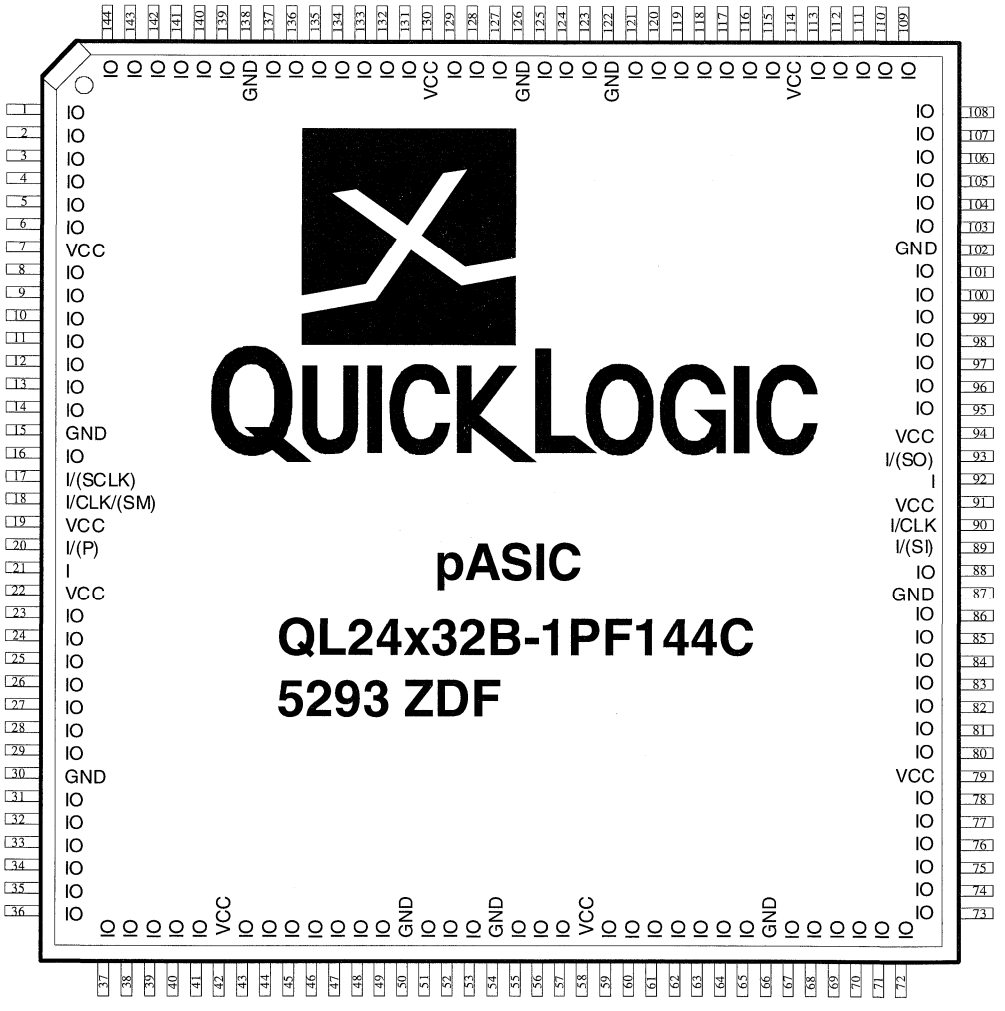
Low-impedance, metal-to-metal, ViaLink interconnect technology provides nonvolatile custom logic capable of operating above 150 MHz. Logic cell delays under 2 ns, combined with input delays of under 1.5 ns and output delays under 3 ns, permit high-density programmable devices to be used with today's fastest CISC and RISC microprocessors and DSPs.

Designs are entered into the QL24x32B using a pASIC Toolkit which combines third-party design entry and simulation tools with QuickLogic's SpDE device-specific place & route and programming software. Ample on-chip routing channels are provided to allow fast, fully automatic place and route of designs using up to 100% of the logic and I/O cells, while maintaining fixed pin-outs.

FEATURES

- Total of 180 I/O pins
  - 172 Bidirectional Input/Output pins
  - 6 Dedicated Input/High-Drive pins
  - 2 Clock/Dedicated input pins with fanout-independent, low-skew clock networks
- PCI 2.0 Compliant I/O's
- Input + logic cell + output delays under 6 ns
- Chip-to-chip operating frequencies up to 110 MHz
- Internal state machine frequencies up to 150 MHz
- Clock skew < 0.5 ns
- Input hysteresis provides high noise immunity
- Built-in scan path permits 100% factory testing of logic and I/O cells and functional testing with Automatic Test Vector Generation (ATVG) software after programming
- 144-pin TQFP upward compatible from QL16x24B
- 0.65µ CMOS process with ViaLink programming technology

Pinout Diagram 144-pin TQFP



Pinout Diagram
208-pin PQFP

PQFP 208 Function/Connector Table

PIN	FUN	PIN	FUN	PIN	FUN	PIN	FUN	PIN	FUN	PIN	FUN	PIN	FUN	PIN	FUN
1	I/O	27	VCC	53	I/O	79	I/O	105	I/O	131	VCC	157	I/O	183	I/O
2	I/O	28	I/P	54	I/O	80	I/O	106	I/O	132	I	158	I/O	184	I/O
3	I/O	29	I	55	I/O	81	I/O	107	I/O	133	I/SO	159	I/O	185	I/O
4	I/O	30	VCC	56	I/O	82	I/O	108	I/O	134	VCC	160	I/O	186	I/O
5	I/O	31	I/O	57	I/O	83	VCC	109	I/O	135	I/O	161	I/O	187	VCC
6	I/O	32	I/O	58	I/O	84	I/O	110	I/O	136	I/O	162	I/O	188	I/O
7	I/O	33	I/O	59	GND	85	I/O	111	I/O	137	I/O	163	GND	189	I/O
8	I/O	34	I/O	60	I/O	86	I/O	112	I/O	138	I/O	164	I/O	190	I/O
9	I/O	35	I/O	61	VCC	87	I/O	113	I/O	139	I/O	165	VCC	191	I/O
10	VCC	36	I/O	62	I/O	88	I/O	114	VCC	140	I/O	166	I/O	192	I/O
11	I/O	37	I/O	63	I/O	89	I/O	115	I/O	141	I/O	167	I/O	193	I/O
12	GND	38	I/O	64	I/O	90	I/O	116	GND	142	I/O	168	I/O	194	I/O
13	I/O	39	I/O	65	I/O	91	I/O	117	I/O	143	I/O	169	I/O	195	I/O
14	I/O	40	I/O	66	I/O	92	I/O	118	I/O	144	I/O	170	I/O	196	I/O
15	I/O	41	VCC	67	I/O	93	I/O	119	I/O	145	VCC	171	I/O	197	I/O
16	I/O	42	I/O	68	I/O	94	I/O	120	I/O	146	I/O	172	I/O	198	I/O
17	I/O	43	GND	69	I/O	95	GND	121	I/O	147	GND	173	I/O	199	GND
18	I/O	44	I/O	70	I/O	96	I/O	122	I/O	148	I/O	174	I/O	200	I/O
19	I/O	45	I/O	71	I/O	97	VCC	123	I/O	149	I/O	175	I/O	201	VCC
20	I/O	46	I/O	72	I/O	98	I/O	124	I/O	150	I/O	176	I/O	202	I/O
21	I/O	47	I/O	73	GND	99	I/O	125	I/O	151	I/O	177	GND	203	I/O
22	I/O	48	I/O	74	I/O	100	I/O	126	I/O	152	I/O	178	I/O	204	I/O
23	GND	49	I/O	75	I/O	101	I/O	127	GND	153	I/O	179	I/O	205	I/O
24	I/O	50	I/O	76	I/O	102	I/O	128	I/O	154	I/O	180	I/O	206	I/O
25	I/Sck	51	I/O	77	I/O	103	I/O	129	I/SI	155	I/O	181	I/O	207	I/O
26	I/Clk	52	I/O	78	GND	104	I/O	130	I/Clk	156	I/O	182	GND	208	I/O

ABSOLUTE MAXIMUM RATINGS

Supply Voltage ..... –0.5 to 7.0V
Input Voltage ..... –0.5 to VCC +0.5V
ESD Pad Protection ..... ±2000V
DC Input Current ..... ±20 mA
Latch-up Immunity ..... ±200 mA

Storage Temperature
Ceramic ..... –65°C to + 150°C
Plastic ..... –40°C to + 125°C
Lead Temperature ..... 300°C

OPERATING RANGE

Symbol	Parameter		Military		Industrial		Commercial		Unit
			Min	Max	Min	Max	Min	Max	
VCC	Supply Voltage		4.5	5.5	4.5	5.5	4.75	5.25	V
TA	Ambient Temperature		-55		-40	85	0	70	°C
TC	Case Temperature			125					°C
K	Delay Factor	-X Speed Grade*	0.39	3.00	0.4	2.75	0.46	2.55	
		-0 Speed Grade	0.39	1.82	0.4	1.67	0.46	1.55	
		-1 Speed Grade*	0.39	1.56	0.4	1.43	0.46	1.33	
		-2 Speed Grade*			0.4	1.35	0.46	1.25	

* Contact QuickLogic Sales for availability.

DC CHARACTERISTICS over operating range

Symbol	Parameter	Conditions	Min	Max	Unit
VIH	Input HIGH Voltage		2.0		V
VIL	Input LOW Voltage			0.8	V
VOH	Output HIGH Voltage	IOH = -4 mA	3.7		V
		IOH = -8 mA	2.4		V
		IOH = -10 µA	VCC-0.1		V
VOL	Output LOW Voltage	IOL = 12 mA		0.4	V
		IOL = 10 µA		0.1	V
II	Input Leakage Current	VI = VCC or GND	-10	10	µA
IOZ	3-State Output Leakage Current	VI = VCC or GND	-10	10	µA
CI	Input Capacitance [1]			10	pF
IOS	Output Short Circuit Current [2]	VO = GND	-10	-90	mA
		VO = VCC	40	160	mA
ICC	Supply Current [3]	VI, VIO = VCC or GND		10	mA

Notes:

[1] Capacitance is sample tested only.

[2] Only one output at a time. Duration should not exceed 30 seconds.

[3] For AC conditions use the formula described in the Data Book, Section 5 — Power vs Operating Frequency.

[4] Stated timing for worst case Propagation Delay over process variation at VCC = 5.0V and TA = 25°C. Multiply by the appropriate Delay Factor, K, for speed grade, voltage and temperature settings as specified in the Operating Range.

[5] These limits are derived from a representative selection of the slowest paths through the pASIC logic cell ***including net delays***. Worst case delay values for specific paths should be determined from timing analysis of your particular design.

AC CHARACTERISTICS at VCC = 5V, TA = 25°C (K = 1.00)

Logic Cell

Symbol	Parameter	Propagation Delays (ns) Fanout [4]				
		1	2	3	4	8
tPD	Combinatorial Delay [5]	1.7	2.2	2.7	3.3	5.5
tSU	Setup Time [5]	2.1	2.1	2.1	2.1	2.1
tH	Hold Time	0.0	0.0	0.0	0.0	0.0
tCLK	Clock to Q Delay	1.0	1.5	1.9	2.7	4.9
tCWHI	Clock High Time	2.0	2.0	2.0	2.0	2.0
tCWLO	Clock Low Time	2.0	2.0	2.0	2.0	2.0
tSET	Set Delay	1.7	2.2	2.7	3.3	5.5
tRESET	Reset Delay	1.5	1.9	2.3	2.8	4.6
tSW	Set Width	1.9	1.9	1.9	1.9	1.9
tRW	Reset Width	1.8	1.8	1.8	1.8	1.8

Input Cells

Symbol	Parameter	Propagation Delays (ns) [4]						
		1	2	3	4	8	12	16
tIN	High Drive Input Delay [6]	3.1	3.2	3.3	3.4	4.4	5.8	6.5
tINI	High Drive Input, Inverting Delay [6]	3.3	3.4	3.5	3.6	4.6	6.0	6.7
tIO	Input Delay (bidirectional pad)	1.4	1.9	2.3	3.0	4.8	6.7	8.5
tGCK	Clock Buffer Delay [7]	2.7	2.8	2.9	3.0	3.1	3.3	3.4
tGCKHI	Clock Buffer Min High [7]	2.0	2.0	2.0	2.0	2.0	2.0	2.0
tGCKLO	Clock Buffer Min Low [7]	2.0	2.0	2.0	2.0	2.0	2.0	2.0

Output Cell

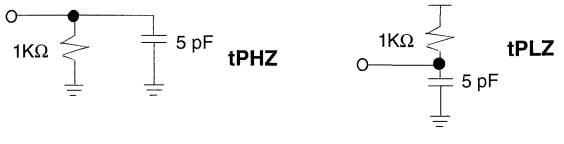
Symbol	Parameter	Propagation Delays (ns) Output Load Capacitance (pF) [4]				
		30	50	75	100	150
tOUTLH	Output Delay Low to High	2.7	3.3	3.8	4.3	5.4
tOUTHL	Output Delay High to Low	2.8	3.6	4.5	5.3	6.9
tPZH	Output Delay Tri-state to High	2.1	2.6	3.1	3.7	4.8
tPZL	Output Delay Tri-state to Low	2.6	3.3	4.1	4.9	6.5
tPHZ	Output Delay High to Tri-state [8]	2.9				
tPLZ	Output Delay Low to Tri-state [8]	3.3				

Notes:

[6] See High Drive Buffer Table for more information.

[7] Clock buffer fanout refers to the maximum number of flip flops per half column. The number of half columns used does not affect clock buffer delay.

[8] The following loads are used for tPXZ:



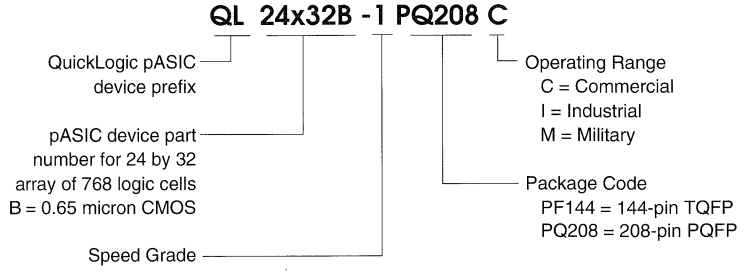
High Drive Buffer

Symbol	Parameter	# High Drives Wired Together	Propagation Delays (ns) Fanout [4]				
			12	24	48	72	96
tIN	High Drive Input Delay	1	5.8	7.2			
		2		5.0	7.1		
		3			5.8	6.7	7.7
		4				5.9	6.8
tINI	High Drive Input, Inverting Delay	1	6.0	7.4			
		2		5.2	7.3		
		3			6.0	6.9	7.9
		4				6.1	7.0

AC Performance

Propagation delays depend on routing, fanout, load capacitance, supply voltage, junction temperature, and process variation. The AC Characteristics are a design guide to provide initial timing estimates at nominal conditions. Worst case estimates are obtained when nominal propagation delays are multiplied by the appropriate Delay Factor, K, as specified in the Delay Factor table (Operating Range). The effects of voltage and temperature variation are illustrated in the graphs on page 2-55, K Factor versus Voltage and Temperature. The SpDE Toolkit incorporates data sheet AC Characteristics into the QDIF database for pre-place-and-route timing analysis. The SpDE Delay Modeler extracts specific timing parameters for precise path analysis or simulation results following place and route. For definition of timing parameters, see page 2-56, Timing Waveforms.

ORDERING INFORMATION



QL8x12BL
*Wild*Cat 1000L
Low Power 3.3 Volt Operation, 1K Gate FPGA

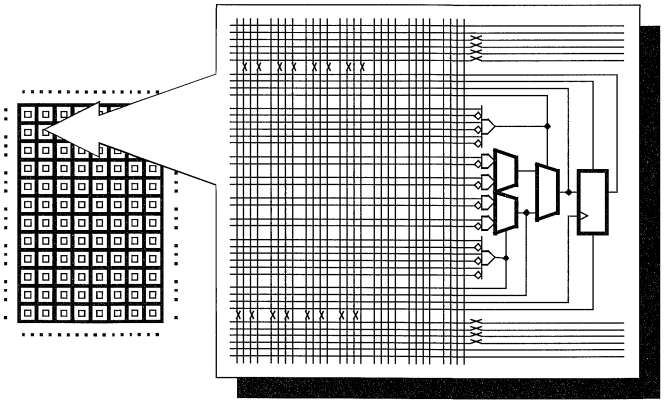
pASIC HIGHLIGHTS

...1000 usable gates, 64 I/O pins

- **High Speed** – ViaLink™ metal-to-metal programmable–via antifuse technology, allows counter speeds over 80 MHz at 3.3 Volt operation.
- **5V Tolerant I/Os** – Support interface to 5 Volt CMOS, NMOS and bipolar devices by sinking up to 12 mA (see IIH specification).
- **High Usable Density** – A 8-by-12 array of 96 logic cells provides 3,000 total available gates, with 1000 typically usable "gate array" gates in 44-pin and 68-pin PLCC packages, and in 100-pin TQFP packages.
- **Compatible with Standard 5.0V product** – The -"L" series is fully pin-out and function compatible with the High Speed 5.0V product. See QL8x12B for pin-out and AC Characteristics.
- **Low-Cost, Easy-to-Use Design Tools** – Designs entered and simulated using QuickLogic's new Quick*Works*™ development environment, or with third-party CAE tools including Viewlogic, Synopsys, Mentor, Cadence and Intergraph. Fast, fully automatic place and route on PC and workstation platforms using QuickLogic software.

QL8x12BL Block Diagram

96 Logic Cells



QUICKLOGIC

▪ = Up to 56 prog. I/O cells, 6 Input high-drive cells, 2 Input/Clk (high-drive) cells

ABSOLUTE MAXIMUM RATINGS

Supply Voltage ..... –0.5 to 7.0V
Input Voltage ..... –0.5 to VCC +0.7V
ESD Pad Protection ..... ±2000V
DC Input Current ..... ±20 mA
Latch-up Immunity ..... ±200 mA

Storage Temperature
Ceramic ..... –65°C to + 150°C
Plastic ..... –40°C to + 125°C
Lead Temperature ..... 300°C

OPERATING RANGE for 3.3 V

Symbol	Parameter		Commercial		Unit
			Min	Max	
VCC	Supply Voltage		3.0	3.6	V
TA	Ambient Temperature		0	70	°C
TC	Case Temperature				°C
K	Delay Factor	-0 Speed Grade	0.46	2.61	
		-1 Speed Grade	0.46	2.23	

DC CHARACTERISTICS over operating range

Symbol	Parameter	Conditions	Min	Max	Unit
VIH	Input HIGH Voltage		2.0		V
VIL	Input LOW Voltage			0.8	V
VOH	Output HIGH Voltage	IOH = -2.4 mA	2.4		V
		IOH = -10 µA	VCC-0.1		V
VOL	Output LOW Voltage	IOL = 4 mA		0.4	V
		IOL = 10 µA		0.1	V
IIH	Input HIGH Current Sink (for tolerance to 5V devices)	VCC+0.6V > VI > VCC		12	mA
II	Input Leakage Current	VI = VCC or GND	-10	10	µA
IOZ	3-State Output Leakage Current	VI = VCC or GND	-10	10	µA
CI	Input Capacitance [1]			10	pF
IOS	Output Short Circuit Current [2]	VO = GND	-5	-50	mA
		VO = VCC	15	100	mA
ICC	Supply Current [3]	VI, VIO = VCC or GND		650	µA

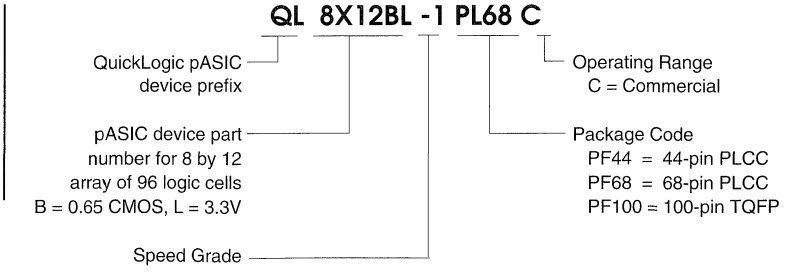
Notes:

[1] Capacitance is sample tested only. CI = 20 pF max on I/(SI).

[2] Only one output at a time. Duration should not exceed 30 seconds.

[3] For AC conditions use the formula described in the Data Book, Section 5 — Power vs Operating Frequency.

ORDERING INFORMATION



QL12x16BL
*Wild*Cat 2000L
Low Power 3.3 Volt Operation, 2K Gate FPGA

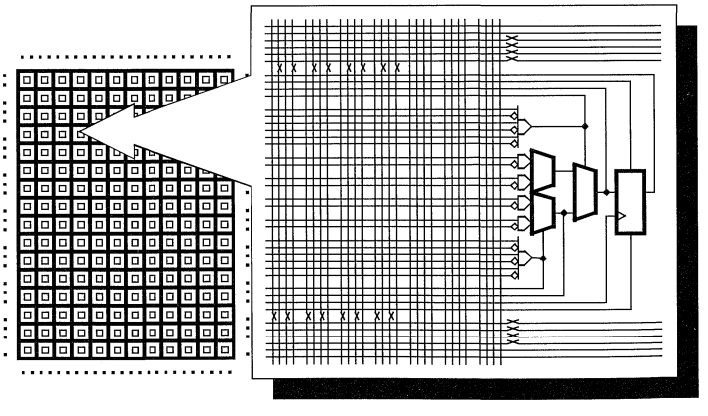
pASIC HIGHLIGHTS

...2000 usable gates, 88 I/O pins

- **High Speed** – ViaLink™ metal-to-metal programmable–via antifuse technology, allows counter speeds over 80 MHz at 3.3 Volt operation.
- **5V Tolerant I/Os** – Support interface to 5 Volt CMOS, NMOS and bipolar devices by sinking up to 12 mA (see IIH specification).
- **High Usable Density** – A 12-by-16 array of 192 logic cells provides 6,000 total available gates, with 2000 typically usable "gate array" gates in 68-pin and 84-pin PLCC packages, and in 100-pin TQFP packages.
- **Compatible with Standard 5.0V product** – The -"L" series is fully pin-out and function compatible with the High Speed 5.0V product. See QL12x16B for pin-out and AC Characteristics.
- **Low-Cost, Easy-to-Use Design Tools** – Designs entered and simulated using QuickLogic's new Quick*Works*™ development environment, or with third-party CAE tools including Viewlogic, Synopsys, Mentor, Cadence and Intergraph. Fast, fully automatic place and route on PC and workstation platforms using QuickLogic software.

QL12x16BL Block Diagram

192 Logic Cells



QUICKLOGIC

▪ = Up to 80 prog. I/O cells, 6 Input high-drive cells, 2 Input/Clk (high-drive) cells

ABSOLUTE MAXIMUM RATINGS

Supply Voltage ..... –0.5 to 7.0V
Input Voltage ..... –0.5 to VCC +0.7V
ESD Pad Protection ..... ±2000V
DC Input Current ..... ±20 mA
Latch-up Immunity ..... ±200 mA

Storage Temperature
Ceramic ..... –65°C to + 150°C
Plastic ..... –40°C to + 125°C
Lead Temperature ..... 300°C

OPERATING RANGE for 3.3 V

Symbol	Parameter		Commercial Min	Commercial Max	Unit
VCC	Supply Voltage		3.0	3.6	V
TA	Ambient Temperature		0	70	°C
TC	Case Temperature				°C
K	Delay Factor	-0 Speed Grade	0.46	2.61	
		-1 Speed Grade	0.46	2.23	

DC CHARACTERISTICS over operating range

Symbol	Parameter	Conditions	Min	Max	Unit
VIH	Input HIGH Voltage		2.0		V
VIL	Input LOW Voltage			0.8	V
VOH	Output HIGH Voltage	IOH = -2.4 mA	2.4		V
		IOH = -10 µA	VCC-0.1		V
VOL	Output LOW Voltage	IOL = 4 mA		0.4	V
		IOL = 10 µA		0.1	V
IIH	Input HIGH Current Sink (for tolerance to 5V devices)	VCC+0.6V > VI > VCC		12	mA
II	Input Leakage Current	VI = VCC or GND	-10	10	µA
IOZ	3-State Output Leakage Current	VI = VCC or GND	-10	10	µA
CI	Input Capacitance [1]			10	pF
IOS	Output Short Circuit Current [2]	VO = GND	-5	-50	mA
		VO = VCC	15	100	mA
ICC	Supply Current [3]	VI, VIO = VCC or GND		650	µA

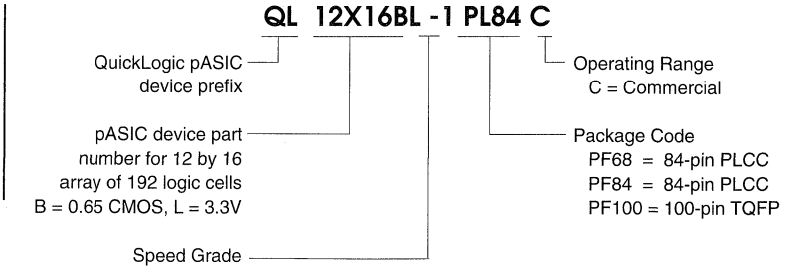
Notes:

[1] Capacitance is sample tested only. CI = 40 pF max on I/(SI) and I/(P).

[2] Only one output at a time. Duration should not exceed 30 seconds.

[3] For AC conditions use the formula described in the Data Book, Section 5 — Power vs Operating Frequency.

ORDERING INFORMATION



QL16x24BL
*Wild*Cat 4000L
Low Power 3.3 Volt Operation, 4K Gate FPGA

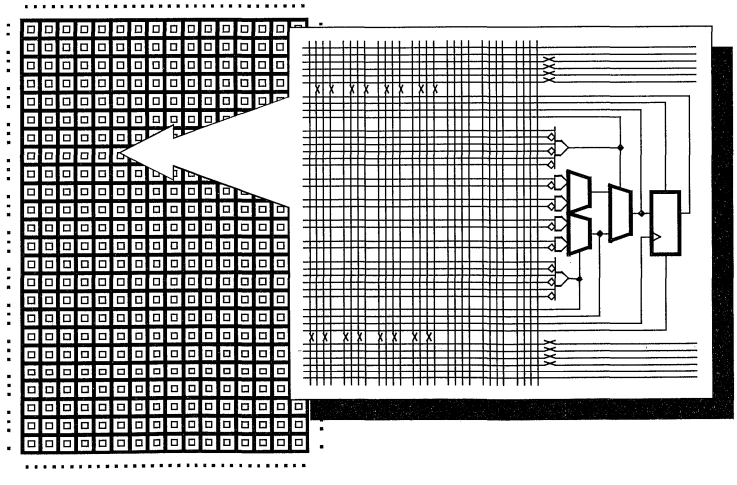
pASIC HIGHLIGHTS

...4000 usable gates, 122 I/Opins

- **High Speed** – ViaLink™ metal-to-metal programmable–via antifuse technology, allows counter speeds over 80 MHz at 3.3 Volt operation.
- **5V Tolerant I/Os** – Support interface to 5 Volt CMOS, NMOS and bipolar devices by sinking up to 12 mA (see IIH specification).
- **High Usable Density** – A 16-by-24 array of 384 logic cells provides 12,000 total available gates, with 4000 typically usable "gate array" gates in 84-pin PLCC packages, and in 100-pin and 144-pin TQFP packages.
- **Compatible with Standard 5.0V product** – The -"L" series is fully pin-out and function compatible with the High Speed 5.0V product. See QL16x24B for pin-out and AC Characteristics.
- **Low-Cost, Easy-to-Use Design Tools** – Designs entered and simulated using QuickLogic's new Quick*Works*™ development environment, or with third-party CAE tools including Viewlogic, Synopsys, Mentor, Cadence and Intergraph. Fast, fully automatic place and route on PC and workstation platforms using QuickLogic software.

QL16x24BL Block Diagram

384Logic Cells



▪ = Up to 114 prog. I/O cells, 6 Input high-drive cells, 2 Input/Clk (high-drive) cells

QUICKLOGIC

ABSOLUTE MAXIMUM RATINGS

Supply Voltage ..... –0.5 to 7.0V
Input Voltage ..... –0.5 to VCC +0.7V
ESD Pad Protection ..... ±2000V
DC Input Current ..... ±20 mA
Latch-up Immunity ..... ±200 mA

Storage Temperature
Ceramic ..... –65°C to + 150°C
Plastic ..... –40°C to + 125°C
Lead Temperature ..... 300°C

OPERATING RANGE for 3.3 V

Symbol	Parameter		Commercial		Unit
			Min	Max	
VCC	Supply Voltage		3.0	3.6	V
TA	Ambient Temperature		0	70	°C
TC	Case Temperature				°C
K	Delay Factor	-0 Speed Grade	0.46	2.61	
		-1 Speed Grade	0.46	2.23	

DC CHARACTERISTICS over operating range

Symbol	Parameter	Conditions	Min	Max	Unit
VIH	Input HIGH Voltage		2.0		V
VIL	Input LOW Voltage			0.8	V
VOH	Output HIGH Voltage	IOH = -2.4 mA	2.4		V
		IOH = -10 µA	VCC-0.1		V
VOL	Output LOW Voltage	IOL = 4 mA		0.4	V
		IOL = 10 µA		0.1	V
IIH	Input HIGH Current Sink (for tolerance to 5V devices)	VCC+0.6V > VI > VCC		12	mA
II	Input Leakage Current	VI = VCC or GND	-10	10	µA
IOZ	3-State Output Leakage Current	VI = VCC or GND	-10	10	µA
CI	Input Capacitance [1]			10	pF
IOS	Output Short Circuit Current [2]	VO = GND	-5	-50	mA
		VO = VCC	15	100	mA
ICC	Supply Current [3]	VI, VIO = VCC or GND		650	µA

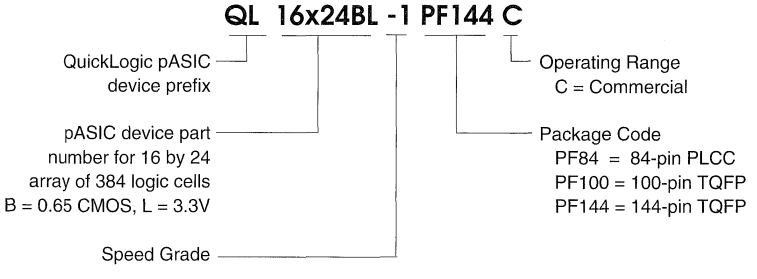
Notes:

[1] Capacitance is sample tested only. CI = 45 pF max on I/(SI) and I/(P).

[2] Only one output at a time. Duration should not exceed 30 seconds.

[3] For AC conditions use the formula described in the Data Book, Section 5 — Power vs Operating Frequency.

ORDERING INFORMATION



QL24x32BL
*Wild*Cat 8000L
Low Power 3.3 Volt Operation, 8K Gate FPGA

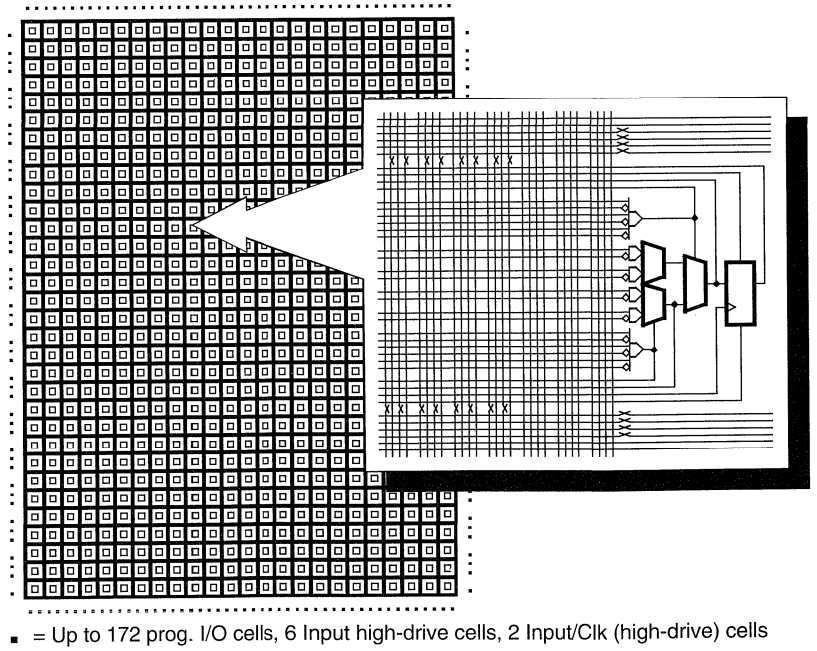
pASIC HIGHLIGHTS

...8000 usable gates, 180 I/O pins

- **High Speed** – ViaLink™ metal-to-metal programmable–via antifuse technology, allows counter speeds over 80 MHz at 3.3 Volt operation.
- **5V Tolerant I/Os** – Support interface to 5 Volt CMOS, NMOS and bipolar devices by sinking up to 12 mA (see IIH specification).
- **High Usable Density** – A 24-by-32 array of 768 logic cells provides 24,000 total available gates, with 8000 typically usable "gate array" gates in 144-pin TQFP packages, and 208-pin metric PQFP packages.
- **Compatible with Standard 5.0V product** – The -"L" series is fully pin-out and function compatible with the High Speed 5.0V product. See QL24x32B for pin-out and AC Characteristics.
- **Low-Cost, Easy-to-Use Design Tools** – Designs entered and simulated using QuickLogic's new Quick*Works*™ development environment, or with third-party CAE tools including Viewlogic, Synopsys, Mentor, Cadence and Intergraph. Fast, fully automatic place and route on PC and workstation platforms using QuickLogic software.

QL24x32BL Block Diagram

768 Logic Cells



■ = Up to 172 prog. I/O cells, 6 Input high-drive cells, 2 Input/Clk (high-drive) cells

QUICKLOGIC

ABSOLUTE MAXIMUM RATINGS

Supply Voltage ..... –0.5 to 7.0V
Input Voltage ..... –0.5 to VCC +0.7V
ESD Pad Protection ..... ±2000V
DC Input Current ..... ±20 mA
Latch-up Immunity ..... ±200 mA

Storage Temperature
Ceramic ..... –65°C to + 150°C
Plastic ..... –40°C to + 125°C
Lead Temperature ..... 300°C

OPERATING RANGE for 3.3 V

Symbol	Parameter		Commercial Min	Commercial Max	Unit
VCC	Supply Voltage		3.0	3.6	V
TA	Ambient Temperature		0	70	°C
TC	Case Temperature				°C
K	Delay Factor	-0 Speed Grade	0.46	2.61	
		-1 Speed Grade	0.46	2.23	

DC CHARACTERISTICS over operating range

Symbol	Parameter	Conditions	Min	Max	Unit
VIH	Input HIGH Voltage		2.0		V
VIL	Input LOW Voltage			0.8	V
VOH	Output HIGH Voltage	IOH = -2.4 mA	2.4		V
		IOH = -10 µA	VCC-0.1		V
VOL	Output LOW Voltage	IOL = 4 mA		0.4	V
		IOL = 10 µA		0.1	V
IIH	Input HIGH Current Sink (for tolerance to 5V devices)	VCC+0.6V > VI > VCC		12	mA
II	Input Leakage Current	VI = VCC or GND	-10	10	µA
IOZ	3-State Output Leakage Current	VI = VCC or GND	-10	10	µA
CI	Input Capacitance [1]			10	pF
IOS	Output Short Circuit Current [2]	VO = GND	-5	-50	mA
		VO = VCC	15	100	mA
ICC	Supply Current [3]	VI, VIO = VCC or GND		650	µA

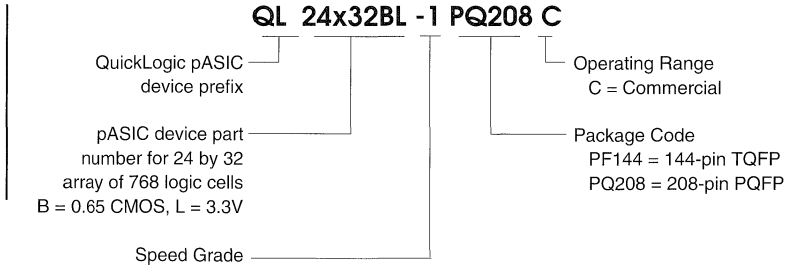
Notes:

[1] Capacitance is sample tested only.

[2] Only one output at a time. Duration should not exceed 30 seconds.

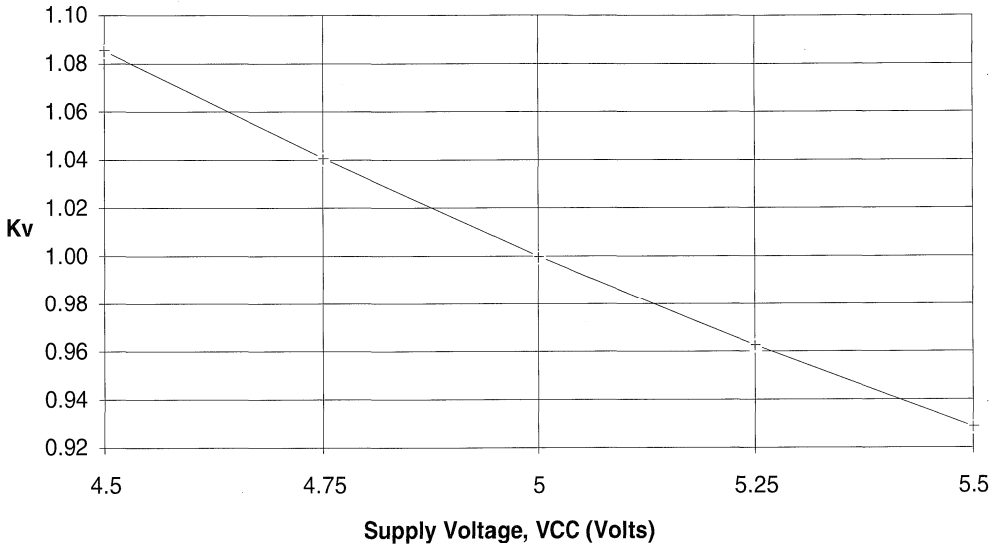
[3] For AC conditions use the formula described in the Data Book, Section 5 — Power vs Operating Frequency.

ORDERING INFORMATION

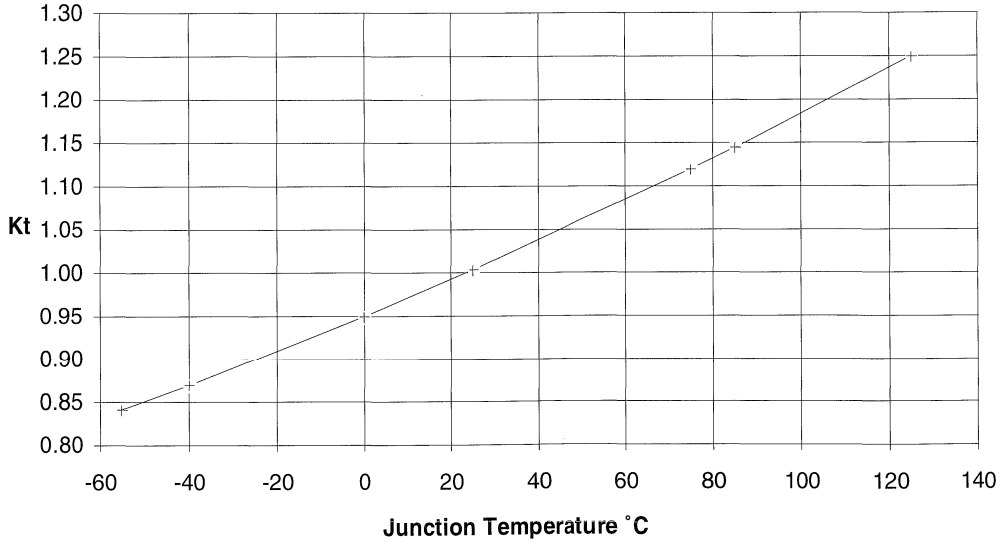


K Factor versus Voltage and Temperature

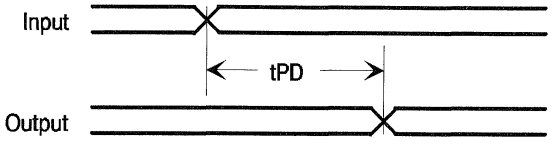
Kv, Voltage Factor versus Vcc, Supply Voltage



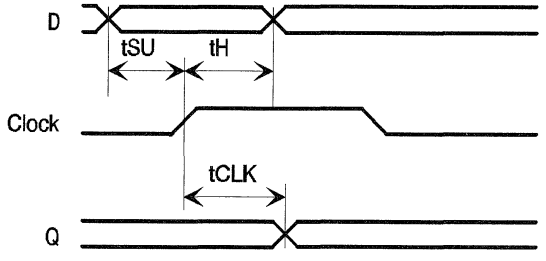
Kt, Temperature Factor versus Temperature



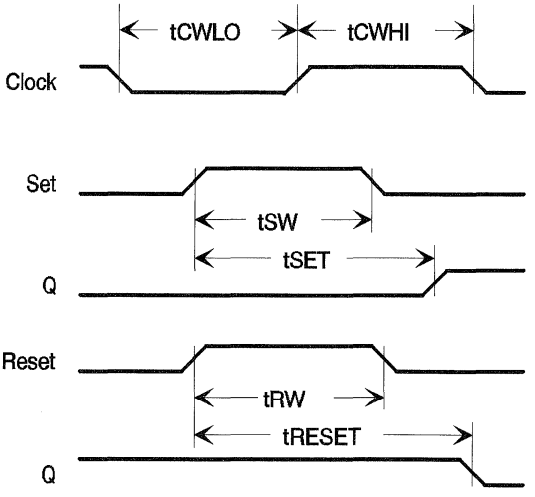
Timing Waveforms



Combinatorial Delay



Setup and Hold Times



Set and Reset Delays

pASIC DEVELOPMENT SOLUTIONS
The Bridge to Connect Your Design Entry Solutions

HIGHLIGHTS

- **"Plugs" into your existing design environment** – it is not necessary to learn a new set of tools.
- **Bridges the gap between design entry and simulation** by accepting input from many third party schematic entry and synthesis tools and generating guaranteed simulation netlists for many of todays high-performance simulation environments.
- **pASIC™ architecture-specific place and route algorithms** to optimize speed and density of the QuickLogic pASIC devices.
- **Extensive logic optimization capabilities** so it is not necessary to become an expert on the QuickLogic architecture.
- **Fast and efficient tools** which can compile fully utilized designs in minutes.

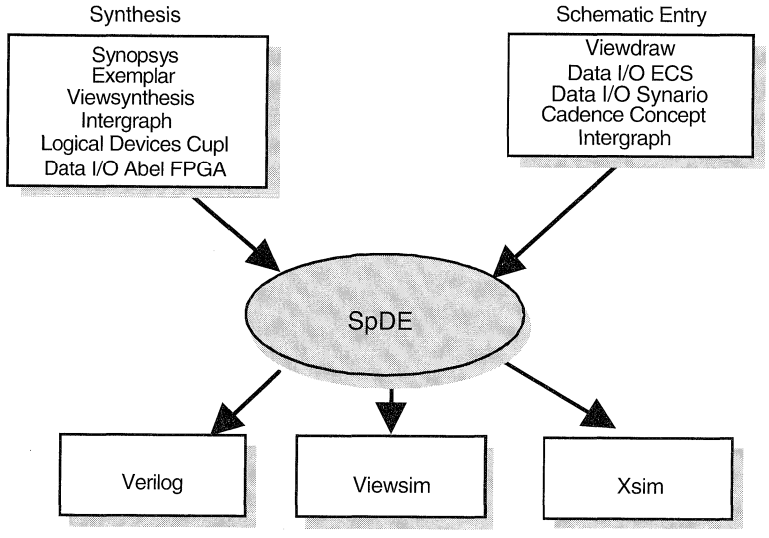
INTRODUCTION

The QuickLogic Seamless pASIC Design Environment (SpDE—pronounced "Speedy") is the most flexible design environment available today. No longer is it necessary to learn a new set of tools to begin working with an FPGA. SpDE provides all the support you need to enter designs using industry standard CAE tools while providing full timing simulation to many environments. This flexability allows the designer to use the "best-of-the-breed" in design entry environments with the SpDE toolkit providing the bridge which makes it all work together.

Using the industry standard EDIF 2 0 0 interchange format enables the designer to unleash the power of QuickLogic FPGA's without the time consuming task to learn new design entry tools. QuickLogic is dedicated to providing third-party plug-in solutions in order to work in your environment, not force you to work in the vendors environment.

QUICKLOGIC

FIGURE 1 SpDE Toolkit is the bridge for your design needs



SpDE Toolkit Software and Hardware Tools

The SpDE Toolkit includes the following software and hardware tools:

- Support for EDIF 2 0 0
- Automatic simulation generation for a variety of todays most popular simulation environments.
- Technology Mapping for efficient design compilation into the pASIC1 architecture
- Automatic placement and routing (APR)
- Static timing path analysis
- Physical layout viewer
- Post-layout delay extraction and back-annotation to a variety of third-party simulators
- Automatic test vector generation (ATVG)
- Device programmer/tester

All applications are tightly integrated within a high-performance design framework. The SpDE toolkit is the most modern and powerful development system available for designing FPGA devices today.

QT/QS-SpDE-PC/SUN(-A)
SpDE Tools
Seamless pASIC Design Environment

HIGHLIGHTS

- **Open Interface to third party tools** – SpDE (pronounced *"speedy"*) supports the EDIF input format for interfacing with many supported design entry tools including Cadence and Synopsys. The QDIF format (QuickLogic Design Interchange Format) is also supported by a large number of third-party vendors including Data I/O, Exemplar Logic, and Viewlogic.
- **Simulation Support** for Verilog, Viewlogic's Viewsim, X-Sim, and the assortment of simulators supported by Logic Modeling's SmartModels.
- **Sophisticated Timing Analysis with the Path Analyzer** – View only the delays you are interested in using a spreadsheet format. If a delay is a little too long, type in the required delay and SpDE will re-run to meet your timing.
- **Accurate Delay Modeling for Guaranteed Timing** – The SpDE Delay Modeler incorporates the state-of-the art Asymptotic Waveform Evaluation (AWE) technique to generate precise timing estimates for all post-layout delays. The Delay Modeler provides the same accuracy as SPICE, at run times that are orders of magnitude faster than SPICE.
- **Fast Optimization, Placement, and Routing tools** – Complete even the most complicated designs in no time. Our tools will automatically place and route a design even if 100% of the logic and I/O resources are used!
- **Interactive Windows Interface** allows communication between different applications for the ideal debugging environment. The X-Sim simulator will send signal values back to the ECS Schematic, and SpDE's Path Analyzer will highlight paths in both SpDE's chip view as well as in the ECS Schematic.

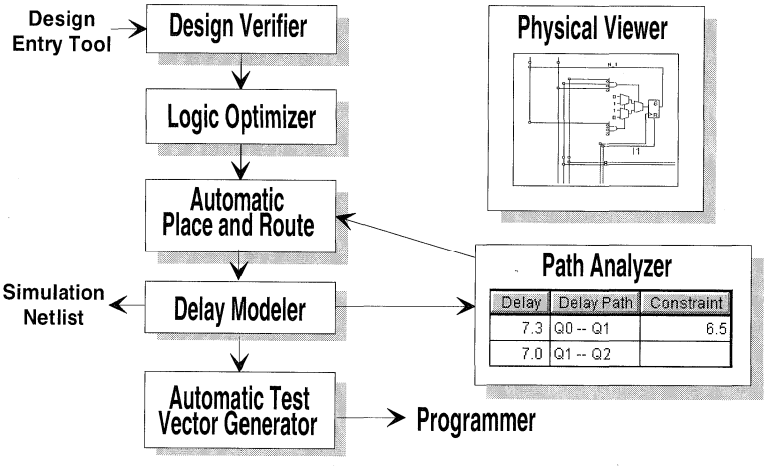
SpDE stands for the Seamless pASIC Design Environment. A design entered in VHDL, Verilog, ABEL, schematics, or a variety of other forms can be loaded into SpDE for optimization, placement, routing, timing analysis, and back annotation into any supported simulation netlist (including Verilog). Finally, the QuickLogic Designer Programmer or another supported third-party programmer can be used to program a QuickLogic pASIC device.

QUICKLOGIC

SpDE tools are available as a complete design kit including a schematic and Boolean design environment with simulation, or as a standalone package for integration into the user's own design environment. This datasheet describes the features of the standalone SpDE product.

SpDE is a collection of design compilation and design analysis tools. Figure 1 illustrates the design process using the SpDE tools.

FIGURE 1 QuickLogic's SpDE tools



The FPGA design process needs to be fast and flexible to support any sort of design constraint. For this reason, FPGA development tools need to be sophisticated, yet very easy-to-use. To this end, QuickLogic has created SpDE as an interactive application, using standard graphical user interfaces such as Microsoft Windows on a PC. SpDE is also available on SUN workstations in the Motif and OpenWindows environments. This allows modern third-party design entry tools to interface seamlessly with QuickLogic SpDE software.

SpDE can accept design netlists from a large number of third party tools. These design tools communicate with SpDE using either the QDIF (QuickLogic Design Interchange Format) or the industry standard EDIF format. These two netlist formats allow compatibility with many third-party tools: Synopsys, Data I/O's ABEL, Exemplar Logic's CORE, Cadence Concept, Data I/O's ECS Design Entry, Data I/O's Synario, Viewlogic's Workview and Powerview design tools, and others.

Once a design has been imported, SpDE performs all necessary verification, optimization, placement, routing, and delay modeling using a complete set of quality tools. The SpDE tools, as previously illustrated in Figure 1, include the following:

- Design Verifier
- Logic Optimization
- Automatic Place and Route
- Delay Modeler
- Path Analyzer
- Automatic Test Vector Generator

Design Verifier

The Design Verifier will carefully analyze each design imported into SpDE. Any design issues or problems that are detected are reported to the user. The Design Verifier detects improperly wired designs, high fanout problems, unused logic, improper pin placement, and many other possible design problems. If ECS design entry is being used, then simply clicking on the Design Verifier's error message will highlight the appropriate portion of the schematic design.

Logic Optimization

SpDE's Logic Optimizer will carefully examine the logic in the design to determine how to map the logic into QuickLogic logic cells. The Level 1 Optimizer, also known as the Technology Mapper, provides the most area-efficient optimization possible. However, for a more predictable optimization algorithm that will leave all nets in the design intact, the Level 0 Optimizer (the Packer) may be chosen.

Automatic Placement and Routing

The Placer uses internally optimized algorithms to arrange the locations of the logic cells and I/O pins on the device for maximum performance. Alternatively, the user can fix the placement of I/O pins and logic cells in the design description if desired. The Placer may also be run in Timing-Driven mode in order to meet timing constraints entered in the Path Analyzer.

The Router is a fully automatic tool. The QuickLogic architecture combines ample routing resources, a simple routing architecture, and a very fast interconnect element. This combination makes the automatic routing of a design possible- even when 100% of the logic cells and I/O pads are used. The Router completes successfully even on designs with all the pins fixed early in the design process.

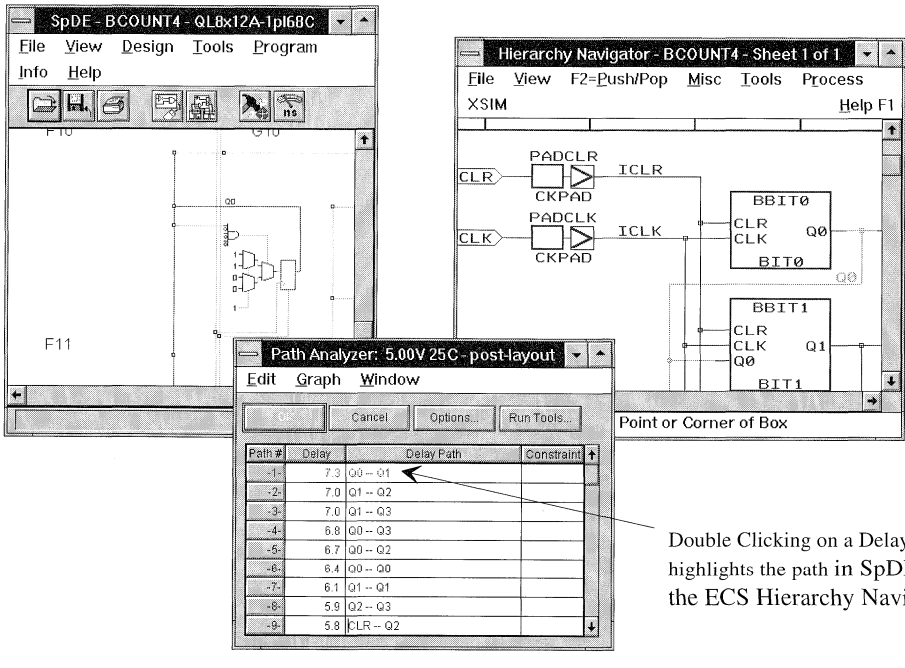
Delay Modeler

SpDE's Delay Modeler calculates delays throughout the device after placement and routing is complete. The Delay Modeler achieves the accuracy of SPICE in orders of magnitude less time with a modern analysis technique called AWE (Asymptotic Waveform Evaluation). The Delay Modeler uses the results of a sequencing algorithm that determines which Via-Links will be programmed and in what order. These accurate, guaranteed delays are used in the Path Analyzer, and when creating a timing netlist for simulation.

Path Analyzer

The Path Analyzer used the results from the Delay Modeler to print the static path delays throughout the chip. The user may select groups of paths (e.g., register to register) or specific paths for viewing. The path delays appear in a spreadsheet format in the Path Analyzer window, allowing for simple analysis. Figure 2 demonstrates how double-clicking on a delay path in the Path Analyzer will highlight the path in SpDE's Physical Viewer and in a supported design entry tool (Data I/O's ECS). Delays may be also be graphed with the Path Analyzer's graphing tool or copied to the clipboard for use in other applications. If the path delays do not meet timing constraints, then the user may specify the delay required; SpDE will re-run the place and route tools to meet the timing constraint.

FIGURE 2
Highlighting with the Path Analyzer



Double Clicking on a Delay Path highlights the path in SpDE and in the ECS Hierarchy Navigator

Physical Viewer

SpDE's Physical Viewer shows the user how the automatic place and route tools have mapped a logic design into the device's resources. It is not a manual editor, but an analysis tool. The designer can inspect the results of optimization, placement, and routing by using the Path Analyzer and net highlighting features within SpDE.

When SpDE is coupled with a compatible design entry tool like ECS from Data I/O, the user can click on nets in SpDE's Physical Viewer and the corresponding net will highlight in the ECS Schematic. In the same manner, when a net is selected in the ECS schematic, the corresponding net in the Physical Viewer will be highlighted. If the Path Analyzer is being used, selecting a path will highlight the path in both the Physical Viewer and the ECS schematic (Figure 2).

Automatic Test Vector Generator

SpDE's ATVG tool will generate test vectors for the design using an internal register scan path. These vectors may be exercised on the device using QuickLogic's Designer Programmer.

Once the design cycle is complete, a QuickLogic pASIC device may be programmed on QuickLogic's Designer Programmer or a supported third-party programmer. SpDE contains a programming interface that is used when programming on the QuickLogic Designer Programmer. If a third party programmer is used, SpDE will create the appropriate programming file.

QuickLogic Designer Programmer

The QuickLogic Designer Programmer is available from QuickLogic either separately or in a variety of package configurations with SpDE and third-party design tools and/or libraries. The Designer Programmer is capable of programming all QuickLogic devices in any package configuration. The programmer can program 84 pin PLCC's and 68 pin PLCC's with the included adapters, and optional adapters can be purchased to program any other supported package type. In addition, the Design Programmer supports a "gang programming" mode, where up to eight programmers can be daisy-chained together with included serial cables. Gang programming allows up to eight devices to be programmed at the same time.

QT/QS-QTL-50-PC/SUN/HP(-A)
QuickTools Toolkit Package
Bundles SpDE Tools and QuickUtilities

HIGHLIGHTS

- **Cost effective bundle** of QuickLogic's SpDE tools and the new QuickUtilities package.
- **Multi-Platform support** - Available for PC, SUN and HP workstation platforms

SpDE Tools

Complete SpDE toolkit as described on pages 3-3 through 3-7, including support for the new *Wild*Cat8000 device.

QuickUtilities

A Utilities package which works seamlessly with QuickLogic's SpDE tools, and includes the following:

- A Logic Re-Optimizer which improves logic utilization by an average of 12% over SpDE tools alone
- Automatic optimal Buffer Generation to improve performance of high fan-out signal paths
- Third party simulator support for Intergraph, Simucad, SusieCAD, and QuickSim
- LPM Reader to accept LPM output from third party tools (Q2/95)

Ordering Information

QS – QTL - 50 – PC – A

QuickLogic Toolkit with:
QT = Both Software & Progmr.
QS = Software Only

Product Designator
QTL = QuickTools Toolkit

Programmer Power Supply Type:
A = USA
B = Europe
C = U.K.

Host Computer
PC = Windows-based PC
SUN = Sun Workstation
HP = Hewlett-Packard

Version Number
Subject to change

QUICKLOGIC

QT/QS-QWK-50-PC(-A)
QuickWorks Toolkit
Complete Design Entry and Simulation Solution

HIGHLIGHTS

- **Microsoft Windows Interface** gives the user an industry standard graphical interface that interactsseamlessly with QuickLogic's SpDE tools and provides a simple and efficient working environment.
- **Optimal Synthesis for Verilog and VHDL** HDLs can be mixed with the SCS Schematic Entry blocks at any level of the design hierarchy.
- **Silos III Verilog Simulator** for both functional and gate level timing simulation. Entry and display can be either textual or graphic waveform.
- **Productivity Focus** extends to the HDL editor which includes context sensitive templates and automatic color coding.
- **Interactive Cross-Probing** - Clicking on nets in the SCS Schematic will highlight the corresponding net in SpDE's Physical Viewer (and vice-versa!). SpDE's Path Analyzer will highlight paths in the SCS Schematic. Silos III Simulation values are updated on the SCS schematic as you click on different points of time in a previous simulation run. SpDE back annotates post-route timing information for Silos III.

INTRODUCTION

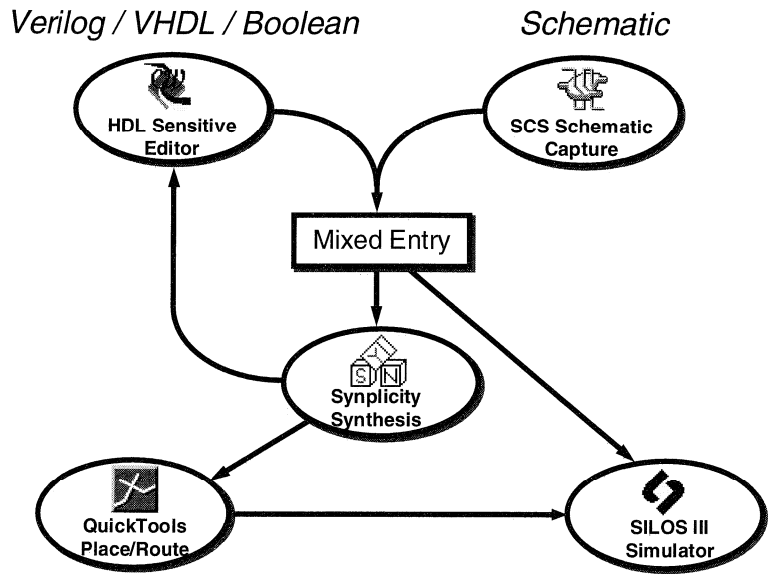
Introducing the next generation in FPGA Design Tools - Quick*Works*. For years QuickLogic has provided an integrated, easy to use toolset to guide designers through their FPGA designs with maximum speed and flexibility. We have continued to live up to this commitment.

The QuickWorks toolkit contains a suite of tools that is unprecedented in today's market, with prominent features such as:

- Verilog® / VHDL / Open ABEL Synthesis
- Context Sensitive HDL Editing
- Color Coded HDL Template Expansion
- Mixed-Mode Design Entry
- Enhanced Schematic Capture
- Comprehensive Verilog Simulation
- Interactive Cross-Probing Between Tools

QUICKLOGIC

FIGURE 1
QuickWorks Design Flow with SpDE



QuickWorks

Design Flow

As illustrated in the Quick*Works* design flow diagram, designers can mix a combination of schematic, boolean, and/or Verilog / VHDL using the Synario Capture System design entry tool. Symbols within the schematic editor can reference boolean files or Verilog files allowing a mixture of schematic design and HDL design. The designer can then simulate functionally (zero-delay) or produce a Verilog netlist, synthesize, place, and route the design, and then simulate with full timing information. The static timing analyzer, timing driven placement and other features (design optimizer and compiler) are still available to aid in the design process.

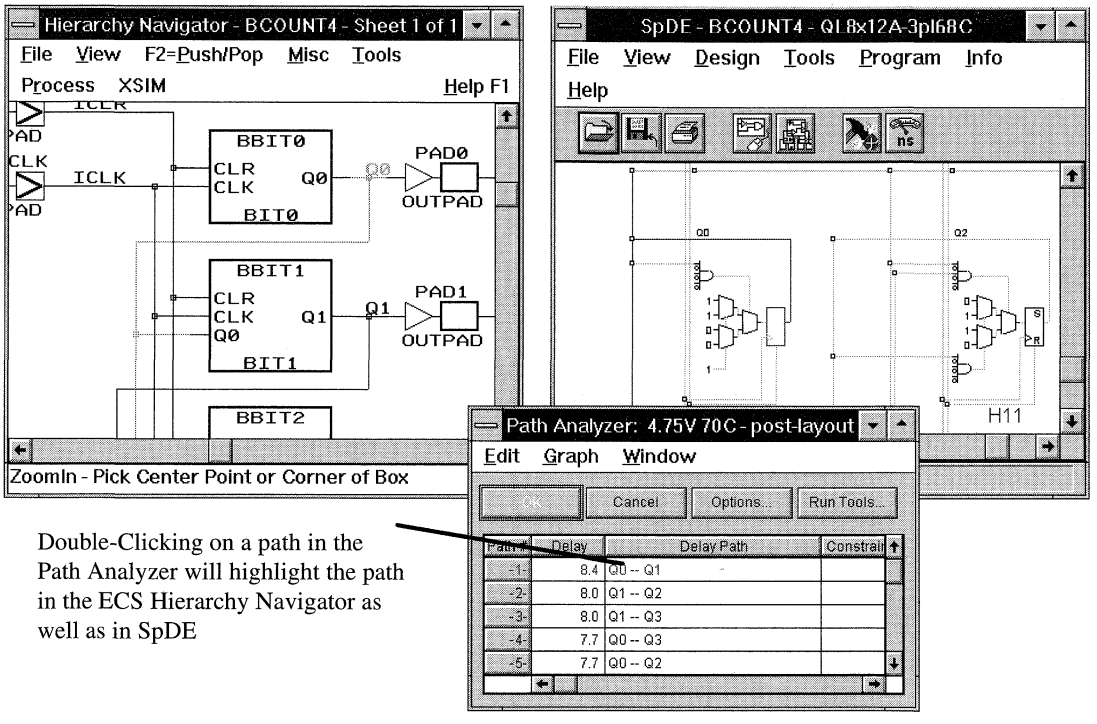
SCHEMATIC CAPTURE

Product Features

Synario Capture System (SCS™)

The SCS Waveform Editor from Data I/O is an easy-to-use Windows schematic capture utility. With the same touch, feel, and functions of the ECS schematic capture we've offered before, SCS offers a wide breadth of enhancements from waveform entry to back-annotation. Mixed mode entry (schematic, boolean, and/or Verilog/VHDL) is supported, providing a variety of entry methods. In addition, an updated Waveform Editor is included for graphical simulation waveform entry, and will export Verilog test-fixtures for simulation.

FIGURE 2
Interactive debugging using SpDE's Path Analyzer with the SCS Hierarchy Navigator



Double-Clicking on a path in the Path Analyzer will highlight the path in the ECS Hierarchy Navigator as well as in SpDE

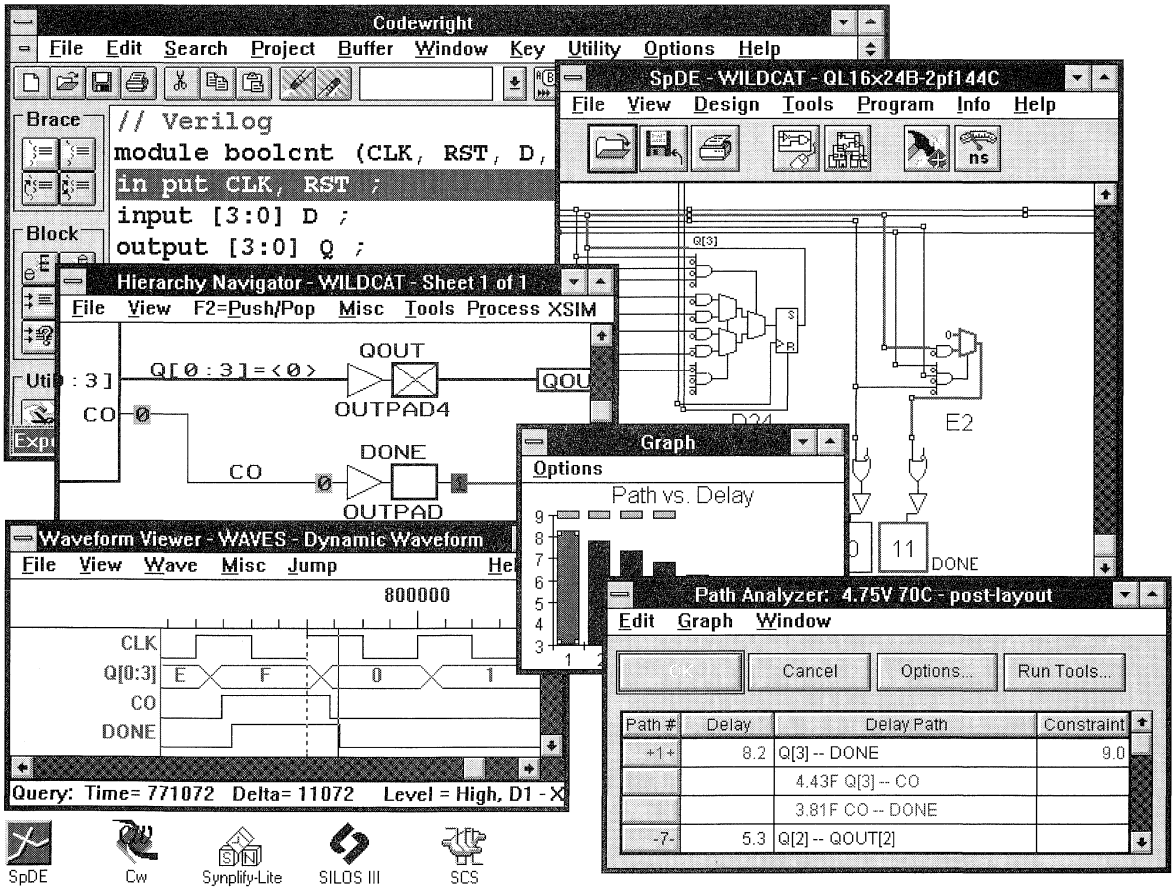
QuickTools 5.0

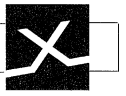
A bundle of SpDE 5.0 and a new utilities package (QuickUtilities). SpDE 5.0 now offers preliminary design support for the new WildCat™ 8000(QL24X32). Improved simulator timing models and faster programming algorithms for all existing devices have been implemented. Programming times have decreased by up to 66%. QuickUtilities provides a new logic re-optimizer which improves logic density by apporximately 12%, and an automatic optimal buffer generator for improving high fan-out signal path performance. The utilities package also includes support for several third-party simulators, such as Intergraph, SusieCAD, and QuickSim. It will include an LPM reader mid 1995.

QUICKBOOLEAN

QuickBoolean is an equation syntax similar to PALASM. The user creates the QuickBoolean file using any text editor, and then creates a symbol for the block using the ECS automatic symbol generator. This QuickBoolean symbol can then be added into an ECS Schematic for incorporation into a design. QuickBoolean equations can be used to describe portions of logic that are more suitable to textual description, such as one-hot state machines, decoding logic, or a set of and-or equations (e.g., PAL equations). When in the Hierarchy Navigator, PUSHing into a QuickBoolean block will open a text editor with the QuickBoolean equation file (Figure 3).

FIGURE 3
Seamless Integration Allows Cross-Probing Between Tools





Synplify-Lite™

SYNTHESIS

A highly integrated HDL synthesis tool. Honestly said, Synplify-Lite is "Simply Better Synthesis™", providing:

a) Quality results - better performance and density implementations than other vendors

b) Ease-of-Use - no vendor specific directives required, just OVI compliant Verilog,

c) Speed - fastest run time of any synthesizer on the market - a 4K design in minutes

d) VHDL and OPEN-ABEL Synthesis added during 1995

TurboWriter™ Editor

TEXT ENTRY

The TurboWriter editor offers a host of features - from emulation of a variety of text editors to context sensitive and color coded entry. Common editors such as vi, Epsilon, or others can be emulated. Context sensitive editing provides a variety of synthesis templates for language constructs, keyword-based template entry and syntax generation. Automatic color coding aids in easy readability and syntax verification. In addition, automatic generation of test-benches aids in Verilog or VHDL simulation.

Silos III™ Verilog Simulator

SIMULATION

Silos III provides the Verilog simulation power demanded by designers. Rated as one of the top simulators by an industry leading magazine, this OVI-compliant simulator is both fast and easy-to-use. Silos III supports graphical waveform entry as well as Verilog test benches. Designers who have previously used QuickLogic's Waveform Editor to enter stimulus can use the same files for Verilog simulation.

QuickWorks requires a minimum of 8 MB of RAM, and a 16 MB Permanent Swap File in Windows. Approximately 30 MB of disk space is necessary for a full installation. The Quick*Works* Design package with QuickLogic Libraries, QuickBoolean, TurboWrite, Synplify-Lite, Synario Capture System and the Silos III Verilog Simulator is available from QuickLogic in two packages.

ORDERING INFORMATION

QuickLogic Product ID	Product Description
QT-QWK-50-PC	Complete Package including SpDE, QuickBoolean, TurboWrite, Synplify-Lite, Synario Capture System and the Silos III Verilog Simulator and a QuickLogic Designer Programmer.
QS-QWK-50-PC	Software-only package with SpDE, QuickBoolean, TurboWrite, Synplify-Lite, Synario Capture System and the Silos III Verilog Simulator.

QS-MEN-SUN/HP
QuickLogic pASIC 1 Family
Mentor "Design Architect/Quicksim II" Libraries

HIGHLIGHTS

- **Design QuickLogic pASIC 1 FPGAs with Design Architect** Schematic Capture (V8.2X) on the Sun & HP platforms - enabling a complete design methodology in the Mentor Graphics environment.
- **Seamless Interface** to a QuickLogic pASIC toolkit through the EDIF 2 0 0 format.
- **Functional and full-timing simulation using QuickSim II.**

INTRODUCTION

Current users of Mentor Graphics Design Architect & QuickSim II tools can now design with the QuickLogic pASIC I family of FPGAs using the QuickLogic libraries and simulation interface.

SUPPORT

The Mentor Graphics Design Architect macro library and interface package provides the library symbols and the interface software to tightly integrate the QuickLogic and Mentor environments. Designs can be entered using the symbol libraries, placed and routed using the QuickLogic SpDE toolkit and simulated using the QuickSim II simulator.

DESIGN FLOW

Design entry is accomplished using the QuickLogic macro library. After the design is captured, behavioral simulation can be performed using QuickSim II. Once simulated, the design is exported to the SpDE tools by generating an EDIF file.

The file is then imported into SpDE to perform the following tasks - verification, logic optimization, placement, routing and timing analysis. The QuickLogic tools also generate a netlist in EDIF format with timing for post-layout simulation with QuickSim II. The designer can then program the device using the QuickLogic Programmer.

CONFIGURATION

This kit (QS-MEN-SUN/HP) is intended for designers who wish to design with the fastest FPGAs using the Mentor Graphics Design Architect (Version 8.25 or higher) schematic capture and the QuickSim II simulator in conjunction with the QuickLogic QuickTools toolkit.

QUICKLOGIC

QS-VL-PC/SUN
QuickLogic pASIC 1 Family
Viewlogic Macrolibrary & Interface

HIGHLIGHTS

- Design QuickLogic pASIC 1 FPGAs with Viewlogic.
- Completely integrated on all Viewlogic Platforms (PC or SUN) — Workview, Workview Pro, Workview Plus, and Powerview.
- Support for Viewdraw, Viewsynthesis, and ViewPLD enabling mixed mode entry and a complete high-level design methodology.
- Precise delay models for Viewsim using Asymptotic Waveform Evaluation (AWE) techniques.
- Back annotated logic values from Viewsim to the Viewdraw schematics. Makes analysis and debugging easy and simple.
- Seamless interface to the QuickLogic pASIC toolkit. Navigate from one environment to the other with a simple push of a button.

INTRODUCTION

Current users of the full Viewlogic tools, no matter what package or what platform, can design for the fastest FPGAs around — QuickLogic's pASICs. With QuickLogic's Viewlogic macro library and interface package, designs can be done on Workview, Workview Pro, Workview Plus, or Powerview with a variety of entry methods — schematics, VHDL or ABEL. The choice is yours. We just make it easy for you. Once the design is done, just enter the QuickLogic Seamless pASIC Design Environment (SpDE — pronounced "speedy") to automatically place and route your designs.

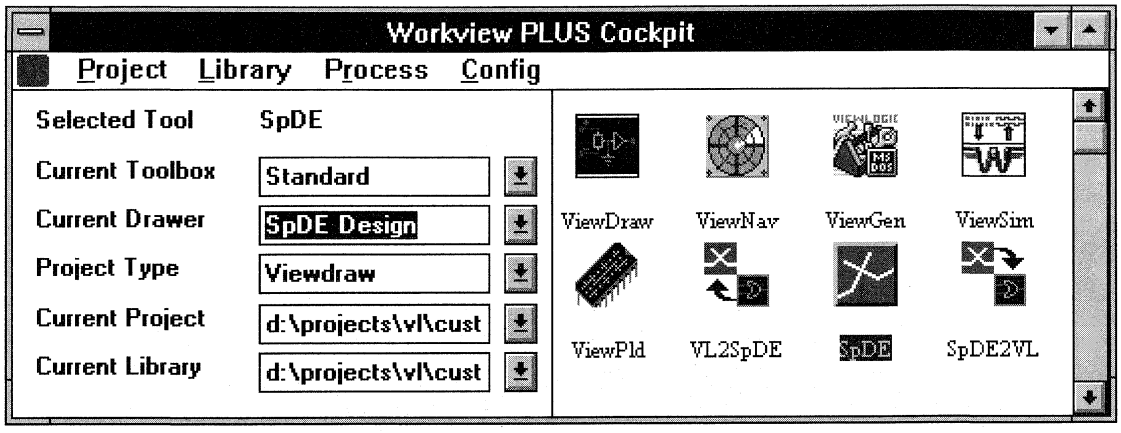
SUPPORT

The ViewLogic macro library and interface package provides the library symbols and the interface software to tightly integrate the QuickLogic and ViewLogic environments. Designs entered through Viewlogic can be simply and directly transferred into SpDE through simple icon selection or menu commands.

And talk about integration...integration is so tight that the designer can move from one environment to the other with hardly a thought. All the SpDE icons are resident in the Cockpit window (Figure 1) for simplicity and continuity.

QUICKLOGIC

FIGURE 1 Integration of SpDE to the Viewlogic Environment



Once in SpDE, it is easy to verify, place, route, and analyze the design. It's as easy as a push of a button. Run everything from the menu commands or navigate through from the cockpit. Continue to have the control and flexibility you are used to — assign fixed pad placement, device selection and package selection.

All the steps necessary for building the simulation files, including creating and processing the WIR file, delay files and net equivalence files, are automatically back-annotated into the Viewlogic environment for Viewsim to use. This, like the rest, has been completely integrated into the menu commands and the cockpit control panel, allowing for a quick and easy process into the simulation phase of the design.

DESIGN FLOW

The following design flow (Figure 2) illustrates the ease in designing a QuickLogic device with Viewlogic. Design entry is no different than that normally used in Viewlogic, no matter what the mode — schematics, VHDL, or Abel. The only difference now is that the QuickLogic macro libraries are used rather than the standard Viewlogic built-in library.

After entering the design, simply export the design to the SpDE tools. This is done by executing the VL2SPDE (Viewlogic to SpDE) tool — either by menu command or icon selection, depending on the package from Viewlogic you are using. This creates a QDIF file which can be imported into the SpDE tools.

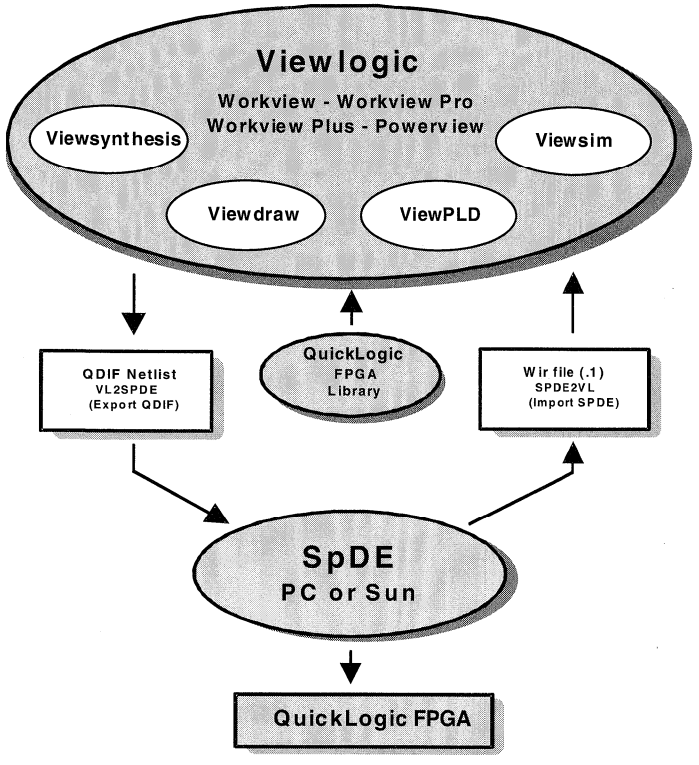


FIGURE 2
QuickLogic Design Flow with Viewlogic

Within SpDE, the design can then be run through the various tools — verification, optimization, placement, route, and analysis for timing and functionality. In addition, the SpDE tools can generate all the necessary files for simulation during the back annotation stage of the process. By executing SPDE2VL (SpDE to ViewLogic) command, a WIR file is created for simulation in ViewSim. Simulation can now be performed or a part can be programmed.

CONFIGURATION

The Viewlogic macro library and interface package from QuickLogic is intended for those of you who now, or plan to, design on the Viewlogic platform — Workview, Workview Pro, Workview Plus or Powerview. With the Viewlogic macro library and interface software (QS-VL-PC)/(QS-VL-SUN) in conjunction with the SpDE software (QS-SPDE-PC)/(QS-SPDE-SUN) you can now design the fastest FPGAs using Viewdraw schematic capture, Viewsynthesis VHDL, or ViewPLD Abel.

QS-INT-PC
QuickLogic pASIC 1 Family
Intergraph "ACEPlus/VeriBest" Libraries

HIGHLIGHTS

- **Design QuickLogic pASIC 1 FPGAs with ACEPlus** Schematic Capture (V12.2) on the PC (Windows 3.1/NT) platform enabling a complete design methodology in the Integraph environment.
- **Seamless Interface** to a QuickLogic pASIC toolkit through the EDIF 2 0 0 format.
- **Functional and full-timing simulation** using VeriBest Verilog simulator.

INTRODUCTION

Current users of the Intergraph ACEPlus and VeriBest tools can now design with the QuickLogic pASIC I family of FPGAs using the QuickLogic libraries and simulation interface.

SUPPORT

The Intergraph ACEPlus macro library and interface package provides the library symbols and the interface software to tightly integrate the QuickLogic and Intergraph environments. Designs can be entered using the symbol libraries, placed and routed using the QuickLogic SpDE toolkit and simulated using the VeriBest Verilog simulator.

DESIGN FLOW

Design entry is accomplished using the QuickLogic macro library. After the design is captured, behavioral simulation can be performed using VeriBest. Once simulated, the design is exported to the SpDE tools by generating an EDIF file.

The file is then imported into SpDE to perform the following tasks - verification, logic optimization, placement, routing and timing analysis. The QuickLogic tools also generate a netlist in Verilog format with timing in SDF format for post-layout simulation with VeriBest. The designer can then program the device using the QuickLogic Programmer.

CONFIGURATION

This kit (QS-INT-PC) is intended for designers who wish to design the fastest FPGAs using the Intergraph Electronics Desktop Manager (Version 12.3) that includes the ACEPlus (Version 12.2) schematic capture system and the VeriBest Verilog (Version 12.3) simulator and the Design Methodology Manager (Version 12.3), in conjunction with the QuickLogic SpDE or QuickTools toolkits.

QUICKLOGIC

QS-SYN-SUN
QuickLogic pASIC 1 Family
Synopsys Macrolibrary & Interface

HIGHLIGHTS

- Design QuickLogic pASIC 1 FPGAs with Synopsys synthesis.
- QuickLogic synthesis libraries are transferrable to any platform supported by Synopsys.
- Support for both VHDL and Verilog HDL standards — enabling a complete high-level design methodology.
- Perform behavioral simulation in the Synopsys environment using VSS.
- Seamless interface to the QuickLogic pASIC toolkit through EDIF 2 0 0 format.

INTRODUCTION

Current users of the Synopsys Design Compiler can continue to use Synopsys with the QuickLogic pASIC 1 family of FPGAs using the Synopsys macro library and interface package.

Designs can be entered in either VHDL or Verilog HDL, compiled in Synopsys, then automatically placed and routed in SpDE, the QuickLogic Seamless pASIC Design Environment.

SUPPORT

The Synopsys macro library and interface package provides the synthesis library symbols and the interface software to intergrate tightly the QuickLogic and Synopsys environments. Designs entered through Synopsys can be simply and directly transferred into SpDE through an EDIF netlist. From here you can easily verify, place, route, and analyze the design.

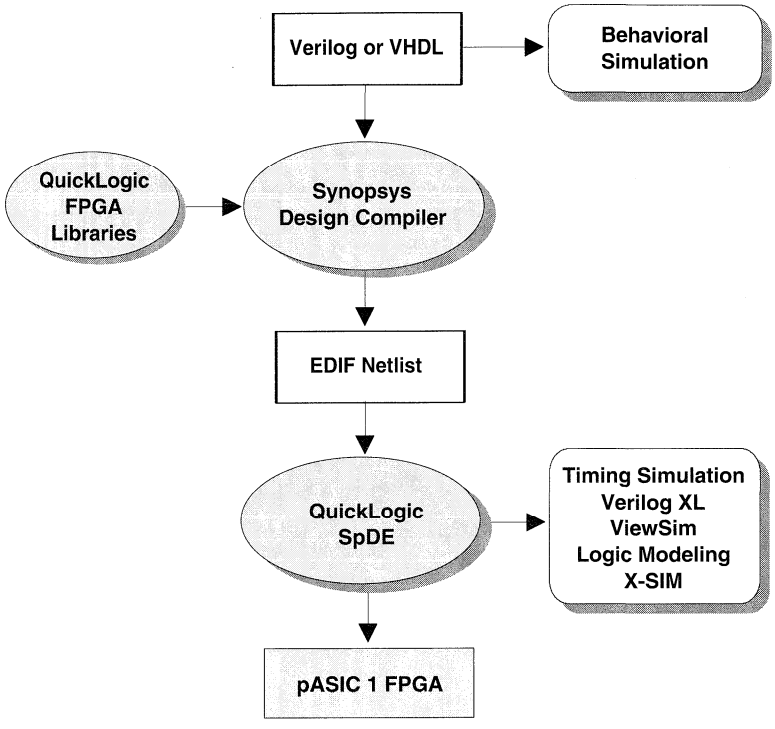
If already familiar with Synopsys, one can continue to do automatic pad insertions, assign fixed pad placement, device selection and package selection.

DESIGN FLOW

The following design flow (Figure 1) illustrates the ease of designing a QuickLogic device with Synopsys. Design entry is no different than that normally experienced with Synopsys, no matter what the mode — VHDL or Verilog HDL. Designs described in these modes are mapped into the QuickLogic macro libraries using the Synopsys Design Compiler. After the compiler maps the HDL behavioral description into the QuickLogic architecture, the compiler produces an EDIF 2 0 0 netlist to be read by the SpDE tools.

QUICKLOGIC

FIGURE 1
QuickLogic Design Flow with Synopsys



The EDIF netlist generated by Synopsys can then be imported into the SpDE tools. Within the SpDE tools, the design can be prepared for implementation — verified, optimized, placed, routed, and analyzed for timing and functionality. In addition, the SpDE tools will generate all the necessary files for simulation during the back annotation stage of the process. At this point, simulation can be performed with various simulators, or a device can be programmed.

CONFIGURATION

The Synopsys package from QuickLogic is intended for those using the Synopsys Design Compiler or planning to soon. With the Synopsys macro library and interface software (QS-SYN-SUN) in conjunction with the SpDE software (QS-SPDE-SUN) you can now design the fastest FPGAs with Synopsys.

QS-CNC-SUN
QuickLogic pASIC 1 Family
Cadence "Concept" Macrolibrary & Interface

HIGHLIGHTS

- Design QuickLogic pASIC 1 FPGAs with Concept Schematic Capture on the Sun — enabling a complete design methodology on the Cadence platform.
- Seamless interface to the QuickLogic pASIC toolkit through EDIF 2 0 0 format.
- Standard Verilog netlist and SDF output to interface with VerilogXL.

INTRODUCTION

With the Concept libraries and the SpDE tools outputting a Verilog netlist and a SDF file, you can design for the QuickLogic pASIC 1 Family of FPGAs in Cadence and simulate it in Verilog.

SUPPORT

The Concept macro library and interface package provides the library symbols and the interface software to intergrate tightly the QuickLogic and Cadence environments. Designs entered through Concept schematic capture can be directly transferred into SpDE through an EDIF netlist. The design can then be verified, placed, routed and analyzed.

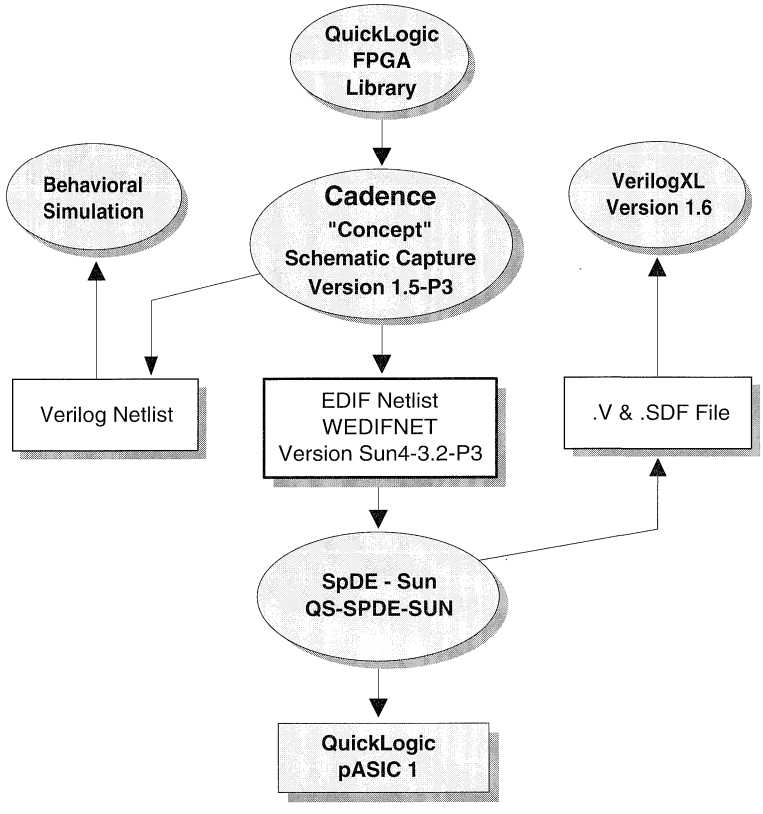
DESIGN FLOW

The design flow (Figure 1) shows the ease of designing a QuickLogic device with Cadence. Design entry is accomplished using the QuickLogic macro libraries.

Once a design is entered, a behavioral simulation can be performed by generating a Verilog netlist. Once simulated, the design is exported to the SpDE tools by generating an EDIF file.

QUICKLOGIC

FIGURE 1
QuickLogic Design Flow with Cadence



This file can then be imported into SpDE to perform all the necessary tasks — verification, optimization, placement, routing, and analysis for timing and functionality. In addition, the SpDE tools will generate all the necessary files for simulation in VerilogXL (.V & .SDF) during the back annotation stage of the process. From here the designer can proceed either to simulation in Verilog XL or to device programming.

CONFIGURATION

The Cadence Concept Interface package from QuickLogic is intended for those using the Concept Schematic Capture from Cadence or planning to use it. With the Concept macro library and interface software (QS-CNC-SUN), in conjunction with the SpDE software (QS-SPDE-SUN), one can now design the fastest FPGAs with Cadence.

pASIC Macro Library

HIGHLIGHTS

- More than 350 Architecturally Optimized Macros
- Includes Simple Gates and Advanced Soft Macros
- Includes Over 100 7400-Series TTL Building Blocks
- SpDE Packs as Many as 4 Macros Into a Single Logic Cell
- SpDE's Logic Optimize maps many simple gates into a single logic cell

INTRODUCTION

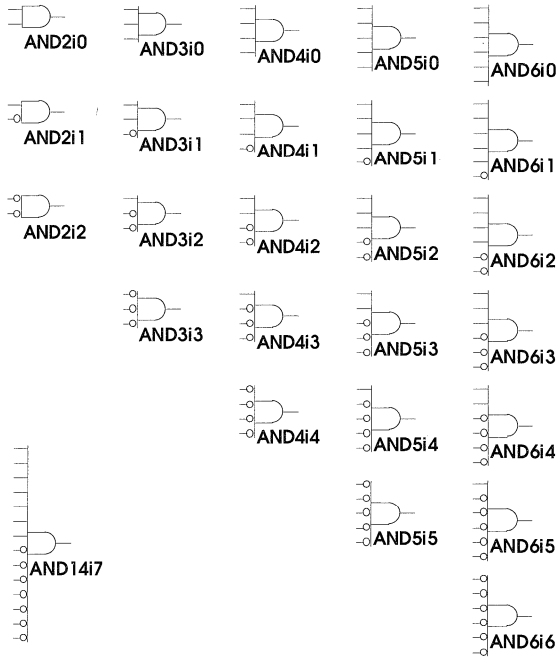
The pASIC Macro Library contains more than 450 macros. While these macros offer a wide range of functions and flexibility, they fall into familiar functional groups. The naming conventions employed in the library are easy to learn and remember—while 450 macros may seem like a daunting number at first, little experience is required to master the use of the library.

The remarkable flexibility of the pASIC logic cell allows literally millions of different hard macros to be created. Therefore, custom hard macros can be created easily using the pASIC logic cell. The logic cell represents the basic building block—all macros are created out of logic cells at the lowest level. Although most designers will never need to design at the logic cell level, this option is available.

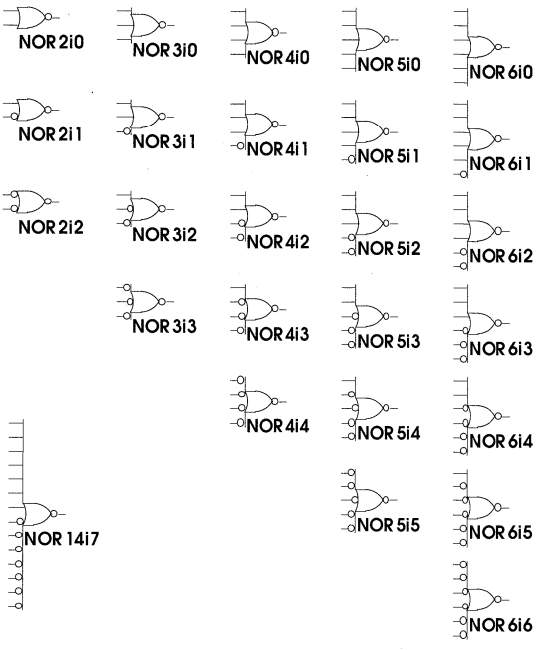
The pASIC Macro Library includes hard macros (macros that fit into one logic cell) and soft macros (macros that fit into multiple logic cells). Many simple hard macros (Multiplexors, AND gates, OR gates, XOR gates, etc.) require only one piece of the Logic Cell's resources. SpDE's Technology Mapper can map many of these simple gates into one Logic Cell, speeding up the design and reducing the number of logic cells needed.

QUICKLOGIC

AND Gates

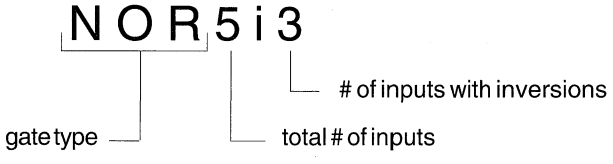


The pASIC Macro Library includes AND, NOR, NAND, and OR gates with two to six inputs. At each input count, all numbers of inversion bubbles are available (for example, 3-input gates are available with 0, 1, 2, and 3 inversion bubbles). The library also features the largest possible AND, NOR, NAND, and OR gates which can be implemented in a single logic cell (**AND14i7**, **NOR14i7**, **NAND13i6**, **OR13i6**).



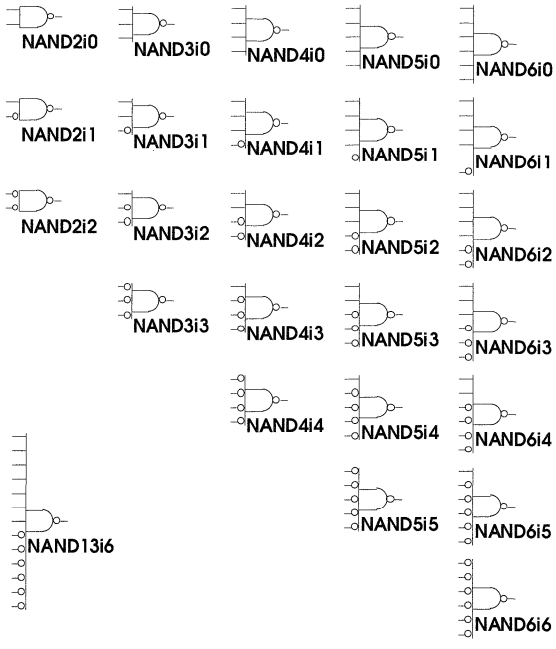
NOR Gates

An easy-to-remember naming convention is employed to identify these gates.

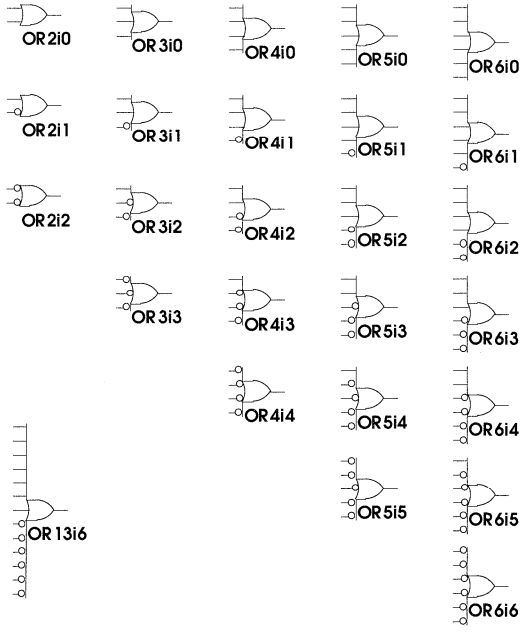


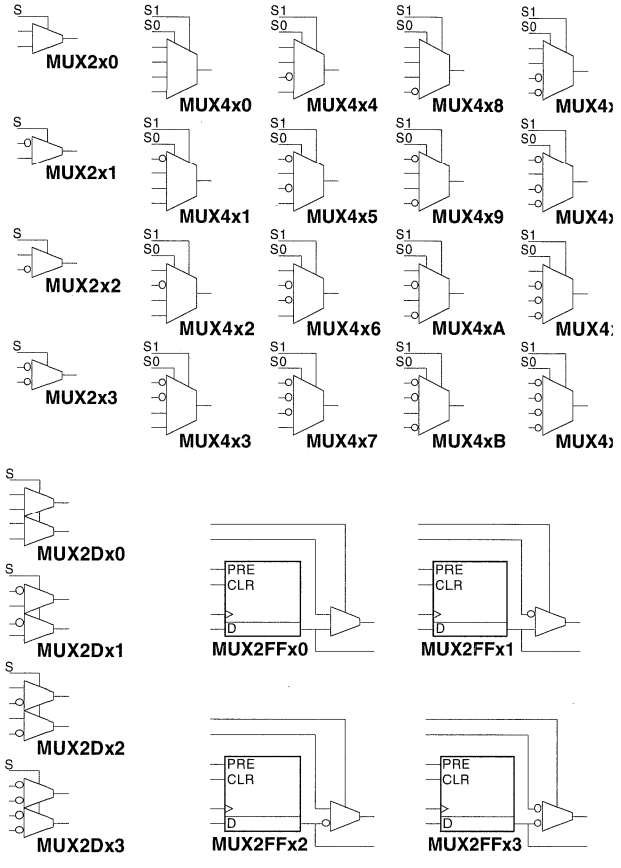
The first field identifies the type of gate—AND, NOR, NAND, OR. The second field specifies the total number of inputs. The 'i' character, which stands for "inverts," serves as a separator. The third field specifies the number of inputs with inversion bubbles.

NAND Gates



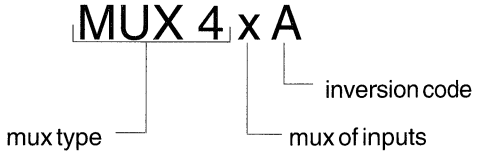
OR Gates





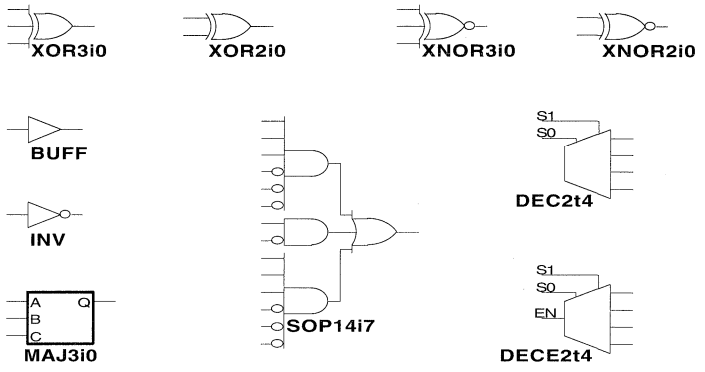
Multiplexers

The Macro Library includes sixteen 4-to-1 multiplexers, representing all possible combinations of inversion bubbles on the four inputs. The naming convention is similar to the AND, NOR, NAND, OR form explained above.



The first field, MUX, identifies this as a multiplexer. The second field specifies the number of inputs. The 'x' character serves as a separator. The third field specifies a hexadecimal code for the pattern of inversion bubbles.

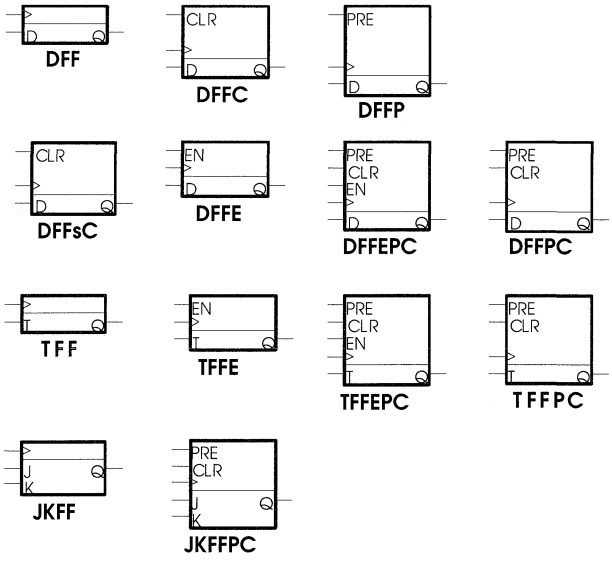
Other Combinational Macros



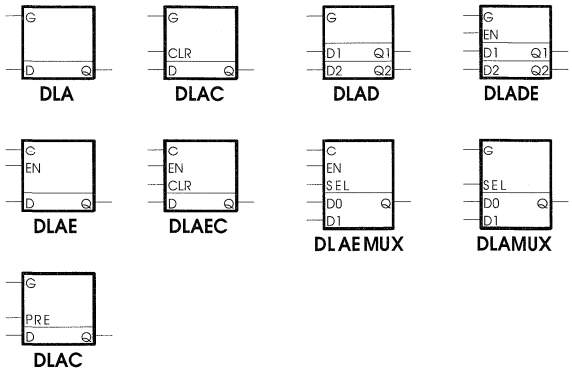
The Macro Library includes two and three input XOR (exclusive-or) and XNOR (exclusive-nor, also known as equivalence) gates. Their names use the same terminology as the AND, NOR, NAND, OR gates.

The Macro Library does not include combination gates (AND-OR, OR-AND, AND-XOR, etc.) present in some macro libraries. There are literally thousands of possible combination gates which can be implemented in a single logic cell, which would make for a fairly large macro library. These combination gates are handled automatically by the Logic Optimizer, as discussed earlier.

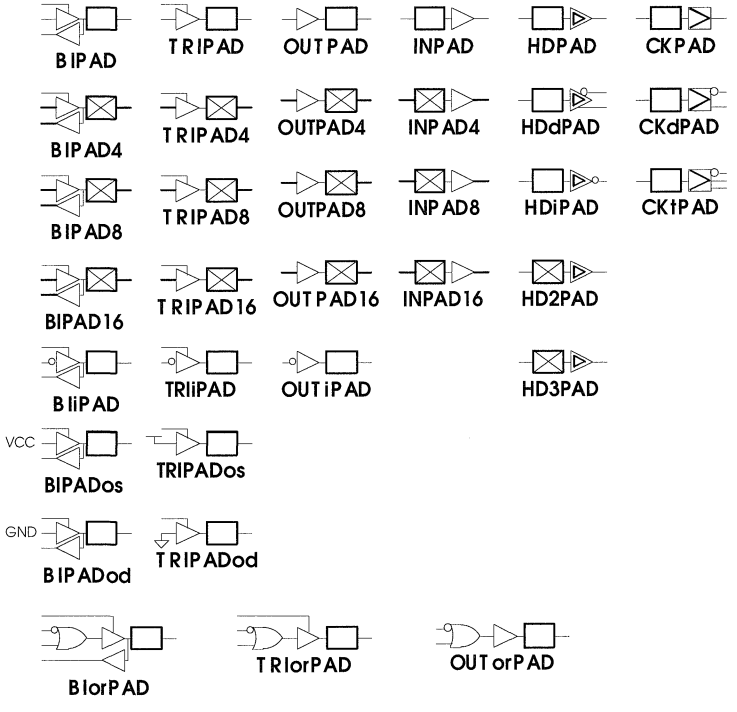
Flip Flops



Latches

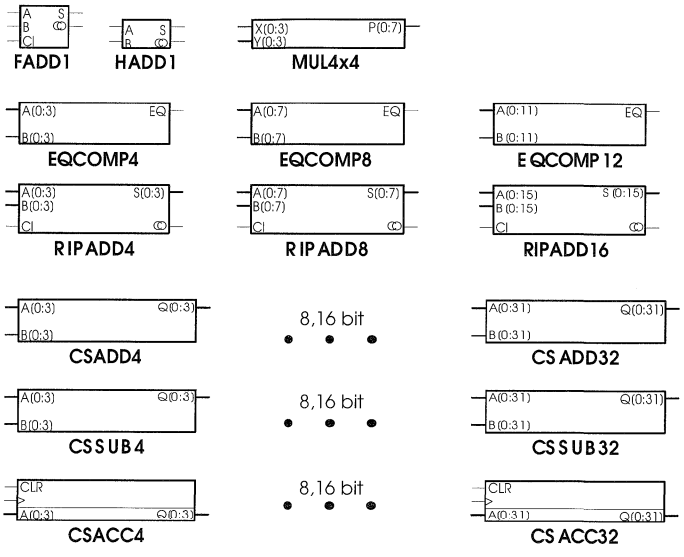


I/O Pads



3 Tools

Arithmetic



The logic cell is well suited for high-speed arithmetic. It includes **HADD1**, a simple half-adder macro which can be implemented in less than one logic cell.

FADD1 is a full-adder which requires two logic cells. This soft macro contains one **XOR3i0** and one **MAJ3i0** macro.

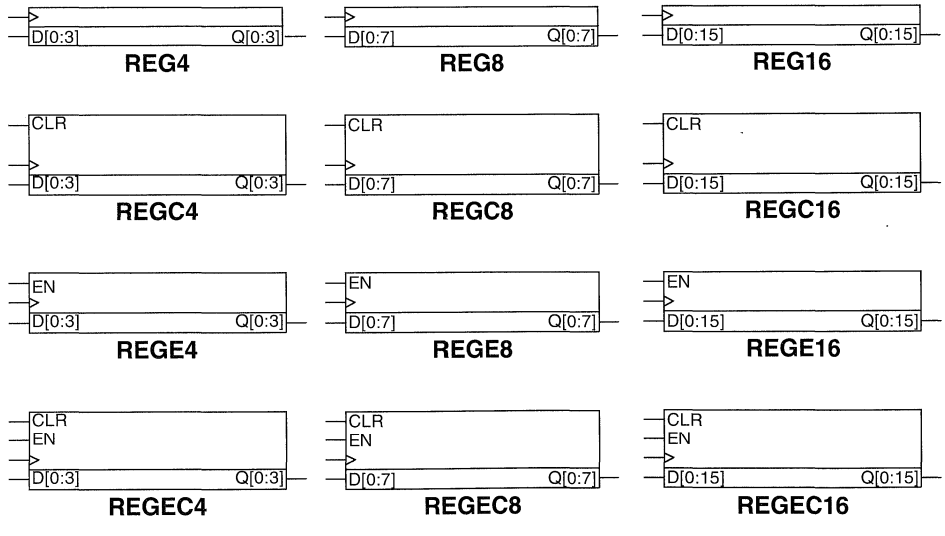
RIPADD4, **RIPADD8**, and **RIPADD16** are ripple-carry adders of 4, 8, and 16 bits, respectively. Each utilizes the **FADD1** full-adder macro in a ripple-carry arrangement. This provides excellent density at the cost of additional logic delays, compared to the conditional sum adders to be introduced shortly.

If high-speed addition is required (16-bit adders at 30 MHz and up) then the conditional-sum adders can be used. These high-speed adders are avaialble in 4,8,16, and 32-bit varieties and named **CSADD4**, **CSADD8**, **CSADD16**, and **CSADD32** respectiveley. If fast accumulators are needed, then the **CSACC** macros can be used. Conditional sum subtractors are also available in the same sizes using the name prefix **CSSUB**.

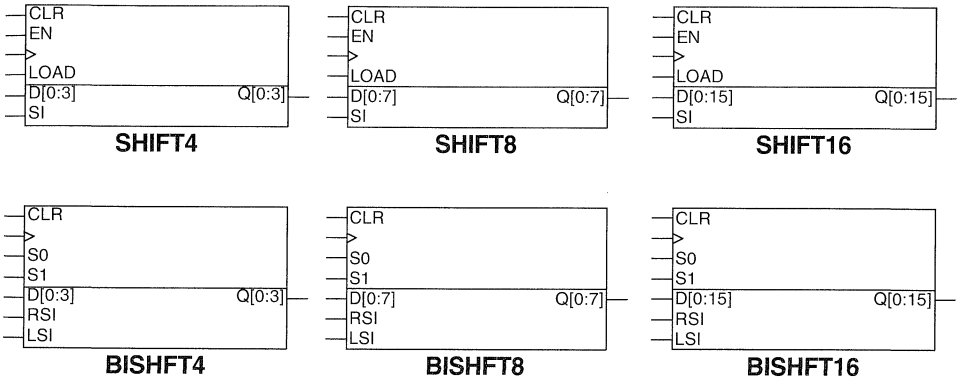
MUL4x4 is a four-by-four multiplier which offers excellent speed and excellent density. This macro's design takes advantage of the XOR utilization property mentioned above—two AND gates and a 3-input XOR gate can be packed into a single logic cell.

	Description	Logic Cells in Critical Path	Logic Cells
HADD1	Half Adder	1	1
FADD1	Full Adder	1	2
RIPADD4	4-bit Ripple Adder	4	8
RIPADD8	8-bit Ripple Adder	8	16
RIPADD16	16-bit Ripple Adder	16	32
CSADD4	4-bit Conditional Sum Adder	2	7
CSADD8	8-bit Conditional Sum Adder	3	20
CSADD16	16-bit Conditional Sum Adder	4	48
CSADD32	32-bit Conditional Sum Adder	5.5	120
CSACC4	4-bit Conditional Sum Accumulator	2	7
CSACC8	8-bit Conditional Sum Accumulator	3	20
CSACC16	16-bit Conditional Sum Accumulator	4	48
CSACC32	32-bit Conditional Sum Accumulator	5.5	120
CSSUB4	4-bit Conditional Sum Subtractor	2	7
CSSUB8	8-bit Conditional Sum Subtractor	3	20
CSSUB16	16-bit Conditional Sum Subtractor	4	48
CSSUB32	32-bit Conditional Sum Subtractor	5.5	120
MUL4X4	4-bit by 4-bit Multiplier	5	30

Registers

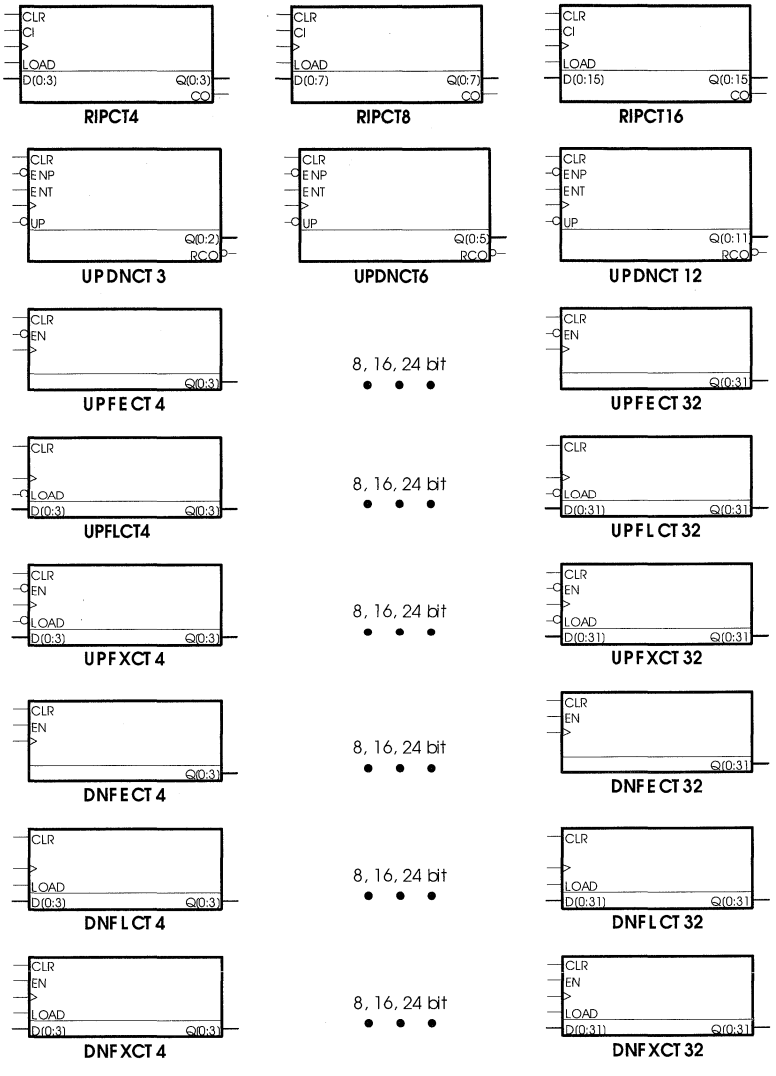


Shift Registers



The architecture can implement extremely fast shift registers. In fact, the shift registers included in the macro library can operate at the maximum logic cell toggle rate.

Counters



The architecture can produce extremely fast counters ("you can count on us"). Many of the counters in the macro library can operate near the maximum logic cell toggle rate.

The Macro Library includes ripple-carry counters (names starting with the prefix RIP) and parallel-carry counters. The following table illustrates the speed-density tradeoffs between these counters.

	Description	Bits	Logic Cells in Critical Path	Logic Cells
RIPCT4	Ripple Counter w/ Load, Enable	4	4	4
RIPCT8	Ripple Counter w/ Load, Enable	8	8	8
RIPCT16	Ripple Counter w/ Load, Enable	16	16	16
UPDNCT3	Up/ Down Counter w/ Load, Enable	3	1	4
UPDNCT6	Up/ Down Counter w/ Load, Enable	6	2	8
UPDNCT12	Up/ Down Counter w/ Load, Enable	12	2	16
UPFECT4	Fast Up Counter w/ Enable	4	1	4
UPFECT8	Fast Up Counter w/ Enable	8	1	11
UPFECT16	Fast Up Counter w/ Enable	16	1	24
UPFECT24	Fast Up Counter w/ Enable	24	1	38
UPFECT32	Fast Up Counter w/ Enable	32	1	52
UPFLCT4	Fast Up Counter w/ Load	4	1	4
UPFLCT8	Fast Up Counter w/ Load	8	1	11
UPFLCT16	Fast Up Counter w/ Load	16	1	24
UPFLCT24	Fast Up Counter w/ Load	24	1	38
UPFLCT32	Fast Up Counter w/ Load	32	1	52
UPFXCT4	Fast Up Counter w/ Load, Enable	4	1.5	4
UPFXCT8	Fast Up Counter w/ Load, Enable	8	1.5	11
UPFXCT16	Fast Up Counter w/ Load, Enable	16	1.5	24
UPFXCT24	Fast Up Counter w/ Load, Enable	24	1.5	36
UPFXCT32	Fast Up Counter w/ Load, Enable	32	1.5	50
DNFECT4	Fast Down Counter w/ Enable	4	1	4
DNFECT8	Fast Down Counter w/ Enable	8	1	11
DNFECT16	Fast Down Counter w/ Enable	16	1	23
DNFECT24	Fast Down Counter w/ Enable	24	1	38
DNFECT32	Fast Down Counter w/ Enable	32	1	51
DNFLCT4	Fast Down Counter w/ Load	4	1	4
DNFLCT8	Fast Down Counter w/ Load	8	1	11
DNFLCT16	Fast Down Counter w/ Load	16	1	23
DNFLCT24	Fast Down Counter w/ Load	24	1	38
DNFLCT32	Fast Down Counter w/ Load	32	1	51
DNFXCT4	Fast Down Counter w/ Load, Enable	4	1.5	4
DNFXCT8	Fast Down Counter w/ Load, Enable	8	1.5	11
DNFXCT16	Fast Down Counter w/ Load, Enable	16	1.5	22
DNFXCT24	Fast Down Counter w/ Load, Enable	24	1.5	36
DNFXCT32	Fast Down Counter w/ Load, Enable	32	1.5	50

The library includes up and down counters utilizing parallel carry designs. The 3-bit up-down counter UPDNCT3 is designed in the spirit of the 7400-series '169. There are two versions of this macro to allow them to be cascaded.

7400-Series TTL Macros

	Description
TTL02	quad 2-input NOR gates
TTL04	hex inverters
TTL08	quad 2-input AND gates
TTL11	triple 3-input AND gates
TTL21	dual 4-input AND gates
TTL27	triple 3-input NOR gates
TTL42q	4-to-10 decoder
TTL49	BCD to 7-segment decoder
TTL74q	dual D FFs with preset & clear
TTL77	4-bit D latch
TTL78q	dual J-K FFs with common clock & clear
TTL85	4-bit magnitude comparator
TTL86	quad 2-input XOR gates
TTL87	4-bit true/complement elements
TTL91	8-bit shift register
TTL98	4-bit data selector/storage register
TTL104q	gated J-K FF with preset & clear
TTL105q	gated J-K FF with preset & clear
TTL107q	dual J-K FFs with clear
TTL109q	dual J-K FFs with preset & clear
TTL116	dual 4-bit D latches with clear
TTL138q	3-to-8 decoder
TTL139q	dual 2-to-4 decoders
TTL145q	BCD to decimal decoder
TTL150	16-to-1 multiplexer
TTL152	8-to-1 multiplexer
TTL153	dual 4-to-1 multiplexers
TTL154q	4-to-16 decoder
TTL157	quad 2-to-1 multiplexers
TTL161	4-bit binary counter with asynchronous clear
TTL163	4-bit binary counter with synchronous clear
TTL164q	8-bit parallel-out shift register
TTL166q	8-bit parallel-load shift register
TTL169	4-bit binary up/down counter
TTL171q	quad D FFs with clear
TTL174q	hex D FFs with clear
TTL180	9-bit odd-even parity generator
TTL194q	4-bit bidirectional shift register
TTL240q	octal inverting tri-state drivers

	Description
TTL244q	octal noninverting tri-state drivers
TTL259	8-bit addressable latches
TTL261	2-bit by 4-bit binary multipliers
TTL268q	hex D latches
TTL273q	octal D FFs with clear
TTL278	4-bit cascadable priority register
TTL279	quad S-R latches
TTL295q	4-bit right-shift left-shift register
TTL365	hex tri-state drivers
TTL366	hex inverting tri-state drivers
TTL367	hex tri-state drivers, 4-bit/2- bit banks
TTL368	hex inverting tri-state drivers, 4-bit/2-bit banks
TTL373q	octal D latches
TTL374q	octal D FFs
TTL375	4-bit latches with dual polarity outputs
TTL376q	quad J-K FFs with clear
TTL395q	4-bit cascadable shift register with clear
TTL396	octal storage register
TTL465	octal tri-state buffers
TTL466	octal inverting tri-state buffers
TTL467	octal tri-state buffers, 4-bit/4-bit banks
TTL468	octal inverting tri-state buffers, 4-bit/4-bit banks
TTL518	8-bit identity comparator
TTL594q	8-bit shift register with output register with clear
TTL595q	8-bit shift register with output register
TTL604q	octal 2-input multiplexed latches
TTL684	8-bit magnitude/identity comparator
TTL686	8-bit magnitude/identity comparator with enable
TTL688	8-bit identity comparator
TTL821	10-bit FFs
TTL822	10-bit inverting FFs
TTL823q	10-bit FFs with enable
TTL841q	10-bit latches
TTL842q	10-bit inverting latches

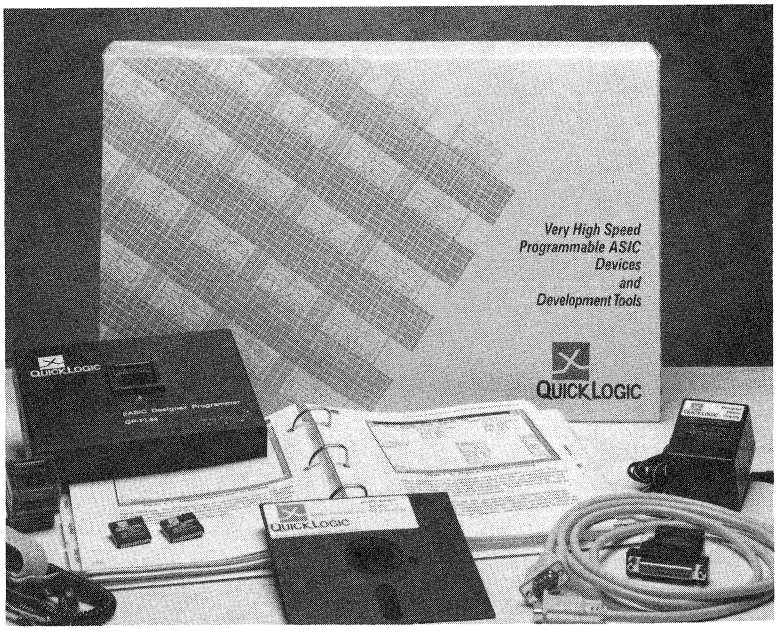
Note: The q suffix indicates that the part is not an exact duplicate of the TTL device. The differences are described in the User's Guide.

QT-DP-PC
Programming/Testing Toolkit

HIGHLIGHTS

- **Combination programmer and tester** – Toolkit contains all hardware and software required to program pASIC devices from a 386/486 PC and then execute ATVG test vectors to verify programming results.
- **Internal Device Scan Path** in pASIC 1 Family allows testing of all registers following programming.
- **Programming/Testing Toolkit** includes cables, an adapter and an antistatic wrist strap.
- **Gangable** – Up to eight programmers may be connected to one PC for volume programming.
- **Intended for users of third-party place and route tools** who need to program QuickLogic devices or for users adding incremental, gangable programmers.

Programming/ Testing Toolkit



QUICKLOGIC

Designer Programmer/ Tester

The Designer Programmer /Tester is a cost-effective programming and testing unit for QuickLogic devices. An 84-pin socket directly accepts 84-lead QL12x16B PLCC devices; other packages/devices are supported by individual socket adapters.

Programming and Testing

The Programmer Interface translates the results of the Automatic Place Route/ Sequencing process into the patterns required to program pASIC devices. These are transmitted, together with automatically generated test vectors, over an RS-232 link to the programmer.

The optimal programming procedure is determined by a SpDE tool called the Sequencer. By coupling the AWE calculations of the Delay Modeler with the results of the Router, the Sequencer optimizes the programming procedure to program ViaLink elements on the most heavily loaded nets to resistance values below 50 ohms. This ensures the shortest possible delays on these critical interconnects.

Automatic Test Vector Generation (ATVG) software analyzes test coverage and generates the test vectors This allows convenient and thorough test coverage of programmed pASIC parts using the internal scan path included in the design of the devices. The degree of test coverage can be easily controlled, and user-specified test vectors generated in the X-SIM simulator can be included. The Programmer Interface downloads the programming element and test vector signals over an RS-232 connection to the Designer Programmer.

Adapter Sockets

A range of adapter sockets are available to adapt the base unit's 84 pin PLCC socket to support all package types offered by QuickLogic. A reference chart to help in ordering any necessary adapters may be found later in this data sheet.

Contents and System Requirements

The Programming/Testing Toolkit contains all necessary hardware and software to program and test the device, minus the PC workstation.

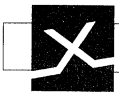
- Designer Programmer/Tester unit
- Automatic Test Vector Generator (ATVG) software
- Programming interface software
- AC adapter and cord (Output 12 VAC, 800 mA. Plug ID 2.1 mm, OD 5.5 mm)
- 9-pin RS-232 cable
- Cable adapter, 25-pin to 9-pin
- 68-pin PLCC adapter
- ESD wrist strap
- All software supplied on 3.5-inch high-density disks; 5.25-inch versions available upon request.

The programming unit operates when connected serially to a PC station running Windows™ 3.1 or NT. The minimum software and hardware requirements for programming the device are as follows:

- MS-DOS (or PC-DOS) version 3.1 or higher
- Microsoft Windows version 3.1 or NT
- 80386/486-based PC
- 8 MBytes RAM
- Windows-supported mouse
- 10 MBytes free disk space
- Available serial port

Order Code	Power	Name
QT-DP-PC-A	USA	Programmer/Tester
QT-DP-PC-B	Europe	Programmer/Tester
QT-DP-PC-C	U.K.	Programmer/Tester

Order Code	DEVICES SUPPORTED	DESCRIPTION
QP-PL44	QL8X12	Adapts 44PLCC To Program/Tester Unit
QP-PL68	QL8X12,QL12x16	Adapts 68PLCC To Program/Tester Unit
QP-CG68	QL8X12	Adapts 68CPGA To Program/Tester Unit
QP-CG84	QL12X16	Adapts 84CPGA To Program/Tester Unit
QP-PF100	QL8X12, QL12X16	Adapts 100TQFP To Program/Tester Unit
QP-CF100	QL12X16	Adapts 100CQFP To Program/Tester Unit
QP-PL4084	QL16X24	Adapts 84PLCC To Program/Tester Unit
QP-PF4100	QL16X24	Adapts 100TQFP To Program/Tester Unit
QP-CF4100	QL16X24	Adapts 100CQFP To Program/Tester Unit
QP-PF4144	QL16X24	Adapts 144TQFP To Program/Tester Unit
QP-CG4144	QL16X24	Adapts 144CPGA To Program/Tester Unit
QP-CF4160	QL16X24	Adapts 160CQFP To Program/Tester Unit
QP-PF8144	QL24X32	Adapts 144TQFP To Program/Tester Unit
QP-PQ8208	QL24X32	Adapts 208PQFP To Program/Tester Unit



QS-QWK-50-PC-EV
"Checkout Your Design in Our FPGA"
Complete and Affordable FPGA Evaluation Kit

HIGHLIGHTS

- **Complete low-cost version** of the Quick*Works* tools for the pASIC™ 1 Family of FPGAs for thorough design evaluation of fit and speed
- **Everything needed** to complete a logic design in any QuickLogic device, including the new WildCat 8000, and 3.3 Volt "L" Series
- **State-of-the-art tools,** based upon Microsoft Windows graphical user interface, include:
  - Data I/O's SCS Synario Schematic Capture System
  - Context sensitive HDL editing
  - Auto-color coding and template expansion
  - Mixed Mode Design entry
  - Optimal Synthesis for Verilog, VHDL, and Open-Abel by Synplicity
  - Place/Route/Timing Modeling/Analysis Environment (SpDE™)
  - Verilog or Waveform Simulation with SimuCad's SILOS III simulator
- **Experience the Productivity** afforded by the seamless integration of tools
- **Provided** with a 30-day license and hotline support

Everything you need to *"Checkout Your Design"*

QUICKLOGIC

No Risk, Low-cost Design Kit

The *"Checkout Your Design in Our FPGA"* Kit is a low cost yet fully functional CAE software package that includes a complete Windows-based design entry system with all the necessary tools to place and route a real-world design and automatically generate accurate timing values. The complete design can be analyzed with the Physical Viewer, Path Timing Analyzer, and the Verilog or Waveform Simulator. This package is intended to show you, without risk, that the speed and capacity of QuickLogic FPGAs will meet your most demanding performance requirements. The kit is provided with a 30-day license and 30-day hotline support.

Quick to Learn pASIC Software

The pASIC Toolkit is the most sophisticated and user-friendly software available anywhere for FPGA design. Based upon Microsoft Windows, the graphical user interface is designed to be easy to use for both the first-time FPGA designer and the PLD design veteran. The kit includes extensive on-line help. Several customers have learned the software in a day and completed their first design over a weekend!

A Complete FPGA Design Package

The *"Checkout Your Design"* Kit is an evaluation package based upon the latest version of the Quick*Works* tools and includes all of the functionality required to implement and analyze your design in any QuickLogic device.

Data I/O SCS Schematic Capture Package

- Schematic Editor
- Symbol Editor
- Hierarchy Navigator
- pASIC Macro Library – 500 library functions available
- Augmented with QuickLogic's QuickBoolean entry
- Mixed mode entry within a schematic: Both HDL & Schematic

Context-Sensitive HDL Entry

- TurboWriter Editor with HDL templates by Saros
- Automatic color coding of keywords, comments, errors, etc...
- Command line expansion with auto indenting
- Automatic Verilog test-bench generation

HDL Synthesis by Synplicity

- Highest quality synthesis available
- Verilog, VHDL and Open-Abel
- Fastest run times: Minutes for a 4,000 gate design

QuickTools, including SpDE (Seamless pASIC Design Environment)

- Logic Optimizers – reduces cell delays and minimizes silicon area
- Placer and Router – 100% automatic using up to 100% of logic and I/O cells
- Physical Viewer – view actual physical layout and cross probe with schematic
- Delay Modeler – SPICE-like accuracy using AWE techniques
- Path Analyzer – simple-to-use table to analyze timing through any path
- Interactive cross-probing between tools

Simucad's SILOS III Verilog or Waveform Simulator

- OVI 2.0 Compliant
- Generate input stimulus in Verilog, or graphically with Waveform editor
- Examine Timing / Functionality with Verilog,or with Waveform Viewer
- Graphically cross probe to logical values displayed on SCS Schematic

The "*Checkout Your Design*" Kit will allow you to implement your entire design and determine that your design fits and runs fast enough before you commit to purchasing a complete system. This package does *not* include the modules necessary to program devices — the Automatic Test Vector Generator or Programming Interface software. The kit comes with a 30-day user license and 30-day hotline support. At the end of this 30-day evaluation period you can upgrade to any QuickLogic toolkit listed below and apply the purchase price of this kit toward a full development system.

QuickLogic pASIC Development Tools

PART NO. **UPGRADE PRODUCT NAME AND CONTENTS**

QT/QS-QWK-50-PC

Complete pASIC Toolkit

- Data I/O SCS Engineering Capture System
- Contexet Sensitive HDL entry
- Verilog & VHDL Synthesis by Synplicity
- Simucad's SILOS III Verilog Simulator
- QuickLogic's QuickTools with SpDE
- Interfaces to third-party tools including: Exemplar, Intergraph, Logic Modeling, SmartModels™, Mentor, SusieCAD
- Programming/Testing Hardware (QT version only)

QT/QS-QTL-50-PC/SUN/HP

QuickTools Toolkit with QuickLogic's SpDE

- QuickLogic SpDE Tools with Path Analyzer
- Level 2 Logic Optimizer
- Interfaces to third-party tools including: Exemplar, Intergraph, Logic Modeling, SmartModels™, Mentor, SusieCAD
- Programming/Testing Hardware (QT version only)

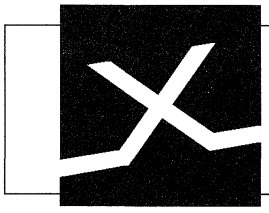
System Requirements

The minimum software and hardware requirements for running the pASIC Toolkit are as follows:

- Microsoft Windows version 3.1 or NT
- 386/486-based PC
- 8 MBytes of RAM (16 MBytes for 24x32B Designs)
- Windows-supported monitor and graphics adapter (EGA, VGA)
- Windows-supported mouse
- 20 MBytes free hard disk space

Summary

QuickLogic is committed to providing the world's fastest FPGAs. We are now making it easy and inexpensive to determine that our FPGAs meet your speed requirements. Try our FPGA evaluation package and *Checkout Your Design!*



pASIC 1 FAMILY
Third-party Design Support

PRODUCT DESCRIPTION	VENDOR	PHONE
DESIGN ENTRY		
ABEL 5.0, VHDL, Synario	Data I/O Corp	(206) 881-6444
Design Compiler, VHDL, Verilog	Synopsys	(415) 962-5000
Viewdraw, Viewsynthesis	Viewlogic	(508) 480-0881
CUPL	Logical Devices	(800) 331-7766
PALASM, ABEL, CUPL, Minc, VHDL	Exemplar Logic	(510) 849-0937
VHDL, Schematic	Intergraph	(205) 730-2000
CONCEPT	Cadence	(408) 493-1234
Engineering Capture System (ECS)	Data I/O Corp	(206) 881-6444
SIMULATION		
Viewsim	Viewlogic	(508) 480-0881
Verilog XL Simulator	Cadence	(408) 943-1234
Simulation Models	Logic Modeling Corp	(503) 690-6900
VHDL, VSS	Synopsys	(415) 962-5000
XSIM	Silicon Automation Sys.	(408) 437-9161
PROGRAMMING		
Unisite, 3900	Data I/O Corp	(206) 881-6444
Sprint, OmniPro	SMS Microsystems	(206) 883-8447
Programming Services	QuickLogic Corp	(408) 987-2000
Programming Services	Axis Components	see Disti Listings
Programming Services	Bell Micro. / Vantage	see Disti Listings
Programming Services	Future Electronics	see Disti Listings
Programming Services	Western Micro Tech.	see Disti Listings

PRODUCT DESCRIPTION	VENDOR	PHONE
SOCKET SUPPLIERS		
Prototype Sockets, TQFP, PLCC	CTI Technologies	(602) 998-1484
Prototype Sockets, TQFP, PLCC	Nepenthe	(415) 496-6666
Prototype Sockets, TQFP, PLCC	Yamaichi Electronics, Inc	(408) 452-0797

Application Note Summary

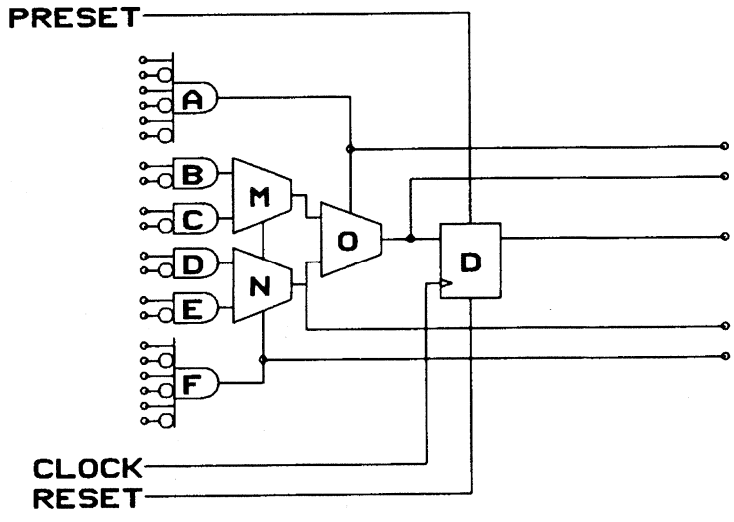
QUICKLOGIC

QAN1
Registers and Latches in the pASIC Architecture

INTRODUCTION

Quicklogic's pASIC™ 1 Family of high-performance FPGAs allows logic function speeds of over 100 MHz. The prime objective of the QuickLogic pASIC 1 Family logic cell is to maximize in-system device speed, while providing the flexibility to integrate both combinatorial and register-intensive designs. For sequential applications, each pASIC 1 logic cell can be configured for latched or registered operation. In fact, for some designs, it is possible to integrate two latches and a register into a single logic cell. Thus, the QL8X12B, which offers 1000 usable gates in 96 logic cells, has a maximum potential capacity of 288 registers and latches. The pASIC logic cell is shown in Figure 1.

FIGURE 1
The pASIC 1 logic cell's 5 outputs offer a new level of flexibility in FPGA design, resulting in very high gate usage



QUICKLOGIC

Each logic cell has a dedicated register that has independent reset and preset functions. One of the unique features of the logic cell is the availability of multiple outputs, which may also be used as feedback paths. These feedback paths, combined with the combinatorial logic in each logic cell, allow it to be configured for a variety of registered and latched applications, while maintaining efficient gate utilization. This QuickNote will demonstrate a few of the many ways the pASIC 1 logic cell can be used to implement latches, registers, and combinations of the two functions.

The pASIC Toolkit is a fully integrated, Windows-based development environment for QuickLogic pASIC 1 FPGAs. The Engineering Capture System (ECS) from Data I/O allows for fast schematic entry of designs. To simplify design entry, the Toolkit comes with an extensive library of macrofunctions including over 50 register and latch functions. These include single and grouped functions, as well as many 7400 series equivalents. Figure 2a shows a few of the basic functions, while Figure 2b gives a listing of the 7400 series register and latch functions included in the library.

FIGURE 2a
Several of the over 50 register and latch macrofunctions included in the pASIC Macro Library

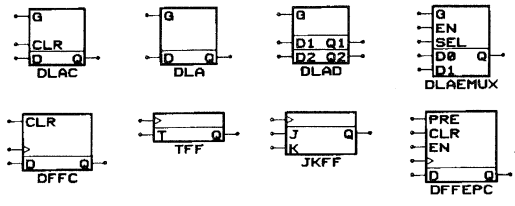


FIGURE 2b
The pASIC Toolkit also offers an extensive library of 7400 series latch and register functions

7474	7477	7478	7491
7498	74104	74105	74107
7409	74116	74171	74174
74194	74268	74273	74278
74279	74373	74374	74376
74395	74396	74594	74604
74821	74822	74823	74841
74842			

For most cases, the macrofunction library will contain all of the functionality required by a particular design.

However, in some cases a customized element may be desired to achieve a specific function. The pASIC Toolkit allows custom functions to be designed at the cell level, or by combining existing library elements. This gives the designer the best of both worlds — high- level functionality from the macro library and control at the gate level when desired.

The multiplexer-based pASIC 1 logic cell, is ideal for implementing latched functions. As shown in Figure 3, a D-type flow-through latch is easily implemented in a single 2-to-1 multiplexer. This function is available in the library as the element DLA.

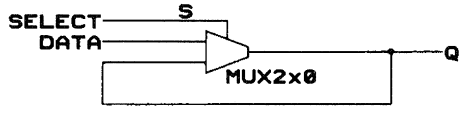


FIGURE 3
A 2-to-1 multiplexer is easily configured as a flow-through latch, taking a fraction of a logic cell

The multiplexer-based logic cell can implement two flow-through latches in a single cell (library element name DLAD). Thus, a function such as the 74373, an 8-bit latch, is implemented in only four logic cells.

Perhaps not so obvious are the benefits gained when the design requires pipelined latches. In this case, for which a logic cell design is shown in Figure 4, the output of the first latch (which is implemented in multiplexer N) feeds the input of the second latch (which is implemented in multiplexer O). This design is easily modified for common enables, true-complement enabling, or inverted data input.

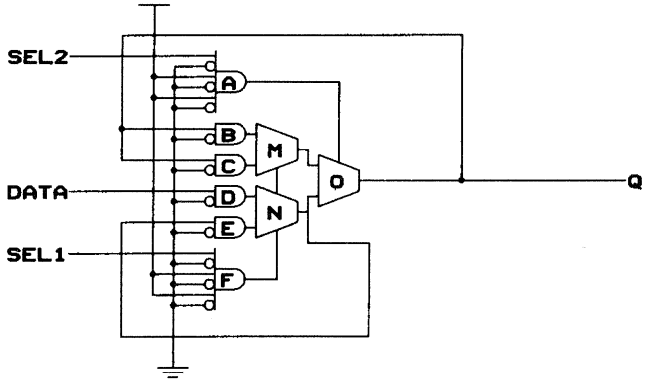


FIGURE 4
Using a cell design such as this one, pipelined latches can be combined into a single logic cell to minimize the latch-to-latch delay

This type of design minimizes the delay path between latches for high operating frequency, while at the same time maximizing logic cell utilization.

The pASIC 1 Family is equally capable when it comes to integrating edge-triggered registers. As mentioned previously, each of the logic cells within a pASIC 1 FPGA comes equipped with a dedicated D-type register that can also be configured for J-K or Toggle operation. These dedicated registers will typically satisfy the register requirements of most designs. However, some designs are register-intensive and will require more than the fixed number of registers within the pASIC device. In this case, it is often possible to put two registers into a single logic cell. An edge-triggered register can be easily constructed from the multiplexer logic in each logic cell. Figure 5 shows how two 2-to-1 multiplexers are connected to create an edge-triggered register. This is the familiar master/slave latch implementation of an edge-triggered flip-flop. Note that this design can be easily modified for negative edge-trigger clocking, simply by moving the inverter so that the enable to the first mux is inverted, and the enable for the second is not. This configuration is

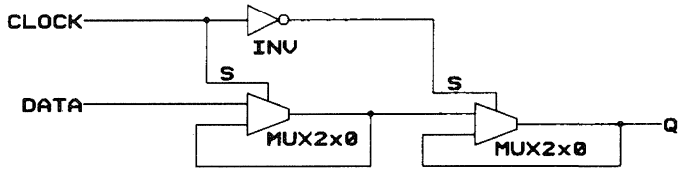


FIGURE 5
Two latches, implemented in multiplexers, are combined in a master/slave configuration to create an edge triggered flip-flop

essentially the pipelined latch configuration that was shown previously, and it is implemented in multiplexer logic, leaving the dedicated register still available. Thus, for cases where one register is feeding another, the pASIC logic cell is capable of integrating two registers into a single cell. Figure 6 shows a cell design for pipelined registers. The first register consists of multiplexer N and multiplexer O in a master/slave latch configuration. This is followed by the dedicated cell register.

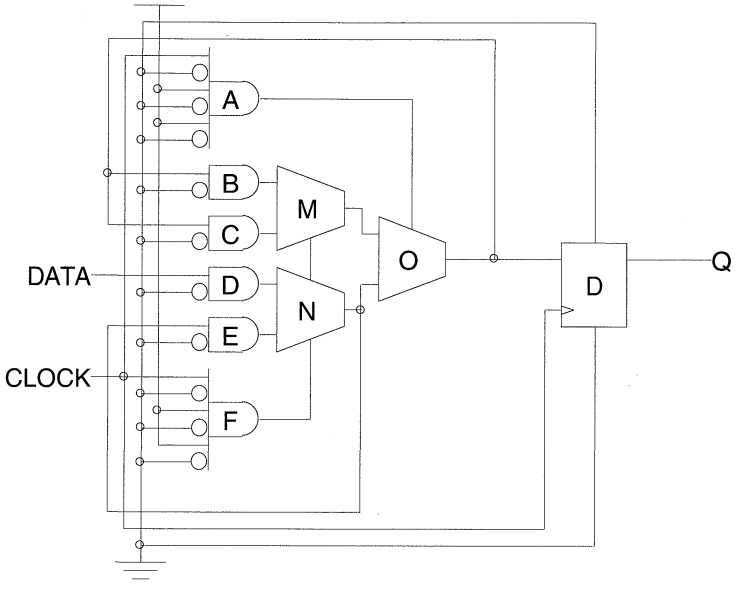
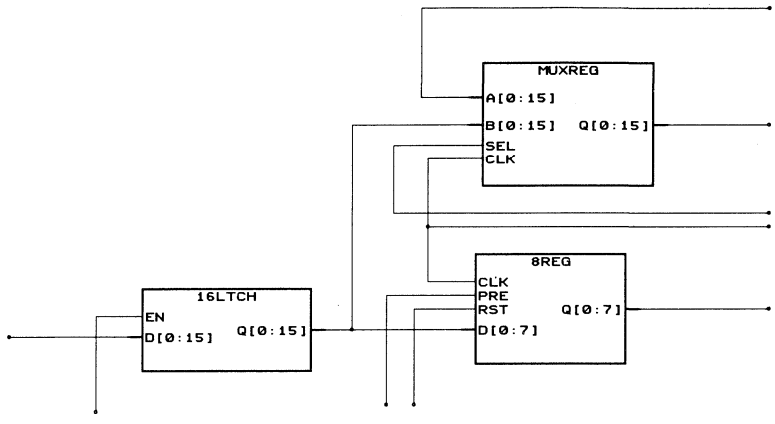


FIGURE 6
A cell design for two edge-triggered registers in a single cell. This allows an 8-bit shift register to be implemented in only 4 logic cells.

A typical example of how this function would be used is an 8-bit shift register. This function is implemented in only four logic cells, thus maximizing the overall gate utilization of the device.

Similarly, designs frequently call for a latch feeding a register. The pASIC Family of FPGAs is capable of integrating this logic into a single logic cell. Conversely, if needed, the logic cell is also capable of handling a design requiring a register feeding into a flow-through latch. Finally, for the designer wishing to take advantage of the full capability of the device, a logic cell can be configured as a pair of latches followed by a register. The register must be fed by one of the latches, but this is not unusual when latching data off a system bus where it is then registered for internal use.

FIGURE 7
A typical 16-bit bus interface, where the data is first latched, and then can be stored in one of two registers



The 16-bit latch plus the 8-bit register can fit into only 8 logic cells, for an average utilization of 22 gates per logic cell.

This QuickNote details a few of the many register and latch functions that can be implemented in the pASIC 1 Family of FPGAs. The flexibility of the logic cell architecture allows for easy integration of multiple latches and registers into a single logic cell, up to 288 in the pASIC QL8X12 1000-gate FPGA. It is this flexibility of the logic cell, combined with the powerful pASIC Toolkit, that allows rapid completion of designs that will operate over 100 MHz, with a fully automatic place and route.

QAN2
Counter Designs in the pASIC Device

HIGHLIGHTS

- **Free running counters** – High-speed counters optimized for binary counting at frequencies in excess of 100 MHz.
- **Counters with added features** – Binary counters with LOAD for data inputs, COUNT ENABLE, UP/DOWN count capability, 3-State output control, synchronous and asynchronous clear inputs.
- **Counter Macro Library** – A comprehensive library of QuickLogic counters exists as ready-made designs for instant systems applications.
- **Counter Design Methodology for pASIC™ devices** – Introduction of techniques to enhance the performance of counter design. How to use look-ahead and pipelined carry to decrease propagation delay between counter modules.

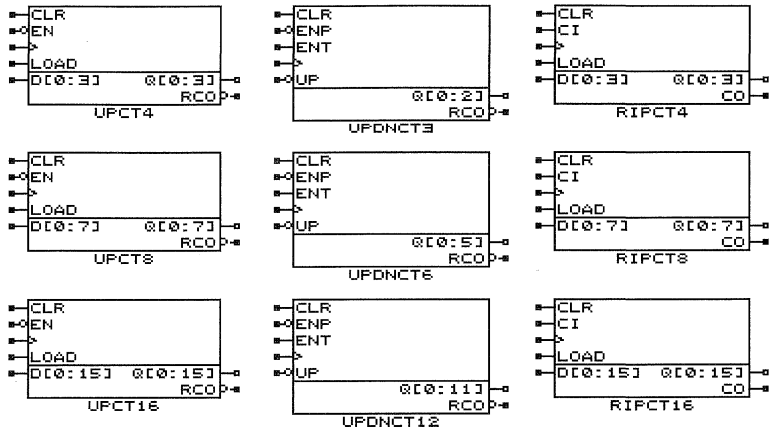
INTRODUCTION

The following application note introduces several high-speed techniques for both counter and state machine design using the QuickLogic pASIC 1 family of FPGA's. This application note uses the QL8x12 1.2 micron device in all of the examples. Since the original publication of this application note, QuickLogic has moved to a 0.65 micron process which has decreased propogation delays by 30%-50% and can achieve worst case counter speeds in excess of 180 Mhz. The techniques displayed here are worthy of merit and the application note has been retained in its original form.

The low-impedance ViaLink™ interconnect element employed in the pASIC device architecture enables higher performance operation than any other FPGA family. This is particularly evident in the design of high-speed counters. The pASIC macro library contains a range of predesigned counters covering a wide variety of needs. Examples of nine of these are shown below. This QuickNote is intended as an overview of alternative approaches to the pASIC design methodology of functions that cannot be satisfied by these counter modules.

QUICKLOGIC

Counters from QuickLogic's Macro Library



Designs in this application note were implemented with the QuickLogic pASIC Toolkit operating under Microsoft Windows™ 3.0 on the PC. This comprehensive set of CAE software includes third-party design entry (ECS from Data I/O, Inc.) and timing and functional simulation (X-SIM from SAS, Inc.). Both operate efficiently and interactively with QuickLogic SpDE place and route, delay modeling and physical viewer tools. The ability to enter and simulate in the same graphical environment provides the user with a quick and efficient way of generating and debugging counter designs.

After SpDE place and route of the pASIC device, timing values may be generated. Compiling net, gate and ViaLink propagation delays provides timing parameters that are a function of the layout and partitioning of the cells in the pASIC device. These timing values can be annotated back into the simulator. Having done this, the designer can evaluate the performance of his counter with regard to maximum count frequency, clock to output, input set up and hold times. If the counter's performance needs improvement the ECS schematic capture environment may be invoked to manually improve the placement of registers or clock input buffers. An optimum placement for counters would have registers placed in a column with clock inputs driven from a clock express line to minimize clock skew.

Figure 1 shows a block diagram of a Moore state machine. The next state of the registered Q outputs is a function of combinatorial inputs gated with the current state of the same Q outputs. A counter is a state machine that conforms to this structure. The state of the inputs combine with the Q registered feedback to provide the next output state. Depending on counter complexity, a design can have combinatorial inputs; CLEAR, HOLD, COUNT ENable, UP/DOWN control, and for loadable counters DATA inputs and a LOAD control.

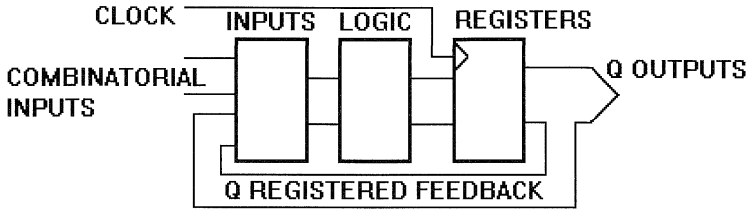


FIGURE 1
Moore State Machine

...State machine design in the pASIC device

Implementation of this type of state machine is well suited to the pASIC device that has internal logic cells comprising logic gates, multiplexers and registers. Universal cell interconnect is possible through vertical and horizontal routing channels and programmable ViaLink sites. Logic and registers combine with interconnect to realize state machines and counters of varying complexity. Position constraints of the registered cells can be entered into the schematic along with the design itself, thus ensuring optimum placement of cell groups.

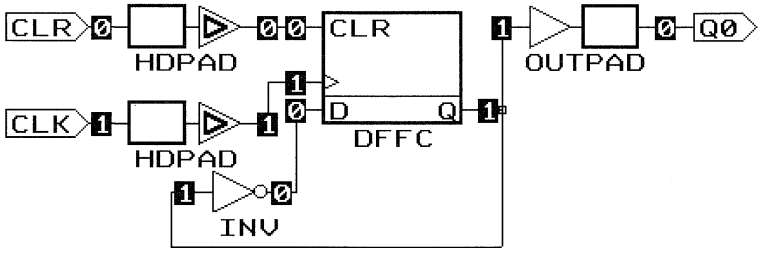
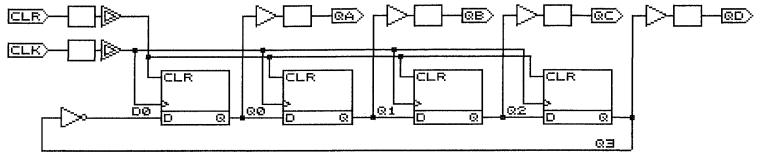


FIGURE 2
Binary Flip-flop

Figure 2 shows a binary flip-flop as entered in the ECS schematic capture system. It conforms to the Moore state machine of Figure 1. The CLR line is used as a direct input to reset the register and the Q0 output is inverted and fed back to drive the D input of the same register which causes it to toggle after each clock rising edge. This circuit forms the least significant bit of a free-running binary counter. To optimize a counter design for maximum performance the designer should take advantage of the high drive input buffer (HDPAD) which rapidly charges/discharges the low capacitance on the dedicated express clock lines. High current drive minimizes clock skew on these lines that cover the entire length and breadth of the pASIC device.

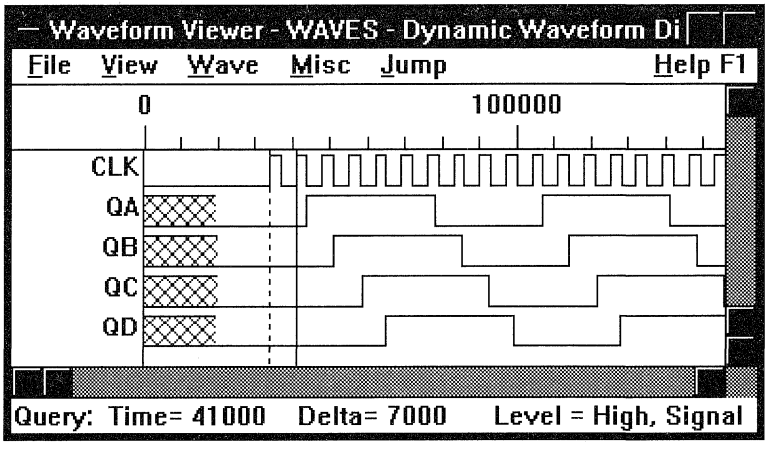
The X-SIM simulator allows logic ONEs and ZEROs to be displayed over the signal lines so the designer can trace the behavior of his system dynamically, as in Figure 2.

FIGURE 3
Four-bit Johnson counter design



The circuit shown in Figure 3 is a Johnson counter implemented in the pASIC device. It consists of four D-type registers linked as a shift register with an inverting feedback from Q3 into the D0 input of register Q0. After the registers have been cleared, the first clock pulse will strobe a logic HIGH into Q0 and a further three clock pulses will propagate that HIGH through the other registers. When Q3 goes HIGH the feedback to D0 will be inverted and a LOW will be clocked into Q0. The subsequent clock pulses will sequentially clock a LOW condition through the shift register as shown in the following simulation. This design comprises four registers and provides eight distinct states through which the counter can transition.

Simulation of the Johnson counter



The advantages of the Johnson counter are found in its simplicity and very high performance. The light capacitive loading on each register output and the lack of combinatorial logic delays in the feedback loop make this design capable of clocking at well over 100 MHz in the pASIC device. A disadvantage is in the number of states through which the counter can transition. If n represents the number of registers in the counter, then the Johnson counter can clock through 2n states as opposed to 2^n found in a binary counter with n registers.

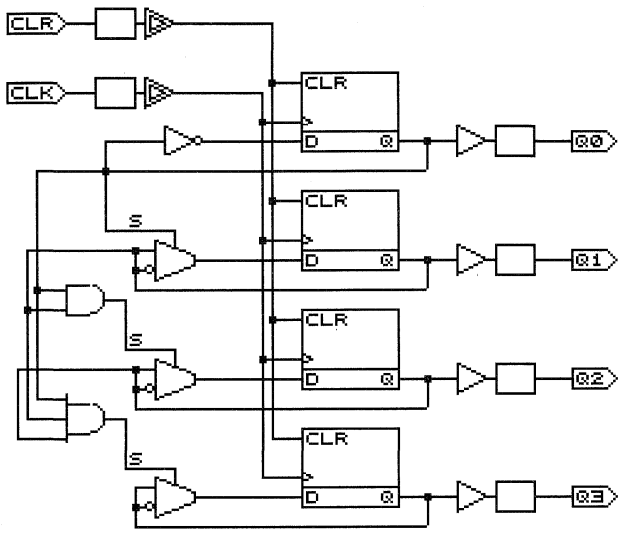
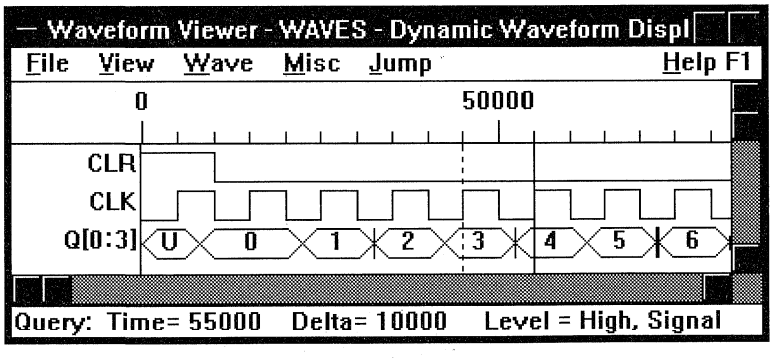


FIGURE 4
Free-running four-bit binary counter with clear

Figure 4 shows a binary counter with an asynchronous CLEAR input. In this design a register is required to maintain or HOLD its current contents until all the lesser significant registers become HIGH. Then the register is required to change its state or TOGGLE after the next clock edge. The least significant stage of this design features the binary flip-flop that is given in Figure 2 and provides the output for Q0.



Simulation waveforms of the four-bit free-running counter clocking at 100 MHz

The TOGGLE and HOLD functions for registered outputs Q1, Q2 and Q3 are achieved with a 2:1 multiplexer feedback. An AND gate control drives the S (Select) input of each multiplexer. When the AND gate output is HIGH, the inverting feedback path through the multiplexer is selected causing the register output to TOGGLE, otherwise a HOLD function is maintained through the non-inverting route. This four-bit counter shows a very efficient use of the pASIC cell because only four of them are required to perform this function. The design when simulated with a clock input with a period of 10,000 pico seconds (10ns), showed 100 MHz counter performance.

A Gray Code sequence allows only one bit in the pattern to change as one state proceeds to the next. This type of encoding can safeguard against simultaneous output driver transition. By definition only one bit can change so only one output buffer will transition after each clock pulse.

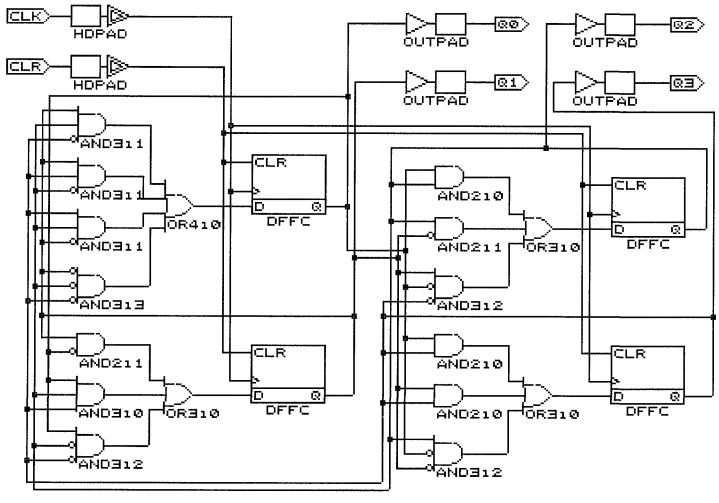
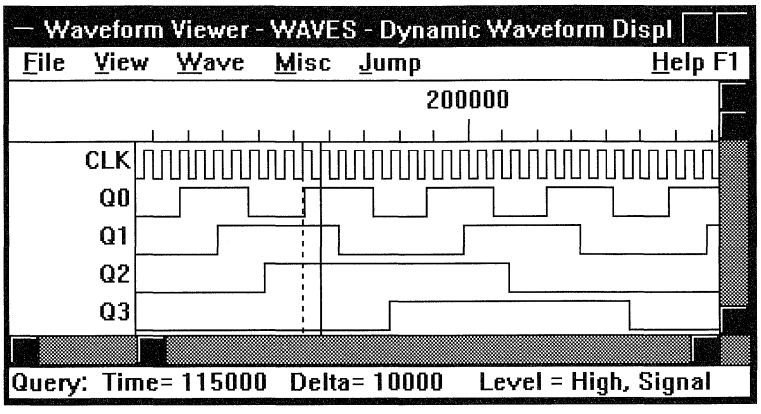


FIGURE 5
Four-bit Gray code counter in pASIC device

Figure 5 shows a Gray code counter as a state machine design. The conventional "sum of product terms" has been used to encode the registered feedback and provide the correct sequence states.



Four-bit Gray code counter simulation

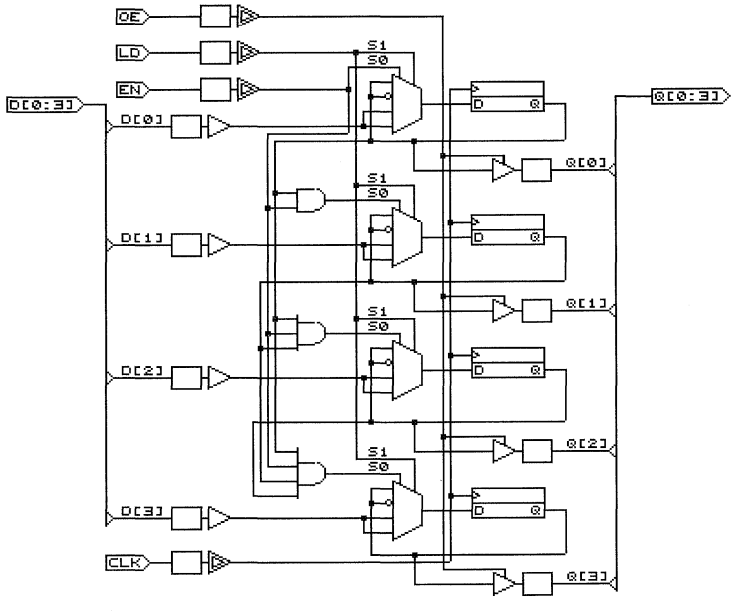


FIGURE 6
Loadable counter with count and output enable

Figure 6 shows a counter having a LOAD/COUNT function selected through 4:1 multiplexers that drive each register in the counter. When the LOAD input is HIGH, the data on the D0:D3 input bus is selected through the multiplexers. The clock rising edge will synchronously load the registers Q0:Q3. To disable a LOAD function, the LD input must be LOW, enabling the COUNT function. The EN input allows the counter to increment or hold. Finally the OE input gives an active HIGH enable to the 3-State output buffers (TRIPAD) given in Figure 6.

Counter simulation load hold 3-State count

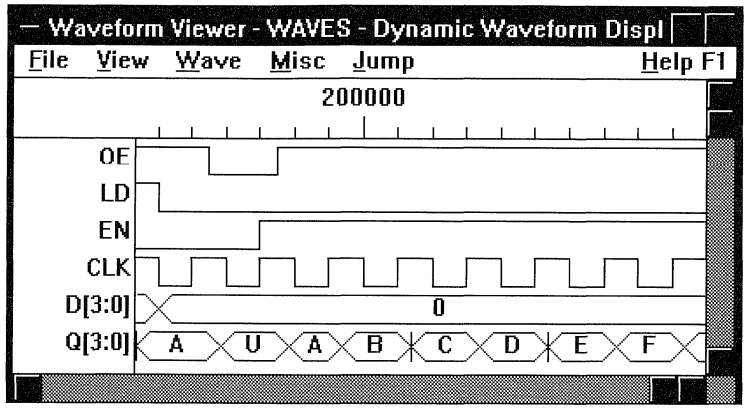
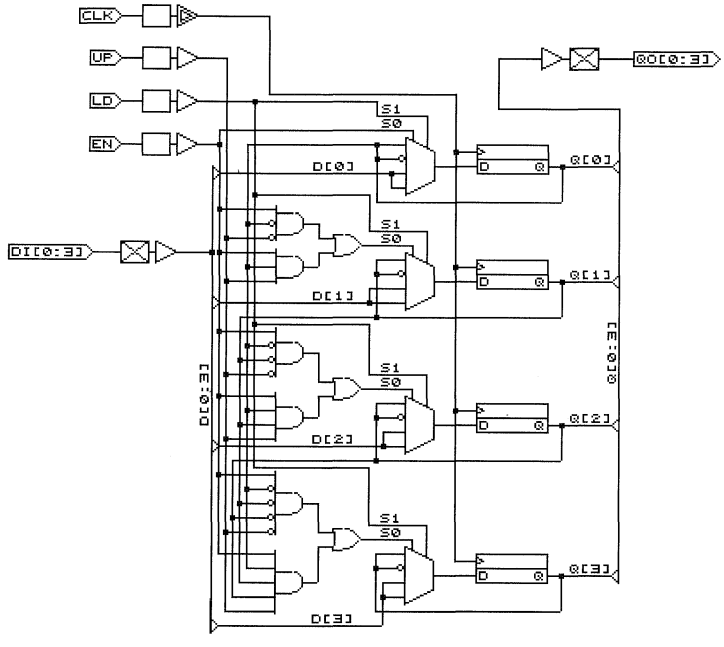


FIGURE 7 Loadable up/down counter



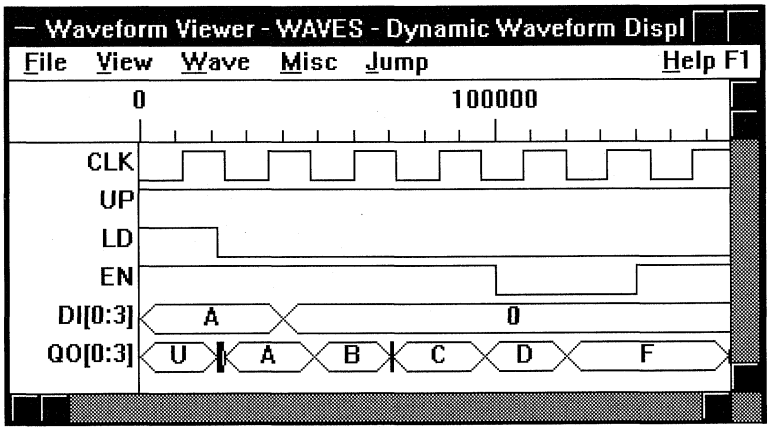
Synchronous loadable up/down counter

Figure 7 shows a binary four-bit loadable UP/DOWN counter with a count enable, EN, input. Data inputs DI[0:3] are tied to a bus and enter the pASIC device through a four-bit wide input buffer. Individual data lines D[0], D[1], D[2], and D[3] drive inputs 3 and 4 of the multiplexer circuits. The LD input,

when HIGH, will select these data lines to drive the D-type register inputs. A synchronous clock loads D[0:3] Data inputs to the registered Q[0:3] outputs. When the registers are loaded, the LD input may be taken inactive LOW. To enable the count function, the EN input must be driven HIGH. This input provides an enabling HIGH to all the AND gates shown in Figure 7. When LOW, this signal maintains a HOLD condition on all four registers. In a binary counter a register is required to TOGGLE after all the less significant registers become HIGH, but this circuit can count either way, UP or DOWN. In a down count a register TOGGLES after all the less significant registers become LOW. The circuit controlling the TOGGLE function of each register comprises a sum of two product terms and drives the S0 input of each multiplexer. One of the two AND gates will be enabled by a logic HIGH on the UP control and the second AND gate enabled for the DOWN count. The UP input is a dual function pin, UP and NOT DOWN. When HIGH, this input selects the UP count and when LOW, selects the down count.

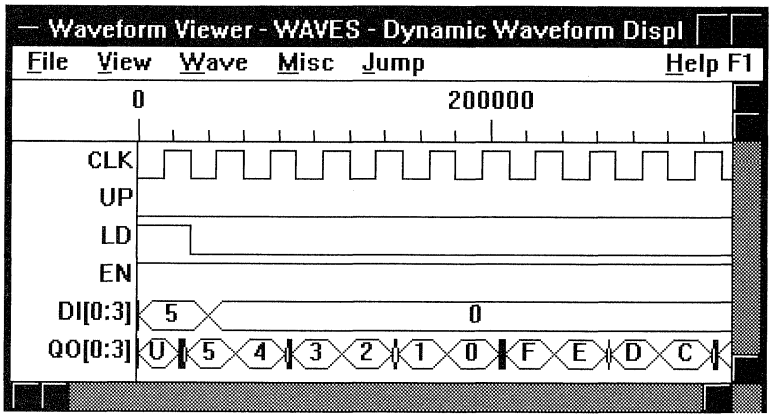
Synchronous loadable up/down counter

It should be noted that in both Figures 6 and 7, buses have been used for data inputs and register outputs. Combining signals on a bus and using components in a group can help simplify the schematic diagrams in complex circuits. For deeper counters such as eight bits, sixteen bits and above, it is recommended that the designer use buses to improve the clarity of the circuit design.



Simulation showing up count, load, count enable

Simulation showing counter load, down count, wrap-around



Eight-bit binary counter with synchronous clear

The design methodology used for integrating deeper counters into the pASIC device employs a technique of interlacing internal registered HIGH and LOW conditions. The wider AND gates in the pASIC cell library have both true and complement inputs up to the widest AND gate which is the AND14i7. This gate has a total of fourteen inputs, seven inverting and seven noninverting. The counter shown in Figure 8 has outputs Q0:Q7 and has been designed such that Q0:Q3 function on LOW logic levels, and Q4:Q7 on HIGHs. When the SCL input is driven active HIGH it will select a logic LOW for registers Q0:Q3, and a HIGH for registers Q4:Q7 via the multiplexer inputs 3 and 4. If registered outputs Q4:Q7 drive inverting output buffers and Q0:Q3 noninverting buffers then all the buffer output pins will be driven LOW. To an external system, this counter is designed to function as a conventional binary counter, but internally the groups Q0:Q3 and Q4:Q7 function on interlaced LOW and HIGH logic group levels, respectively.

During synchronous count operations, the AND gates controlling the TOGGLE function also require interlaced true and complement inputs. Just as Q4:Q7 require inverting output buffers, the same internal TOGGLE control AND gates invert Q4:Q7 signals. An example is given in the TOGGLE control gate for Q6 which combines Q0:Q3 and NOT Q4 and NOT Q5 as an AND gate function of all six input variables.

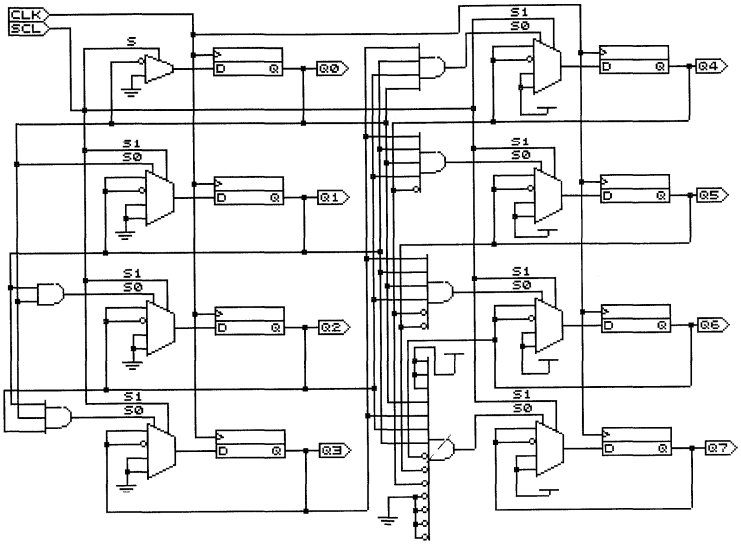


FIGURE 8
Eight-bit binary counter

...Tie unused logic inputs to Vcc or GND

The TOGGLE function to register Q7 uses the 14i7 AND gate. In any circuit design the unused inverting or noninverting inputs must always be tied to Vcc or GND. This is shown for the 14i7 AND gate in Figure 8. Any unused input to a logic cell can be tied to create a permanent HIGH or LOW enable condition for the other signal inputs. The way in which the designer interlaces the logic HIGH and LOW conditions in the counter design is also important. An obvious way would be to consider even outputs Q0, Q2, Q4 ... functioning on logic HIGH levels and odd outputs Q1, Q3, Q5 ... on logic LOWs. This way of interlacing a counter is on a bit-by-bit basis and could cause problems if there were a requirement for the registers to be bussed onto pin driver groups. Interlacing a four-bit-wide group , Q0:Q3, and Q4:Q7 allows a bus of four bits wide to drive a pin driver group four bits in width. Interlacing on alternate bits would prevent the designer taking advantage of the bus feature.

The circuit given in Figure 8 can be made into a symbol and used as an eight-bit counter module or block with CLK and SCL inputs. The design has no count ENable input and is a free-running counter toggling on each clock cycle. A designer can develop deeper counters, 16, 24 and 32 bits by combining eight-bit counter modules. If a 16-bit counter is created from two eight-bit modules, then the higher order eight-bit module must be prevented from incrementing until the lower order counter has reached its final count. An ENable input is an additional control that is required by the higher order byte counter. The counter shown in Figure 8 may be developed to incorporate an additional ENable input. If each TOGGLE AND gate has one additional input to control the TOGGLE or HOLD function, then that control can be used

as the ENable input. An AND gate will be required for the register and multiplexer combination driving Q1. The S0 input of the multiplexer will be selected by Q0 AND the ENable function. For Q0, a four-input multiplexer should be added with the ENable input driving its S0 input and the S1 input being driven by SCL. This multiplexer has the same connections as Q1, Q2 and Q3.

Pipelined carry generation

The way in which a carry output is generated, from the lower order eight-bit counter, and propagated to the ENable input of the next stage is crucial in determining the performance of the two stages. One method would be to detect the final count of the lower order counter and feed an enabling signal to the next stage. This requires one clock period for gating and propagation. A more subtle approach would be to detect the count value prior to the final count. This penultimate carry bit is generated one clock pulse before the last count, so a D-type register is required to delay its propagation to the next eight-bit stage. Carry generation and propagation can be extended over two clock cycles, effectively halving the delay path.

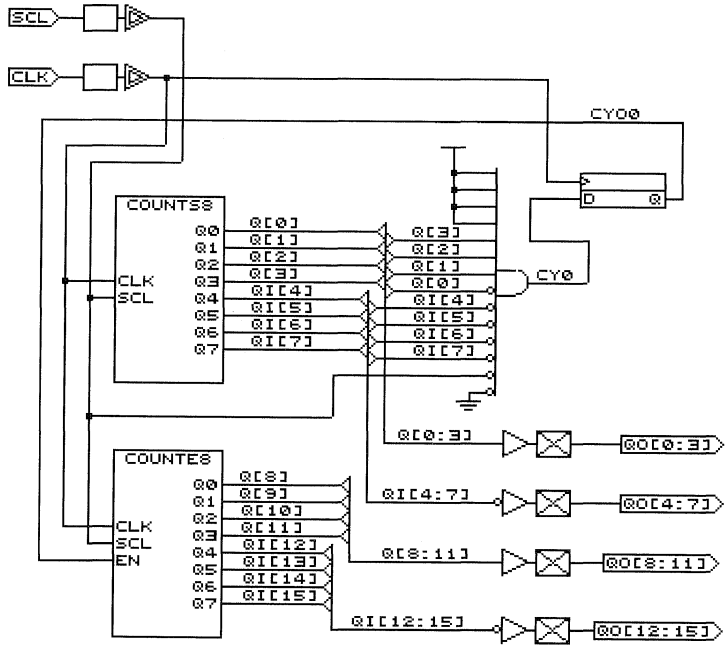


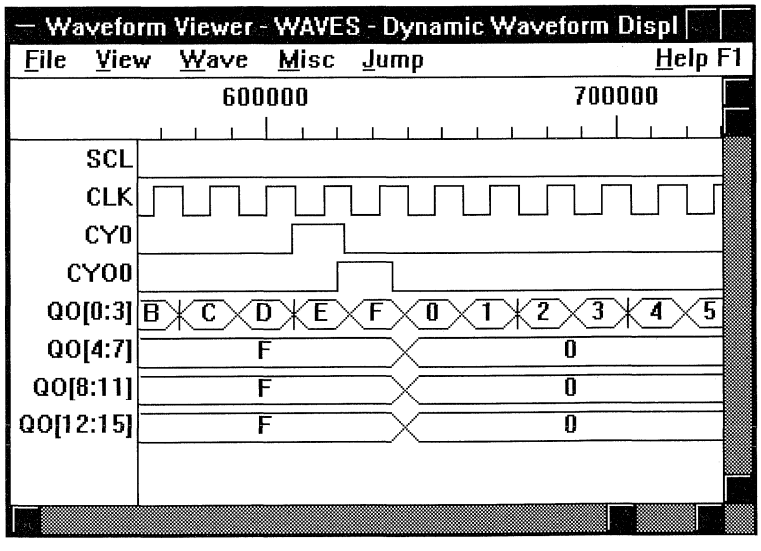
FIGURE 9
16-bit counter from two eight-bit modules and a pipelined look-ahead carry

One AND gate and a D-type register combination is used to decode and generate and pipeline the carry CY0/CYO0. This is decoded from the least significant eight-bit counter output bus. The technique is also called 'carry anticipate' and its application is to reduce the carry propagation delay to the input of the next counter stage. The eight-bit counter module in Figure 8 is given in Figure 9 as a macro COUNTS8. An additional eight-bit counter has been added to provide outputs QO8-QO15. The macro COUNTS8 is identical to COUNTE8 apart from one additional input. The EN input is an active HIGH count enable input and is driven from the look-ahead pipeline carry register. This counter will HOLD its current contents when EN is LOW and increment when EN goes HIGH. From Figure 9 the bus nets Q[0:3] and Q[8:11] are non-inverted while QI[4:7] and QI[12:15] give inverted outputs. The output buffers correct the logic polarity to provide a conventional binary output code.

When considering the performance of this counter design it should be noted that the internal set up time for the registered outputs Q[8:11] and QI[12:15], of module COUNTE8, is 256 clock cycles. The propagation of the carry bit then becomes the critical performance factor in the linking of the two counters.

...Carry generation and propagation is critical in determining performance

The simulation shows the sixteen-bit counter outputs and the look-ahead carry CYO which decodes the hexadecimal count 'FFFE'. The D-type register delays this signal by one clock edge, so its output CYO0 goes HIGH when the counter reaches 'FFFF'. The counter outputs "roll over" to '0000' after the next clock transition and a logic zero is clocked into the carry register. So carry generation and propagation takes two clock cycles.



Simulation of the 16-bit counter showing the pipelined carry

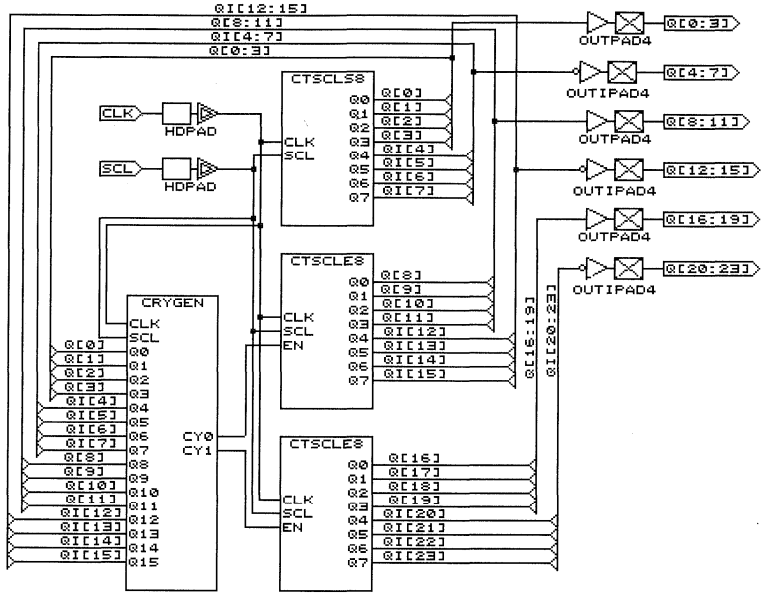


FIGURE 10
24-bit counter form three eight-bit modules

Look-ahead carry for a 24-bit counter

Figure 10 takes the counter depth to 24 bits using a similar look-ahead technique. In this example a carry generator macro CRYGEN has been created to decode the count before the penultimate count. If 'FD' hexadecimal is decoded from the least significant byte, as opposed to 'FE' then two registers are required to pipeline the early carry. Its propagation delay is then extended to three clock cycles increasing the time available to enable the higher order counters. The higher order byte counters in the chain have a set up time through internal feedback of 256 count periods. This makes the look-ahead carry generation propagation delay the significant determining factor in the counter's overall performance.

For a 32-bit design, the designer could decode 'FC' from the least significant eight-bit stage and pipeline the carry through three registered stages if it gives any benefit to the system performance. In a design that uses the technique of a pipelined carry it should be noted that a synchronous clear should flush the carry registers as well as the counter registers. The CRYGEN MACRO has a SCL (synchronous clear) input to perform this function, Figure 11.

Additional considerations

For deep loadable counters that use look-ahead carry, the designer should distinguish between a count and a load function. When the count ENable is valid, the carry bit is generated from the look-ahead condition at the registered outputs. For a LOAD, the carry input is created from a HIGH condition on all the data inputs of the lesser significant stage.

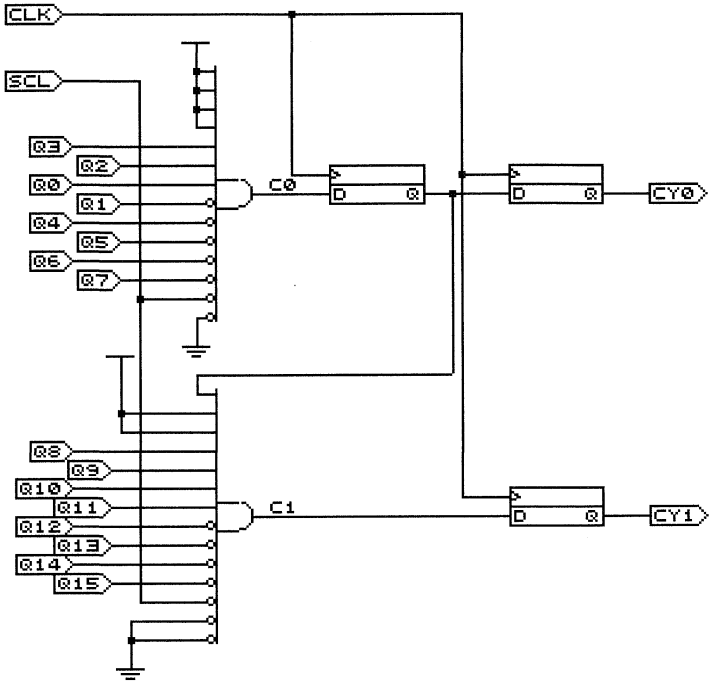
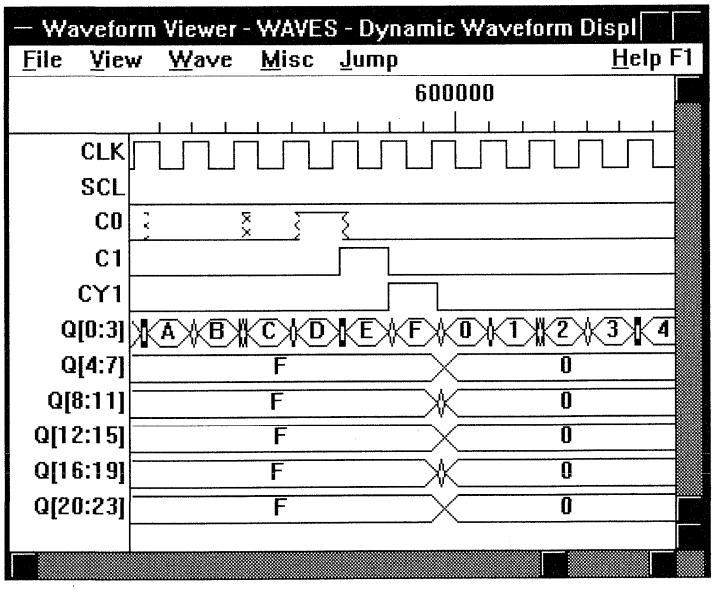


FIGURE 11
Pipelined look-ahead carry generator CRYGEN



Simulation of the 24-bit counter

QAN4
Fast Accumulators

There are many methods of designing adders and accumulators. The style adopted for the QuickLogic accumulators is called conditional sum addition. This style of adder takes advantage of the versatility of the QuickLogic Logic Cell, and incorporates a variety of high-speed design techniques.

...accumulators operate between 50 and 75 MHz

The conditional sum technique used and improvements made in the design to optimize both speed and density are presented in this application note. The QuickLogic macro library features accumulators using these design techniques. These macros operate between 50 and 75 MHz.

Before going into the details of the design, it is best to introduce the basic building block of the conditional sum adder. With traditional adders, a long carry chain caused very long propagation delays. In the conditional sum adder two "assumed" sums and carries are generated for every bit of the adder -- one assuming the carry-in is one, the other assuming the carry-in is zero. This allows for faster propagation, since the carry from lower bits only is needed in the last stage of the adder to select the proper sum. The basic building block of the adder is shown in Figure 1.

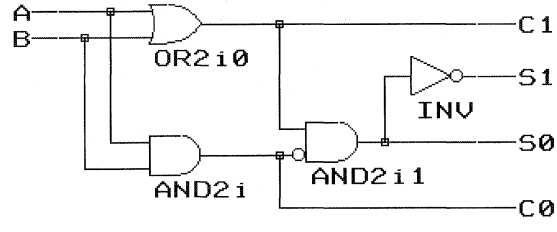
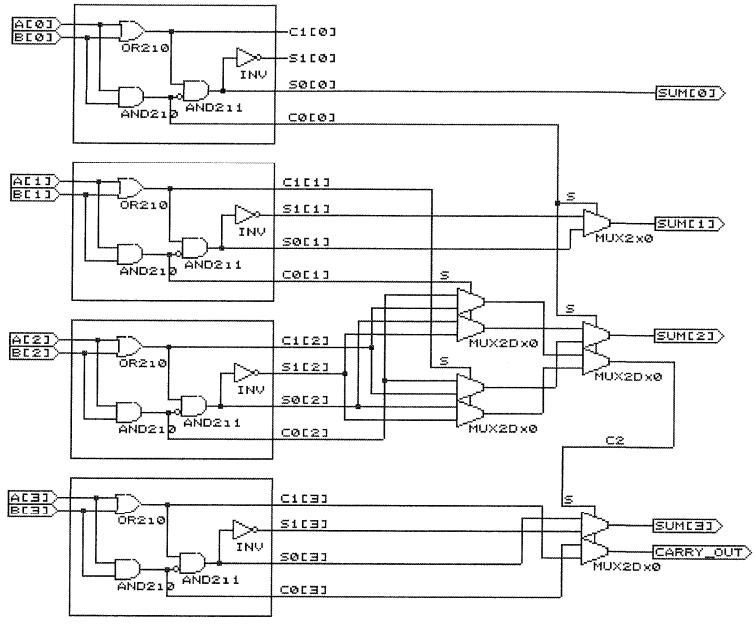


FIGURE 1
Conditional Sum Building Block

In this logic circuit, C0 is the carry if the carry-in (from all previous bits) is 0, and C1 is the carry if the carry-in is 1. The sums are labeled in the same manner. The conditional sum technique is demonstrated fully in the first stage of development for a four-bit adder shown in Figure 2. At this point, the adder requires 17 cells and is four cell-levels deep in the critical path.

QUICKLOGIC

FIGURE 2
Starting Design



Notice that the design in Figure 2 has already been slightly optimized by the proper selection of macro library components — a dual 2:1 MUX was used wherever two 2:1 MUXs have the same select line. This MUX2Dx0 macro fits two 2:1 MUXs into a single logic cell.

The SUM[1] output is generated by selecting between the two sums (S1[1] and S0[1]) based on the value of the carry from the zero bit. The same basic logic works for higher bits. Sums and carries are generated for each bit of the adder, and then a series of multiplexing is performed to determine which sum or carry is appropriate to propagate to the output.

For the first stage of optimization, look for opportunities to move bubbles in order to pack more logic into the logic cell. Notice in the boxed areas of Figure 2 that an OR2i0 and an AND2i0 are feeding an AND gate. By changing the OR to a NOR and bubbling the appropriate input of the AND gate, the QuickLogic Packer tool will be able to combine all three gates into a logic cell. (Remember that the NOR2i0 is the same as the AND2i2 by DeMorgan's Theorem, so the NOR gate will be able to fit into the AND fragment.) Figure 3 demonstrates these changes and shows the symbol CSABIT that will be used to represent this logic.

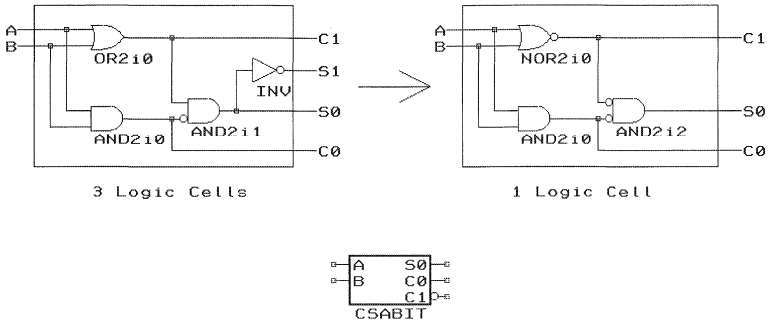


FIGURE 3
Optimizing the Adder Building Block

By using a NOR gate instead of an OR gate for C1 we create the inverted output. This is satisfactory, however, because the inversion can be picked up at a later stage with a bubbled input. Also, the inverter creating S1 is unnecessary, since the inverted signal (S1) can be obtained by using the noninverted signal (S0) with a bubbled input. The use of the CSABIT macro results in the circuit of Figure 4. This circuit uses nine logic cells and has a critical path through four logic cell levels.

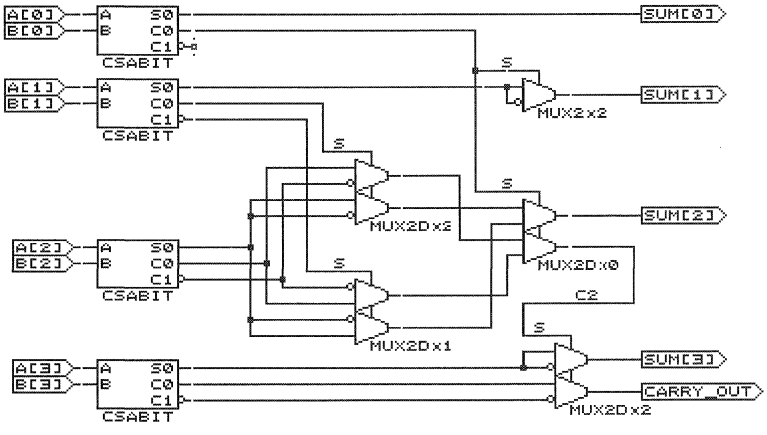
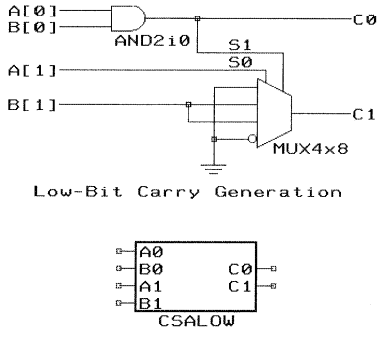


FIGURE 4
4-Bit Adder with the CSABIT macro

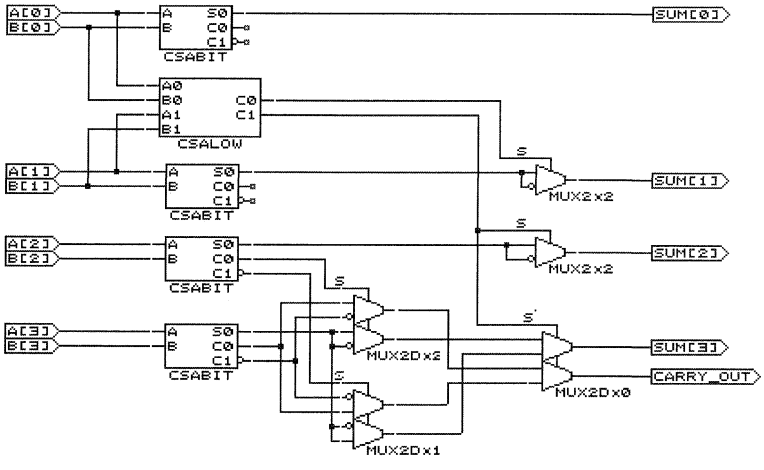
The next step in the optimization process entails a good understanding of the QuickLogic Logic Cell . Since a carry-in is not used in this accumulator, it is possible to produce the true carry from the first two bits in one Logic Cell, as shown in Figure 5.

FIGURE 5
Generating the true C0 and C1 in one cell



In the CSALOW macro, C0 is the carry from just bit 0 of the input, C1 is the carry from bits 0 and 1. The AND gate will be packed into the same cell as the MUX by the Packer, since the two input AND can easily fit into one of the AND fragments. The effect of this early generation of C1 is a 'shifting' of the bits of the adder resulting in the new schematic of Figure 6. In this form, the design uses nine logic cells and goes through a maximum of three logic cell levels.

FIGURE 6
"Shifted" 4-bit Adder Using CSALOW



Notice that the worst case path in this design must go through three logic cells. This would be through the inputs A[2],B[2],A[3], or B[3], then through the CSABIT cell, through two cell levels of dual two-input MUXes, and to the outputs SUM[3] or CARRY_OUT. We could greatly improve the speed of this design if we could reduce this worst case path to two logic cells. One method of achieving this is to modify this design so that it does not create a carry out. This results in the logic for SUM[3] shown in Figure 7.

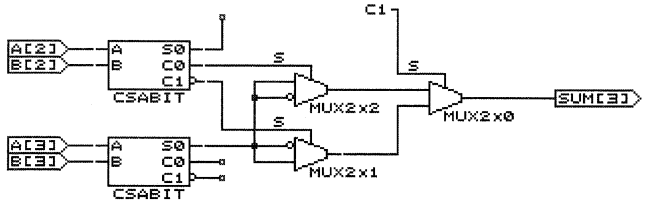


FIGURE 7
Removing the Carry-Out

The two multiplexers in the middle of Figure 7 share common inputs — the sum (of A[3] and B[3]) and its inversion. This configuration allows us to exchange the MUX inputs and the select lines such that we end with two multiplexers with a common select line, as shown in Figure 8. The entire logic structure can then be shown to reduce into a single 4:1 MUX. Figure 8 demonstrates this optimization.

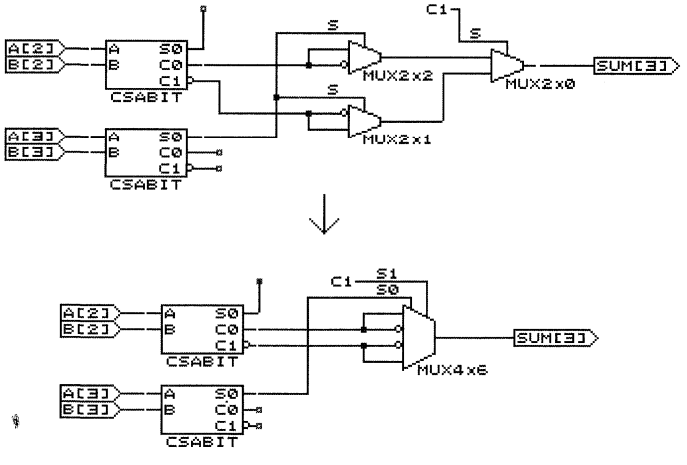
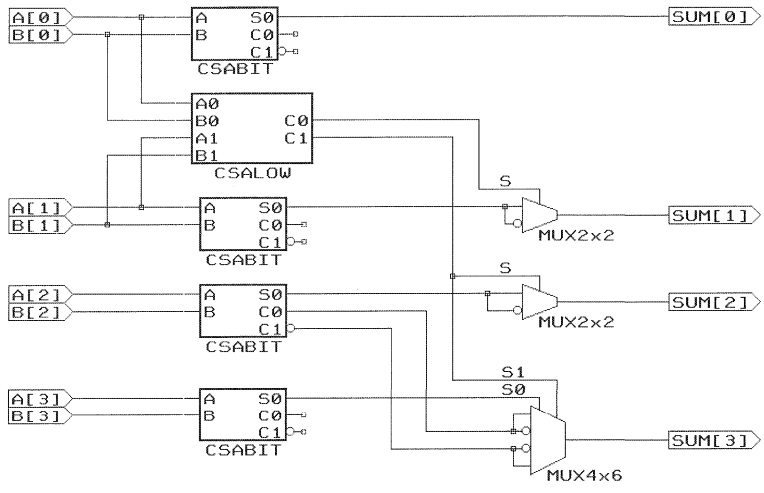


FIGURE 8
Generating SUM[3] in two logic cell levels

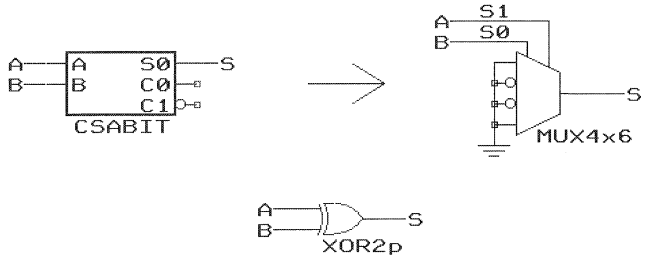
After this change, the circuit appears as in Figure 9. This circuit uses eight logic cells and has a critical path through two logic cell levels.

FIGURE 9
4-bit Adder with SUM[3] optimized



This may appear to be the stopping point for optimization, because two cell levels is the minimum in which a four-bit adder can be implemented. However, there is still an opportunity for improvement. Each of the CSABIT symbols has a fanin of 2 for both the A and B inputs. We can reduce the fanout of the [0], [1], and [3] bits by using an XOR2p macro (an optimized XOR gate). Only the XOR sum function is needed for these bits since the C0 and C1_ outputs of the CSABIT macro are not used. The XOR2p macro (shown in Figure 10) is performance optimized by reducing the fanout on the inputs to one and always taking the fastest path through the Logic Cell (through the AND fragments instead of the MUX fragment).

FIGURE 10
The XOR2p macro



The XOR2p gate can also be used in place of the MUX's driving SUM[1] and SUM[2] because these MUXes are performing XOR functions.

The final design is shown in Figure 11. Implemented in a QL 12x16 device this critical path is 11 ns (worst-case commercial conditions). The optimized design uses eight cells and the critical path passes through no more than two logic cell levels.

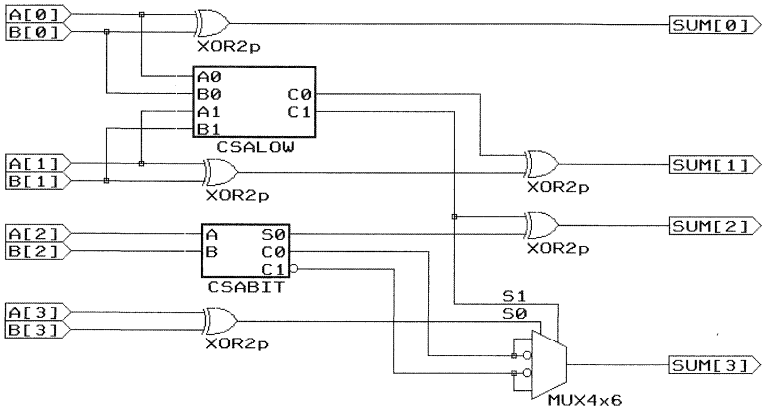


FIGURE 11
4-bit Adder - Final Design

Notice that this adder can be changed into an accumulator by simply putting D flip-flops on the outputs and tying the Q's of the flip-flops back to the B inputs. The last level of logic will fit into the same logic cell as the flip-flops, so density is not compromised.

The design optimization techniques used in this accumulator can be applied to any design —

- Reduce fanout
- Implement critical functions in a minimum number of logic cells
- Remove inverters and push bubbles
- Use efficient macros, such as the dual 2:1 MUX
- Determine that the longest path is optimized for speed

QAN5
DRAM Controller for the TI TMS32C30

Mike Dini

INTRODUCTION

This application note describes the key functions and design considerations for a DRAM controller optimized for the Texas Instruments TMS32C30-28 Digital Signal Processor. A system block diagram implementing the design in a QuickLogic QL8x12B high-speed FPGA is shown in Figure 1.

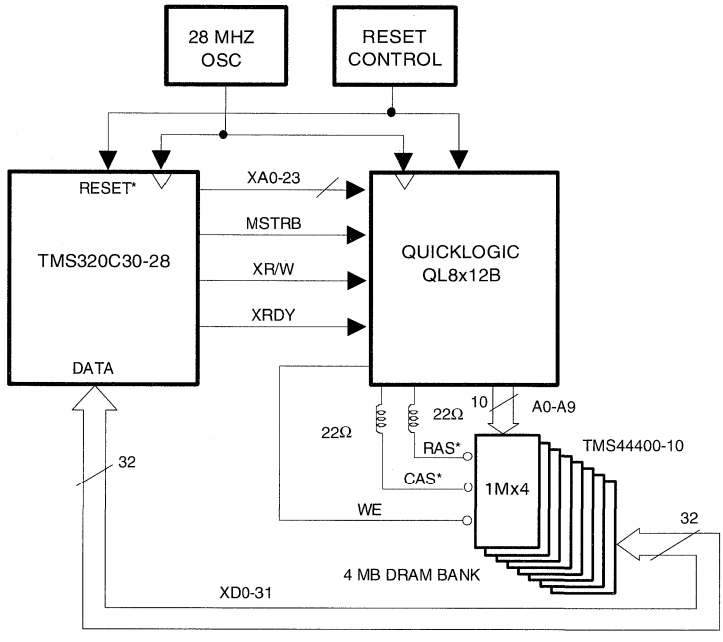


FIGURE 1 System Block Diagram of a DRAM Controller for the TMS32C30 Using the QL8x12B

DRAM OVERVIEW

Dynamic RAM devices are available in a wide variety of sizes and organizations. The 4-Megabit chip alone is offered in 4Mx1, 1Mx4, 512Kx8 and other configurations; 16-Megabit and larger chips will soon be available and in even more varieties. DRAMs can currently be accessed in a variety of modes: page mode, nibble mode, or static column mode. Each of these options differ in how the data is accessed within a row after the assertion of the Row Address Strobe (RAS*) signal. Thus, the data bandwidth can be significantly increased depending upon the specific application and mode incorporated. It is unlikely that integrated circuit vendors will create standard DRAM controllers to efficiently serve all of these different possible configurations.

QUICKLOGIC

DRAM addresses are multiplexed to reduce the number of package pins required, thus minimizing cost. The DRAM timing cycle begins when the first set of addresses is applied to the multiplexed address pins and the RAS* signal is asserted. After a minimum hold time tRAH (t row address hold) the addresses are switched and the Column Address Strobe (CAS*) signal is asserted. During a read cycle, data appears on the output pins after a time tCAC (CAS* access time) or tRAC (RAS* access time), depending on whether the time tRCD (RAS* to CAS* delay) was satisfied. The cycle terminates when both RAS* and CAS* are de-asserted. A delay of tRP (RAS* precharge) is required to let the chip recover before another access can be initiated.

The 4-Megabit DRAM requires a refresh cycle to be executed on every row every 16 milliseconds. In addition, any access to a memory location refreshes that row of memory. All 1-, 4-, and 16-Megabit DRAMs have a refresh option called CAS* before RAS*. To serve this option a counter internal to the DRAM keeps track of the row status and automatically increments the row address to be refreshed after each CAS* before RAS* refresh cycle. This eliminates the need for an external counter as required in the RAS* Only Refresh mode. RAS* Only Refresh is a holdover from earlier DRAM architectures and is not recommended for new designs.

OPTIMIZED DRAM CONTROLLERS

To take into account the logic and timing requirements described above, a DRAM controller must support the basic functions shown in the block diagram of Figure 2. Key elements include:

- Address Multiplexers to switch between the row and column addresses.
- A Refresh Counter to request refresh cycles.
- A RAS*/CAS* Control state machine to arbitrate between processor read/write demands and refresh cycles, as well as to generate the appropriate cycling of the RAS* and CAS* signals.
- An Address Decoder to determine if the processor has correctly addressed the memory chip.
- Address Latches are also required for processors, such as the Intel 80x86 family, with multiplexed address and data lines.

Speed is of the utmost importance in designing memory controllers. The key issue in designing an optimized DRAM controller is to keep the total propagation delay through these blocks as short as possible such that unnecessary wait states need not be inserted into the processor cycle. Until recently, such optimized DRAM controllers were designed using several low-density TTL packages, a delay line, and the fastest PAL devices available. Today this functionality and speed can be packed into a single QuickLogic high-speed FPGA with plenty of room to include other logic functions. Power consumption and printed circuit board space are reduced dramatically and significant cost savings can be realized. Design and debug is faster and easier with QuickLogic's FPGAs because of the low-cost, interactive CAE tools available to support the QuickLogic devices.

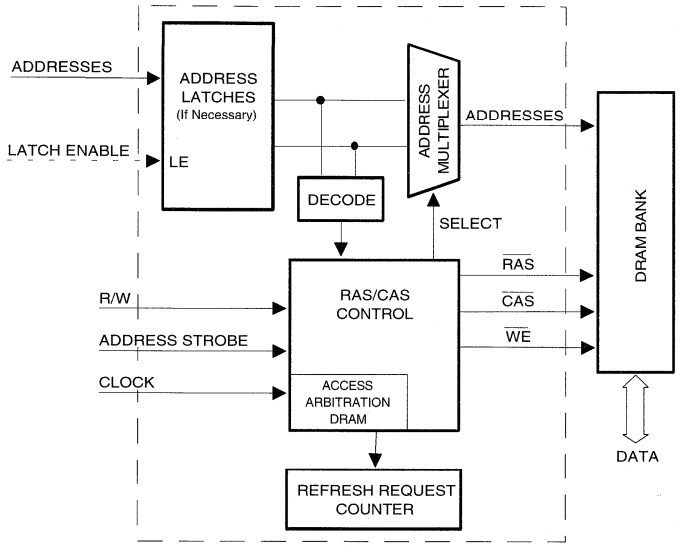


FIGURE 2
Basic Functions of a DRAM Controller

A 4-MEGABIT DRAM CONTROLLER FOR THE TMS32C30

The remainder of this application note describes the design of a 4-Megabit DRAM controller for the 28 MHz version of the TMS32C30 Digital Signal Processor based on the general block diagram of Figure 2.

Memory read and write timing diagrams shown in the TMS32C30 literature reference all timing to clock signals H1 and H3 which are derived from dividing the processor clock, CLKIN, by two. As it is most straightforward to sequence a DRAM controller from the CLKIN signal and because the QuickLogic FPGAs can easily run at the processor clock rate, this is the recommended design approach. It is therefore necessary to create a timing diagram that shows the read and write timing relationships between CLKIN, H1 and H3 (Figure 3).

During a read operation, the TMS32C20-28 must have valid data within 42 nanoseconds of Master Strobe signal (MSTRB) being asserted. This cannot be achieved using a standard DRAM timing cycle. One wait state must be added to both the read and the write cycles. All possible timing cycles to satisfy these requirements are shown in the waveforms of Figure 3. Figure 3a shows read/write cycles for the processor. CAS* before RAS* refresh cycles are illustrated in Figure 3b. A hidden refresh cycle—a read or write cycle immediately followed by a CAS* before RAS* refresh cycle—is shown in Figure 3c. As read and write cycles differ only in the polarity of WE* (write being active low and read being high), the timing relationships between RAS* and CAS* remain the same for both read and write operations.

FIGURE 3a Read/Write Waveforms for the TMS32C30 Processor

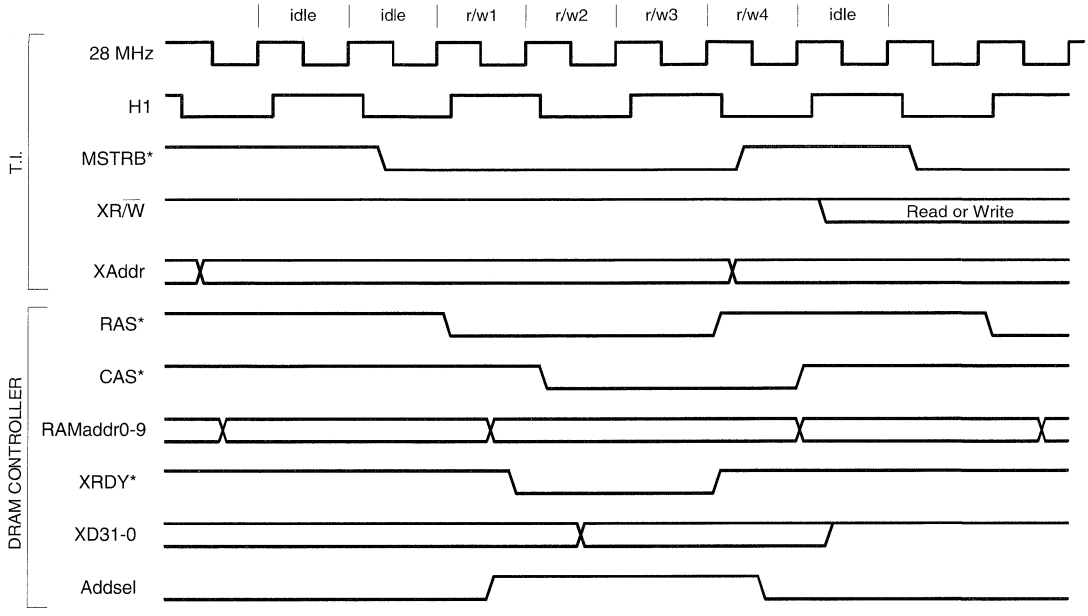


FIGURE 3b CAS before RAS Refresh Waveforms

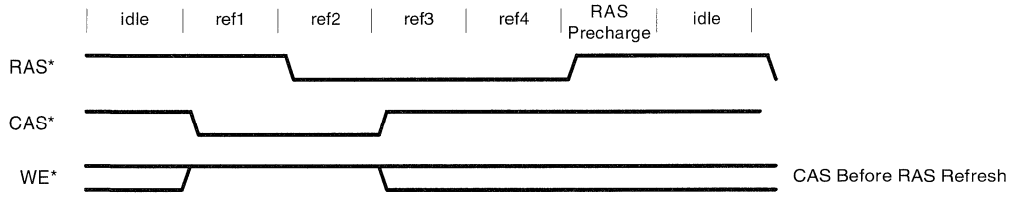
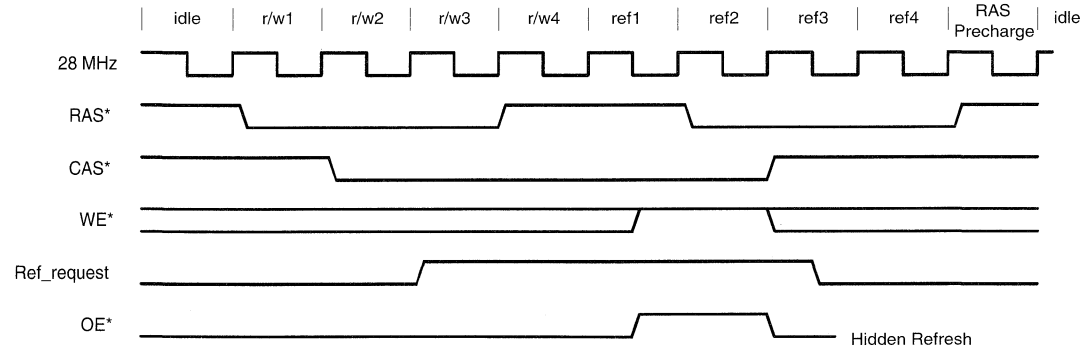


FIGURE 3c Hidden Refresh Cycle Waveforms



From these timing waveforms the DRAM Access State Transition Diagram for the RAS*/CAS* Controller State Machine is determined in Figure 4a. This is one of the more difficult tasks in the design. Because of the numerous timing parameters and restrictions in memory access, enormous care must be taken in generating the RAS* and CAS* signals. Timing must be checked carefully and thoroughly to verify that all timing constraints are adhered to and that no violations occur. Figure 4b displays the state definitions for the various states and indicates the active signals occurring in each state.

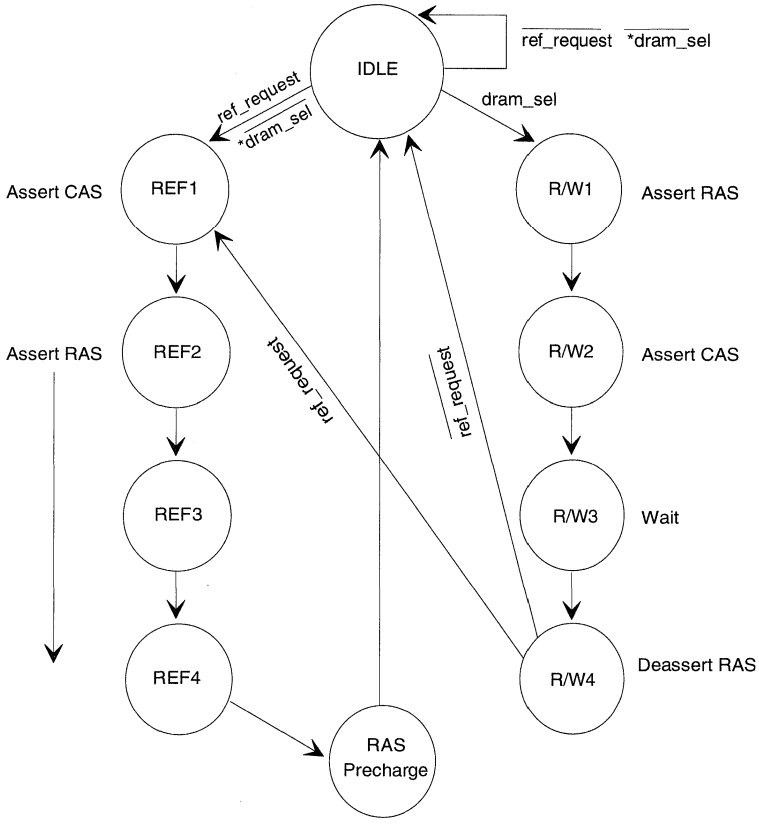


FIGURE 4a
DRAM Access State Diagram for TMS32C30 DSP Processor

Idle:	No activity; RAS* & CAS* deasserted
R/W1:	RAS* asserted
R/W2:	RAS* & CAS* asserted
R/W3:	Wait state; RAS* & CAS* asserted
R/W4:	CAS* asserted; RAS* deasserted
REF1:	CAS* asserted; WE* deasserted
REF2:	RAS* & CAS* asserted; WE* deasserted; Ref_Request deasserted
REF3:	RAS* asserted; CAS* deasserted
REF4:	RAS* asserted
RAS PRE:	Precharge; RAS* & CAS* deasserted

FIGURE 4b
State Definition for DRAM State Machine

TABLE 1
Data I/O ABEL File For State Machine

```
     module dramcntl
     title ' Mike Dini  27 Jun 1992'

                H,L,Z,C,X = 1,0,.Z,.C,.X.;
"inputs:
     clock                                                    pin;
     ref_request                                              pin;
     dram_sel                                                 pin;

"outputs:

     r_w_1,r_w_2,r_w_3,r_w_4                                  pin;
     refresh_1,refresh_2,refresh_3,refresh_4                  pin;
     ras_precharge                                            pin;
     idle                                                     pin;
     r_w_1,r_w_2,r_w_3,r_w_4                  ISTYPE    'REG_t , BUFFER';
     refresh_1,refresh_2,refresh_3,refresh_4  ISTYPE    'REG_t , BUFFER';
     refresh_2                                ISTYPE    'REG_t , BUFFER';
     refresh_3                                ISTYPE    'REG_t , BUFFER';
     refresh_4                                ISTYPE    'REG_t , BUFFER';
     ras_precharge                            ISTYPE    'REG_t , BUFFER';
     idle                                     ISTYPE    'REG_t , BUFFER';

dram_control - [r_w_1,r_w_2,r_w_3,r_w_4,refresh_1,refresh_2,refresh_3,
                refresh_4,ras_precharge,idle]

     s_idle           = [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];
     swr_w_1          = [ 1, 0, 0, 0, 0, 0, 0, 0, 0, 1];
     swr_w_2          = [ 0, 1, 0, 0, 0, 0, 0, 0, 0, 1];
     swr_w_3          = [ 0, 0, 1, 0, 0, 0, 0, 0, 0, 1];
     swr_w_4          = [ 0, 0, 0, 1, 0, 0, 0, 0, 0, 1];
     srefresh_1       = [ 0, 0, 0, 0, 1, 0, 0, 0, 0, 1];
     srefresh_2       = [ 0, 0, 0, 0, 0, 1, 0, 0, 0, 1];
     srefresh_3       = [ 0, 0, 0, 0, 0, 0, 1, 0, 0, 1];
     srefresh_4       = [ 0, 0, 0, 0, 0, 0, 0, 1, 0, 1];
     sras_precharge   = [ 0, 0, 0, 0, 0, 0, 0, 0, 1, 1];

equations

     "r_w_1.CK         = clock;
     "r_w_2.CK         = clock;
     "r_w_3.CK         = clock;
     "r_w_4.CK         = clock;
     "refresh_1.CK     = clock;
     "refresh_2.CK     = clock;
     "refresh_3.CK     = clock;
     "refresh_4.CK     = clock;
     "ras_precharge.CK = clock;
     "idle.CK          = clock;

     @DCSET
state_diagram dram_control
     state s_idle :
        IF  dram_sel THEN  sr_w_1
        ELSE IF ref_request THEN srefresh_1
             ELSE s_idle

     state sr_w_1 :   GOTO  sr_w_2;
     state sr_w_2 :   GOTO  sr_w_3;
     state sr_w_3 :   GOTO  sr_w_4;

     state  sr_w_4 :
        IF  ref_request THEN srefresh_1
        ELSE s_idle;

     state srefresh_1 :  GOTO   srefresh_2;
     state srefresh_2 :  GOTO   srefresh_3;
     state srefresh_3 :  GOTO   srefresh_4;
     state srefresh_4 :  GOTO   sras_precharge;
     state sras_precharge : GOTO s_idle;

end
```

The state transition diagram is then translated into an equation description format. Table 1 is a Data I/O ABEL language version of the state machine. At the time of writing no mechanism was available to input ABEL equations directly into the QuickLogic SpDE place-and-route tools. This application note therefore shows the equations used as the source document for drawing logic schematics. This is quite straightforward and is still the preferred approach for many designers. However, with the subsequent release of ABEL-FPGA and Exemplar support for the QuickLogic architecture, equation input can now be used directly if so desired.

MAJOR CONTROLLER FUNCTIONS

The three key controller functions are the RAS*/CAS* Controller State Machine, the Refresh Counter, and the Address Multiplexing and Decoding blocks. Design considerations for these functions are outlined in this section.

The State Machine

The RAS*/CAS* state machine was designed using the One-Hot state encoding technique. This technique employs a one flip-flop per state in building the state machine versus the traditional binary or gray code encoding method. By using this method, the strengths of the QuickLogic architecture are exploited and the register-rich environment is utilized, enhancing speed and efficiency. Thus, a performance-optimized design is achieved with no state decoding logic required and reduced levels of logic to each flip-flop.

The circuit schematic (drawn in the Data I/O ECS schematic capture package included in the QuickLogic pASIC toolkit) for the One-Hot encoded RAS*/CAS* state machine is shown in Figure 5. One common concern in the designing of state machines deals with illegal conditions corrupting state-dependent operations. This occurs whenever invalid or undefined states exist. Therefore, it is crucial to avoid such situations. Initialization is onc arca in which this may occur. To ease the task of initializing the state machine, the idle state is defined as having all the flip-flops cleared. Therefore, by simply clearing the flip-flops, the state machine is re-initialized to idle. By definition of One-Hot, another illegal condition is whenever two or more flip-flops are active. A reset pin on both the TMS32C30 and the QL8x12B Controller provides a means to rectify any illegal conditions. The Reset signal is synchronized to the 28 MHz clock. Following good design practice, every flip-flop in this circuit is set to a known state at power up and during reset.

Refresh Signal Generation

The 100 nanosecond, 4-Megabit (1Mx4) DRAM used for this design requires that all 1024 rows be refreshed every 16 milliseconds. This is satisfied by executing a CAS* before RAS* refresh cycle every 15.6 microseconds, or every 437 (calculated by 0x1b5) 28 MHz clock periods. The refresh-request, free-running counter shown in Figure 6 sets a D-type flip-flop on its terminal count. The flip-flop output, Ref-request, signals the state controller that a refresh cycle is required. If the state machine is in an idle state, a refresh cycle is started immediately. If both a refresh cycle and a processor access cycle have been requested, the processor has priority and a read or write cycle is executed, followed by a hidden refresh cycle on

completion of the processor access. If a refresh request arrives during a processor read or write cycle, a hidden refresh cycle is executed. When the state machine passes through the state ref-2, the refresh request flip-flop is cleared. Note that during a CAS* before RAS* refresh cycle, the DRAM input WE* must be inactive, or the DRAM enters a test mode.

Address Multiplexers

Address multiplexers switch addresses from the row address to the column address prior to assertion of the CAS* signal. This occurs on the falling edge of the 28 MHz clock after assertion of RAS*. This is the simplest means of satisfying the row address hold time, tRAH, of 12 to 15 nanoseconds and ensures that the column address is valid before assertion of CAS*. The address multiplexer circuit for the DRAM controller is shown in Figure 7. Addresses from the TMS32C30 are connected to the multiplexers in a configuration that ensures that the maximum number of memory rows are accessed during the execution of sequential instructions. This is a conservative design practice that allows more rows to be accessed, and therefore more refreshing of memory, as a read or write to any column within a row refreshes the entire row. It is not essential in this case as the design meets the refresh cycle requirements.

Output Switching Noise Considerations

In the block diagram of Figure 1, the QL8x12 drives eight RAS* and CAS* loads. Even though the QuickLogic FPGA has carefully designed output circuits to minimize output switching noise in the form of ground bounce (up to 48 outputs can be switched simultaneously with less than 1 volt of peak to peak noise), good design practice is to insert a series resistor between the driver and its load. Since excessive bounce can cause unreliable operation of the DRAM, a 22 ohm resistor in combination with the DRAM input capacitance is used to form a low pass filter to suppress bounce on these strobe lines. For additional reliability, series resistors can also be placed on the address lines.

A 4 MEGABIT DRAM CONTROLLER FOR THE TMS32C30

In following good design practices, this design is completely synchronous. All transitions are made based upon the 28 MHz clock edge. No clocks are gated nor are gates ever used to generate delays, therefore any gate can be replaced with a faster gate with no impact on the desired operation. As the 28 MHz clock has a fifty-fifty duty cycle, the design can be mapped to any comparable speed, or faster, technology and will still function reliably. No changes are necessary to the design for it to be mapped to a gate array or standard cell technology for high volume production.

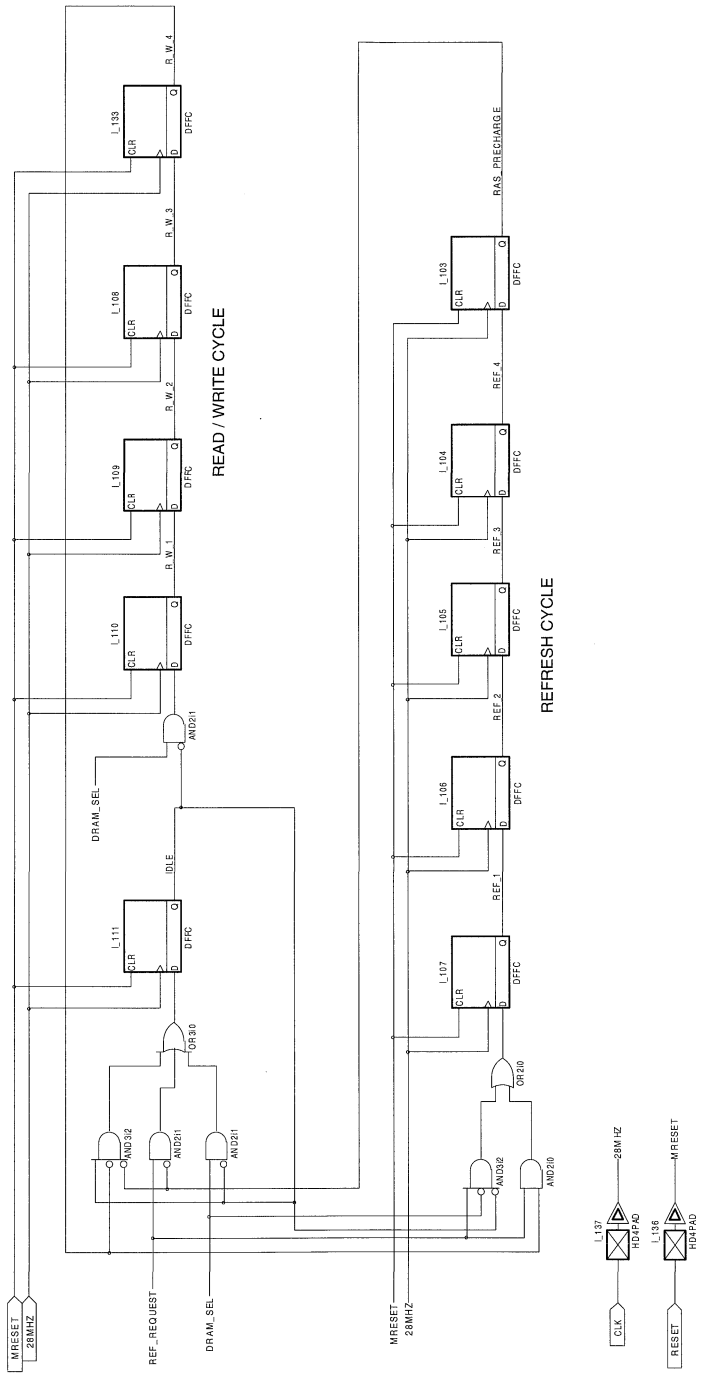
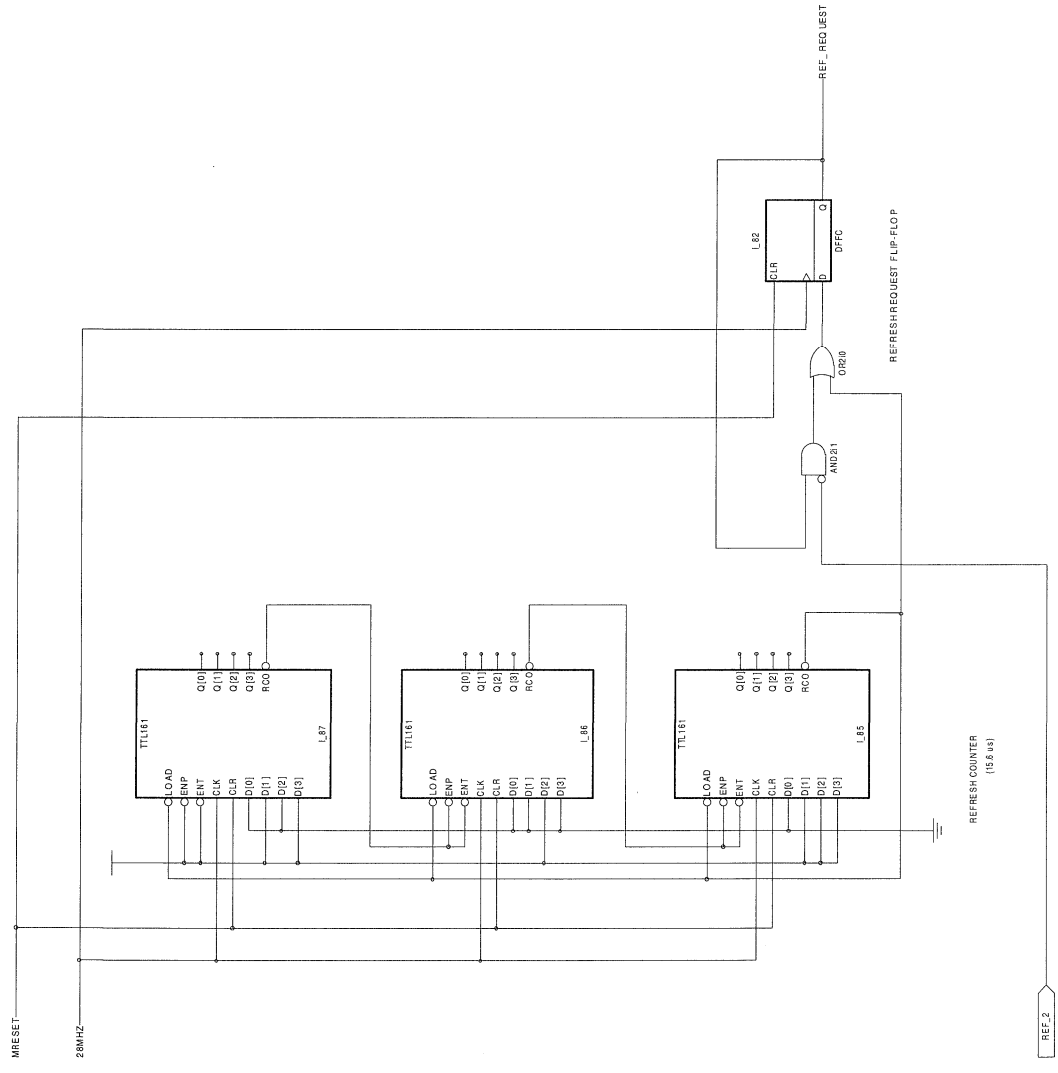


FIGURE 5 Schematic of One-Hot Encoded RAS/CAS State Machine

FIGURE 6 Schematic of Refresh-Request, Free-Running Counter



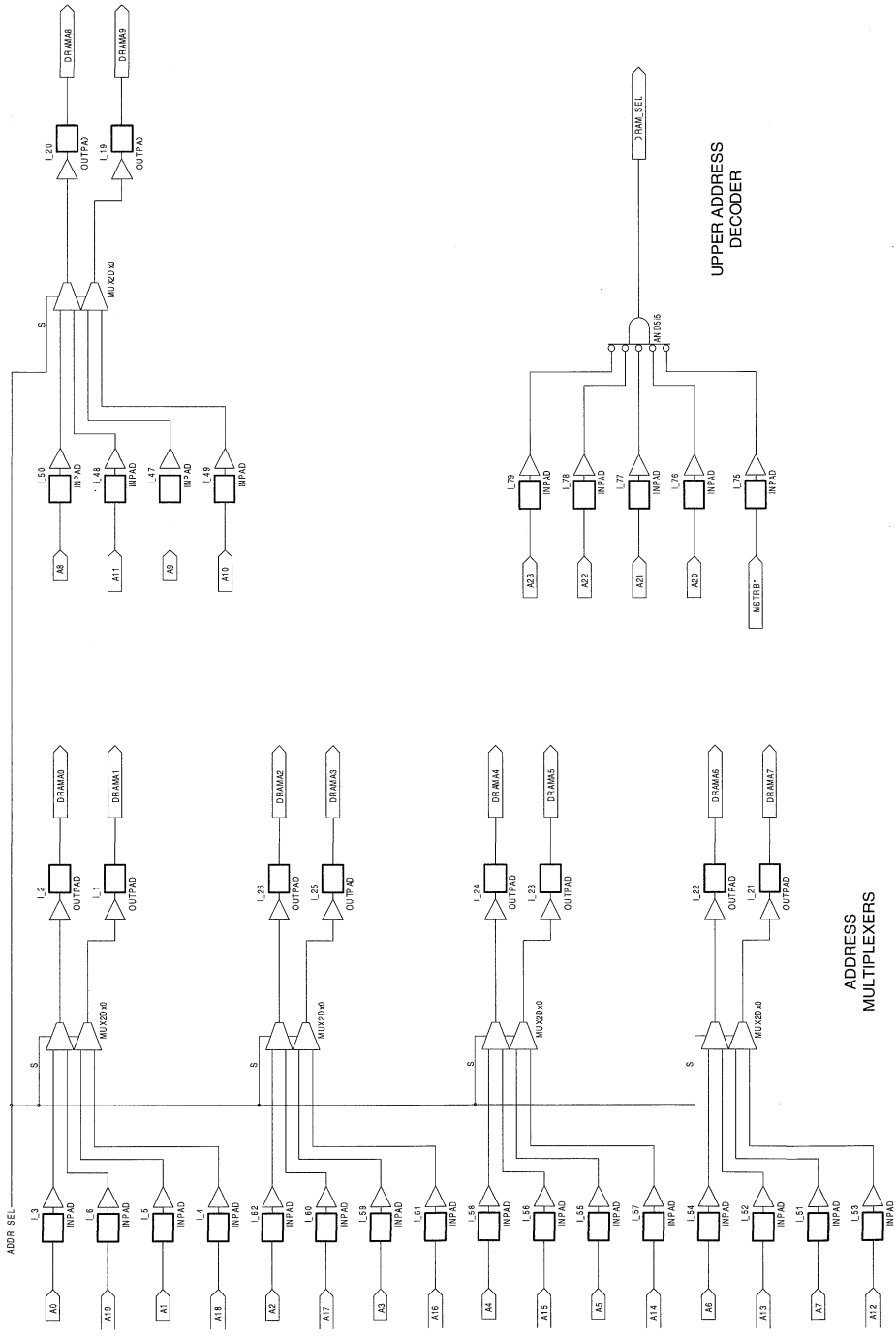
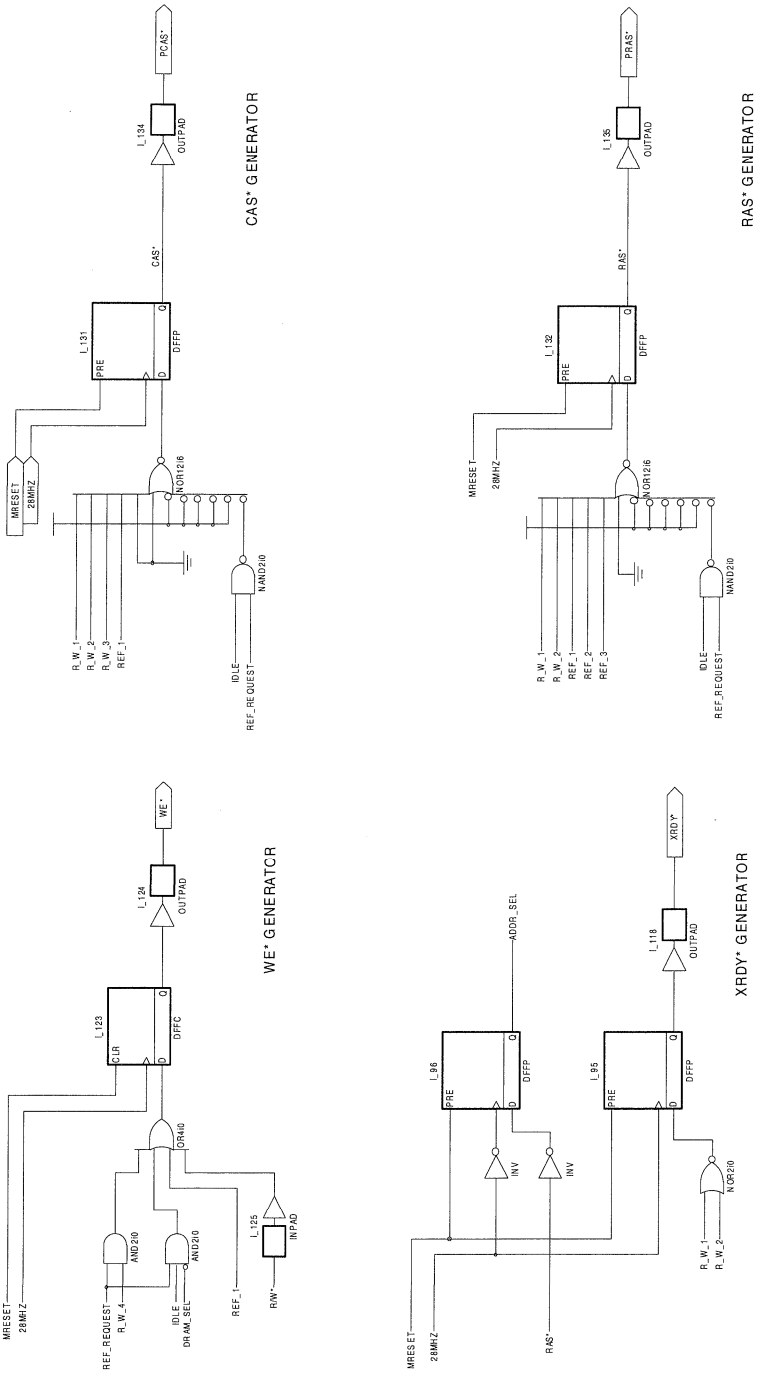


Figure 7a Schematics of Address Multiplexers and Upper Address Decoder

FIGURE 7b Schematics of RAS, CAS, WE and XRDY Output Drivers



QAN6
Page Mode DRAM Controller for 486DX2

1.0 SUMMARY

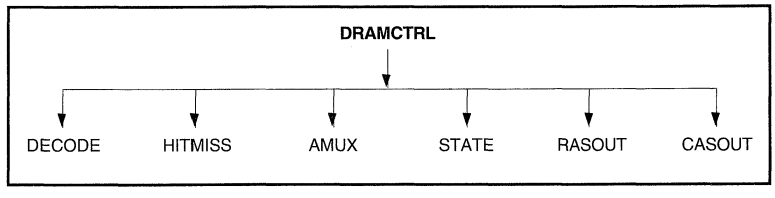
The Intel 486™ DX microprocessor is one of today's most advanced microprocessors. In addition to its popularity in personal computer applications it is increasingly chosen as the host in a wide variety of workstations.

Today, many designs are migrating to the Intel 486DX2, which is compatible with the 486DX, yet offers an internal clock rate at twice the external clock rate. These processors continue to integrate more and more functionality onto the chip to improve performance and decrease system form factors. However, because of the additional pin count and the continually changing DRAM market, DRAM controllers are typically not integrated with the processor, leaving the design up to the system engineer.

Interfaces to 66 MHz 486DX2 microprocessor

This application note presents an example of a high-performance page-mode DRAM controller implemented in a QuickLogic QL12x16 FPGA which interfaces to a 66 MHz 486DX2 microprocessor. The function integrates the address decoding and multiplexing, page hit/miss detection, a basic controller state machine, and the RAS/CAS output logic into a TQFP (Thin Quad Flat Pack) or an 84-pin PLCC package. Designs of this complexity and speed will typically require up to fifteen high-speed 22V10 PLD packages.

DRAMCTRL Major Function Blocks



The design illustrates the critical paths generally associated with a high-frequency DRAM controller and shows the advantages of the QuickLogic pASIC® 1 architecture in serving this class of applications. It was created using the QuickLogic pASIC Toolkit based on the Data I/O ECS schematic entry package. Copies of the detailed design schematics are available on request from QuickLogic (QS-QAN6). All timing delays quoted are worst case values for the 1.0 micron, QL12x16-0 device over the commercial operating range. QuickLogic's latest .65 micron QL12x16B exhibits significantly faster timing (20% - 30% faster).

QUICKLOGIC

2.0 BRIEF DRAM OVERVIEW

The acronym 'DRAM' stands for Dynamic Random Access Memory. The term 'dynamic' comes from the internal implementation of each bit storage cell. This type of cell uses capacitance to store the bit value instead of combinatorial feedback as is used in a static type latch. Using a capacitance to store the charge allows for a minimum number of transistors to achieve the highest possible density. By keeping the transistor count to a minimum, DRAMs can provide the lowest cost per bit in a random access memory. There is, however a price to pay for this low-cost solution.

Since total isolation from leakage current is not possible, charge stored on a bit cell capacitance will leak off over time. For this reason the charge on the capacitance must be restored to a full value periodically. This is known as refreshing the DRAM and some method of refreshing must be supported in the system design.

DRAMs have a second unique feature used to reduce overall cost and that is the use of a multiplexed address bus. Typical DRAMs today require on the order of 20 or more address lines. This, coupled with data lines, would require a fairly large package — increasing package cost, board space, and ultimately board cost, DRAMs use two strobe signals (i.e., RAS and CAS) to latch in the address. Half of the address is first latched using the RAS signal — this is known as latching the row. The second half of the address is latched using the CAS signal — this is known as latching the column.

There are a variety of timing specs associated with the row and column strobes that must be met for proper operation. The most common specs referred to are tRP, tRAC and tCAC. These represent specs for the RAS precharge time, RAS access time and CAS access time. The RAS access time is the time it takes for data to become valid from the RAS edge. This spec is normally used as the designator for the speed of the DRAM and is typically the most difficult timing to meet.

The CAS access time is also a data valid time, but from the CAS edge rather than the RAS edge. Both specs must be met before data will be valid. Current DRAMs typically have a tCAC value that is about one-fourth the tRAC value.

Due to the ever increasing need to always have faster access times and higher performance, DRAM manufacturers have come up with many different modes that help to improve the access time for certain applications. One of the most common modes used today is page mode. Page mode allows successive addresses to the same page to keep RAS active while the data access time is totally dependent on tCAC (and some address timing specs) for subsequent cycles. Thus, each cycle following the leadoff cycle is able to execute at a much higher rate. However, the logic to implement a page mode controller is significantly more complicated, as it requires keeping a latched copy of the previous page address and comparing it to subsequent cycles. When the page does not match, the RAS output must be precharged as defined by the tRP spec. Once tRP has been met, a new cycle with a new page may be started. The end result is a slower cycle for page misses and a faster cycle for page hits. Generally, the use of page mode gives an overall increase in performance.

3.0 DESIGN OBJECTIVE

In addition to the many modes available in DRAMs, today there exist many different internal configurations that may use asymmetrical combinations of row and column address bits. This makes support for multiple DRAMs even more difficult. Additionally, DRAMs come in an assortment of data out bits and the number of RAS, CAS, and WE signals it may need. All of these variations can make for a very complex design when designing a DRAM controller.

The majority of system designs using an Intel 486DX2 require large amounts of local DRAM memory, and thus, some sort of DRAM cycle controller. The performance of this controller is of primary concern since it will greatly affect the overall system performance. Additionally, board space that the controller takes up is also of concern. It is no longer acceptable to use discrete devices of small integration in order to achieve high performance. A single-chip solution is almost always required and generally includes additional board-level functions.

The objective of this design is to provide the highest performance DRAM interface for a 486DX2 at 33 MHz while using minimal board space and readily available DRAMs. The desired specifications for this controller are:

- Support 486DX2 burst mode cycles of 5-2-2-2
- Staggered refresh cycles using CAS before RAS refresh mode
- Bank size configurability via mode select pins
- CAS wait state selection via mode select pin giving 5-3-3-3
- Fast page mode cycle support
- DRAM select decode with range programmability

4.0 DESCRIPTION OF DESIGN PROBLEM

Speed is always the primary concern when designing DRAM controllers. With today's microprocessors running at 33 MHz and higher, circuit delays of a few nanoseconds may mean additional wait states to the processor access. Additionally, market requirements typically dictate support for multiple DRAM sizes, organizations, and speed. Designing a DRAM controller to accommodate these needs directly translates into additional levels of logic and additional delays.

The design presented in this application note supports two different sizes of banks that can be mixed across the four banks. The major problem areas and critical speed paths in a design of this type are typically:

- Bank decode logic
- Row/Column address mux delay
- Page compare logic and cycle start delay
- Output logic delay and variation

Bank Decode Delay

The bank decode is by far the most complicated section of logic. This may seem surprising, since in the past, it has typically been the simplest part of the design. No longer is the bank decode simply a 2-to-4 decoder of two address lines. If you choose to support different sizes of banks simultaneously, the address decode gets significantly more complicated.

The design presented here supports two sizes of banks simultaneously, provided the larger memory is mapped to the smallest address possible. Support for any combination of banks becomes even more difficult, but still possible. The possible configurations reduce down into five distinct cases. Within each case, it is possible to have additional cases that do not have all banks populated. Table 1 shows the configurations supported in this application note in more detail.

Note that the decode uses four address bits, and which bits are used is dependent upon the configuration.

TABLE 1
Bank Decode Table

CASE	ADDRESS				BANK NUMBER				RANGE SIZE (MEG)
	25	24	23	22	0	1	2	3	
1	–	–	0	0	4				4, 8, 12, 16
	–	–	0	1		4			
	–	–	1	0			4		
	–	–	1	1				4	
2	–	0	X	X	16				16, 20, 24, 28
	–	1	0	0		4			
	–	1	0	1			4		
	–	1	1	0				4	
3	0	0	X	X	16				16, 32, 36, 40
	0	1	X	X		16			
	1	0	0	0			4		
	1	0	0	1				4	
4	0	0	X	X	16				16, 32, 48, 52
	0	1	X	X		16			
	1	0	X	X			16		
	1	1	0	0				4	
5	0	0	X	X	16				16, 32, 48, 64
	0	1	X	X		16			
	1	0	X	X			16		
	1	1	X	X				16	

4.2 Row/Column Address Mux Delay

The row/column address mux directly affects how early the address can be driven to the DRAM. The fastest and simplest 2-to-1 mux is generally used to switch between row and column. Any additional delay on the address mux causes both the row and column address to be delayed. This, in turn, translates

to either delaying RAS and CAS (adding additional wait states) or a violation of address setup times to RAS and CAS.

Support for multiple DRAM sizes and organizations presents yet another problem that requires further levels of logic in the address mux. This is due to the fact that different sized DRAMs will require different address bits in the row address. This can be observed by first looking at the configuration for a 1M bit DRAM (256Kx4) organized in the system as 256Kx32. In this configuration, a depth of 256K would need to be addressed by the row and column address. The width of 32 would be addressed by 4-byte enables.

Addressing of 256K requires 18 address lines — generally nine row bits and nine column bits. If we are designing the address mux (MA[0:10]) around this specific requirement, we would have the assignments shown in the first row of Table 2.

TABLE 2 Row Address Map

DRAM DEPTH	ROW/ COLUMN	10	9	8	7	6	5	4	3	2	1	0
256K	9x9	22	20	19	18	17	16	15	14	13	12	11
1M	10x10	22	20	19	18	17	16	15	14	13	12	21

Since A1 and A0 are used up in the byte selection, addressing to the DRAM starts at A2 and goes up to A19. Note that the highest order bit of the column is A10, with the row starting at A11. To support a DRAM of depth 512K (organized 10x9), A20 simply needs to be driven on an additional row address — MA9. However, to support a depth of one meg (organized 10x10), A21 needs to be driven on either the row or column. The easiest solution would be to add it to the column address on MA9. However, this would fragment the page of the DRAM into two halves, lowering the page mode performance.

In order to keep the page contiguous, A21 must be driven in the row address. This is accomplished by replacing A11 on the row address with A21, and driving A11 on the unused column address. As such, the address mux for MA0 now becomes a 3-to-1 mux selecting between A2, A11, and now A21. As mentioned previously, if the delay incurred for each additional mux input is too large, the cycle will require an additional wait state.

4.3 Page Compare and Cycle Detect Delays

The start of page mode DRAM cycle is typically gated by:

- DRAM address range decode select
- Page hit/miss detection

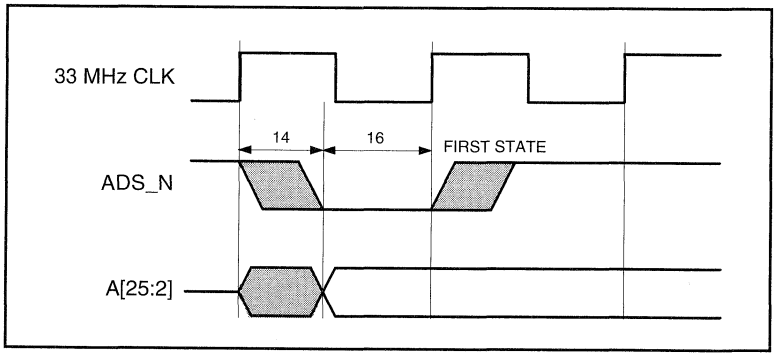
The DRAM address range decode determines if the address of the cycle goes to DRAM or some other device. This decode can range from simply matching upper address lines to 0, or actually comparing an address

between two registers depending upon the granularity of the banks and the combinations of bank population. Both mechanisms typically require several levels of logic to implement.

The page hit/miss detection determines whether the page of the pending cycle is equal to the page of the previous cycle. If the page value is different, then the row of the DRAM will need to be precharged. This typically consists of a comparator implemented as several levels of XNOR gates. Both the address range decode and the page hit/miss detection depend upon a valid address from the processor. The total delay before starting the cycle will then be the maximum sum of the processor address delay with either the range decode delay or the page detect delay.

The processor address delay is given in the 33 MHz data sheet to be 14 ns. In order for the state machine to start the DRAM cycle on the first following clock edge, both the range detect and the page detect must be performed and allow for a setup time to a latch in less than 16 ns (see Figure 1). Performing XOR combinations can be quite costly in terms of delay and depend upon the number of address bits needing comparison.

FIGURE 1
Cycle Detect Delays



4.4 Output Logic Delays and Variation

There is typically very little margin for variation among the output pin timings of a DRAM controller. Today's DRAM timing specifications are all done with respect to each other, i.e., pin-to-pin timing and not pin-to-clock. Any variation of one pin may greatly affect its timing in relation to another pin. Figure 2 illustrates a typical requirement among RAS, DATA, and MA.

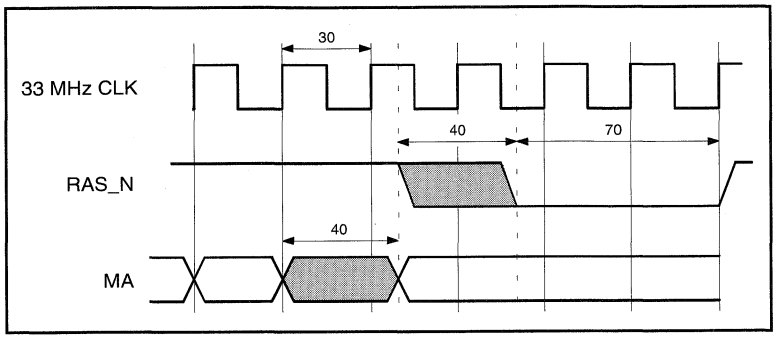


FIGURE 2 RAS/DATA/MA (Version 1)

As shown in Figure 2, RAS must be active 70 ns prior to the end of the DRAM cycle. If the variation in output timing is 40 ns, then the point for activating RAS would be designed to be 110 ns prior to the end of the cycle. This would ensure that the slowest manufactured part would still provide the needed 70 ns prior to the end of the cycle. If the variation of MA is also 40 ns (and this delay does not track with the RAS delay), then MA would need to be designed to be valid 150 ns before the end of the cycle. If the variation in delay for RAS and MA can be kept to 20 ns, then MA would need to be valid 110 ns prior to the end of the cycle (see Figure 3). This difference of 40 ns would mean the difference in one wait state to the processor at 33 MHz. The same scenario also applies to MA and CAS, WE, CAS, and other timings.

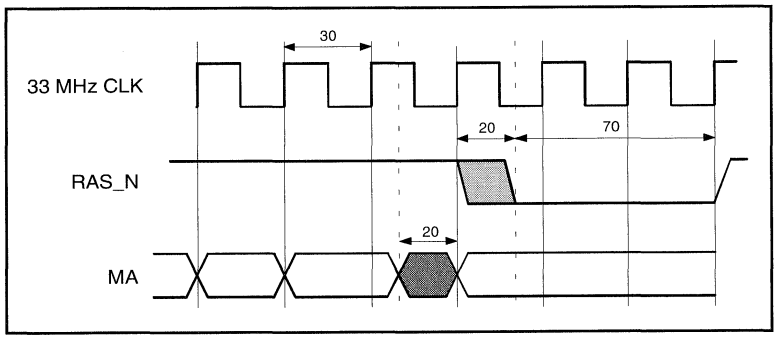


FIGURE 3 RAS/DATA/MA (Version 2)

The key to keeping the output delays and variation in delay to a minimum is by using fast logic and placing the controlling signal as close to an output pin as possible (i.e., so there are a minimum number of gate delays from the clock to the output pad). The most common way of doing this is to place the output latch as the very last device in the path. This means a separate latch for each output signal. The abundance of flip-flops coupled with the extremely fast logic makes QuickLogic's 12x16-0 FPGA an ideal solution.

5.0 DESCRIPTION OF DESIGN

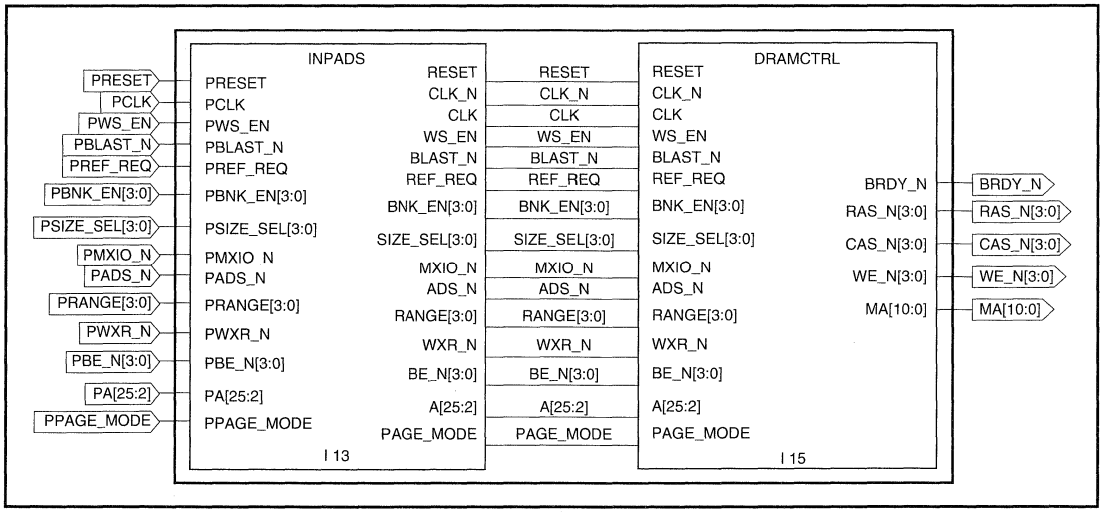
5.1 Overview

A Page-Mode DRAM Controller meeting the demands of the design problems outlined in the prior pages was implemented in an 84-pin version of the QuickLogic QL12x16 2000-usable gate FPGA. Complete design schematics for this controller are available from QuickLogic. The remainder of this application note describes the functions of each of the blocks in the design.

Figure 4 is a top view of the controller as created in the Data I/O ECS schematic capture tool. It consists of the Processor and DRAM Interface Signals and two blocks containing the input pads (INPADS) and the DRAM controller logic (DRAMCTRL). Tables 3 and 4 list the signal name and description of all the I/O pins.

Within the DRAMCTRL block of Figure 5, are six major logic blocks. The schematic hierarchy diagram of Figure 6 shows that six more blocks are embedded at lower levels. For example, the REFRESH block is contained inside the STATE block (see Figure 7).

FIGURE 4
DRAM Controller



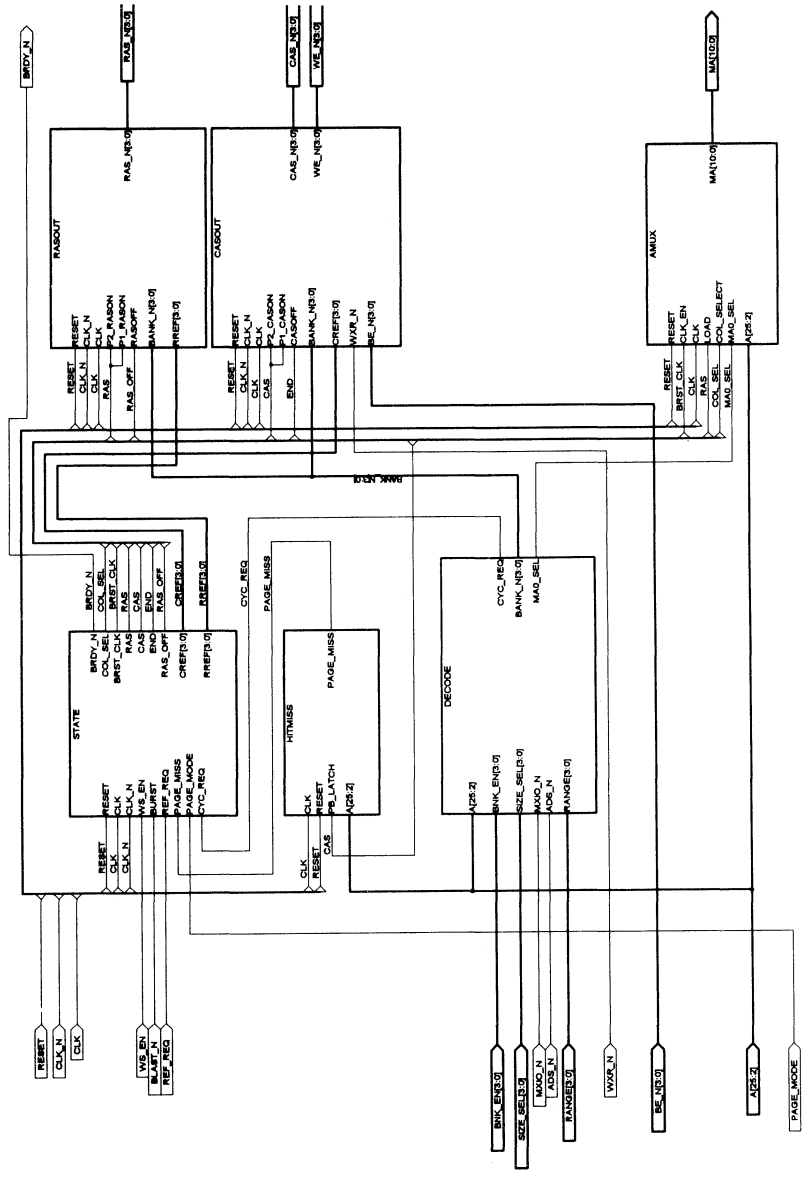
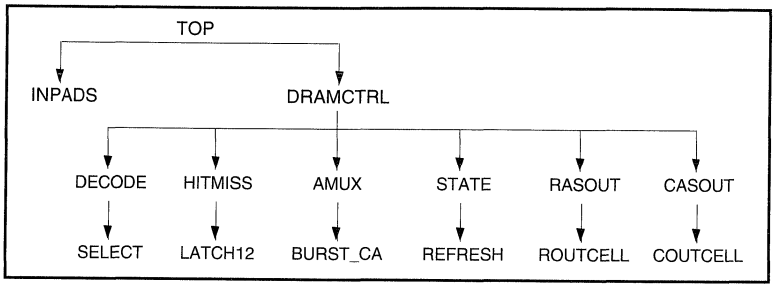


FIGURE 5 DRAMCTRL Major Function Blocks

FIGURE 6
Schematic Hierarchy



A description of all twelve upper and lower level blocks follows:

DECODE – Address Decode: This block takes care of decoding the address range to determine if it is the DRAM range, and also bank selection.

SELECT – Cycle Select Logic: This block consists of a 4-bit range comparator and logic that takes in ADS_N (Address Strobe) and MXIO_N (Memory/Input/ Output Status Signal) to determine the start of a cycle.

HITMISS – Page Hit/Miss Detection: This block detects whether the page address of the current cycle matches the page cycle of the last address. The output status is used as an input to the state machine.

LATCH12 – 12-Bit Register Latch: This block takes the 12 address lines that make up the page and latches them.

STATE – State Machine: Controls the entire DRAM controller. The state machine takes input of the form 'cycle request,' 'refresh request,' 'page miss,' and other configuration information to determine the cycle type and control the outputs accordingly.

REFRESH – This block is within the state machine block and is essentially the state machine for the refresh cycles.

AMUX – Address Multiplexer: This block takes care of driving the correct row and column address to the DRAM.

BURST_CA – Burst Column Address Generator: Within the AMUX block this block takes care of generating the correct CA[3:2] when burst cycles are performed.

INPADS – Input Buffers: This block contains all input pad declarations for the chip.

RASOUT – RAS Output Logic: This block clocks out the appropriate RAS_N signal at the correct time. It instantiates 4 ROUTCELL blocks to accomplish this.

ROUTCELL – RAS Out Cell: Takes the set and reset signals from the state machine and combines them with both CLK and CLK_N to clock out the RAS_N signals. Additionally, this block uses the bank select signals as enables.

CASOUT – CAS Output Logic: Similar to RAS Output Logic only used for the CAS lines. This block also contains logic to drive the WE_N[3:0] signals.

COUTCELL – CAS Out Cello: Almost identical to the ROUTCELL with a slight difference on the reset logic.

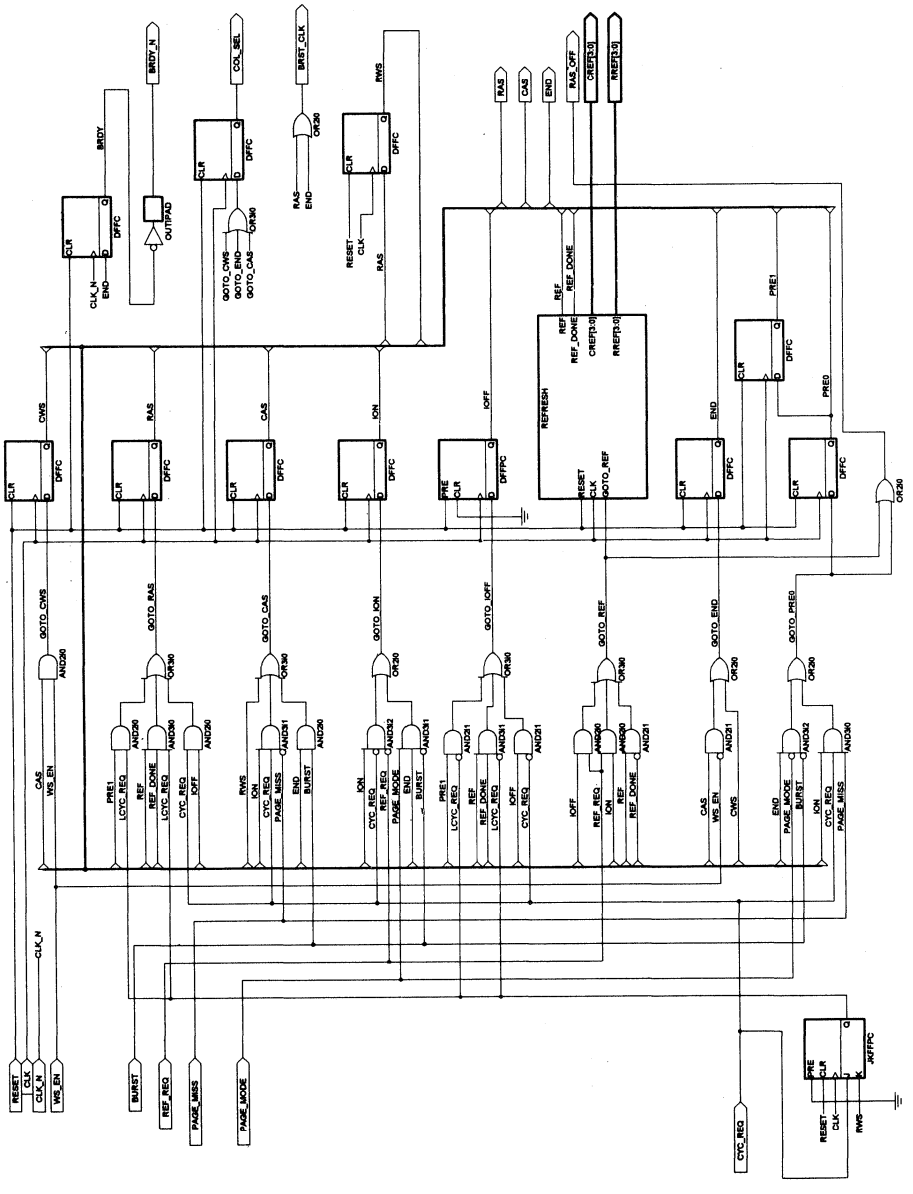


FIGURE 7 State Machine Logic Function

5.2 Interface Signals

TABLE 3 Processor Interface

SIGNAL	I/O	DESCRIPTION
A[25:2]	I	Processor Address lines: Address line inputs from the 486DX2. They are driven active on the rising edge of the clock and are guaranteed to be valid 14 ns later. The address is driven throughout the entire cycle.
ADS_N	I	Address Strobe: This signal is used to indicate the start of a cycle from the 486DX2. It has identical timing with the address. However, it will remain active for only one clock.
BE_N[3:0]	I	Byte Enables: Indicates which bytes of the current cycle are to be accessed. They have identical timing with the address.
BLAST_N	I	Last Burst Cycle: This signal indicates when the current cycle is the last of a burst cycle. Cycles that are not burst cycles will always have BLAST_N active.
CLK	I	33 MHz Clock: This is the clock input used to clock internal logic.
RESET	I	Reset Signal: Used to reset all logic internally.
BRDY_N	O	Burst Ready: This signal indicates the completion of the cycle. Since this device (DRAM Controller) supports burst cycles when requested, BRDY_N will always be used.
MxIO_N	I	Memory/Input/Output Status Signal: This signal indicates whether the current cycle is a memory cycle or an I/O cycle. It has identical timing to the address.
WxR_N	I	Write/Read Status Signal: This signal indicates whether the current cycle is a write cycle or a read cycle. It has identical timing to the address.
REFREQ	I	Refresh Request: This signal indicates a refresh cycle is desired by the external system. The assumptions here are that arbitration for the bus has already occurred outside this controller.
WS_EN	I	Wait State Enable: This signal is a programming option to configure the controller with a wait state during CAS active. When active, CAS will have approximately two clocks of access time. When inactive CAS will have approximately one clock of access time.
RANGE[3:0]	I	DRAM Range: These inputs indicate the overall size of the DRAM memory range. These four bits represent the number of 4 Meg blocks of memory.
BNK_EN[3:0]	I	Bank Enables: These four signals represent which banks are enabled.
SIZE_SEL[3:0]	I	Bank Size Selects: These four signals represent the size of each bank of memory. When inactive, the size of the bank is 4 megabytes. When active, the size is 16 megabytes.
PAGE_MODE	I	Enable Page Mode: This programming option is used to enable page mode. Because of the timing restrictions, worst case design indicates a critical path in the page mode detection logic. However, at lower frequencies or with fast parts, page mode may still be used. When inactive, the critical path is removed from the logic.

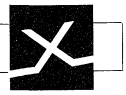


TABLE 4 DRAM Interface

SIGNAL	I/O	DESCRIPTION
RAS_N[3:0]	O	Row Address Strobes: These signals are used to strobe in the row address for each of the banks.
CAS_N[3:0]	O	Column Address Strobes: These signals are used to strobe in the column address for each of the banks.
WE_N[3:0]	O	Byte Write Enables: These four signals represent write enables for each of the bytes in a DWORD.
MA[10:0]	O	DRAM Row/Col Memory Address: These 11 outputs are used to present both the row and column to the DRAM.

5.3 Cycle Detection Logic

Cycles in the design are started by the address being within the range programmed on the RANGE[3:0] bits, ADS_N going active, and MXIO_N indicating memory. When these three events occur the signal CYC_REQ goes active indicating to the state machine to start a cycle.

The first attempt at designing this used a macro for the TTL range comparator 74684. Since this was an 8-bit comparator, four of the comparisons were redundant and were tied inactive. The output of the comparator was then combinatorially ANDed with ADS_N and MXIO_N. The use of the extra logic needed for 8-bit comparison and the following level of decode made this path painfully slow. It was not possible to detect the cycle by the next clock edge after ADS_N had gone active.

By reducing the comparator to a 4-bit range comparator — and combining the ADS_N and MXIO_N logic into the comparator, it was possible to speed this path up from 18 ns to 8 ns, thus making it functional. Paying close attention to which gates fit directly in one level of a logic cell also helped. An inverter was needed in the ADS_N path in order to combine it with the comparator logic. However, since ADS_N will be active well before the range comparison is complete, it is not in the critical path.

5.4 Page Compare Logic

This page hit/miss logic is similar to the range compare, except that it must be performed over 12 address lines and it only needs to match the value. However, because of the need to compare 12 lines down to one output, this path is very slow. Under worst case conditions for the QL12x16-0 this path is simulated to 27.6 ns.

The delay of 27.6 ns was well above the allotment of 15 ns for page hit/miss logic. However, under best case conditions this path is within margin. For this reason, an option was added to keep the page mode access. This option is controlled by enabling the input PAGE_MODE. When active, the state machine will assume the page compare logic is fast enough and will use the output to determine state transitions. If PAGE_MODE is inactive, then the PAGE_MISS signal will be masked off in the state machine.

Design Tip

If page mode is strongly desired under all conditions at 33 MHz, then the state machine could be modified to support this. By designing in a one clock delay at the start of the cycle in which no activity is performed, the page hit/miss logic has an additional 30 ns to complete. This of course, increases all cycles by one clock, but if the cycle is a page hit, the RAS and RWS state are bypassed — saving two clocks. There is still a net gain of one clock.

5.5 Bank Decode Logic

The bank decode logic consists of a static section and a dynamic section. The static section decodes the choices programmed on the SIZE_SEL[3:0] pins and the BNK_EN[3:0] pins. The programming on these pins comes down to five possible cases. The delay on this logic is irrelevant since it is static.

The dynamic section takes the five case signals and combines them with the address to determine the correct bank. The delay through this logic must be less than 30 ns in order to be valid when RAN_N is activated. The worst case simulation came to 15 ns.

Design Tip

If more pins are needed for a different implementation, the size and enable pins could be replaced with the case signals directly, which would reduce eight pins to five pins.

5.6 Row/Column Address Mux

The row/column address mux consists of several of 2-to-1 multiplexers that select the appropriate address bits to drive for the row and column. The selection between row and column is made using the COL_SEL signal that comes from the state machine.

Tables 5 and 6 show the various DRAM configurations that the multiplexer is capable of supporting. However, only two of the address mux modes are supported directly in the bank decoder.

The 1M depth and the 4M depth give bank sizes of 4M and 16M respectively, which are supported in the bank decoder. The other address mux modes could be used provided only one bank was populated. The bank decoder could also be redesigned to support different decoding combinations. Both MA0 and MA1 require additional levels of muxing for row address bits. MA0 must support A11 and A21, while MA1 must support A12 and A23.

DRAM DEPTH	ROW/ COLUMN	10	9	8	7	6	5	4	3	2	1	0
256K	9x9	22	20	19	18	17	16	15	14	13	12	11
512K	10x9	22	20	19	18	17	16	15	14	13	12	11
1M	10x10	22	20	19	18	17	16	15	14	13	12	21
2M	11x10	22	20	19	18	17	16	15	14	13	12	21
4M	11x11	22	20	19	18	17	16	15	14	13	23	21

TABLE 5 Row Address Map

DRAM DEPTH	ROW/ COLUMN	10	9	8	7	6	5	4	3	2	1	0
256K	9x9	12	11	10	9	8	7	6	5	4	3	2
512K	10x9	12	11	10	9	8	7	6	5	4	3	2
1M	10x10	12	11	10	9	8	7	6	5	4	3	2
2M	11x10	12	11	10	9	8	7	6	5	4	3	2
4M	11x11	12	11	10	9	8	7	6	5	4	3	2

TABLE 6 Column Address Map

Burst cycles on the 486 require external logic to generate A[3:2]. Depending on the first address, the subsequent values may either be an incremented address or decremented address. The BURST_CA block takes care of detecting the first address, latching the address, and generating subsequent CA[3:2] bits of the burst cycle. Table 7 shows the burst address sequence.

BURST ADDRESS SEQUENCE A[3:2]				
FIRST	SECOND	THIRD	FOURTH	COMMENTS
00	01	10	11	Increment
01	00	11	10	Decrement
10	11	00	01	Increment
11	10	01	00	Decrement

TABLE 7 486DX2 Burst Address Sequence

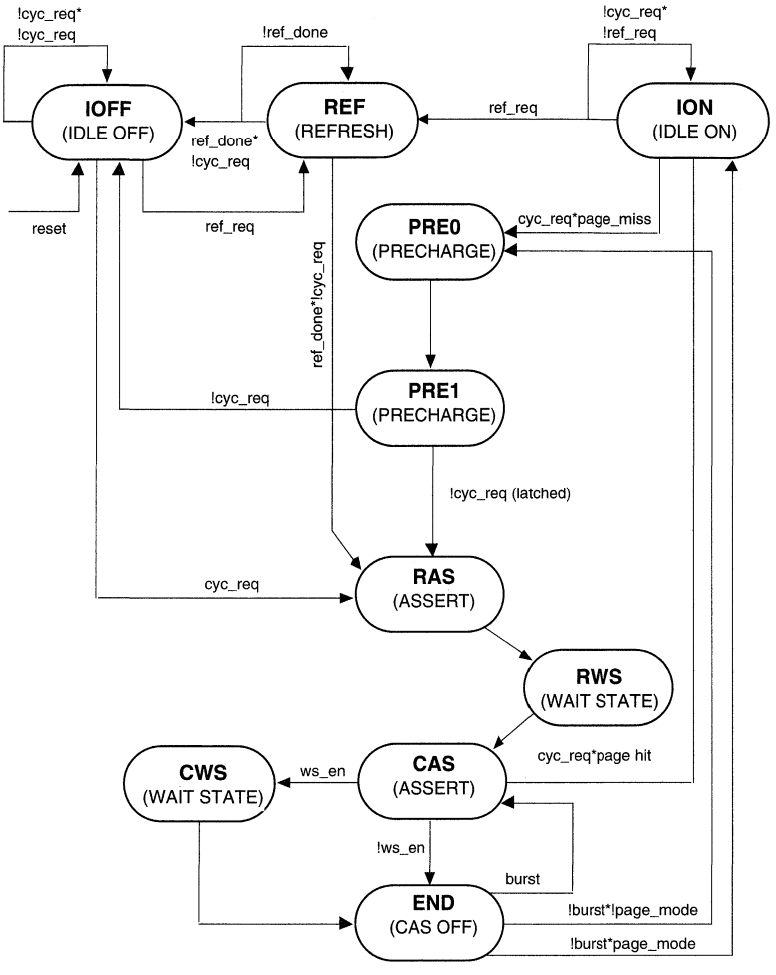
5.7 State Machine

The state machine contains all the control logic for the DRAM controller. In most cases a state output may be used directly for the desired control function. However, some control functions require additional logic to combine several states to create a control signal.

The state machine resets to the IOFF state. In this state RAS is inactive. Requests to leave this state may consist of either a refresh request or a processor cycle request. When a processor cycle request is made, a transition to the RAS state will occur where RAS is activated. The cycle will then proceed through RWS, CAS (where it asserts CAS) and END. The end state is the last state of a single data transaction and is used to deactivate CAS. If burst mode is indicated by the processor, then following cycles may transition directly back to the CAS state to reactivate CAS.

Refresh cycles occur in the REF state. A separate state machine controls the actual refresh cycles. The main state machine will remain in the REF state until a REF_DONE sinal is received from the refresh state machine. At this point all the RAS_N outputs are precharged, thus the machine waits for the next cycle request in the IOFF state.

If page mode is not enabled (and not bursting), transitions from the END state will reset the RAS_N output. If page mode is enabled, then RAS will not be reset and a transition to the idle on state will be made (ION). Departure from ION will occur for refresh request and processor cycle request. For the latter, a transition to either PRE0 or CAS will be made depending upon the page miss status. Page miss cycles from the ION state will require precharging in PRE0 and PRE1 states. Figure 8 shows the state transition table.



Normal Cycle
(Including Burst)

FIGURE 8
DRAM State Machine

The states and Cycle State Machine Equations are defined as follows:

IOFF – Idle RAS Off State: This state is an idle state with RAS inactive. This state is entered at reset and from the refresh state when no cycles are pending. This state may also be entered from the PRE1 state if page_mode is not enabled and no cycle is pending.

IOFF = IOFF•/CYC_REQ
+ REF•REF_DONE•/CYC_REQ

ION – Idle RAS Active State: This state is identical to the IOFF state except that one of the RAS_N outputs is active. This state is only used if page_mode is enabled. If a refresh request occurs while in this state, then a transition to the REF state is made. The REF state makes sure the lines get precharged before starting the refresh cycle. When a regular cycle request is made, a transition to either the precharge states or the CAS state is made depending upon whether the page address matches the previous cycle page address.

ION = ION•/CYC_REQ•/REF_REQ
+ END•/BURST•PAGE_MODE

PRE0 – RAS Precharge State 0: This state is the first of two when precharging the RAS_N outputs.

PRE0 = ION •CYC_REQ•PAGE_MISS
+ END•/BURST•/PAGE_MODE

PRE1 – RAS Precharge State 1: This state is the second of two when precharging the RAS_N outputs. Transitions out of this state will always be to the RAS state when page_mode is enabled. However, if page_mode is not enabled, then the precharged states may not have been entered as a result of a cycle request (as in the case of page_mode), and therefore, the state of cyc_req must be checked to determine between IOFF and RAS.

PRE1 = PRE0

RAS – Assert RAS Active State: This state is used to activate the correct RAS_N output. It can be entered from three different states. RAS may be entered from IOFF directly whenever a cycle request is made. In this scenario the RAS_N outputs are already precharged. Likewise, if a request comes near the completion of the refresh cycle, then a direct transition from REF to RAS is possible. The REF state assures that all RAS_N outputs are precharged when leaving. The third entry point to RAS occurs from the PRE1 state when a request is pending.

RAS = PRE1•LCYC_REQ
+ REF•REF_DONE•LCYC_REQ
+ IOFF•CYC_REQ

RWS – RAS Wait State: This state is needed to meet RAS active time specs. It will always be entered directly from the RAS state and transition to the CAS state.

RWS = RAS

CAS – Assert CAS Active State: This state is entered from one of three different states. It may be entered directly from RWS for a page miss cycle, or from ION for a page hit cycle. Additionally, the CAS state may be entered from the END state when the 486 is bursting bus cycles. This situation is similar to a page hit cycle with no idle clocks and no need to detect the hit/miss. Although this cycle is essentially a page hit cycle, it will be operational in both page mode and non page mode since it does not require detecting the page hit/miss status.

CAS = RWS
+ ION•CYC_REQ•PAGE_HIT
+ END•BURST

CWS – CAS Active Wait State (also called Wait State 0): This state is similar to the RWS state in that it is used to extend the CAS access time. This state is entered from the CAS state whenever the wait state enable pin (WS_EN) is active. When in use, the CAS active time will be on the order of two clocks, and burst and page hit cycles will be approximately three clocks.

CWS = CAS•WS_EN

REF – Refresh Active State: This state is entered when a refresh cycle is needed. It can be entered from the ION state or the IOFF state. It is assumed in the design that arbitration for the bus takes place outside the chip, and that the ref_req signal is mutually exclusive with the bus cycle ADS_N going active. Once in the REF state, the machine will remain until refresh is complete. Once the refresh cycle is complete, all RAS_N outputs are precharged and ready to access a new bank. The machine can then directly move to the RAS state if a cycle has begun. If no cycle is pending, then a transition to the IOFF state is made so that the RAS_N outputs will remain precharged.

REF = REF•/REF_DONE
+ ION•REF_REQ (MUTUALLY EXCLUSIVE WITH CYC_REQ)
+ IOFF•REF_REQ (MUTUALLY EXCLUSIVE WITH CYC_REQ)

END – Last CAS Active State: This state is the last state of the bus cycle. It is primarily used to reset the CAS_N outputs, and sometimes the RAS_N outputs if needed. It is used to activate the correct CAS_N output.

END = CAS•/WS_EN
+ CWS

Additionally, there are several output signals based on different transitions that are defined as follows:

BRDY_N – Burst Ready Signal: This signal is simply a delayed version of the END state. Since the ready signal to the 486DX2 must straddle the rising edge of the clock, it is best to clock the ready with the falling edge of the clock.

COL_SEL – Column Select Signal: This signal is used in the address multiplexer to select between row and column. When active, it selects the column as the name implies. The default for this signal is inactive. It is driven active during the CAS, CWS, and END states.

BURST_CLK – Burst Clock Signal: This signal is used to change the column address bits used for successive cycles in a burst transaction. This signal will go active during the RAS state to latch in the first value. It will then go active on each END state to clock the next value.

RAS_OFF – RAS Off Signal: This signal is used to reset the RAS outputs at the end of a cycle when not in page mode, and also during idle when in page mode. It goes active when the next state of the state machine will be PRE0 or REF.

CAS-Before-RAS Refresh

The refresh cycles are controlled by the refresh state machine contained in the REFRESH block. This block consists primarily of a shift register triggered by entering the REF state of the cycle state machine.

The type of refresh cycle employed by the controller is known as CAS-before-RAS. This type of refresh cycle is lowest in power and requires the least amount of external logic. By activating CAS first and then RAS, the DRAM detects that a refresh cycle is intended and therefore uses an internal refresh row to address the row. Additionally, the DRAM will know to not drive the output buffers of the DRAM.

Activating refresh to all banks simultaneously can cause huge power surges and results in a difficult system design. The controller presented here staggers the accesses to each bank by one clock — thus keeping power surges to a minimum.

The control signals to enable refresh for each RAS and CAS of each bank are created by detecting different points in the shift register. For example, the bank 0 CAS will go active when the second latch in the shift register goes active, and will go inactive when the fifth latch goes active. The logic is then shifted for each of the following banks. At the end of the shift register a signal goes active for one clock to indicate the completion of the refresh cycle. At this point the last bank has been precharged for one clock and will have been precharged for two clocks by the time the cycle state machine begins a cycle.

5.8 Output Logic

The output logic consists of logic required to clock RAS, CAS, and WE. The implementation used here uses a generous amount of latches (which are plentiful in the QL 12x16) to clock out RAS and CAS on phase 2 clock edges while deactivating on phase 1. Two different cells, COUTCELL and ROUTCELL, are repeated multiple times and wired to the appropriate bank enables to achieve this.

The basic outcell (ROUTCELL and COUTCELL) uses a simple D flip-flop to start activation on phase 2 (by clocking with CLK_N). A second JK flip-flop is used to continue driving the output for the rest of the cycle. While the D FF remains active for only one clock, the JK FF will remain active until the reset signal comes along. The outputs of these two latches and the refresh signal are ORed together to create the final output signal.

Design Tip

Potential areas for improvement would be reconfiguring the logic such that the NOR gate could be removed. This could be accomplished by using one JK FF clocked on phase 2 which would have separate set and reset signals for refresh cycles combined with the set and reset of processor cycles. The only downside would be the deactivation of the signal in phase 2 instead of phase 1. While this would be fine for RAS, CAS would have a problem on page mode and burst cycles meeting.

6.0 PHYSICAL VIEW OF COMPLETE DESIGN

Figure 9 shows a typical view of how the controller design is implemented in the QL12x16 FPGA logic cell architecture. It occupies 122 out of the 192 available logic cells and 73 of the 76 I/O pins.

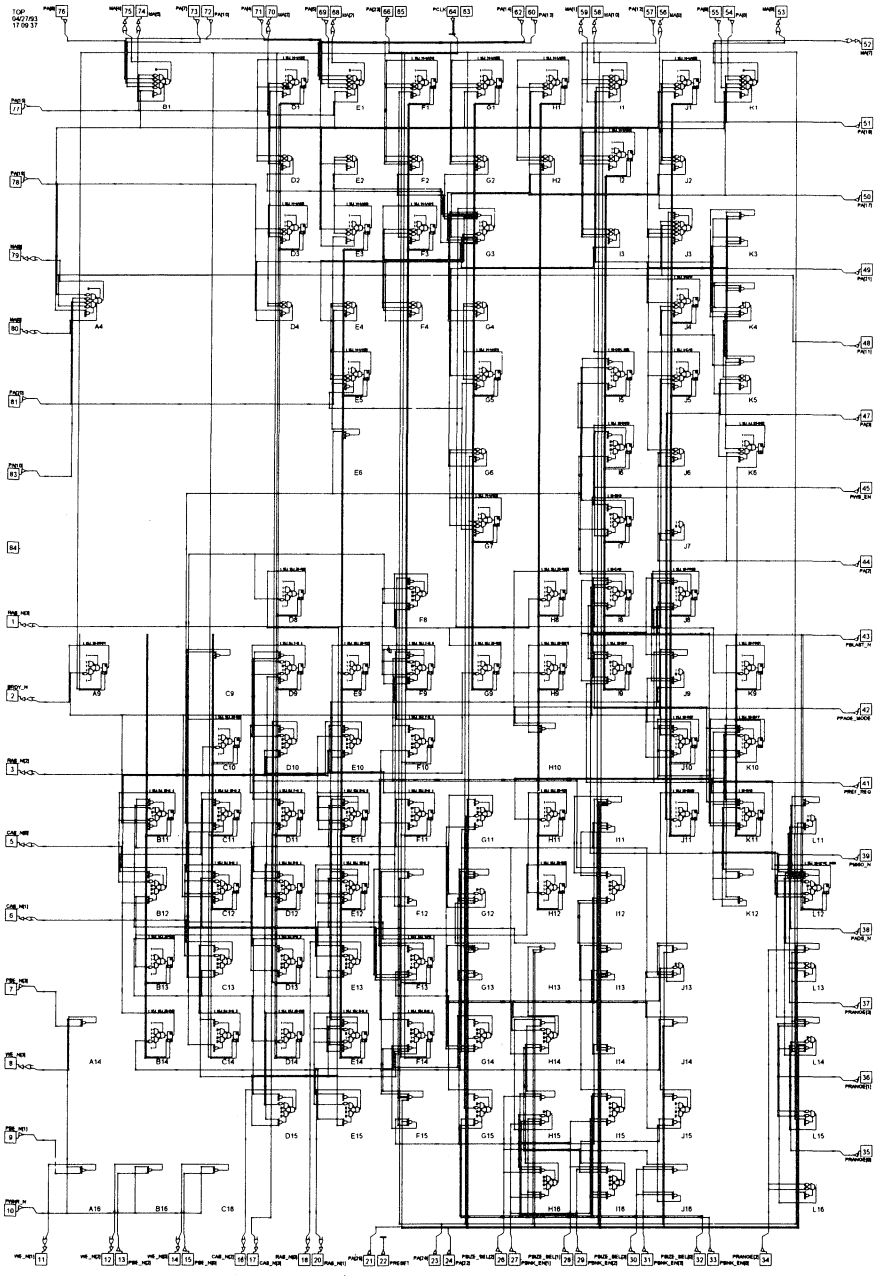


FIGURE 9 Pinout and Cell Interconnect

7.0 QUICKLOGIC TOOLS AND FPGA ARCHITECTURE

The QuickLogic FPGA development tools are completely integrated under Microsoft Windows making use of hierarchical links between all aspects of the design with a familiar user interface.

The QuickLogic development tools were extremely advantageous in the development of this application note. The schedule available for completing this design was extremely tight. The tight schedule, coupled with the fact of not ever having used the tools initially created a feeling of extreme apprehension. However, the intuitive nature and ease of use of the interface made it possible to be up to speed within hours.

Design Entry

The design entry phase consisted of using the Engineering Capture System. This schematic entry package was well suited for hierarchical design, making it possible to easily design generic cells which could be repeated multiple times at a higher level. The hierarchical nature of this tool also made it very easy to observe all connections within the design.

The built-in symbol generator and error checker made creation of symbols very straightforward and painless. At any time when the design of a cell changed, which happened continuously, all that was needed was a quick click on the CREATE-SYMBOL option to replace the old symbol. If at any time there were errors between the symbol and the schematic, the error checker would point them out.

Design Verification

I generally used the design simulator after placing and routing to perform both logic verification and timing verification simultaneously. The simple process of back annotation made it easy to verify the design with post layout delay information. The waveform tool made it easy to enter stimulus to the design, and observe the output during simulation.

FPGA Architecture

The FPGA architecture made it possible to achieve a fast decode circuit with minimal clock to output skews — both of which were extremely important in this design. The direct outputs from each of the first-level gates allowed for minimal propagation delay in sections of logic that did not require the entire logic block. Additionally, input selection into the last mux of the logic block made it possible to implement the address mux much faster than standard AND-OR architecture. The end result was a complete high-speed DRAM controller design that was accomplished in several weeks — versus several months with custom designs.

QAN7
FPGA Cache Controller for the 486DX

Russ Lindgren

HIGHLIGHTS

- Zero wait state operation
- Flexible addressing supports cache RAM sizes from 128K to 1024K
- Look Aside implementation – no main memory speed penalty for cache misses
- Parallel design – concurrent access of Tag and Cache Lookup

SUMMARY

This application note presents an example of a versatile, high-performance cache controller using two QuickLogic QL12x16 FPGAs (see Figure 1). The cache controller is designed to interface with a 66 MHz 486DX2 microprocessor. Due to the performance characteristics of the QuickLogic pASIC™ 1 Family of FPGAs (10 ns pin-pin), the design can support cache 20 ns static RAMs of virtually any size and type up to 1 Megabyte. The cache controller can be 2 or 4-way set-associative and features zero wait-state operation.

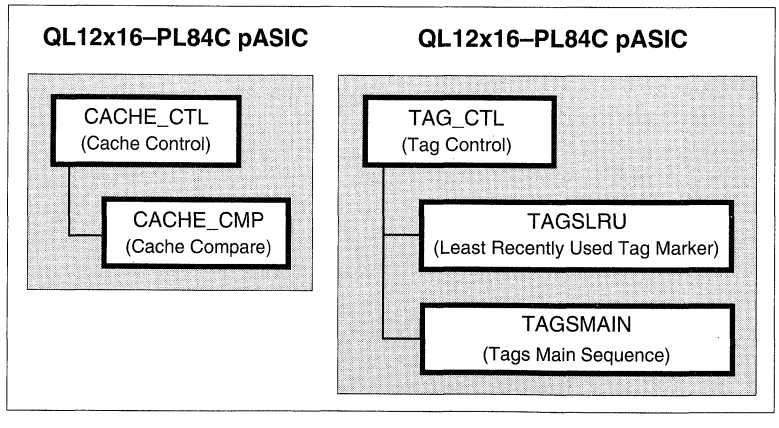


FIGURE 1
Cache Controller, the CACH_CTL and TAG_CTL Function Blocks

QUICKLOGIC

By integrating the CPU, MMU, 8K cache, and clock doubler within a single device, the 486DX2 has an internal processor and memory system that can operate at 66 MHz with a 33 MHz clock. Because the internal cache operates at 66 MHz, it can simultaneously feed data to the CPU and MMU while initiating data prefetch from either a cache or main memory. One way to increase performance is through a large, external data and instruction cache. This versatile cache closely matches the data structure of the onboard cache and can pipeline bursts of data into the onboard cache at the 32-bit data width. The design presented in this application note illustrates the techniques needed to achieve zero wait state operation and highlights the advantages of QuickLogic's pASIC devices for implementing processor to memory logic.

The cache controller design presented here is representative of the cache-control logic needed to provide efficient system operation with high-performance processors like the 66 MHz 486-DX2. This application note outlines the advantages of implementing cache control logic in the QuickLogic pASIC architecture. The design was created using the Data I/O ECS schematic capture package and the X-SIM simulator integrated with the QuickLogic SpDE place and route tools. Copies of the detailed design files are available on QuickLogic BBS or on request from QuickLogic (QS-QAN7).

DESIGN OBJECTIVES

- Zero wait state operation (with a 33 MHz CPU clock)
- Fast and dense cache coverage of the full 4 GByte address space
- Cache size from 128K to 1024K supports different RAM configurations
- Set associative, 2-Way design
- Parallel access of Tags and cache data

The cache controller design presented in this application note is fully associative and can cache any block in the entire 4 gigabyte address range of the 486 CPU, because it supports up to 30 address lines and 32-bit wide data. The tag addresses are 16 bits wide, plus 2 bits for burst addresses, so with 12 set bits the cache can store any address value. Twelve set bits translates (through the use of 12 address lines accessing the tag RAM) into 128 KBytes of cache RAM, because the number of bytes in the cache RAM array is always 16 times the size of the number of tags and because there are two tags for each set address. However, the advantage of using more cache memory is denser cache coverage of main memory. Note that although the CPU can access 32 bits of linear addressing, the lowest two address bits, AD0 and AD1 are not supplied directly because the 'usual' access is via 32-bit wide data. Access to individual bytes is handled by the CPU with four 'byte enable pins.'

DESCRIPTION OF DESIGN PROBLEM

In a single processor design, the 486DX2 supports DMA and bus snooping through an external address line which allows other bus masters, such as a DMA controller, to drive the address pins of the 486 with the memory address being modified. The on-chip cache controller will then invalidate matching tags in the cache. By mimicking this feature of the CPU, an external cache controller can easily maintain valid copies of main memory, even during DMA access. Because a memory area may change dynamically, as in a frame buffer for video capture, those noncacheable addresses are flagged with a special pin (KEN #) that can disable the cache for those areas. Because the 486DX2 has a single bus for both data and code, a cache controller must support both data and code accesses. When caching ROM and code areas, a write protect line can keep the cache from being invalidated by processor writes, increasing code performance.

One of the best ways to increase the cache performance of the 486DX2 is for the external cache controller to closely match the structure of the internal 8K cache, different principally in cache size. The internal cache of the 486DX2 is 4-way set-associative that holds 128 sets of four lines each. Each data line is 128 bits long, or four 32-bit words. The external cache controller needs to maintain this line size per tag and to support a fast, burst memory access mode of the CPU.

When accessing data in the burst mode, the CPU only needs to output one address to access the four 32-bit words per line, because the locations of the subsequent accesses are predefined. Because the 486 may first access any one of the 32-bit segments of the line, the external cache controller must replicate this predefined access sequence of the 486DX2 (see Table 1).

TABLE 1 CPU Access Sequence

FIRST ADDRESS	SECOND ADDRESS	THIRD ADDRESS	FOURTH ADDRESS
0	4	8	C
4	0	C	8
8	C	0	4
C	8	4	0

Timing Considerations for Zero Wait State Operation of a Large Set-Associative Cache

Designing a large, set-associative cache for the 66 MHz 486DX2 requires minimizing data delays and paralleling operations to achieve true zero wait state performance. With the external CPU bus clocked at 33 MHz, the 486DX2 data book specifies that approximately 40 ns is available from the falling edge of the address enable signal (ADS#) to the first valid 32-bit wide data access from memory (this time is approximate because the CPU clock doesn't necessarily have a perfect 50 % duty cycle). This parameter applies to the burst ready signal (BRDY#) as well as to the bidirectional data lines. The burst ready

signal comes from the external cache controller and indicates that the cache is ready to deliver a line of data. For zero wait state burst operation, the next 32-bit word must be on the data lines with each subsequent clock, within a period of 30 ns. Even with the use of fast, static Tag and Cache RAMs which offer 12 and 20 ns access times respectively, sequential operation would give only 8 ns (i.e., 40 –(12+20)) for address latching and tag compare, an unattainable spec for TTL levels (see Table 2). How can zero wait states be accomplished?

The key is to parallel these delays so that the tag match and valid signals don't depend upon this sequence of RAM access delays and to begin the access to the cache RAMs as soon as possible. Because the cache RAMs are addressed in a set-associative manner (this is described in detail later), you only have to determine which tag address matches the current address and then output the lower (one or two) bits of the cache address, or better, simply output the appropriate output enable to the cache RAM bank. The upper bits are the same since they are determined by the set address. For a 2-way cache, if it's not one, it has to be the other. And if the cache doesn't have the correct data, there's no penalty for accessing the look aside cache because you also begin the access to the main memory in parallel with the cache process (see Table 3).

TABLE 2
Sequential Access of Tag and Cache RAMs

START: ADS # LOW	OPERATION DELAY TIME (ns)	CACHE DELAY TIME (ns)
Address Latched	10	10
Tag RAM Access	10	20
Compare Tag and Address	15	35
Cache Access	20	55
Total Delay (One Wait State)		55

TABLE 3
Parallel Access of Tag and Cache RAMs

START: ADS # LOW	DELAY (ns)	START: ADS # LOW	DELAY (ns)	CACHE DELAY TIME (ns)
Address Latched	10	Address Latched	10	10
Cache Access	20	Tag RAM Access	10	20
		Compare Tag	15	35
		Cache Output Enable	5	40
Total Delay (Zero Wait States)				40

Yet another approach worth mentioning is connecting the Tag RAMs directly to the CPU bus, circumventing the address latch delay. This technique works if you use tag RAMs with an internal ALE signal or if the cache controller logic can perform a read-write cycle in under one clock cycle (30 ns). In either case, you need to depend on fast decoding logic, and most FPGAs other than QuickLogic's pASIC devices can't handle this speed grade of the 486DX2 CPU.

DESCRIPTION OF DESIGN SOLUTION

The design of the cache controller is able to provide cache coverage for the full address space of the 486 CPU with many different configurations of tag and cache RAM. To support a wide range of different RAM memories, configuration bits are used to determine address decoding of the tag and cache RAMs. Functionally, the cache controller design can be configured for 2-way or a 4-way set-associative operation. The tags are fully associative, meaning that they can be mapped to any memory address of the CPU (naturally, IOs are not cached). Because a cache controller is I/O intensive, the design is implemented in two QuickLogic, 84-pin (PLCC) 12x16 devices. Figure 2 presents the schematic connections between these two FPGA packages. The logic for interfacing to the CPU and Memory and for maintaining the tag RAMs resides in one device called TAG_CTL. The sub-schematic for the TAG_CTL block is shown in Figure 3. The second device is labeled CACH_CTL and covers address/tag comparison and control of the addressing for the cache RAMs. The CACH_CTL device can be doubled up to provide 4-way set-associative operation. Together these two devices comprise a complete cache controller fully implemented in QuickLogic's pASIC family.

How Does a Set-Associative Cache Operate?

A group of the lower address bits of the CPU, such as A4 to A16, supply the address to a 'look up' table of address tags. When the CPU outputs a new address (with the ADS# strobe), the cache controller looks up the tags in the tag RAMs, compares them with the CPU address, and determines a cache hit or miss. For a cache hit, the hit tag address is passed to the cache RAMs and used to sequentially access a line of data (a line is composed of four, 32-bit words). For a cache miss, the least recently used tag (for a set) is replaced with the upper portion of the current CPU address and successive 32-bit words are written into the cache memory in parallel with the CPU reads. The state diagram representing the main state machine used to control these operations (TAGSMAIN) is shown in Figure 4. This state machine was implemented in one-hot fashion (one state per flip-flop) to take advantage of the QuickLogic architecture.

FIGURE 2
Schematic Diagram of CACHE_CTL and TAG_CTL QuickLogic pASICs

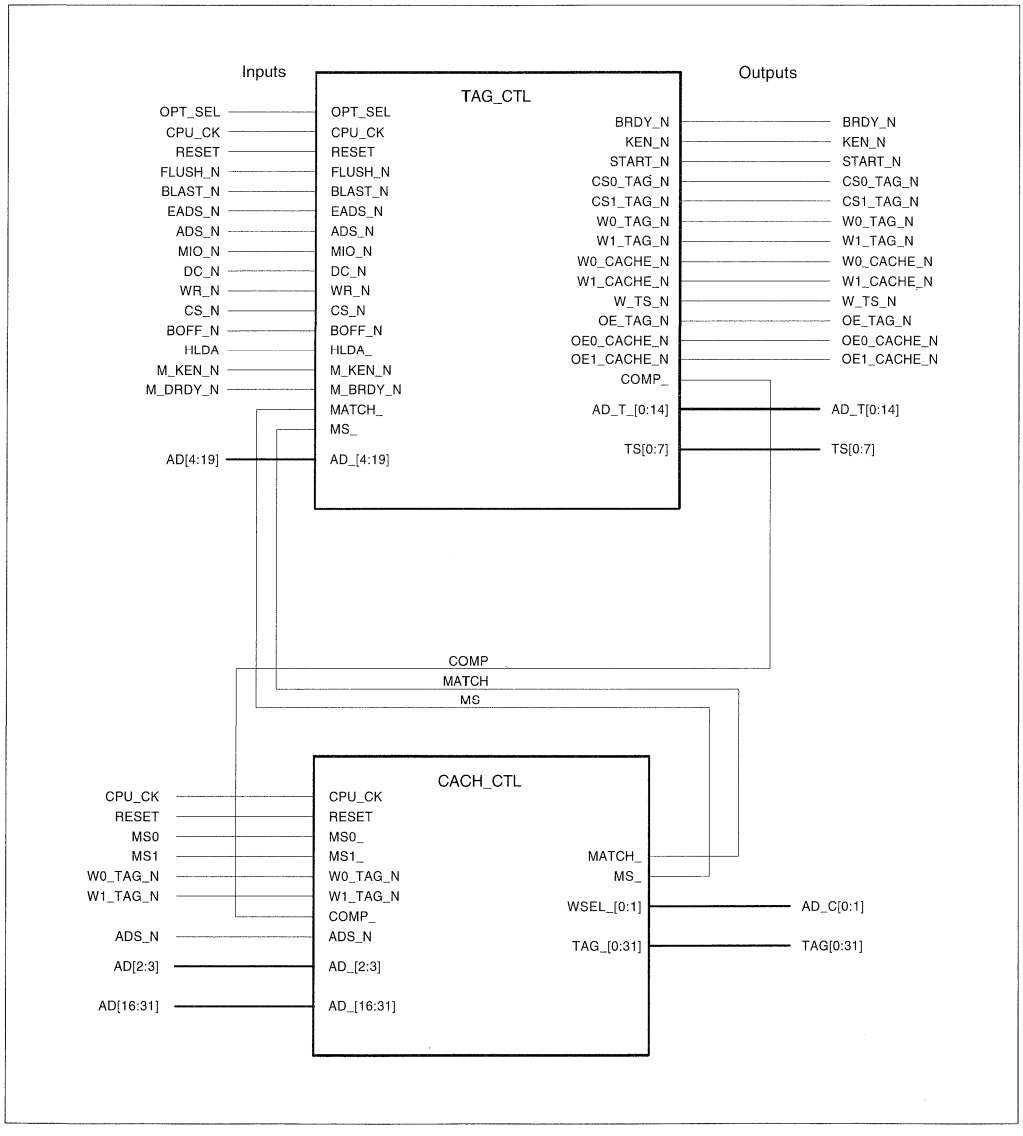


FIGURE 3
TAG_CTL Subschematic (Major Elements)

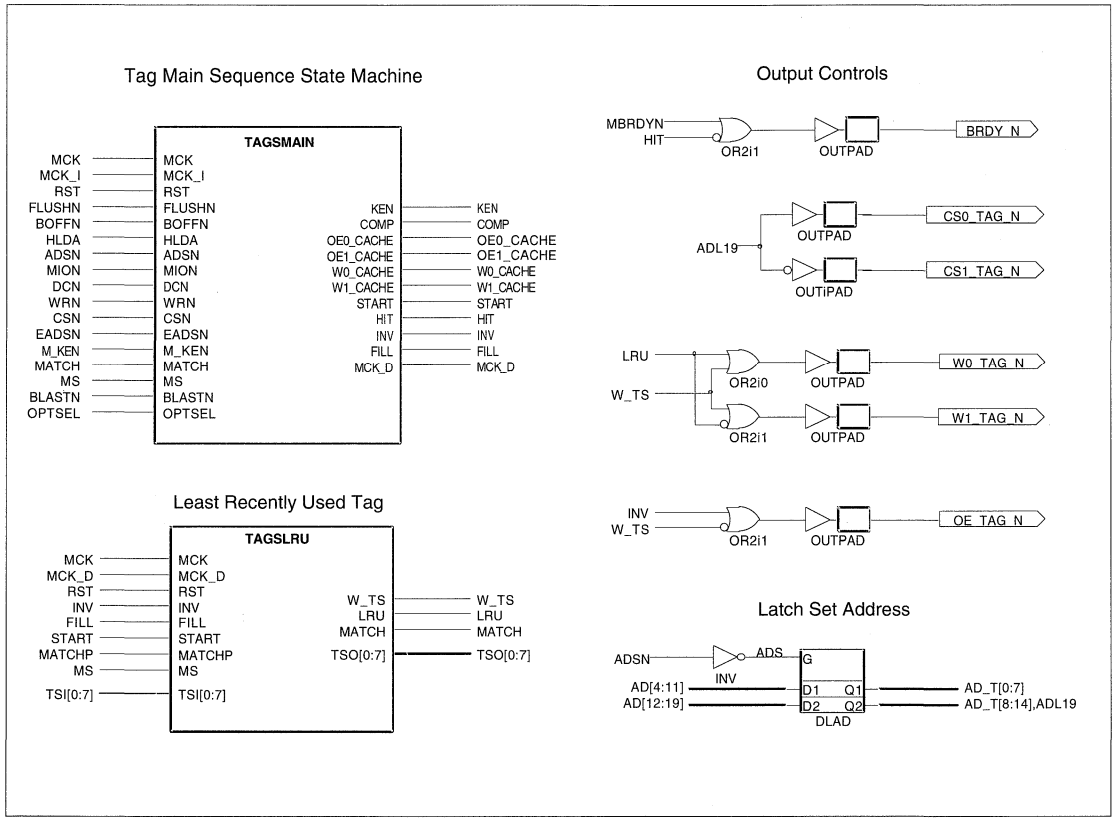
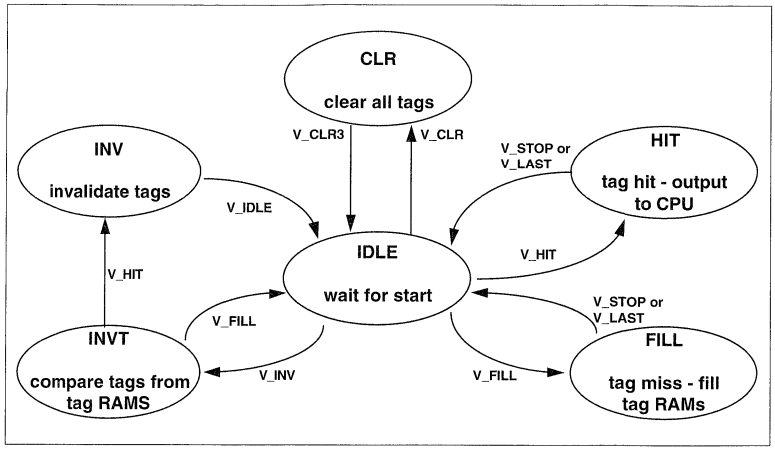


FIGURE 4
TAG Main Sequence (TAGSMAIN) State Machine Diagram



Although at first glance it may seem odd that the lower addresses are used as sets, to access the cache address tags, this approach provides the easiest, most efficient means to cover the entire address space of the 486 without knowing the actual size of the main memory. Each set refers to a group of 2 (for 2-way) or 4 (for 4-way) tags available at each set address. Because each tag can be up to 16 bits wide, a group of two tags are 32 bits wide and a group of four tags is 64 bits wide. Thus, the width of the tag RAM determines the number of tags available for each set. A valid bit and last access bits are maintained for each tag, and stored in a separate tag status RAM. Through the use of an 8-bit status RAM, independent of the tag RAMs, the cache controller can use tags up 16 bits long. After a 'RESET' or a 'FLUSH' command is issued, all the tags are sequentially 'invalidated' before the cache resumes operation. During the tag flush operation, all memory accesses generate a START command to main memory. To support DMA access, the cache controller supports bus snooping in parallel with the 486 CPU. When an external bus master writes directly to memory, it must activate the EADS# line and supply an address, signifying that the contents of that address have changed. The address is compared to the tags just like a CPU access, but a cache hit causes that tag to be invalidated. Also, an external input, KEN#, allows external decoding logic to lockout memory areas that dynamically change and are not cacheable.

The cache controller design does not directly support CPU writes and a CPU write to a cached line invalidates that line. This is because the CPU often writes to only one byte of the 32-bit memory. Due to pin constraints (which will change with the availability of 100 pin QFPs and larger for the QL 16x24B), only 32-bit access of cache RAM could be supported.

For updating the cache, an LRU (least recently used) algorithm is implemented. This algorithm is quite simple for a 2-way cache, because if the last hit occurred in one way, the other way is the least recently used. When the cache controller determines a tag hit, it stores the single bit LRU flag for that set, the 'way' for the hit. Physically, the LRU flag reflects whether the upper or lower 16 bits of the 32-bit wide tag RAM's contained the last hit, where a zero means the lower bits and a one the upper bits. The least recently used way is then the complement of that flag and determines which tag RAM bank receives the new value.

Cache and Tag Ram Design, Size and Organization

By using up to 16 bits per tag, it's possible to handle cache memory sizes up to 1 Megabyte. This maximum cache RAM configuration uses 16 address lines and two burst lines (2^(16+2) = 256K), and requires pairs of 128K x 8 static RAMs. Note that the cache is always 32 bits wide. If parity is needed, it can be generated by additional parity generators and stored in the cache by using RAMs that are 9 bits wide. The cache RAMs used in any configuration must have an access time less than or equal to 20 ns (this allows 10 ns for delay times through the QuickLogic cache chips). The wide variety of acceptable RAM types show how versatile the QuickLogic pASIC architecture is for memory design. The number of bytes in the Cache RAM array is always 16 times the size of the number of tags.

The Tag RAMs must be a different type (to achieve maximum performance), and are based on 10 ns 8K x 9 or 16K x 4 static RAMs. The acceptable RAM types shown in Tables 4 and 5 are listed by cache and tag RAM sizes. Table 6 shows how the size of the cache corresponds to different tag RAM configurations.

TABLE 4 Cache RAM Organization (20 ns max. access time)

CACHE SIZE (KBYTES)	RAM ORGANIZATION	# ICs	ADDRESS LINES	SELECT LINES
1024	128K x 8	8	17	2
512	128K x 8	4	17	0
256	64K x 4	8	16	0
256	32K x 8	4	15	2
128	2 x 4K x 16	8	12	4
128	8K x 8	16	13	4 [1]

[1] Using the pASIC output drivers, simultaneously driving more than a total of 12 cache and tag RAMs may slow access time.

TABLE 5 Tag RAM Organization (15 ns max. access time)

# TAGS (x1024)	RAM ORGANIZATION	# TAG RAMs	STATUS RAM ORGANIZATION	ADDRESS LINES	SELECT LINES
128	32K x 9	8	32K x 8 x 2	15	2
64	32K x 9	4	32K x 8	15	0
32	8K x 9	8	32K x 8	13	2
16	8K x 8	4	8K x 8	13	0
16	2 x 4K x 16	2	8K x 8	12	0

TABLE 6 Tag RAM Correspondence to Cache RAM Size

# TAGS (x1024)	RAM ORGANIZATION (TAG RAM)	# ICs	TAG WIDTH	CACHE SIZE (KBYTES)
128	32K x 9 x 2	8	36	1024
64	32K x 9	4	36	512
32	8K x 9	8	136 (or 72)	256
16	8K x 8	4	32	256
16	2 x 4K x 16	2	32	128

CPU Control Lines and the Cache Controller

The following descriptions include signals connecting directly to the 486DX2 processor (CPU). Note the '#' suffix always indicates an active low signal and no suffix indicates an active high signal. Each signal in the design retains the same name(s) used in the 486DX2 data sheets, with the exception that in the schematic the '#' is replaced by "N' for compatibility with the XSIM simulator. In each state description, the initiating signal(s) are listed following the 'state name' to help understand the functionality. Note that DWORD means one 32-bit word. CPU CK always refers to the 33 MHz, master CPU clock.

Reset or Flush — RESET or FLUSH #

These operations clear the Tag Ram buffers by invalidating all data. Reset also performs internal initialization of all registers.

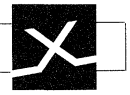
Start External Address Cycle — ADS #

Latches 'Set', 'Tag' and Burst addresses from the CPU.

Retrieves Tags from Cache memory, compares Tags for cache hit.

Moves to Burst Out state machine on hit, -or-

Moves to Memory Start on miss.



Burst Out — BRDY # (directly affects CPU BRDY #)

Informs processor data now available for burst line fills.

New data DWORD output on each successive state until BLAST # from CPU.

Memory State — START # (and KEN #)

Fills cache line(s) synchronously with CPU access when KEN # is asserted.

Cache Invalidation — EADS #

When EADS # is asserted, it stops any other cache memory process.

Latches external address for 'Set' and 'Tag' addresses

Tests for cache hit

Invalidates line(s) on hit

Requires two CPU CK cycles to read-modify-write invalid Tag address on hit

Write Invalidation — ADS #, WP and WR (see CPU State table)

This implementation doesn't support write-through to the cache, instead the line is invalidated, unless WP (write protect) is asserted

To decode whether the cache is active during the current processor state, a CPU state table (refer to Table 7) is used to determine the operation of the cycle in progress. This approach simplifies adding functionality to the device by allowing different branches for each CPU state-cycle. The CPU states are decoded and used to determine the state transitions in the TAGSMAIN state machine (Figure 4).

TABLE 7 CPU State (ADS # Asserted)

M/IO	D/C #	W/R #	CPU STATE
1	0	0	Code Read (if KEN no. asserted)
1	1	0	Memory Read (if KEN no. asserted)
1	1	1	Memory Write (Invalidates cache line on

Cache Process Overrides

Certain bus-mastering signals require immediate cessation of some or all CPU bus activity or affect the linked operation of the address and data bus. These signals are noted along with their effects on cache operation.

HLDA

HLDA is the active high hold acknowledge signal generated by the CPU in response to a HOLD request. HLDA means that the CPU is relinquishing control of the bus to another bus master. Although this operation may not be used for this single processor implementation, support logic is provided in the cache controller. HLDA has the same effect on cache operation as BOFF #.

BOFF #

When asserted, all data and address signal must be released into a tri-state mode by the beginning of the next CPU CK cycle. If a cache operation (including burst cache line fills) is in progress, it is immediately terminated. Note that for cache fills, this requires that the validation bit be asserted (as OK) upon completion of all burst line fills. Since the validation bit is stored internal to the cache controller implemented in QuickLogic's pASIC ICs, this carries no additional cycle penalties as are usually found in most look-aside cache controller configurations that store the validation bit with the Tag in SRAM. Note also that BOFFIN #, an internal signal to the cache controller is defined by: BOFFIN # = ! (!BOFF # + HLDA).

AHOLD

Used to invalidate the CPU internal cache address, AHOLD is a request to the CPU to release only the Address bus in the next cycle. Because this signal doesn't affect the control and data lines and the cache controller latches the addresses internally, this signal does not directly affect operation of the cache controller. Usually this request is generated by a DMA or EISA bus controller prior to asserting EADS #, which is used to invalidate internal cache lines.

ADVANTAGES OF QUICKLOGIC DESIGN TOOLS

What sets the QuickLogic design tools apart from other FPGA development tools is the combination of integrated delay tracking, error checking and design management. Wherever you are in the design process, it's always possible to open another window to the Hierarchy Navigator and the X-Sim simulator to check the functionality of a design. Every check will use either estimated or guaranteed delays, depending upon whether the design has been placed and routed within a pASIC device.

Design hierarchies are first created with the schematic editor which provides sheet-level error checking and symbol creation. By capturing schematics containing just the IO ports for each symbol or logic block in the design, the complete top-down schematic hierarchy can first be drawn and then checked with the Hierarchy Navigator. Once the schematic hierarchy has been created, you can then begin testing the design. Testing the design is composed of three steps — loading in the design hierarchy, generating the stimulus and test vectors and running the simulation. Cross probing is dynamic and supported across all these tools. For example, to view the simulation results for any net, simply select probe from the navigator menu and click on the net. The current simulation value is displayed on the schematic by a virtual LED and the waveform is drawn in the waveform viewer.

In the design of the i486 cache controller, three levels of simulation are used to develop and verify the design. The 'CPU-Test' schematic is used to functionally simulate the interaction of the i486 CPU and the cache. The next level down in the hierarchy is the 'CACHE_Q' schematic, used to functionally simulate the interaction of the cache controller as a whole with the cache RAMs. One further level down is the 'CACHE_R' schematic, that groups together the signals of the two chip cache controller designs 'CACH_CTL' and 'TAG_CTL', both implemented in the 84-pin QL12x16 FPGA. Copies of the detailed design files are available on the QuickLogic BBS or on request from QuickLogic (QS-QAN7). Since these two designs can be placed and routed within a pASIC device, you can simulate them using the guaranteed post layout delays and compare the results to the functional simulations.

Probably the most vexing portion of any FPGA design is placing and routing the design within the selected component, but again, the automated SpDE toolset coupled with a fast, flexible device architecture make this process productive and direct. When the logic design (in the QDIF format) is read into SpDE, it's checked for single-ended nets, too many IO pins and similar errors, before running the place and route. For each of the designs presented, these tools yielded 100% completion with acceptable delays in thirty minutes. These routing delays were then back annotated for accurate simulation. One feature that proved useful for simulating the capacitive effects of the large cache memory is the ability to place custom capacitive loads up to 150 pF on the output nets, that made it possible to verify that the TAG_CTL pASIC could supply sufficient drive for high-speed cache operation.

ALTERNATIVE SOLUTIONS AND THEIR DRAWBACKS

The successful design of a i486 cache controller with FPGAs requires consistent delays for the address and tag inputs. For the tag comparison to work, 48 lines need to route to the address comparator logic block, which determines a cache hit or miss within 10 ns of receiving the tag addresses. Although some of the new static RAM-based FPGAs can offer reasonable performance over earlier generations, the routing resources of these devices are still too inconsistent for the type of wide, high-speed bus performance needed. In addition, the tools for these FPGAs are generally substantially more expensive, require much longer run times for place and route, do not offer 100% automatic place and route at high utilization, and do not support the interactive capabilities of the QuickLogic tools.

Complex PLDs offer delay matching for multiple nets, but they lack either the density or speed required by high speed-microprocessors. To implement this cache controller design in most EPLDs you would need partitioning software that could break a large design into many smaller components, but even with the best possible mapping, you would still incur additional delay penalties from the added input and output buffers. Overall, the QuickLogic pASIC devices offer the best solution to this problem.

QAN8
Video Support Logic and VME Interface

Dan Le Vasseur

INTRODUCTION

This application note describes how a video board based on the TMS34020 Graphics Processor was designed using a QuickLogic pASIC® device to implement a complete VME slave interface and all the high-speed video support logic required by the LSI video components.

A design that was previously implemented using a variety of high-speed PALs, a MACH device, and some 7400-series chips was packed into a single FPGA. The QuickLogic pASIC Toolkit plus an FPGA synthesis tool were used to convert the existing design files into a form that would allow easy integration with new circuitry so a considerable amount of design time was saved.

Statement of the Problem

The intent of this project was to update a video board using a TMS34010 as a graphics processor with a TMS34020, and then consolidate the rest of the design into as few chips as possible using a single FPGA on a 6U VME board (with 2 DIN connectors). The prior design was implemented with an assortment of 20- to 44-pin programmable devices. Shrinking the overall design would allow the incorporation of substantially more video memory on the fixed size of the 6U board. This required an FPGA with both a flexible architecture, to accommodate the previous PLD designs, and high speed, to implement video support logic operating at 40 MHz. Many FPGAs claim high system speeds, but due to architectural limitations cannot deliver the rated performance when filled with a complex design.

Another challenge of the task was selecting the tools; the development tools for the FPGA would have to be easy to learn and use due to the short time available. I was required to take a design that was already implemented in a handful of small-scale programmable devices, upgrade it, and convert it into a single-FPGA solution as quickly as possible.

QUICKLOGIC

Design Approach

The QuickLogic part QL12x16-2PL84C was selected to replace a MACH device, three PALs and some 7400-type logic since a fast pin-to-pin delay was required to support the time-critical video-related and memory-support features. However, this selection was also contingent on the quality of the development tools.

Because the QuickLogic tools are based on the Windows Graphical User Interface, all applications were intuitive and easy to learn and use. In addition, Windows made it easy to integrate multiple, third-party tools into a single environment. Even DOS-based tools such as the Exemplar Logic Synthesis System can be run in a DOS window.

The Exemplar Logic Synthesis System was used to convert the ABEL PAL files used in the earlier TMS34010-based design into files that could be combined into a single device by the QuickLogic pASIC toolkit. (Future releases of ABEL will permit direct input to the QuickLogic system without the need for synthesis tools.)

In summary, the approach taken in this application was as follows. The portion of the system that was being upgraded from the prior design, consisting of PALs and a MACH part, was modified and processed by Data I/O's ABEL compiler and optimizer producing <.tt2> files. (Any device dependencies must first be removed from the ABEL source code.) The <.tt2> files were fed into Exemplar Logic's toolset and individual <.qdf> files were produced. The remaining portions of the new design were drawn using QuickLogic's schematic editor. The <.qdf> files from the Exemplar Logic toolset were included in the schematic as symbols using the QuickLogic <.qdf> schematic converter. Then the schematic was converted to a single <.qdf> file representing the entire design that was then placed, routed, and simulated in the pASIC environment.

Variations on the above scenario are also possible. The individual PAL files can be combined at the equation level using Exemplar's design integration language, Exemplar Integration Language (EIL). The biggest advantage of this approach is that product term sharing can occur earlier in the design process, leading to a more optimized implementation. Other problems, such as increased fan-out that can occur after combining separate <.qdf> files, can also be avoided using this technique. However, if the separate PAL files contain mutually exclusive logic, the earlier approach used above should produce acceptable results with a minimal effort. Also note that if you are converting PAL designs that implement gated clocks, asynchronous functions, or combinational feedback loops, you should modify them to avoid introducing problems into the FPGA implementation.

Figure 1 is a functional block diagram of the board.

Overall Design Description

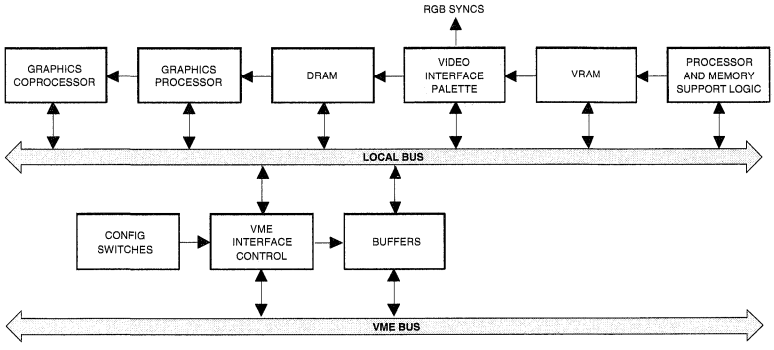


FIGURE 1
Board Block Diagram

The Graphics Processor and memory support logic consist of address decoding on the local multiplexed address/data bus, decoding of memory access types, and deriving RAS signals from the local control bus. The Video Interface Palette, DRAM, and VRAM are devices hanging on the local multiplexed address/data bus, and decoding is required to select them, although most of the control lines are provided by the Graphics Processor. Additionally, several VRAM control lines are derived from the local bus cycles, for example, the recognition of serial-register transfers, Graphics Processor interrupt vector fetch, and host cycles.

The clock circuitry consists of a clock for the synchronous state machine logic in the VME control logic, the Graphics Processor, Video Palette, and pASIC device. Additional clocks are required for the Video Palette dot clocks needed to support various pixel data rates. An ECL clock is required to support the higher resolution screen formats.

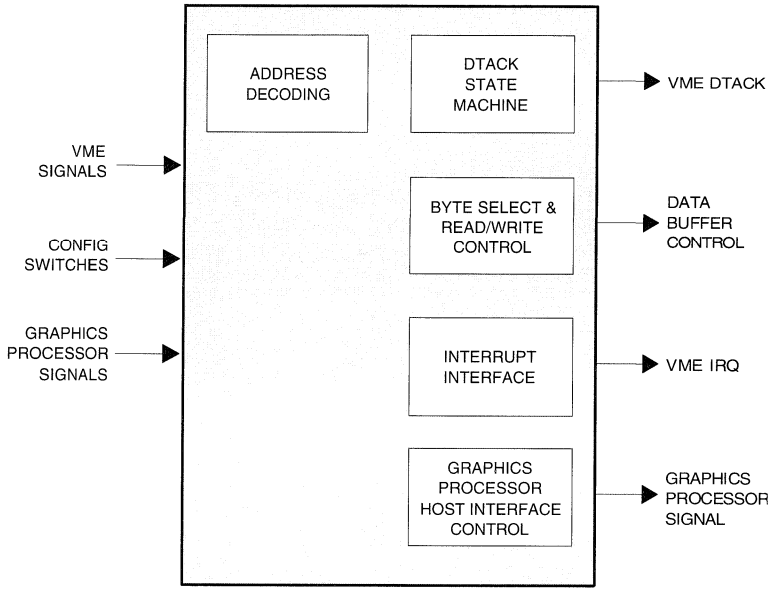
pASIC Implementation: VME Interface

The VME interface control logic is implemented in the pASIC device. The VME signals interfacing to the pASIC device fall into several groups: the Data Transfer Bus lines (DTB), the Priority Interrupt bus lines, and the Utility bus lines. The DTB lines consist of the address lines A01-A31, the address modifier lines AM0-AM5, the data strobe lines DS0* and DS1*, long word select LORD* the data lines D00-D31, address strobe AS*, bus error BERR*, data transfer acknowledge DTACK*, and a read/write select line WRITE*. The Priority Interrupt bus lines consist of interrupt request lines IRQ1*-IRQ7*, interrupt acknowledge line IACK*, interrupt acknowledge daisy-chain input line IACKI*N*, and interrupt acknowledge daisy-chain output line IACKOUT*. The Utility bus lines that are relevant to this board are system reset SYSRESET* and system fail SYSFAIL*. SYSRESET* is the global reset line for the VME bus, while SYSFAIL* is used to interrupt the Graphics Processor to display a "system fail" message.

Most of these lines are driven by the bus master; only DTACK* and BERR* are driven by the video board. Both are open-collector lines.

Figure 2 is a functional block diagram of the VME Interface logic implemented in the pASIC. The major components in the diagram are the DTACK* handshake state machine, the Graphics Processor host interface and byte select logic, the interrupt vector control and IACKIN*/IACKOUT* state machine and logic, and reset synchronizer.

FIGURE 2
VME Interface Control Block Diagram



The Graphics Processor and memory support logic are implemented in the pASIC device.

The local bus on the board consists of the time-multiplexed address/data lines which can be driven from the VME bus buffers via the host interface or the Graphics Processor. The chip select decoding logic implemented in the pASIC chip selects the appropriate memory bank being accessed or the palette chip. Additional functions implemented are decoding of the shift register transfer, host, and refresh cycles and controlling the memory chips accordingly. The fast pin-to-pin speeds of the pASIC are required to perform these tasks.

An additional feature of the pASIC device and toolset that proved valuable, particularly regarding the video logic, was that the timing delays remained relatively constant and predictable from one iteration of place-and-route to the next as the design was being modified. Going through multiple passes did not appear to affect the device timing significantly. The device has ample routing

resources such that the design could be added to without disturbing prior routing. Although the pASIC placement and routing is completely automated, producing acceptable results, the only manual intervention taken was the assignment of a limited number of pins. This was only done to facilitate PCB interconnections for the memory control signals and not an attempt to control the device timing.

Figure 3 is a functional block diagram of the video logic implemented in the pASIC. The major components in the diagram are the local bus address latch and address decoding, DRAM and VRAM control, and the palette chip control.

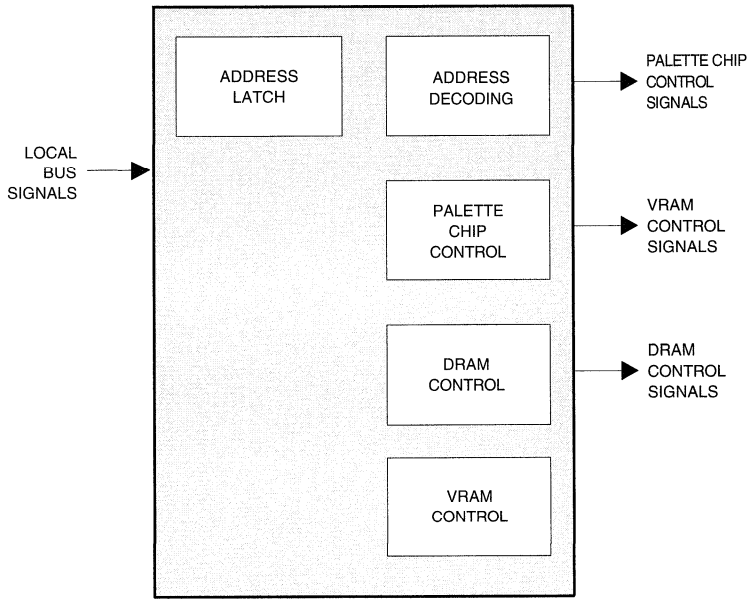


FIGURE 3
Video and Memory Support Logic Block Diagram

Device Utilization

Of the 76 signal pins available on the QL12x16-2PL84C, 74 were used, leaving two spares that could be used to bring out internal nodes for testing. Of the 192 logic cells available, the VME control logic and interrupt state machine used 64 cells, or 33%, the video support circuitry required 24 cells, or 13%; and miscellaneous TTL circuitry used four cells, or 2%. Over half (52%) of the pASIC's logic cells were unused. Regarding timing, as determined by the QuickLogic waveform simulator, typical pin-to-pin delays of substantial video decoding functions (without any manual tweaking) were on the order of nine nanoseconds, acceptable for this application. The substantial unused resource of logic cells could be used to further optimize the timing.

This article describes an application of a pASIC device (the QL12x16-2PL84C) where speed and board space utilization were the primary physical concerns. Existing design files from a prior board consisting of a variety of high-speed PALs, a MACH part, and some 7400-series chips was integrated with new circuitry and packed into a single FPGA using the QuickLogic pASIC development tools, thereby saving a considerable amount of design time.

SUMMARY

The toolset used involved the ABEL PLD compiler, the Exemplar Logic Synthesis System and the pASIC toolkit. The pASIC toolkit was not only used to integrate component files produced by Exemplar, but also to generate part of the schematic design, conduct simulation, and program the part.

QAN9
Optimizing pASIC Architecture Designs

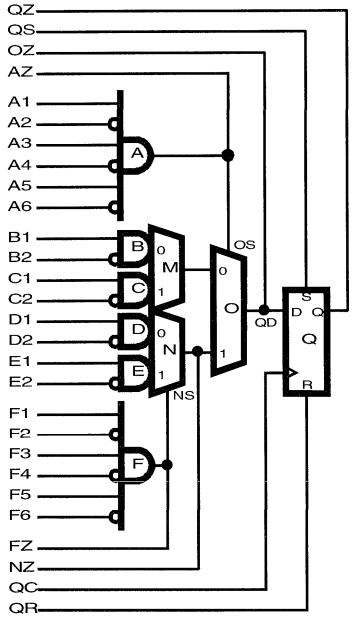
DESIGNING FOR SPEED

Many FPGAs require careful study of the device architecture. Beginner designers may find that such devices require tremendous effort to achieve the originally anticipated result. Often, expert designers are needed to meet the original design goals. QuickLogic's pASIC-1 devices utilize a fast and flexible architecture that eliminates the need for extensive knowledge of the underlying architecture. Using QuickLogic's state-of-the-art SpDE (Seamless pASIC Design Environment) software tools, the beginner designer can achieve expert-level results on the first project.

Background: The logic cell

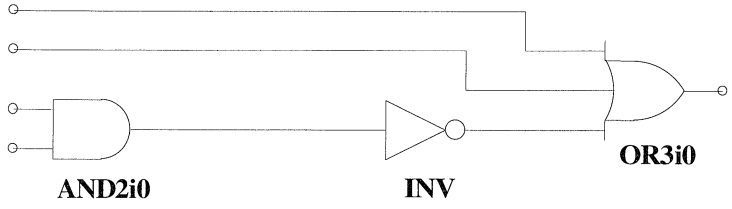
The QuickLogic logic cell has 20 combinatorial inputs, which allows the creation of wide gating logic. Each logic cell also includes a D flip-flop. The logic cell has five outputs—this increases the utilization and flexibility of the logic cell by allowing multiple macro gates to be packed into a single logic cell. Another valuable feature of the logic cell is that any function of three input variables can be accommodated in the multiplexing section.

FIGURE 1 The logic cell



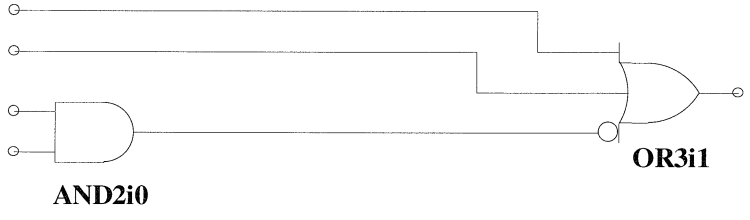
The pASIC architecture provides great flexibility in the use of inverted polarity inputs and outputs. Consider Figure 2, which shows one implementation of a simple circuit using an inverter.

FIGURE 2
Inverting logic



If implemented exactly as drawn, this circuit would require more than one logic cell and incur two logic cell delays. SpDE's Technology Mapper tool will "push" the inversion bubble, as shown in Figure 3, use less than one logic cell and incur only one logic cell delay. The Technology Mapper performs bubble pushing and gate collapsing—reducing complex logic functions into a single logic cell—in order to achieve expert-level performance and density without the need for expert-level knowledge of the logic cell.

FIGURE 3
Use a bubbled input



Speed Estimation

Speed estimation includes logic cell delay and net delay. Other FPGAs have highly unpredictable routing delays that make the task of speed estimation difficult. Because net delays in these devices vary wildly, designers are often unpleasantly surprised at the substantial difference between estimated speed and actual speed.

The pASIC architecture is built upon QuickLogic's ViaLink interconnect. The ViaLink provides an exceptionally low interconnect impedance (20 to 50 ohms), which results in an exceptionally fast routing structure. Unlike other FPGAs, this routing structure produces fast, intuitive, and predictable routing delays.

Logic cell delay is determined by the path through the cell (in other words, which inputs and output are used). The shortest path, 2.0 ns, is through either AND gate labeled A or F in Figure 1. The longest path, 3.6 ns, is through the multiplexer labeled O. Study of hundreds of customer designs has produced a statistical mean of 2.4 ns for the logic cell delay.

Figure 4 shows the results of a statistical analysis of QuickLogic net delay versus fanout. The plot illustrates a remarkably linear relationship between net delay and fanout. Much like a masked gate array (but very unlike other FPGAs), user intuition can be applied to accurately predict the net delay.

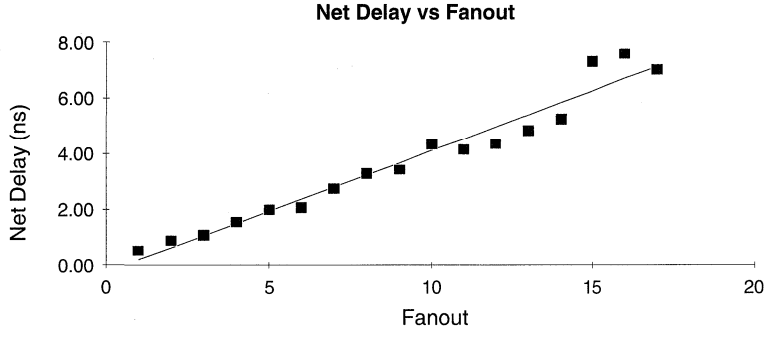


FIGURE 4
Net delay vs fanout

As an illustration, consider a typical fanout of three. From Figure 4 this fanout would result in a net delay of 1.05 ns. A single level of logic in this example would have a propagation delay of 3.45 ns (2.4 ns logic cell delay plus 1.05 ns net delay). For a more complex example, consider the 16-bit adder featured in the pASIC Macro Library. There are four levels of logic cells in the critical path of this circuit, so a reasonable estimate of its propagation delay would be 13.80 ns. The actual delay is 14.20 ns—the estimate is off by less than 4 percent!

These examples illustrate combinatorial delays. Calculations for circuits using the logic cell flip-flops add 2.4 ns for the Clock-to-Q of the register.

Fanout and Buffering

As discussed above, higher fanouts result in larger net delays. Very high-speed designs can be optimized by limiting the fanouts of nets in the critical paths. The SpDE tools provide fanout warnings when the design is first imported. Table 1 shows the fanout limits with respect to driver type.

TABLE 1 Fanout Warning Values

Driver Type	Fanout
INPAD	13
HDPAD	24
ANY GATE	13

These parameters are a warning to the user that for high-speed purposes some type of buffering technique might be desirable.

Example 1 gives a sample calculation of buffering versus nonbuffering net delay. This example is a comparison of a single logic cell driving a fanout of 16 versus the same logic cell driving two buffers, each with a fanout of 8. All numbers are for the QuickLogic QL8x12A.

Example 1 "To buffer or not to buffer"

Single net calculation	Using two buffers with fanout of 8
Fanout = 16	Fanout of 2 = 0.60 ns
	Buffer delay = 2.40 ns
	Fanout of 8 = 3.10 ns
Delay = 6.71 ns	Total Delay = 6.10 ns

In this example, buffering techniques are a "win" at fanouts of between 14 and 16.

Buffering Techniques

There are several buffering techniques that may be employed to increase performance in designs that have high-fanout internally generated signals. Figure 5 gives an example of selective buffering. This method is used to help separate critical paths from non-critical paths. This technique limits the fanout of speed critical paths by isolating the non-critical paths with a buffer.

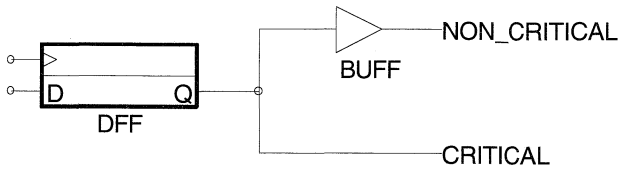


FIGURE 5
Buffer noncritical signals

Pros	Cons
• Easy to implement • Requires little additional logic • Does not increase fan-in	• Only useful if high-fanout net can be split into critical and noncritical sections

If all destinations on a high-fanout net are critical, the selective buffering technique offers little benefit. Another technique is the duplication of logic in order to break-up or distribute fanout. This technique—called paralleling—is illustrated in Figure 6.

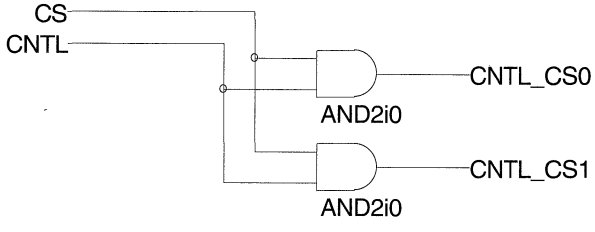


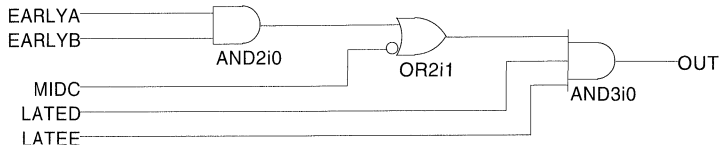
FIGURE 6
Use duplicate logic

Pros	Cons
• Cuts fanout in half (or better) • Does not add additional levels of logic	• Increases fan-in • May utilize as much as one full additional logic cell

High-speed Combinatorial Logic

An extremely effective technique for speeding up multilevel combinatorial logic is the optimization of late arriving signals. Figure 7 demonstrates the late arriving signals entering the last stage of logic while the early arriving signals enter the first stages of logic. In this manner, the varying propagation delays balance the varying arrival times of the signals. The signals EARLYA and EARLYB have the longest path to traverse. The signals MIDC, LATED and LATEE have successively shorter paths and therefore can arrive later than EARLYA and EARLYB.

FIGURE 7
Apply late arriving signals to latter stages of decode logic



Pros	Cons
• Requires no additional logic	• Only useful in certain circuits

High-speed State Machines

pASICs are very well suited for high-speed state machine applications. The wide fan-in of the logic cell allows the creation of wide gating functions—a single logic cell can implement a 14-input AND gate or a 13-input OR gate. These large AND and OR gates can be used to implement conventional sum-of-products state machines in just two levels of logic cells. These state machines can be designed for speeds of over 70 MHz (as described in the Speed Estimation section, circuits with two logic cell delays are extremely fast).

Sum-of-products state machines use binary state bits to encode the state (these state machines are also known as encoded state machines). An alternative to this traditional technique is the direct encoding of the state using a bit-per-state. These state machines, commonly called one-hot state machines because only one of the state bits is active at any time, are well suited to register-rich FPGA architectures.

The pASIC logic cell can implement a 4-to-1 multiplexer with 6-input AND gates on each of the select lines. Because of the small AND gates at the multiplexer inputs (labeled B, C, D, E in Figure 1), the 4-to-1 multiplexer section can implement any function of three variables. This structure—two 6-input AND gates feeding any 3-input gate—is extremely useful in the design of one-hot state machines. As demonstrated in Figure 8, one-hot state machines are compact and easy to test and debug. Implemented in a single level of logic cells, one-hot state machines can run at over 100 MHz in pASIC devices.

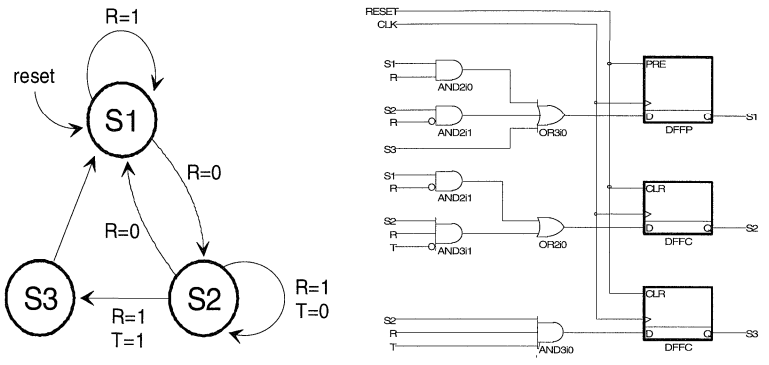


FIGURE 8
Use one-hot state encoding

DESIGNING FOR RELIABILITY

Good design practices produce reliable circuits, whether the design is an FPGA, an ASIC, or a printed circuit board full of chips. The high-speed capabilities of pASIC devices places a premium on these good design practices. Below are a few tips on good design practices that will help to ensure a reliable pASIC design.

Synchronous Design

Asynchronous circuits can cause reliability problems in any design, especially when an asynchronous signal is used to control the clock, set, or reset on a flip-flop. As an example, consider the circuit shown in Figure 9. In this design, the counter is reset asynchronously by logic that decodes a terminal count value.

This is a four-bit counter that should asynchronously reset the registers when a hexadecimal count value of D is reached. As shown in Figure 10, the counter resets at the transition of 7 to 8 because a glitch is generated on RCO due to skew on Q0-Q3.

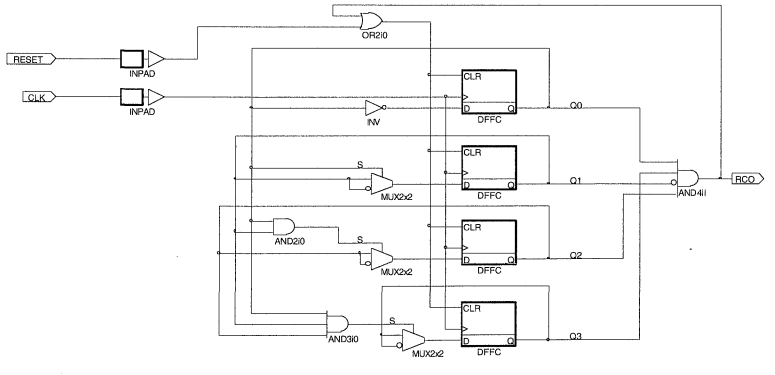
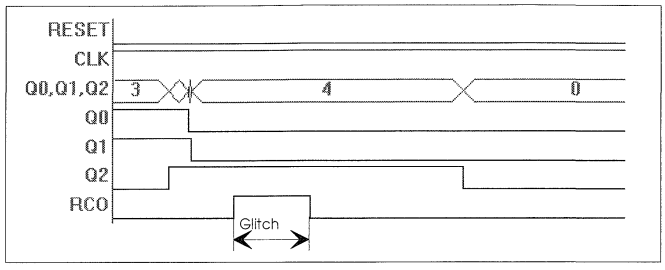


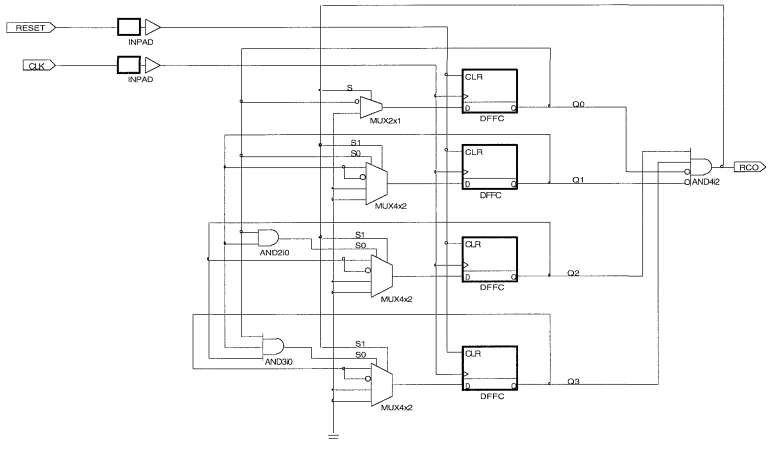
FIGURE 9
Example of asynchronous reset

Figure 10
Circuit failure caused by decode glitch



This circuit can be made fully synchronous, as shown in Figure 11. The decode now "anticipates" a reset value of C, one cycle early. This decode is then used to synchronously reset the counter.

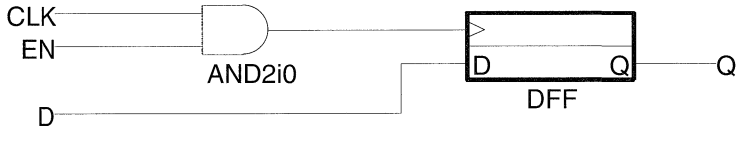
Figure 11
Synchronous decoded clear



Avoid Gated Clocks

Typically, gated clocks are used as clock enables, as shown in Figure 12.

Figure 12
Avoid gated clocks



This circuit can cause false clock signals to the flip-flop, even if CLK and EN are both synchronous signals. If CLK and EN arrive at the AND gate at slightly different times, a "false" clock signal will be produced. Furthermore, this circuit does not take advantage of the pASIC's high-speed clock networks, and places an additional level of logic in what may be a critical path.

Figure 13 shows the same circuit with a modified implementation which will:

- use a high-speed clock network
- implement in one logic cell
- not produce "false" clocks

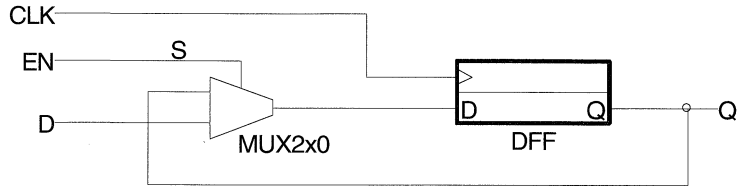


Figure 13
Use multiplexers

This implementation utilizes a high-speed clock network—a dedicated resource that is fanout independent and has exceptionally low skew. The EN signal selects either the current value Q or the new value D. The flip-flop is clocked synchronously on every cycle. This implementation is the DFFE macro in the pASIC macro library.

SUMMARY

Designing with QuickLogic pASIC devices is easy. The abundant, regular, high-speed routing structure, coupled with the automatic place and route tools, provides exceptional performance without manual intervention. The flexible and fast logic cell, coupled with the automatic logic optimization tools, provides excellent performance and capacity without the need to memorize the idiosyncrasies of the architecture. The pASIC architecture "behaves" like a masked gate array, and optimized designs are produced by following the same guidelines one would follow for a masked gate array—

- reduce the number of logic cell levels
- reduce the fanout on timing-critical nets
- apply late arriving signals to latter stages of decode logic
- use synchronous design techniques
- avoid gated clocks

QAN10
Peripheral Component Interconnect (PCI) Using the QL16x24B FPGA

Charles Geber

1.0 SUMMARY

The Peripheral Component Interconnect (PCI) is a recently-defined standard Local Bus for high-speed processors and peripheral controllers. PCI is intended to meet the local bus requirements of next generation high-performance computer systems for several years, and has been adopted by several key processor architectures and numerous system integrators. Digital Equipment (DEC Alpha), Motorola (PowerPC), and Intel (Pentium) are only a few industry players who have embraced this new bus interface for their processors and future desktop computers.

PCI was originally intended for communication between high-bandwidth LSI devices on the processor motherboard. However, with the introduction of processors such as the Pentium, PowerPC, and DEC Alpha, and the increasing need to directly interface to peripherals, the PCI definition was recently expanded to support add-in modularity via I/O connectors. Suppliers of high-speed custom subsystems (such as signal and image-processing, data encryption, etc.) can now directly interface to PCI to obtain the highest level of system performance.

In addition to its high-speed bus definition, PCI contains a sophisticated Configuration Space register block that can be used by system software for autoconfiguration of peripheral controllers. It provides the glueless interface to high-performance peripherals (SCSI, IDE, LAN, graphics, video) that the industry has long awaited. The power of the PCI bus and associated Configuration Space support allows designers to plan for the next generation of high-performance personal computers, eliminating the bottleneck of standard expansion busses.

This application note describes a complete PCI interface implemented in a single QuickLogic QL16x24B FPGA. The user side of the interface has been designed for a generalized 32-bit device with a typical READY and READ/WRITE-strobe handshake sequence; 24 bits of user device address have also been provided. The large logic and pinout capabilities of the QL16x24B device are key to providing the necessary interface functionality in a single FPGA device. In addition, the extremely fast I/O pads and internal logic can accommodate the stringent system timing requirements of the 33 MHz PCI bus.

QUICKLOGIC

The design implements a fully PCI-compliant interface that utilizes the PCI burst transfer mode for highest data throughput. All required PCI Configuration Space registers have been implemented in a highly modular structure; readers may simply modify the necessary fixed-value registers to contain the vendor, device, and revision identification for a specific product.

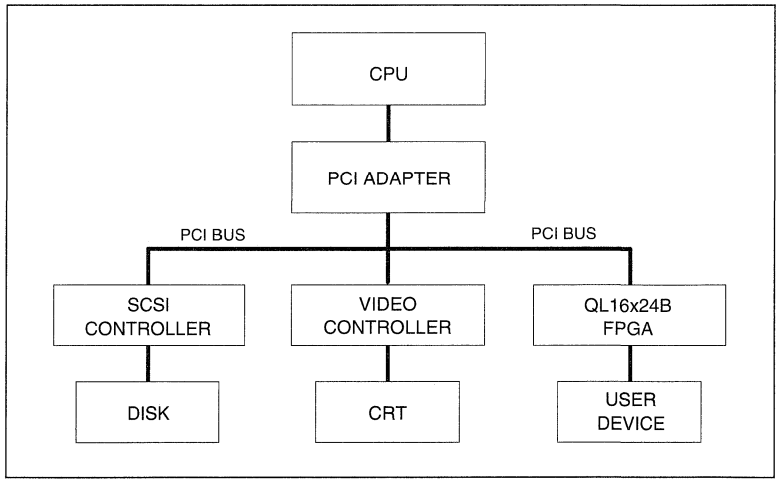
While portions of this application note may appear to only address specific areas of the PCI interface, the general design concept described may be applied to a variety of applications for various processors and peripherals. The design files and schematics are available from QuickLogic and can be easily modified to your particular needs. If you have any questions regarding this application note or other technical questions, please call the QuickLogic Customer Engineering Hotline at (408) 987-2100.

2.0 SYSTEM OVERVIEW

2.1 System Topology

Figure 1 indicates a typical PCI system topology. The CPU is coupled to the PCI bus via the indicated adapter. The figure indicates that a high-speed disk and video controller reside on the PCI bus as well as a user device, interfaced by the QuickLogic FPGA device described in this application note.

FIGURE 1
System Topology



2.2 PCI Bus Overview

The PCI Local Bus is a high-performance 32-bit (or 64-bit) bus with multiplexed address and data lines. It is designed to interconnect high-performance processors and peripheral controllers for a wide variety of next-generation computer systems.

The following is a brief overview of the PCI Local Bus. For a complete, detailed description the reader should consult the *PCI Local Bus Specification*, distributed by the PCI Special Interest Group.[1]

Address Space

PCI defines three address spaces: Configuration Space, defined to support PCI hardware configuration, and Memory and I/O Spaces, which are used by the attached devices for actual data transfer. Address decoding on PCI is distributed, wherein each device is responsible for decoding its own address. PCI supports two address decoding schemes: positive decoding, where a device looks for an address in its assigned range, and subtractive decoding, where a single bus-coupler device accepts all addresses that were not positively decoded by some other device. With either scheme, a device indicates that it has decoded its address and thus claims the transaction by asserting the tristate signal DEVSEL#.

Configuration Space

PCI provides for totally software-driven initialization and configuration via the Configuration Space. PCI devices are required to allocate 256 bytes of configuration registers for this purpose. (NOTE: not all of these bytes need be implemented in physical logic if a read-value of zero can be generated.) Accesses to the Configuration Space require external address decoding via the IDSEL control pin which functions as a unique "chip-select" for each device.

Data Transfer

The basic bus transfer mechanism on PCI is a burst transaction, composed of an address phase and one or more data phases. The first clock cycle on which the FRAME# control signal is asserted establishes the address phase of the transaction. FRAME# will remain active to indicate each data phase to follow, and will deassert at the start of the final burst data phase. Handshake signals for each data phase include IRDY# (Initiator Ready, indicating the Bus Master is supplying write-data or is ready for read-data) and TRDY# (Target Ready, indicating that the target device has accepted the write-data or is supplying the read-data).

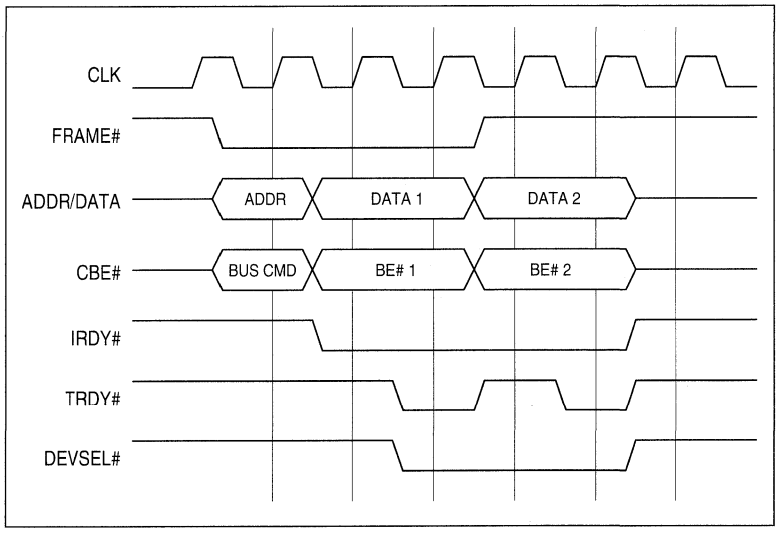
[1]PCI Special Interest Group
M/S HF3-15A5200 N.E. Elam Young Parkway
Hillsboro, Oregon 97124-6497
(503) 696-2000

Devices connected to PCI implement Bus Commands, which indicate the type of transaction that the Bus Master is requesting. Bus Commands are encoded in the 4-bit CBE# lines during the address phase of Bus transactions. During the data phases the CBE# lines contain byte-enables for the word transfers.

Byte lane swapping is not done on PCI since all PCI compliant devices must connect to all 32 address/data bits for address decode purposes. This means that bytes will always appear in their natural byte lane, based upon byte address. In addition, PCI does not support automatic bus sizing for 8- or 16-bit transfers: the byte-enables alone are used to determine which word bytes carry meaningful data.

The maximum transfer rate of the PCI is one 32-bit word every 30 ns, or 132 MB/sec. Basic timing diagrams for PCI write and read transactions are shown in Figures 2 and 3.

FIGURE 2
PCI Write Transaction



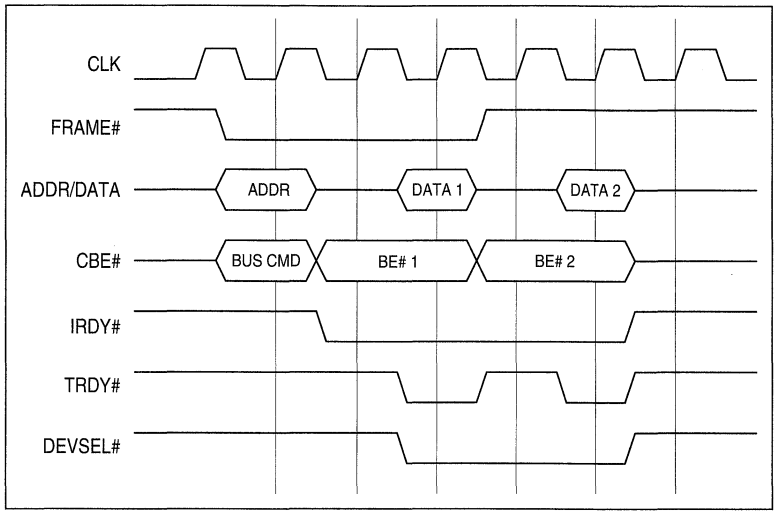


FIGURE 3
PCI Read Transaction

3.0 DESIGN OBJECTIVES

The design presented in this Application Note will accomplish the following objectives:

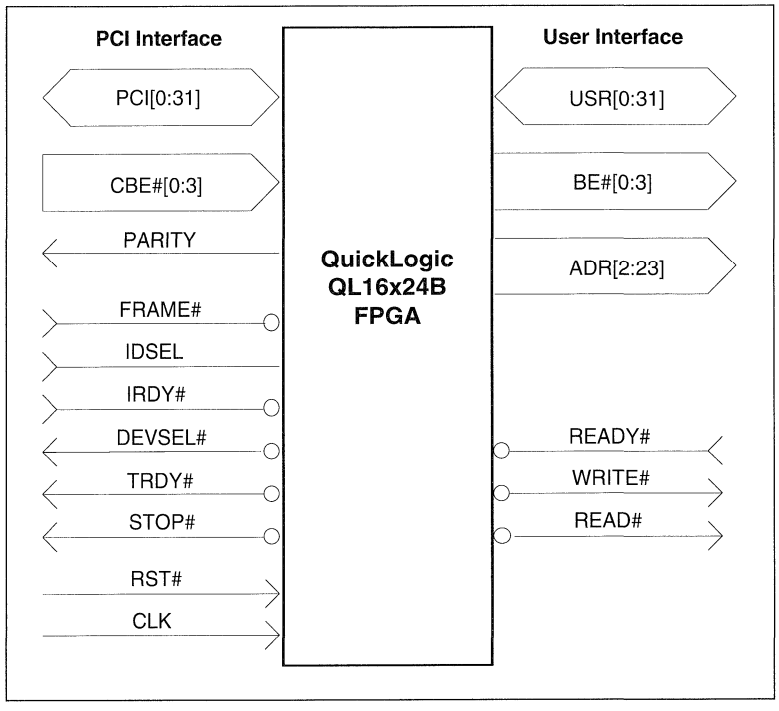
- Single FPGA device for complete PCI interface to general 32-bit user device
- Fully PCI compliant:
  - Configuration Space register file
  - Device addressing via Base Address Register
  - Data Transfer handshaking
- 16 MB Demultiplexed Address Lines for user device
- Simple READY/READ-WRITE Strobe user device handshake
- High-speed burst-mode data transfers for read and write
- Modular hierarchical design for easy modification to fit specific user needs
- Efficient use of QuickLogic FPGA architectural advantages

4.0 DESCRIPTION OF DESIGN PROBLEM

4.1 External Interface Signals

Figure 4 indicates the external signals between the FPGA and the PCI Bus and user device. The function of each signal will now be described. Note that only the PCI signals that are relevant to this application are presented.

FIGURE 4 FPGA External Interface Signals



PCI Interface Signals

PCI[0:31]	Multiplexed Address and Data Bus
CBE#[0:3]	Multiplexed Bus Command and Byte Enables
PARITY	Even parity across PCI[0:31] and CBE[0:3]
FRAME#	Indicates the beginning and duration of a data frame
IDSEL	Used as chip select during Configuration transactions
IRDY#	Indicates that the bus master is ready to transfer data
RST#	Master reset signal
CLK	Master clock signal
DEVSEL#	Driven by FPGA to indicate it has decoded its address
TRDY#	Driven by FPGA to indicate it is ready to transfer data
STOP#	Driven by FPGA to indicate it wishes to stop a (burst) transfer

The PARITY signal is bidirectional; parity generation is required by all PCI devices but its detection is not. The PARITY signal is active one cycle after its corresponding data transfer to allow pipeline parity-generation and detection to be implemented. Parity detection has been omitted in this application note for the purposes of simplicity.

The three PCI control signals driven by the FPGA (DEVSEL#, TRDY#, and STOP#) are defined in the PCI specification to be Sustained Tri-State, meaning that after being driven low (active), the FPGA must drive the signal high (inactive) for at least one clock time before letting it float.

User Interface Signals

USR[0:31]	Bidirectional user data bus
BE#[0:3]	User byte enables for write and read transactions
ADR[2:23]	User address bus
READY#	Indicates that the user device is ready to transfer data
WRITE#	User write strobe
READ#	User read strobe

Note that this design assumes a separate WRITE# and READ# strobe; a simple alternative design might use a common STROBE# signal accompanied by a R/W# indicator.

The PCI interface defines transactions for memory and I/O address spaces. In the I/O address space, all address lines are valid; in the memory address space, address bits [0:1] contain burst direction information and should not be considered valid address information. This application note describes a memory-space design and thus address signals ADR[0:1] are not needed by the user device.

The design assumes that the user device will be running synchronously to the PCI Bus Master Clock. However, connecting the user device and the FPGA to the PCI CLK signal would exceed the allowable fan-in capacity of one CMOS load. Thus, the system-level design should utilize a CMOS Phase-Locked-Loop type clock buffer to provide zero-skew copies of the PCI master clock signal to both the FPGA and the user device.

4.2 PCI Bus Commands

During the address phase of a PCI Bus transaction the CBE#[0:3] lines contain the Bus Command, indicating the type of transaction that will occur. The defined Bus Commands are indicated in Table 1:

TABLE 1
PCI Bus Commands

CBE#[0:3]	Command Type
0000	Interrupt Acknowledge
0001	Special Cycle
0010	I/O Read
0011	I/O Write
0100	Reserved
0101	Reserved
0110	Memory Read*
0111	Memory Write*
1000	Reserved
1001	Reserved
1010	Configuration Read*
1011	Configuration Write*
1100	Memory Read Multiple*
1101	Dual Address Cycle
1110	Memory Read Line*
1111	Memory Write and Invalidate*

The Bus Commands marked with an asterisk are utilized in this design. If the reader wishes to map the device into the PCI I/O address space, a schematic module may be simply modified to decode the appropriate Bus Command transactions.

4.3 PCI Configuration Space

The PCI Configuration Space is divided into a predefined header region (64 bytes) and a device dependent region (192 bytes). A PCI-compliant device is not required to implement all of the registers; all unimplemented registers, however, must return a value of zero when read. The format of the header register region is shown in Figure 5:

Device ID*		Vendor ID*		00h
Status*		Command*		04h
Class Code*			Revision ID*	08h
BIST	Header Type	Latency Timer	Cache Line Size	0Ch
Base Address Register 0*				10h
Base Address Register 1				14h
Base Address Register 2				18h
Base Address Register 3				1Ch
Base Address Register 4				20h
Base Address Register 5				24h
Reserved				28h
Reserved				2Ch
Expansion ROM Base Address				30h
Reserved				34h
Reserved				38h
Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line	3Ch

FIGURE 5 Configuration Space Header

Registers marked with an asterisk in Figure 5 have been implemented in this design. All unimplemented registers, including the 192-byte device-dependent area, will return a read value of zero.

Of the seven indicated implemented header registers, four deal with device identification, and thus are read-only. The reader will simply modify the skeleton form of these registers provided in the application note design to uniquely identify the target device. The following briefly describes the contents of these registers; more detailed information can be referenced in the *PCI Local Bus Specification*.

Vendor ID Identifies the manufacturer of the device, allocated by PCI SIG

Device ID Identifies the particular device type, allocated by the vendor

Revision ID Identifies the revision, allocated by the vendor

Class Code Identifies the generic device function. (See detailed specification.)

The remaining three registers have read-write capability and provide the following functionality:

Command	Controls the device's ability to respond to PCI cycles
Status	Records status information for PCI bus-related events
Base	Specifies the base address of the device (assigned by the O/S)

The Command register is cleared at power-on reset to logically disconnect the device from all PCI transactions except Configuration Space reads and writes. This design implements Command Register Bits 0 and 1 which control the device's response to I/O and memory space accesses. All other Command register bits have an implicit value of zero.

Bits [10:9] of the Status register indicate the speed at which the device is capable of decoding its own PCI address. The PCI specification offers codings for Fast, Medium, and Slow response: this design utilizes the Medium decode speed. (Fast speed is not obtainable in FPGA technology using synchronous design practices.)

The Base Register is also used by the operating system to determine the Memory or I/O space requirements of a device. In this design, the user device is assumed to have a 16 MB address space (24 bits); thus the upper 8 bits of the Base Register will be implemented with read-write capability and the lower 24 bits will always return a read value of zero. If the reader wishes to expand or contract the address space allocated to their device, the size of the Base Register and its associated address comparator can be adjusted accordingly.

5.0 DESCRIPTION OF DESIGN SOLUTION

5.1 Top Level of Design Hierarchy

Schematic PC132BIT represents the top level of the FPGA design, containing six primary modules. An overview of the module functionality will now be presented. Note that all external FPGA pad signals have been made accessible at the top hierarchy level for simulation purposes.

Module PCIIFACE

At the top of Sheet #1 is the PCIIFACE module which interfaces to the 32-bit PCI multiplexed address and data bus. The bus interface is implemented via bidirectional tristate pads whose outputs are enabled via signal PCI_ENABLE. Data received from the PCI bus is clocked into a register before being sent to all other areas of the design; this architecture is dictated by the tight PCI data setup specifications (7 ns).

Also included in this module is a 36-bit-wide parity generator for PCI read operations. Note that the generated parity and its associated tristate enable signal are delayed from the data bus contents by one clock cycle, as required.

The RESET signal will ensure that no PCI signals are being driven after power-on.

Module USRIFACE

In the center of the sheet is the USRIFACE module which interfaces to the 32-bit user data bus. This module simply contains the bidirectional tristate pads, enabled by the indicated USR_ENABLE signal.

Module CFGSPACE

At the bottom of the sheet is the CFGSPACE module which implements the required registers of the PCI Configuration Space. A 4-to-1 multiplexer, controlled by address bits, generates Configuration Space read data on bus CFG[0:31]. The output of this multiplexer will be clamped to zero for any other (non-implemented) register read operation. In addition, the contents of the Command and Base Registers are brought out as discrete busses for use by other modules.

Signal BVALID indicates that the base register has been loaded by the operating system software and will be used to qualify the base address comparator output. At power-on the RESET signal will ensure that BVALID is false.

The CFGSPACE is comprised of five sub-modules that implement static-value registers, a Configuration Space beam address register, and a Configuration Space Command Register.

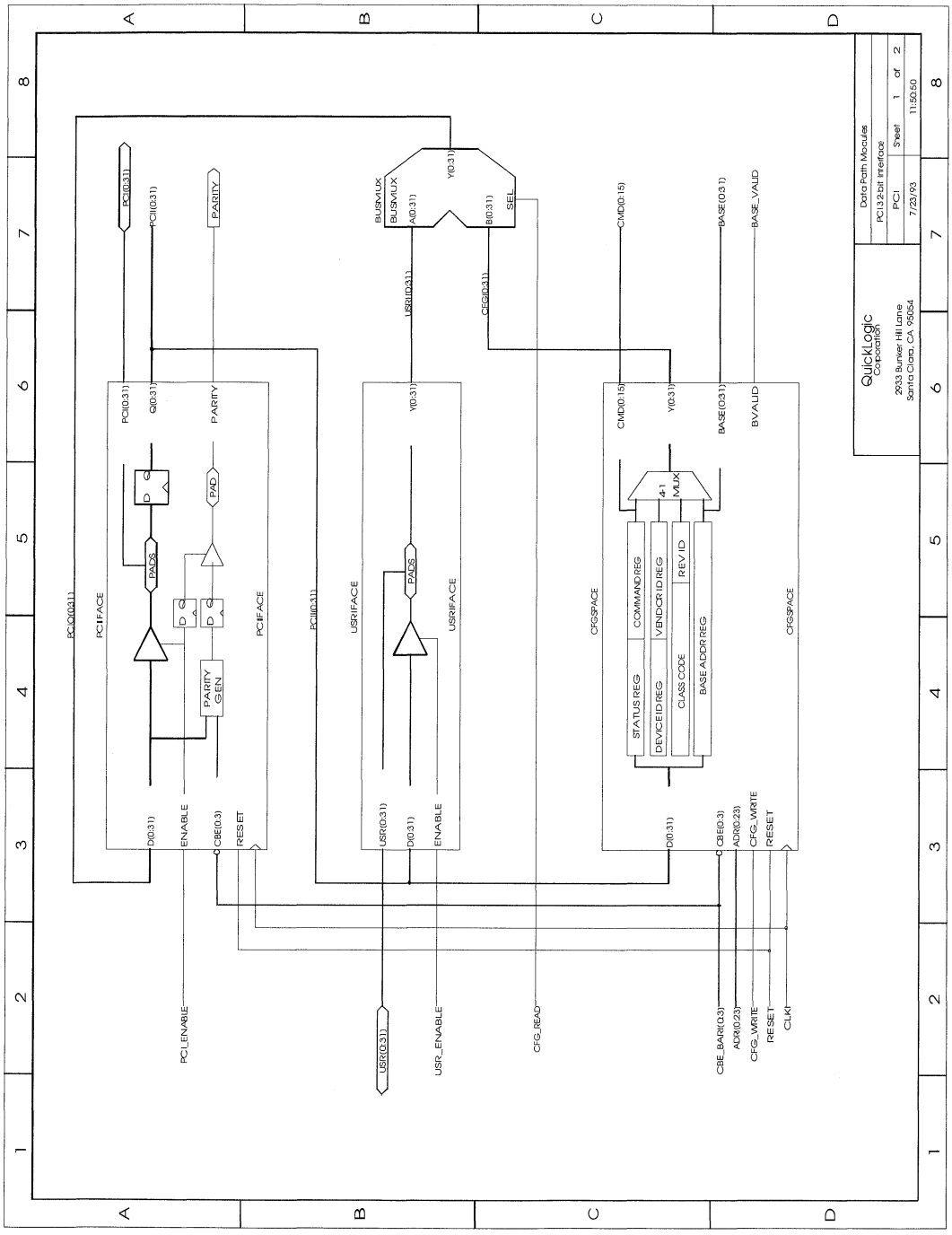
Module BUSMUX

At the right of the sheet is the BUSMUX module which selects the user read-data bus (USRI[0:31]) or the Configuration Space read-data bus (CFG[0:31]) for PCI read operations.

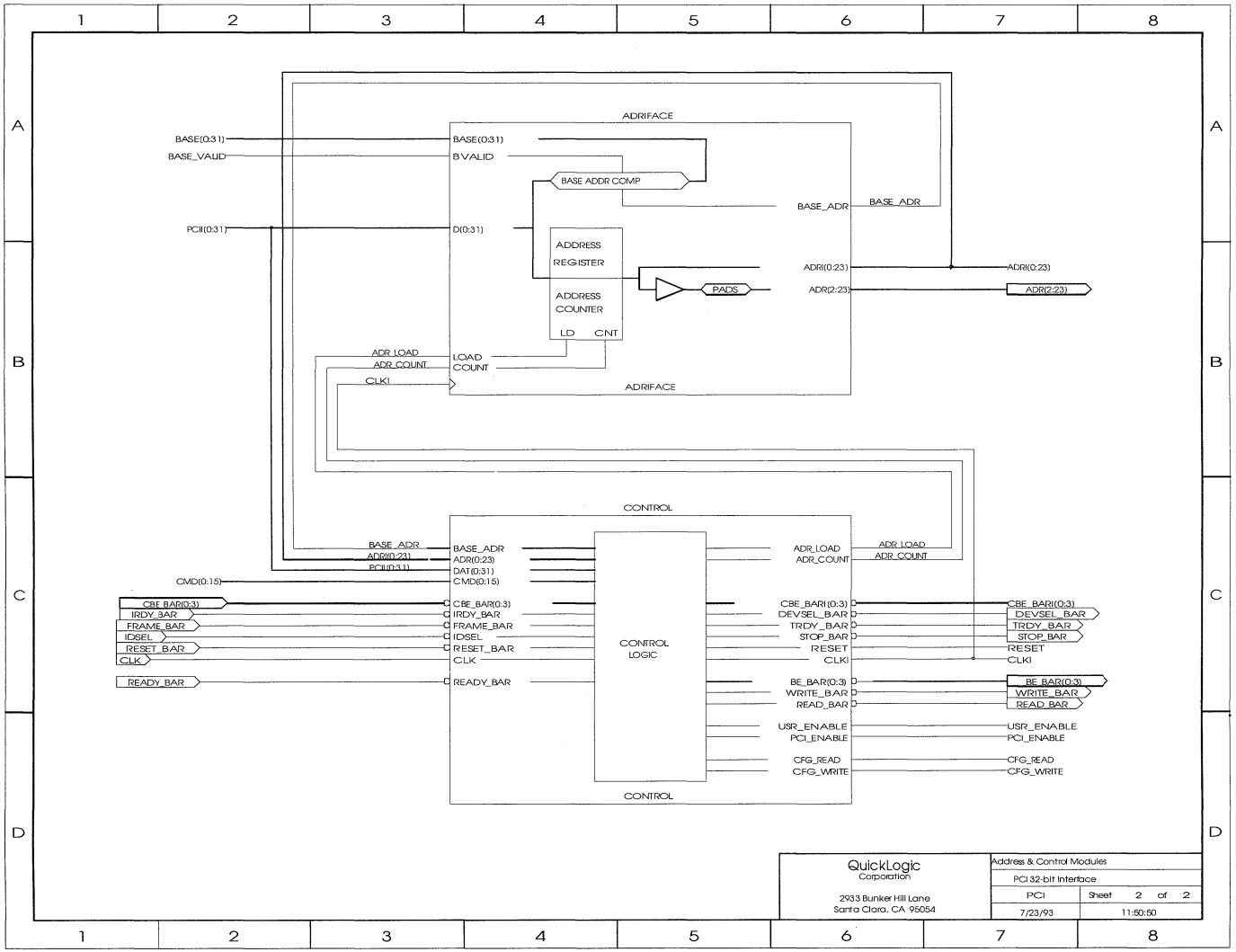
Module ADRIFACE

At the top of Sheet #2 is the ADRIFACE module which loads the address present on the PCI multiplexed address/data bus during the address phase of the bus transaction. Some bits of this register are implemented as an up-counter to provide address auto-incrementing during burst operations.

This module also contains the Base Address Comparator, which compares the contents of the Configuration Space Base Register with the PCI address. Note that the address input of the comparator is taken from the bus interface, not the address register, to allow a faster address decode. Thus, address loading and decoding can take place in the same clock cycle instead of in consecutive cycles. As mentioned previously, the address comparison is conditioned by the BVALID signal.



PCI 32-bit Interface - Data Path Modules (Sheet 1)



PCI 32-bit Interface - Address and Control Modules (Sheet 2)

Module CONTROL

At the bottom of the sheet is the CONTROL module which generates external handshake signals for the PCI and User interfaces and internal control signals for all of the hierarchical modules just described.

The CONTROL module is comprised of six sub-modules:

PCICPADS	Contains the input and tristate output pads for the PC2 control signals.
USRCPADS	Contains the I/O pads for the PC2 control signals.
CBEDECOD	Decodes the Bus Command.
PCICNTRL	Implements control logic for the PC2 interface.
CFGCNTRL	Provides control logic for the Configuration Space.
USRCNTRL	Provides control logic for one user interface.

5.2 Bus Transaction Timing Diagrams

The TIMING32 "schematic" module contains nine timing diagrams that represent the relevant PCI Bus transactions for this design. In these diagrams, signal names in **CAPS** are external signals (PCI or User) as described in Figure 4; signal names in **lower_case** are internal control signals within and/or between the FPGA functional modules. PCI signals names ending in '#' are low-true polarity.

Timing Sheet #1 Single Configuration Write

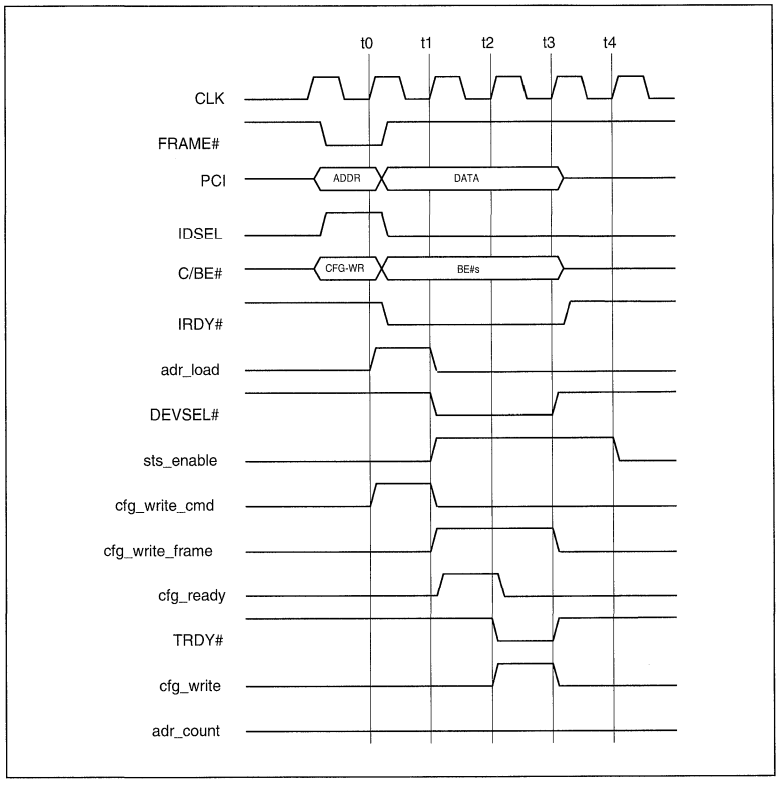
The PCI Configuration Write transaction begins at clock edge *t0* with the following PCI Interface conditions:

FRAME#	Is asserted to signal the start of a frame
PCI[0:31]	Contains to the address value
IDSEL	Is asserted to address the device for configuration
CBE#	Indicates a Configuration Write Bus Command

The CONTROL Module decodes this condition and generates signals **adr_load** to load the address value into the address register/counter and **cfg_write_cmd** to indicate that a Configuration Write Command has been received.

At clock edge *t1*, **FRAME#** is deasserted to indicate that a single (non-burst) data word will be transferred, whose value is now present on the **PCI** bus accompanied by byte-enable signals on the **CBE#** bus. **IRDY#** is also asserted to indicate that valid write-data is present. The FPGA asserts **DEVSEL#** to claim the transaction for the device and **sts_enable** to turn on the PCI control signal tristate drivers. State variable **cfg_write_frame** is set to indicate that a Configuration Write transaction is in progress.

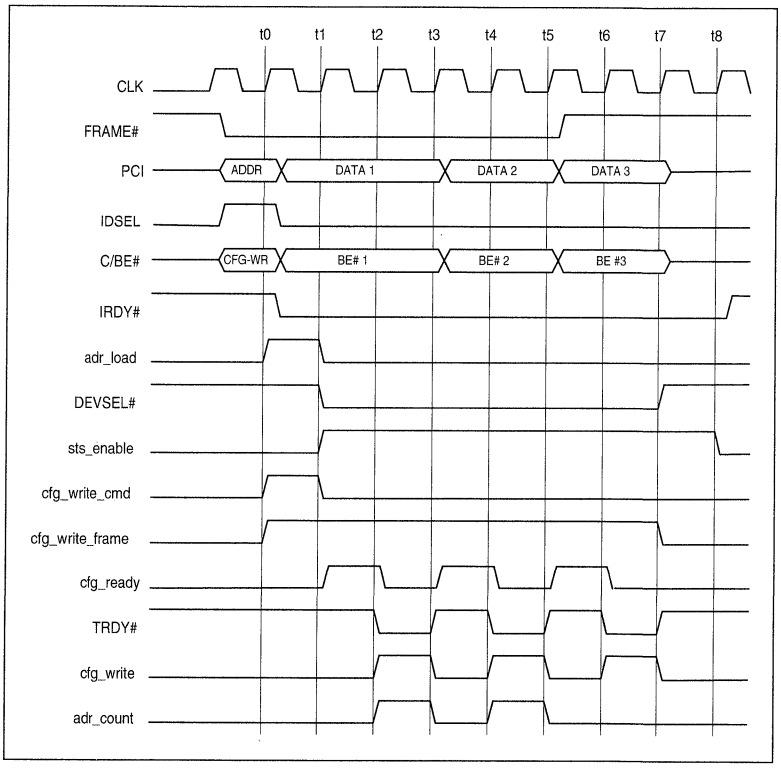
At clock edge *t2*, **TRDY#** signals that the device has accepted the write-data. Also generated is signal **cfg_write** to actually write the data into the addressed Configuration Space register. Clock edge *t3* marks the end of the transaction: signals **cfg_write_frame** and **DEVSEL#** are deasserted. Note that signal **sts_enable** remains asserted until clock edge *t4* to drive the PCI control lines to the active deasserted state before floating them, as required by the PCI specification.



Timing Sheet #2 Burst Configuration Write

This transaction begins exactly as the non-burst example just discussed; at clock edge *t1*, however, **FRAME#** remains asserted to indicate that more data will follow. This condition will cause **adr_count** to be generated at clock edge *t2* to increment the address register for the next data word.

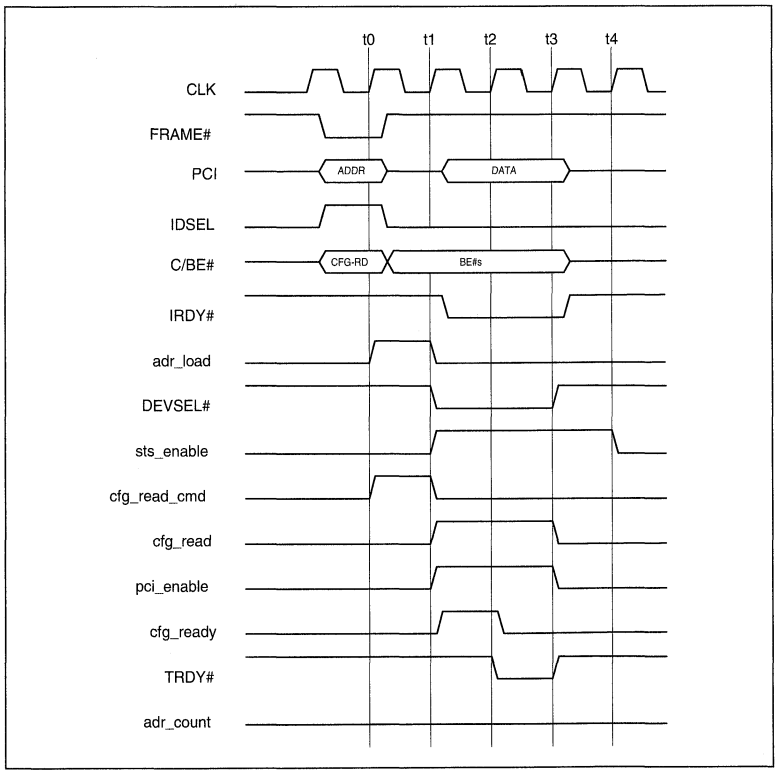
In this example **FRAME#** is deasserted at clock edge *t5* to indicate that the last data word of the (3-word) burst will follow. Clock edge *t7* marks the end of the frame, at which time **DEVSEL#** and **cfg_write_frame** are cleared. Signal **sts_enable** remains until additional clock edge *t8* as before.



Timing Sheet #3 Single Configuration Read

This transaction begins in a similar manner to the Configuration Write, with the different **CBE#** Bus Command decoded by signal **cfg_read_cmd**. Assertion of **IRDY#**, indicating that the PCI bus master is ready to receive data, is typically held off for an additional cycle for bus turnaround.

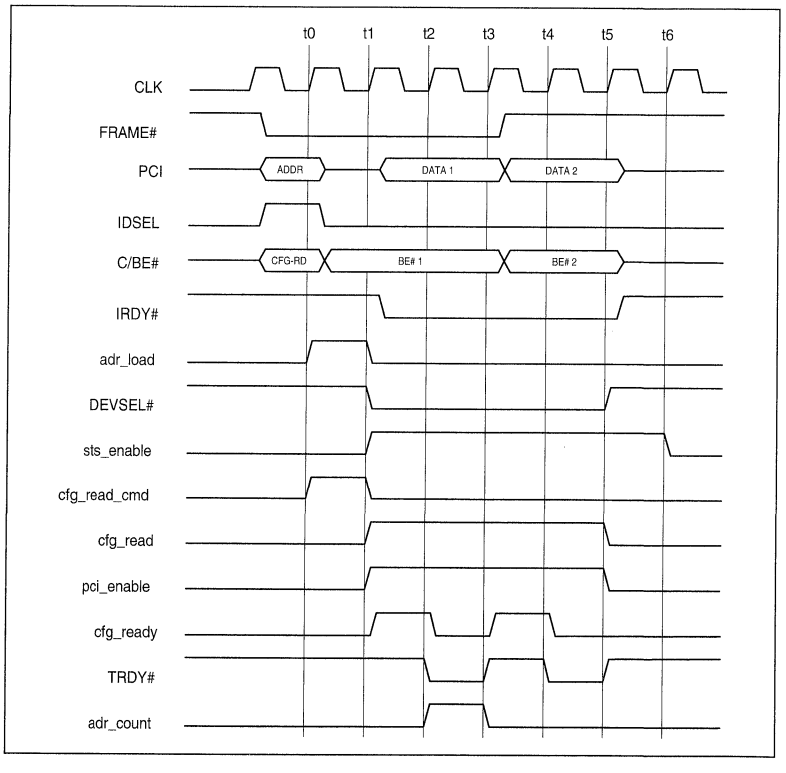
At clock edge *t1*, state variable **cfg_read** indicates that the transaction is in progress and steers the BUSMUX multiplexor to supply Configuration Space data to the **PCI** bus drivers, which are enabled by signal **pci_enable**. Address bits captured at this clock edge will begin selecting the proper Configuration Space register within the CFGSPACE module. At clock edge *t2*, **TRDY#** is generated to inform the PCI that the requested read data is available, causing the frame to end at clock edge t3 with the deassertion of **DEVSEL#**. Signal **sts_enable** performs its usual function of driving the control signals to the unasserted state for an additional clock cycle.



Timing Sheet #4 Burst Configuration Read

This transaction begins in the same manner as its single-word counterpart, with signal **FRAME#** remaining asserted past clock edge *t1* to indicate that more data is being requested. At clock edge *t2*, signal **adr_count** is generated to advance the address counter to the next word of the requested transfer.

In this example **FRAME#** is deasserted at clock edge *t3* to indicate that the last word is being requested. This leads to the final **TRDY#** signal at *t4* followed by the frame-end deassertion of **DEVSEL#**, **pci_enable**, and **cfg_read** at *t5*.
Note that the PCI bus specification allows for data to be transferred on consecutive cycles, however, the delays through the various read-data multiplexors limit the FPGA to the indicated 2-cycle burst data rate for configuration read data (at the 33 MHz PCI system clock rate).



Timing Sheet #5 Single User Write

The PCI Memory (User) Write transaction begins at clock edge t0 with the following PCI Interface conditions:

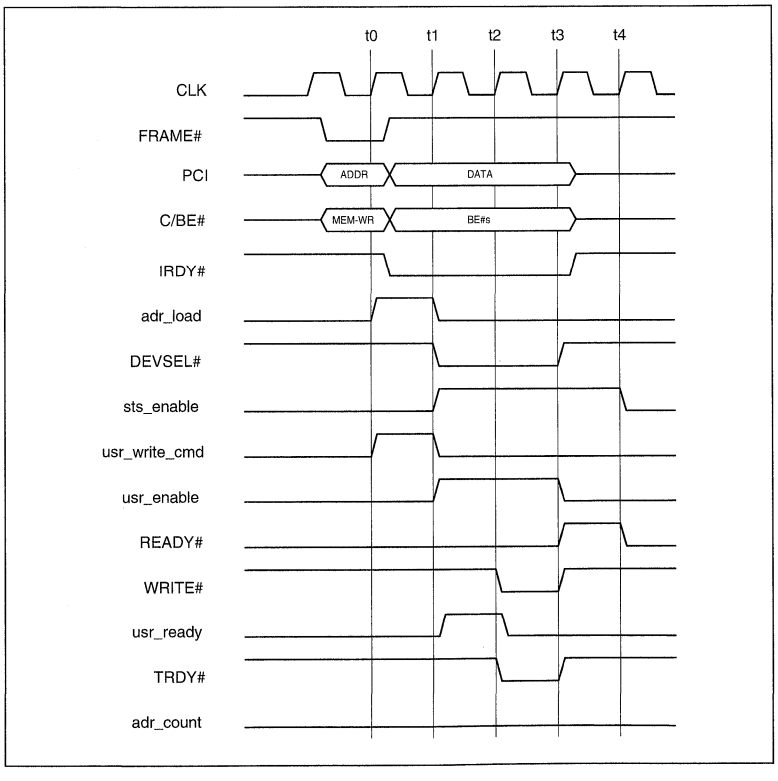
FRAME#	Is asserted to signal the start of a frame.
PCI[0:31]	Contains the address value.
CBE#	Indicates a Memory Write Bus Command.

The CONTROL Module decodes this condition and generates signals **adr_load** to load the address value into the address register/counter and **usr_write_cmd** to indicate that a Memory Write Command has been received.

At clock edge *t1*, **FRAME#** is deasserted to indicate that a single data word will be transferred; its value is now present on the **PCI** bus, accompanied by byte-enable signals on the **CBE#** bus. **IRDY#** is also asserted to indicate that valid write-data is present. The FPGA asserts **DEVSEL#** to claim the transaction for the device and **sts_enable** turns on the PCI control-signal tristate drivers. State variable **usr_enable** is set to indicate that a User Write transaction is in progress; this signal will also enable the output drivers to the **USR** data bus pads.

In this example **READY#** indicates that the user device is ready for data, causing the user **WRITE#** strobe and PCI control signal **TRDY#** to be generated at clock edge *t2*. (If this were not the case, the generation of these two signals would be delayed until an active **READY#** signal was detected).

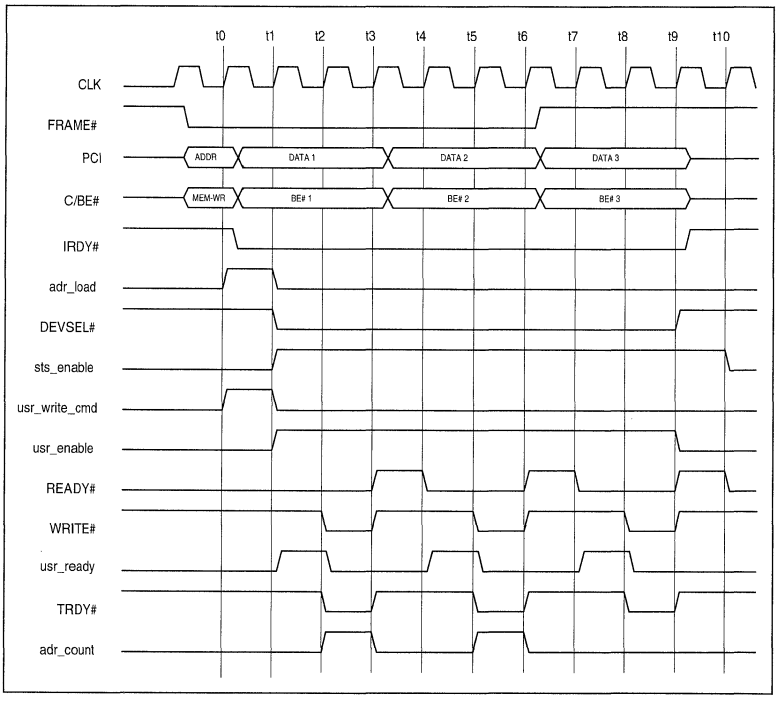
Clock edge *t3* marks the end of the frame with the FPGA's deassertion of **DEVSEL#** and **usr_enable**. The example also indicates that the user device deasserts **READY#** for one cycle as it is processing the just-received **WRITE#** strobe.



Timing Sheet #6
Burst User Write

This transaction begins in the same manner as its single-word counterpart. As in the burst Configuration Write transaction, signal **FRAME#** will remain asserted past clock edge *t1* to indicate the burst condition. At clock edge *t2*, the **WRITE#** and **TRDY#** signals are generated for the first data word, as well as **adr_count** to advance the address counter. The user device deasserts **READY#** for one cycle at clock edge *t3* to process the first data word.

The reassertion of **READY#** indicates that the user interface is ready for the next word of data and allows the second **WRITE#**, **TRDY#**, and **adr_count** strobes to be generated at clock edge *t5*. In this example **FRAME#** is then deasserted, indicating that the last word of the burst will follow.The final strobes are then generated at clock edge *t8*, once again following the indicated **READY#** state of the user device. (If the user device requires several cycles to process each **WRITE#** strobe rather than the single cycle indicated in the example, the strobes for the successive data word will be delayed accordingly.)

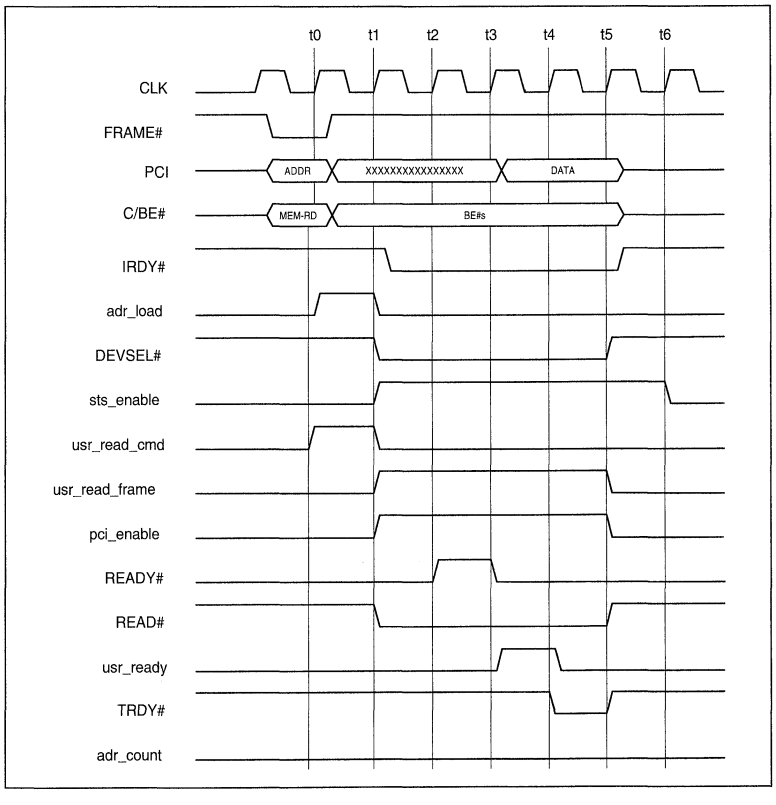


Timing Sheet #7
Single User Read

This transaction begins in a similar manner to the Single User Write, with different **CBE#** Bus Command decoded by signal **usr_read_cmd**. At clock edge *t1*, state variable **usr_read_frame** indicates the transaction in progress, while signal **pci_enable** turns on the FPGA's PCI bus drivers. The indicated **READY#** state of the user interface allows the **READ#** user read strobe to be generated at this time. (If the user device had not been ready at clock edge *t1*, the **READ#** strobe would be delayed until the ready condition existed.)

At clock edge *t2* the **READY#** signal deasserts, indicating that the user device is fetching the requested read-data. In this example, the user fetch operation has completed at clock edge *t3* as indicated by the return of an active **READY#** signal. The returned user data begins driving the **PCI** data bus at this time, as indicated.

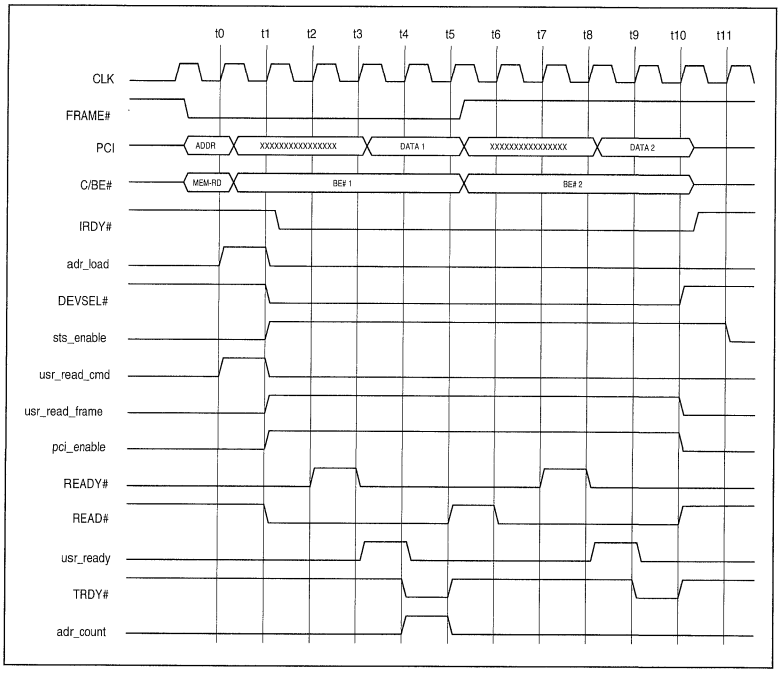
TRDY# is signalled at clock edge *t4* to inform the PCI bus that the requested memory read-data is present. The frame then ends at clock edge *t5* with the deassertion of external signals **DEVSEL#** and **READ#**, and internal signals **usr_read_frame** and **pci_enable**. As always, signal **sts_enable** remains asserted for an extra clock cycle.



Timing Sheet #8 Burst User Read

This transaction begins in the same manner as its single-word counterpart just described, with signal **FRAME#** remaining asserted past clock edge *t1* to indicate that more than one word will be requested. At clock edge *t4*, signal **adr_count** is generated to increment the address counter for the second word of the burst.

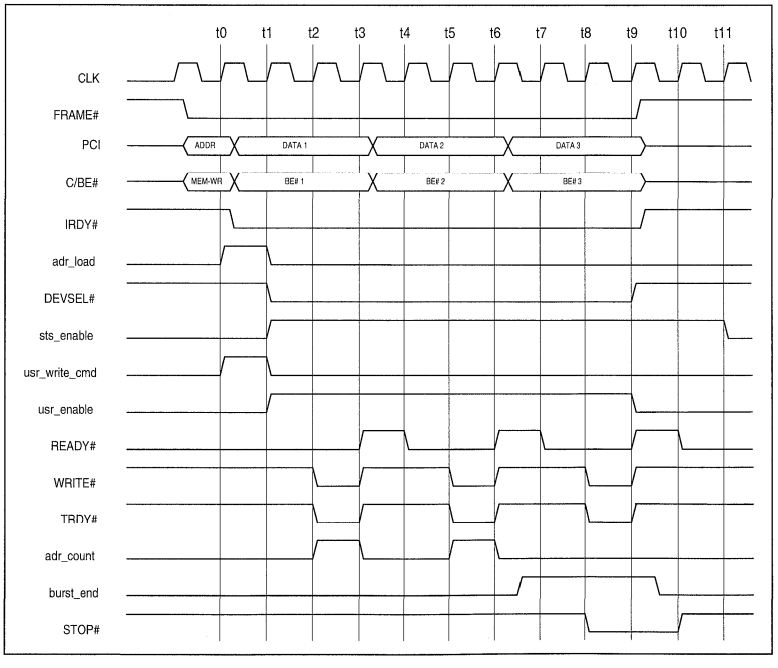
At clock edge *t5* the **READ#** user read strobe is momentarily deasserted to inform the user device that the first data word has been accepted by the FPGA (and the PCI bus). In this example, **FRAME#** is also deasserted to indicate that the last word of the burst is now being requested.
The reassertion of the **READ#** strobe at clock edge *t6* will instruct the user device to begin fetching the second read-data word, which will be processed in the same manner as the first. The frame ends at clock edge *t10* with the deassertion of all signals except the delayed **sts_enable**, as usual.



Timing Sheet #9 Burst User Write with Stop

The data transfer aspect of this transaction is identical to the Burst User Write transaction described in Timing Sheet #6. The added assertion of the PCI control signal **STOP#** is derived from the architecture of the Address Register/Counter contained ADRIFACE Module. The word-address size of this unit is 22 bits, 8 of which are contained in an up-counter; the remaining 14 bits are simply loaded into a register. If a PCI burst transfer crosses the boundary formed by the upper address of the counter, **STOP#** must be signalled to inform the PCI that the burst transfer cannot be continued at this time. The PCI will then terminate the frame and restart a new frame with the next address of the burst to be accessed. All 22 bits of this new address will then be loaded into the counter/register, effectively crossing the counter address boundary. (It should be noted that a typical PCI burst consists of about four to 32 data words. Thus the 256-word capability of this design already represents "overkill," but is possible due to the high-performance capability of the QuickLogic library counter modules.)

In the timing diagram, the assertion of **TRDY#** at clock edge *t8* is accompanied by **STOP#** because **FRAME#** indicates that more data exists in the burst and **burst_end** indicates that the address counter is at its terminal count (11111111b). The PCI will respond to the **STOP#** condition be terminating the frame as early as clock edge *t9*. Signal **STOP#** must remain asserted until after the device has verified that the frame has been terminated; signal **sts_enable** must also be suitably delayed for proper **STOP#** drive timing.



6.0 THE QUICKLOGIC ADVANTAGE

Hierarchical Design Entry

The use of hierarchical design modules, efficiently provided by the QuickLogic Engineering Capture System, yields a final design that is easily modified by the author and understood by others. Entry-time error checking within the schematic editor greatly speeds the creation of an error-free interconnect among the hierarchical modules. The Hierarchy Navigator provides a final check of this interconnection.

Design Verifier

The Design Verifier provides further analysis of the design, indicating potential fanout problems before physical layout is performed. For this application, the wide (32-bit) data paths can generate large fanout requirements for control signals, which will be readily apparent at this design stage.

Logic Optimizer

The Technology Mapper module of the Seamless pASIC Design Environment (SpDE) can greatly reduce the logic requirements of the design. In particular, the use of static registers in the Configuration Space module of the PCI interface will be efficiently reduced, particularly for those static bits with a zero value (which is a significant number). This allows the application note design to contain dummy register values which will be modified for the user's application with a minimum of logic overhead. In addition, because the efficient manner in which the logic is packed, cell utilization is reduced to 45% leaving an abundant amount of room for other logic functions.

Functional Simulation

Functional simulation can be performed before the physical Place and Route stage to yield a high level of confidence in the functional design. For this design, it is particularly useful to confirm the impact of user-device handshake speed on the PCI bus transaction rate. Efficient graphical entry of simulation stimulus waveforms speeds this process.

Place and Route

A significant advantage of the QuickLogic Architecture's ample routing resources is the flexibility of I/O pin placement. The PCI Local Bus Specification has defined a recommended device pinout to eliminate printed-circuit board feed-throughs to the PCI connector. The flexible internal routing resources have enabled this design to be 95% compatible with these recommendations, differing only by slight repositioning of a few PCI control signals, to accomodate the stringent system setup-time and hold-time parameters.

Path Analyzer

The static-timing Path Analyzer is very useful for evaluating the setup and hold time parameters of the PCI Bus Specification, as well as the propagation delays within the internal logic of the FPGA. These values, in turn, will indicate the number of "wait-states" (if any) that need to be present to accommodate PCI and user timing requirements.

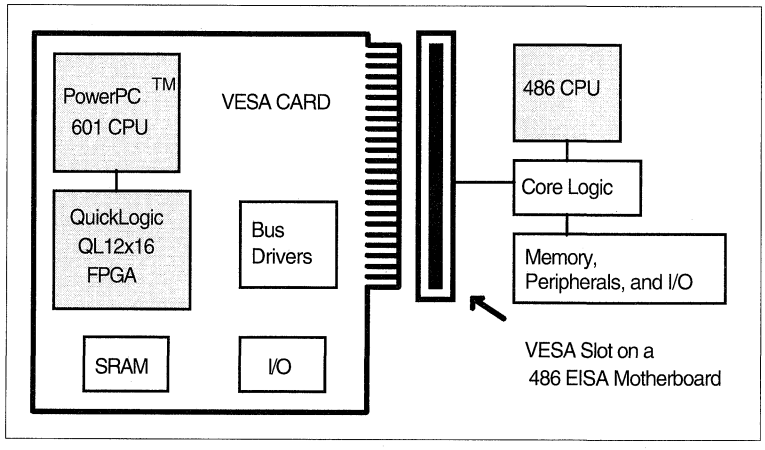
In summary, QuickLogic provides a solution that is not only quick, but that is easy, powerful, and efficient.

QAN11
PowerPC™ 601 CPU Interface to VESA Bus

HIGHLIGHTS

- QuickLogic QL12x16 device controls system interface logic connecting PowerPC™ 601 CPU to OPTI chipset that supports PC/AT standard
- Fast 33 MHz VESA bus operation
- Converts 64-bit 601 CPU data cycles into 32-bit cycles for VESA bus
- Maintains 601 cache coherency during EISA master mode DMA cycles

FIGURE 1
QuickLogic FPGA interfacing PowerPC 601 to VESA Bus



Summary

The PowerPC™ 601 Software Development Platform is a prototyping and development tool offered by Motorola to aid its OEM customers in software development for the PowerPC 601 Processor. This development platform consists of three main components: the 601 Processor Board on a VESA card, a standard 80486 EISA motherboard with a VESA slot, and the enclosure, power supply and peripherals needed to complete the computer system. The 601 Processor Board contains four primary functional blocks: the PowerPC™ 601 processor with a 66 MHz clock, the system interface

QUICKLOGIC

logic running in a QL12x16 pASIC at 66 MHz, onboard I/O devices (such as serial interfaces for debugging) and onboard memory to provide the PowerPC™ 601 CPU with 64-bit wide data storage separate from the memory available on the 486/EISA motherboard. During operation, either the PowerPC™ 601 or the 80486 CPU can serve as the permanent bus master for the motherboard chipset logic.

The PowerPC™ 601 CPU supports a little endian mode that can emulate the hardware operation of a 80486 CPU when provided with corresponding system interface logic. The approach described in this application note translates the control signals, 32-bit address bus and the 64-bit wide data bus of the PowerPC™ 601 onto a VESA bus modified for a permanent bus master. The system interface logic used in this system, the PowerPC™ 601 Software Development Platform, fits into a single QL12x16 pASIC. The system interface quickly executes complex state machines to perform the data path translation, a task Quicklogic's pASIC high-speed FPGAs uniquely handle with minimum delay. Complete design information is available from Quicklogic Applications (408-987-2100).

Design Objectives

- Place into a single QL12x16 pASIC all the system logic for bus arbitration, status encoding and control of the data path and on-board RAM.
- Interface the PowerPC™ 601 CPU with a 80486 compatible OPTI chipset using VESA and EISA interfaces to optimize speed/density tradeoffs.
- Support the standard PC/AT memory map through custom I/O extensions in the PowerPC™ 601 CPU's upper address space.

The system interface logic of the processor card makes the PowerPC™ 601 appear to the core logic (the OPTi chipset) as a 80486, which includes memory and I/O mapping as well as DMA and level 2 cache support. The guiding rule for maintaining system compatibility is, if a 80486 supports a function then a mechanism in the system interface logic must provide for an equivalent 601 cycle. From the system's point of view, no difference exists between the 601 and the 80486 processors.

Description of Design Problem

Three main problems arise when connecting the PowerPC™ 601 CPU to the VESA bus. First, because the VESA bus closely matches the 80486 CPU's 32-bit bus, the system interface must change the 64-bit data bus of the PowerPC™ to match the bus operations of the 80486. Even though the PowerPC™ 601 supports a little-endian byte mode similar to the 80486's, address alignment poses special difficulties. Second, to support a permanent bus master other then the 80486, the standard VESA bus requires customizations to add interrupt and DMA functions, which functions require support in the system interface. Finally, to simulate the I/O space of the x86

family, the system interface logic for the PowerPC™ 601 must map that I/O space into the 601's memory space without conflicting with the standard PC/AT memory map.

Although the 601 can utilize a write back cache mechanism, the system interface does not support this function due to problems maintaining cache coherency with the OPTi chipset. Instead, the system initialization code reconfigures the 601 level one cache as write-through. The 601 can also support pipelined addresses and loosely coupled address/data phases, but the system interface logic will only run a fixed length, non-overlapped address phase and varied length data phase. This non-optimized scheme simplifies the state machines in the system interface logic and eases the timing constraints imposed by the OPTi chip set.

VESA Customization

The design objective calls for either CPU to control the motherboard independently. The VESA local bus connector contains non-multiplexed lines for 32-bit address and data, and control lines supporting memory and I/O transfers. Because the REQ/GRANT signals of the VESA bus will not support alternate bus master that's permanently in control of the bus, the system logic needs to manage the HLD/HLDA signal pairs between the CPU and the OPTi chipset. In the modified VESA bus, the signals occupy unused pins on the bus connector. Note that the custom arbitration and control semaphores used to enable/disable each CPU are part of the system interface logic in the pASIC device. The design targets system initialization with the 80486, including loading of the PowerPC™ 601 boot RAMs, and then permanent master operation by the PowerPC™ 601 CPU.

On-Board Memory

The system interface logic controls the 64-bit wide on-board memory located on the 601 processor board and arbitrates 32-bit access from the VESA side with the 64-bit access from the PowerPC™ 601 CPU. The interface logic converts each 32-bit access from the 80486 (read or write) into upper and lower long word accesses. This conversion allows either processor (i.e. the 80486 or the PowerPC™ 601) to access the memory, as controlled by the signal *CBIT0* that's internal to the system interface logic. For development purposes, the 80486 can load the on-board memory with the initialization code for the 601 processor from disk, and then switch system control to the 601.

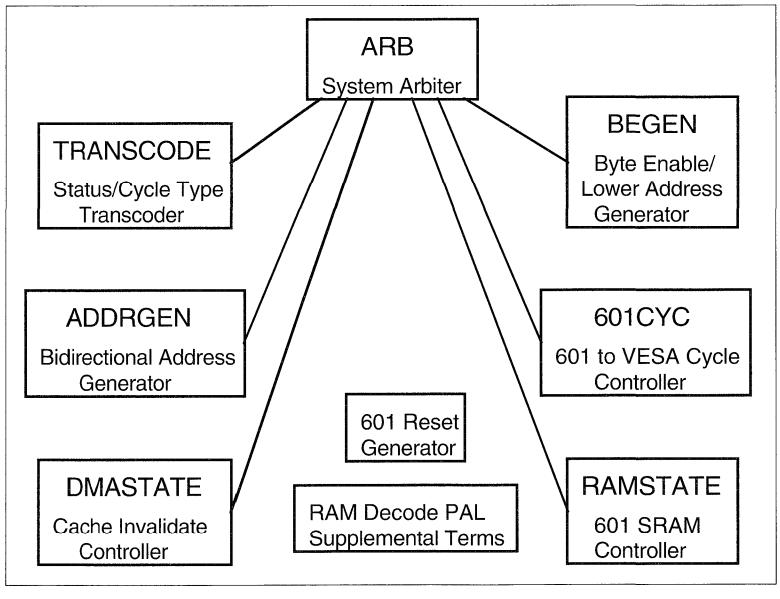
Off-Board Memory

When active, the PowerPC™ 601 processor can access the full 64 MB of system memory, level 2 cache and memory mapped video in the EISA/ISA slots. The system interface logic maps the 601 memory access to the appropriate motherboard memory address. This translation of cycle types and byte ordering is one of the major logic functions of the system interface logic. The logic also translates DMA cycles from the motherboard to the 601 processor, as needed to maintain cache coherency, by driving the addresses and cycle types back to the 601. This guarantees cache coherency even during EISA master mode DMA cycles. The 601 processor, during certain states such as filling a cache line, may assert the signal ARTRY#, which means the current snoop cycle must restart. Thus, the system interface logic must monitor this signal during snoop cycles and rerun the cycle.

Description of Design Solution

A single QL12x16 pASIC contains virtually all of the timing, control and bus steering logic. This device is ideal for interfacing the signals from the PowerPC™ 601 CPU to the VESA bus because of its low internal propagation delays. The system interface logic in the pASIC generates the data bus control signals; high speed TTL bus-buffers provide the actual bus control logic. The logic internal to the pASIC device falls into nine major functional blocks as indicated in Figure 2. Detailed design files can be obtained from QuickLogic Applications (408-987-2100).

FIGURE 2
PowerPC™ 601 to VESA Interface Logic in the QL12x16



601 Reset Generator

The 601 Reset Generator creates the signals SRESET and HRESET for the PowerPC™ 601 processor. Both signals are active low. HRESET is asserted for pASIC reset or CBIT8 and is deasserted after the rising edge of LCLK when both inputs are inactive. SRESET is activated by pASIC reset, the BREAK switch or NMI. Both BREAK and NMI are deglitched and activate a 4 bit counter which guarantees a 16 cycle SRESET pulse width.

System Arbiter

The System Arbiter (ARB) regulates which CPU is in control of the system and preempts the controlling CPU for DMA or Master cycles. CBIT0 tells the arbiter which CPU to activate. The arbiter is designed to prevent preemption of a cycle in process and avoid deadlock conditions. GRANT to the PowerPC™ 601 CPU is deasserted each cycle to prevent loss of address and cycle type information.

Status/Cycle Type Transcoder

The Status/Cycle Type Transcoder (TRANSCODE) takes the PowerPC™ 601 CPU cycle type information, decodes the most significant bit of the CPU's address and generates the 80486-like status to the chipset. Memory cycles are generated for the PowerPC™ 601's reads or writes to the lower half of its address space. 601TC0 controls D/C status for memory cycles. The I/O cycles for the AT bus are generated from the PowerPC™ 601's reads or writes to 8xxxxxxx address range. Interrupt acknowledge cycles are generated for reads from the Axxxxxxx address range. The Transcoder will not generate 80486 special cycles. A latched transceiver direction control signal is also generated by the module. Direction is determined by which CPU is in control as well as that CPU's read /write status. Address only cycles are decoded for use in generation of AACK- by the 601CYC module.

601 SRAM Controller

The 601 SRAM Control (RAMSTATE) determines the read and write cycles to the SRAM local to the PowerPC™ 601's bus (601SRAM). RAMSTATE has two state machines; one for the PowerPC™ 601 cycles and another for 80486 cycles. Outputs from both state machines are combined to generate output enables and write enables to the SRAM. The PowerPC™ 601 state machine has a counter for burst cycles, logic to generate TA- for the CPU and increment addresses to the RAM. The 80486 state machine will generate read and write control strobes to the 601 SRAM in addition to enabling and generating LRDY- to the VESA bus. A latched burst signal is generated for use in other modules, along with a cycle active signal (WHOA) used to disable GRANT to the PowerPC™ 601 and inhibit arbitration in the ARB module.

601 to VESA Cycle Controller

The first state machine in the 601 to VESA Cycle Controller (601CYC) controls the translation of the PowerPC™ 601 bus cycles to the AT bus cycles. It will handle all single, multiple (nonaligned transfers and transfers larger than 32 bits) and burst transfers. The state machine performs multiple AT bus cycles when necessary to allow for bus size differences. It converts the PowerPC™ 601 burst cycles into two 80486 burst cycles and also generates BSTACK, which is combined in RAMSTATE to create TA-. The 80486 signals ADS- and BLAST- are also derived from this state machine.

The second state machine in the 601 to AT Cycle Controller generates IOR- and IOW- to the PowerPC™ 601 I/O devices (DUARTs and LED port). A three bit counter is used to extend the strobes to meet timing requirments for the DUART I/O devices.

Data bus gating and clocking signals are also generated in the 601CYC module. The data and address bus are operated in their transparent mode when the 80486 is active. Signals from RAMSTATE are read in to provide data bus gating. Data bus clocking is only required when the 601 is running and are generated only when this state machine is activated. Signals from RAMSTATE and the PowerPC™ 601 I/O state machine are ORed together to generate AACK- to the PowerPC™ 601 CPU. The module also creates ADRLATOUT to latch the PowerPC™ 601 address (both internally and externally) for the required AT bus cycles.

Bidirectional Address Generator

The Bidirectional Address Generator (ADDRGEN) contains a loadable adder/accumulator. The logic generates A2 through A4 by loading the accumulator with either the 80486 or the PowerPC™ 601 address and recirculating this address through an adder. One aspect of the PowerPC™ 601 little endian mode that deserves special explanation is on how the CPU handles aligned multibyte transfers. During these transfers, the CPU places big-endian addresses on it's bus, which the control logic needs to convert to little-endian addresses. To perform this function, the control logic uses a recirculating adder. This adder can increment by 1, add 2 or add 4 to generate the address for multiple and burst cycles. For 8-byte transfers, the address signal A29 is non-inverted. The module also creates the multiple cycle decode signal DOUBLE for use by the 601CYC logic.

Byte Enable/Lower Address Generator

The Byte Enable/Lower Address Generator (BEGEN) creates the byte enables for the VESA bus using the PowerPC™ 601 address and size information. Byte enables for both cycles of a two cycle transfer are decoded and latched. Depending upon whether the transfer is aligned, a mux then selects the first cycle set or the second cycle set. Byte enables from the 80486 are encoded into A31, A30 and size information (TSIZE[0:2]) for external generation of SRAM write enables. There is a little endian to big endian conversion implemented in the encoding mechanism.

Cache Invalidate Controller

The Cache Invalidate Controller (DMASTATE) generates TS- and AACK- for the PowerPC™ 601 cache invalidation during the AT bus DMA and Master memory write cycles. A mechanism has been implemented to re-try the invalidation cycle when requested by the PowerPC™ 601 CPU.

RAM Decode PAL Supplemental Terms

Because additional terms were needed in the write enable generation PAL, the logic block Supplemental Terms for RAM Decode PAL was added. These terms are generated in the pASIC to supplement those in the PAL. NBURST is active for 8-byte transfers and HELP is used to expand the terms for RWE1- (generated in the external PAL).

Author Information

Russ Lindgren, consultant and author, has worked in the area of CAE tools and electronic systems design for the past twelve years. Russ is the principal of Program Systems and Associate Editor for Personal Engineering and Instrumentation News (PE & IN). He can be reached through PE & IN at 603-427-1377 or via Internet at 71551.2062@compuserve.com.

Jeff Owens, designer and a contributing author, can be reached at Up To Date Technologies.

QAN12
Designing With Abel & Palasm

Abel & Palasm With QuickLogic's SpDE

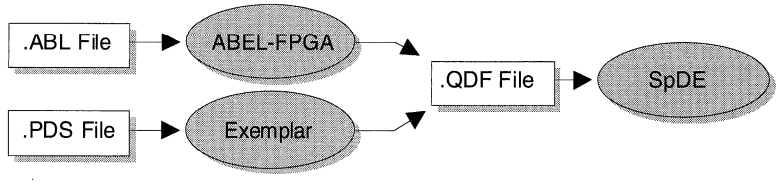
There are two general design methods of using the Abel and Palasm hardware descriptive languages (HDL) with SpDE. The two methods differ in how a complete pASIC design is defined - in a single HDL file or in multiple HDL files. You can generate an entire pASIC design in a single file or use multiple files with a top level schematic. However, whenever using an HDL you ***must*** enter through a third party tool.

Single Module Design

The basic design flow is as follows:

1. Create single HDL file for complete design.
2. Use the respective third party tools (Abel-FPGA, Exemplar)
   a) Compile
   b) Optimize
   c) Fit (generate QDF file)
3. Import QDF file into SpDE and QuickLogic tools.

FIGURE 1
Single Module pASIC Design



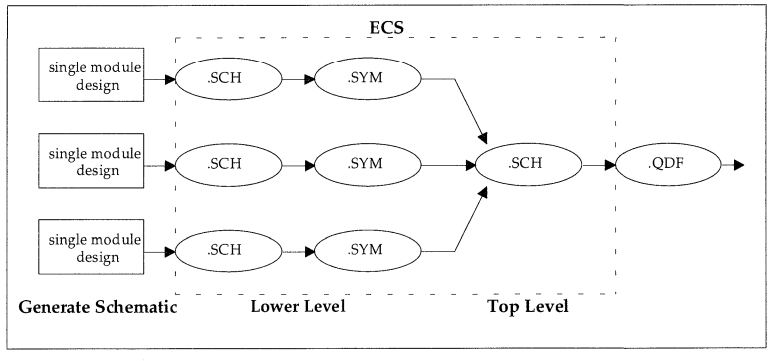
QUICKLOGIC

Multi-Module Design

For designs that contain multiple input files:

1. Follow single module flow for each HDL file.
2. Generate schematic for each module.
3. Create a symbol for each schematic.
4. Create a hierarchical design with a single top level schematic.
   a) Insert newly created symbols
   b) Add any other symbols from library
   c) Add various interconnects and pads
   d) Generate hierarchy
5. Generate an overall single QDF file (export QDF).
6. Enter SpDE as if it were a single module design (import QDF).

FIGURE 2
Multi-Module pASIC Design



Conclusion

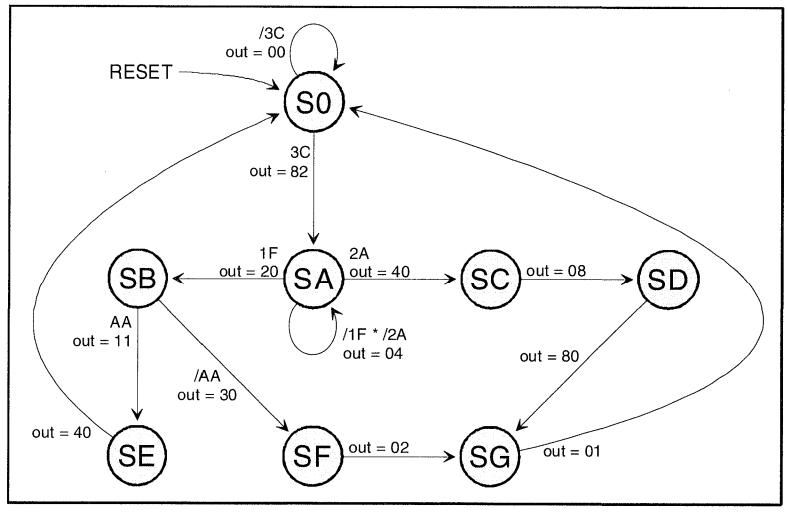
Thus, whether using a single design file or multiple files, the objective is to obtain a single QDF file that defines the entire design. Once a single QDF is generated and imported into SpDE, the design flow is as with any QuickLogic Design. The design would need to be placed and routed and any analysis that is desired can be performed.

QAN13
Implementing a One-Hot State Machine using QuickBoolean

INTRODUCTION

State Machines form the heart of many applications—DRAM and DMA controllers, cache controllers, bus arbitrators, for example. Numerous techniques are employed to implement state machines; two of the most popular are encoded state and one-hot encoding. The one-hot encoding technique (also known as bit-per-state encoding) was popularized by gate array and FPGA devices over the past 10 years. This technique employs one flip-flop for each state. In a manner of thinking, one-hot directly implements the state transition diagram by substituting a flip-flop for each state bubble.

FIGURE 1
State Transition Diagram



While schematic capture is well suited to the one-hot encoding technique, boolean entry is an alternative that can save time and streamline future modifications. The QuickBoolean tool allows blocks of boolean-described logic to be mixed with blocks of schematic-described logic. All of the logic in a design can be described as boolean, or one or more blocks of boolean can be connected with one or more blocks of schematic. The user is free to choose the best entry technique for each block in the design.

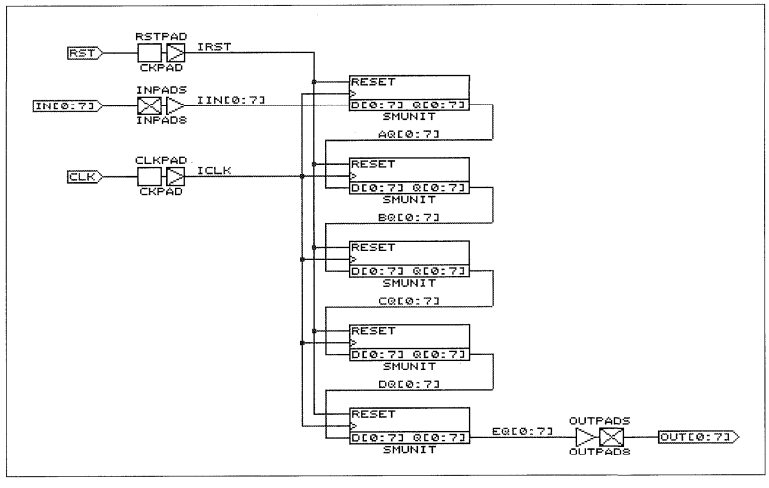
QUICKLOGIC

Design Example

PREP® Benchmark Number Three describes a typical Mealy state machine, and serves as an illustrative example of implementing one-hot state machines using QuickBoolean. Figure 1 is the state transition diagram for this state machine. The state machine has eight states and 12 transitions. Transitions are determined by an eight bit input; where applied, these values are shown along the transition arcs. The 8-bit output is registered; these values are given along the transition arcs as *"out = x"*.

The PREP benchmarking methodology allows the state machine to be implemented as any black box adhering to the specifications of the state machine. These state machines are replicated until the device is full. The state machines are connected in a daisy-chain arrangement, as shown in Figure 2.

FIGURE 2
State Machines Connected to Fill Device



The schematic shown in Figure 2 includes input and output pads selected from the pASIC Macro Library. The state machine block **SMUNIT** is a symbol that can be created with the automatic Create Symbol feature or the Symbol Editor supplied with the pASIC Toolkit. One of QuickBoolean's goals was ease of use—streamline the process so that adding a block of boolean is just as easy as adding a block of schematic. For example, the symbol **SMUNIT** defines the inputs and outputs of the state machine; no further definition of inputs or outputs or "pins" is required.

Boolean Entry

QuickBoolean is based on the popular PALASM® syntax (historical observers will note that PALASM was invented by one of QuickLogic's founders). The syntax has been extended a bit, for example, to handle multiple clocks and multiple resets. The boolean definition of the state machine is entered in the **SMUNIT.QEQ** file using any handy text editor. The first section of the file defines the global clock and reset.

```
.Clkf clk
.Rstf reset
```

The keywords **.clkf** and **.rstf** define the global clock and reset for this block of boolean (the keyword **.setf** defines the default set); **clk** and **reset** correspond to the names of the input pins on the **SMUNIT** symbol. These two lines define the default connections for all flip-flops in this block.

The next section supplies the equations for decoding the eight input bits **D[0:7]**.

```
d1f =  d[0] * d[1] * d[2] * d[3] * d[4] * /d[5] * /d[6] * /d[7]
d2a = /d[0] * d[1] * /d[2] * d[3] * /d[4] * d[5] * /d[6] * /d[7]
d3c = /d[0] * /d[1] * d[2] * d[3] * d[4] * d[5] * /d[6] * /d[7]
daa = /d[0] * d[1] * /d[2] * d[3] * /d[4] * d[5] * /d[6] * d[7]
```

There are four values of interest used by the state machine. These four decoded terms, as well as the eight registered state bits defined next, are internal nodes. They do not appear on the **SMUNIT** symbol as outputs. QuickBoolean does not require a declaration of any kind for internal nodes.

The next section of the file supplies the equations for the eight states.

```
s0 := s0 * /d3c + se + sg
sa := s0 * d3c + sa * /d1f * /d2a
sb := sa * d1f
sc := sa * d2a
sd := sc
se := sb * daa
sf  := sb * /daa
sg := sd + sf
```

Each product term defines one transition into the state. The product terms are summed for all possible transitions into the state. Referring back to Figure 1, there are two transition arcs into state **SA**; the equation above has a corresponding two product terms. As demonstrated by this compact set of equations, one-hot encoding results in a boolean description that looks a lot like a higher-level behavioral description.

The next section defines the reset handling for the **S0** state.

```
s0.rstf = gnd
s0.setf = reset
```

The first line overrides the global default defined above, and the second line insures that the **S0** flip-flop is asserted at reset.

The final section of the file supplies the equations for the registered output bits **Q[0:7]**.

```
q[0] := sb * daa + sg
q[1] := s0 * d3c + sf
q[2] := sa * /d1f * /d2a
q[3] := sc
q[4] := sb
q[5] := sa * d1f + sb * /daa
q[6] := sa * d2a + se
q[7] := s0 * d3c + sd
```

Arriving at these equations does require a bit of old-style boolean arithmetic, because QuickBoolean does not include a global boolean optimizer.

The entire **SMUNIT.QEQ** definition is a total of 24 lines. The pASIC Toolkit includes the Hierarchy Navigator, a tool that allows the user to "push" and "pop" through the design hierarchy. When the user clicks on a schematic block, the underlying schematic is displayed. When the user clicks on a boolean block, the underlying boolean is displayed.

Boolean Processing

In keeping with the ease of use goal, QuickBoolean does not require a separate "compilation" step for processing of the boolean input. The boolean input is processed automatically when the design netlist is created with the **Export QDIF** command.

As mentioned earlier, QuickBoolean does not include a global boolean optimizer. Commercial tools available from third-party vendors (for example, ABEL from Data I/O and CORE from Exemplar Logic) utilize sophisticated multi-level optimization techniques. QuickBoolean directly translates the boolean input into small AND and OR gates, which are then mapped into pASIC logic cells. In this regard, QuickBoolean is somewhat of a WYSIWYG boolean tool—compact equations will result in compact implementations, while large equations will likely result in large implementations.

The pASIC Toolkit includes a powerful Technology Mapper, which is essential to produce good results with QuickBoolean. The Technology Mapper is selected by clicking on SpDE's **Tools** menu and selecting **Options General**. In the **Logic Optimizer** section, select **Level 1 - Technology Map** before running the tools.

Results

The five implementations of the state machine utilize 85 logic cells. This is actually better than the finely tuned schematic design submitted by QuickLogic to PREP in early 1993, which requires 90 logic cells. Both implementations have three levels of logic cells in the critical path. The schematic design, however, does feature slightly lower fanouts and results in a 15% performance advantage over the boolean design.

QuickBoolean is not intended to replace schematic capture. Many circuits are best described with a schematic and a selection of macros from the pASIC Macro Library. Other circuits are best described with boolean. The pASIC Toolkit with QuickBoolean allows the user to mix and match boolean where it is preferred and schematic where it is preferred.

QAN14
Analyzing & Optimizing Performance using QuickBoolean

INTRODUCTION

QuickBoolean is a boolean entry tool tightly integrated with QuickLogic's Seamless pASIC Design Environment (SpDE). QuickBoolean allows blocks of boolean to be mixed with blocks of schematics quickly and easily. This tool is included as part of the pASIC Toolkit.

Like all textual entry methods, QuickBoolean has the advantage of abstraction—the logic description is separate and independent from the logic implementation. However, this abstraction also serves to complicate the matter of analyzing the performance of the design. In a design entered entirely as schematic blocks, factors such as fanouts and levels of logic are graphically apparent to the designer (SpDE's Path Analyzer utilizes cross-probing to graphically highlight the fanouts and levels of logic along a critical path). In a design entered entirely or partially as boolean blocks these factors are no longer apparent; and optimization (such as SpDE's Technology Mapper) further complicates the analysis process.

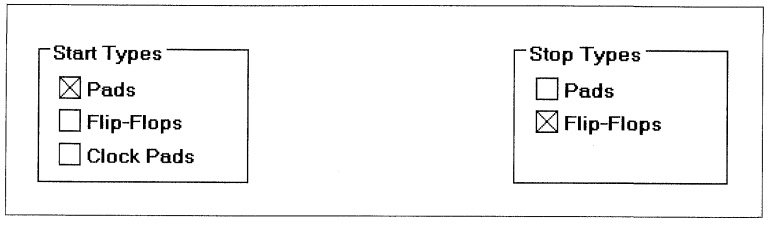
This note describes the use of SpDE's Path Analyzer tool in analyzing the performance of designs using QuickBoolean. Methods and techniques are also presented for optimizing performance in QuickBoolean designs. These optimization techniques are in no way exhaustive; knowledge, creativity, intuition, and experimentation are widely encouraged. The path delays mentioned in this note are based on QuickLogic's 1.0 micron pASIC devices. The latest .65 micron devices offer significant speed improvements (20%-30% faster).

Measuring Operating Parameters

The **Design Techniques** chapter of the ***pASIC Toolkit User's Guide*** discusses key calculations—clock skew, operating frequency, setup time, and hold time. This note will focus on setup time for illustrative purposes, although these analysis and optimization techniques apply equally to the other key calculations. The Path Analyzer Options dialog box facilitates the easy selection of setup time measurements, as shown in Figure 1. Setting the **Start Types** to only **Pads** and the **Stop Types** to only **Flip-Flops** will limit the Path Analyzer results to critical paths that start at input pads and end at flip-flops.

QUICKLOGIC

FIGURE 1
Path Analyzer Options Settings



As described in the aforementioned Chapter 12, the setup time is calculated as

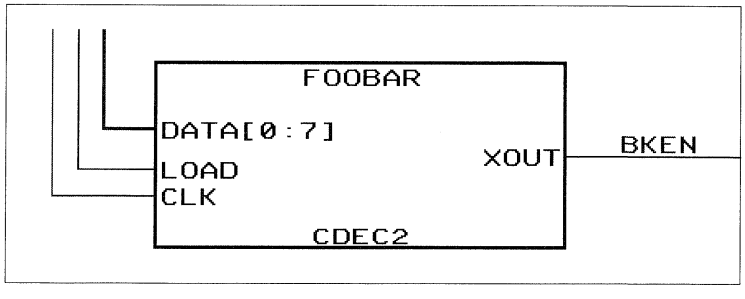
$$t_{setup} = pads\_to\_ffs - clock\_to\_ffs$$

Using the pASIC's internally buffered clock network (via the **CKPAD** macro), the *clock_to_ffs* term is fanout independent, allowing this term to be considered fixed. Optimizing the setup time, therefore, is an exercise in optimizing the *pads_to_ffs* term in this equation. With the settings shown in Figure 1, this term is displayed as the Path Analyzer results.

Signal Naming

QuickBoolean maintains signal names from boolean blocks throughout the hierarchy. If an equation is an output from the boolean block it is considered an external signal, and it will be given the name of the net attached to the output pin on the boolean symbol. In the example shown in Figure 2, the **XOUT** output from the boolean symbol is attached to a net named **BKEN**; the Path Analyzer report will use the net name **BKEN**. Furthermore, because the signal is external to the boolean block, the cross-probing function will highlight the **BKEN** net, making identification even easier.

FIGURE 2
External Signal Naming



Internal signals are given fully hierarchical net names—the hierarchical instance name of the boolean block is concatenated with the internal signal name. Consider an equation for the boolean block shown in Figure 2—

```
q[0] := load * data[0] + /load * q[0]
```

Because this signal is internal to the boolean block, with the instance name **CDEC2**, the Path Analyzer report will use the net name **CDEC2.Q[0]**. Note that automatic instance names (such as **I_42**) are given to unnamed blocks. Supplying meaningful instance names will result in a more meaningful Path Analyzer report.

Example One

The circuit in this example is a 10-bit address decoder. Consider the Path Analyzer report shown in Figure 3; the critical path is 10.6 ns. The critical path has been expanded (by double-clicking on the **1** button in the **Path #** column) to display its component trails—

- 2.10 ns propagation delay through input pad
- 5.58 ns routing & loading delay from input pad
- 2.87 ns setup time through logic cell

Path #	Delay	Delay Path	Constraint
+1+	10.6	DX[5] -- IBOF	
		2.10R DX[5] -- X[5]	
		5.58R X[5] -- DEC3.OUT0F-OZ_	
		2.87F DEC3.OUT0F-OZ_ -- IBOF	
-2-	10.3	DX[3] -- IBF0	
-3-	10.2	DX[5] -- IBF0	
-4-	10.0	DX[1] -- IBF0	

FIGURE 3
Path Analyzer Report for Example One

Using the SpDE's Hilight Net tool, the Physical Viewer shows that the input pad (net **X[5]**) drives a fanout of 4 loads. This is a relatively low fanout, so buffering and paralleling techniques (discussed below) will not improve the 5.58 ns routing & loading delay. Furthermore, there is only one logic cell level in the critical path, so there is no opportunity to lower the logic levels by optimizing the equations.

The critical path can be improved, however, using Timing-Driven Placement (see Chapter 12 of the ***pASIC Toolkit User's Guide*** for more information). Typically, this feature is employed to meet a particular design specification, such as a 10.0 ns delay, by applying the design specification in the constraint column. For illustrative purposes, a constraint of 1.0 ns is applied, which will yield the lowest possible setup time.

FIGURE 4
Example One After Timing-Driven Placement

Path #	Delay	Delay Path	Constraint
+1+	9.5	DX[5] -- IBOF	1.0
		2.10R DX[5] -- X[5]	
		4.51R X[5] -- DEC3.OUT0F-OZ_	
		2.87F DEC3.OUT0F-OZ_ -- IBOF	
-2-	9.2	DX[3] -- IBF0	1.0
-3-	9.2	DX[5] -- IBF0	1.0
-4-	8.7	DX[3] -- IBOF	1.0

As shown in Figure 4, Timing-Driven Placement improves the setup time by 1.1 ns (the Physical Viewer shows that the flip-flops have moved closer to the input pads). The *clock_to_ffs* time in this example is 5.5 ns, so we have a setup time

$$t_{setup} = pads\_to\_ffs - clock\_to\_ffs = 9.5 - 5.5 = 4.0$$

of 4.0 ns.

Example One Summary

In circuits with only one logic cell level and low fanouts, typically the greatest improvement in critical path delay is achieved with Timing-Driven Placement.

Example Two

The circuit in this example is a state machine. Figure 5 shows an 18.8 ns critical path with its component trails—

- 2.10 ns propagation delay through input pad
- 4.76 ns routing & loading delay from input pad
- 2.87 ns propagation through first logic cell, plus routing & loading delay
- 6.15 ns propagation through second logic cell, plus routing & loading delay
- 2.87 ns setup time through third logic cell

Path #	Delay	Delay Path	Constraint
+1+	18.8	IN[3] -- QB1.SC	
		2.10R IN[3] -- IIN[3]	
		4.76R IIN[3] -- QB1.D2A-INT0	
		2.87F QB1.D2A-INT0 -- QB1.D2A	
		6.15F QB1.D2A -- QB1.SC-OZ_	
		2.87F QB1.SC-OZ_ -- QB1.SC	
-2-	18.6	IN[3] -- AQ[6]	

FIGURE 5
Path Analyzer Report for Example Two

Inspection with the Physical Viewer shows that the input pad (net **IIN[3]**) drives 4 loads, the first logic cell (net **QB1.D2A-INT0**) drives 1 load, and the second logic cell (net **QB1.D2A**) drives 4 loads. As was the case in Example One, the fanouts are relatively low. Buffering and paralleling techniques will probably not yield substantial improvement (as noted below, however, these techniques are circuit dependent; a given design with similar critical path measurements may result in genuine improvement).

Unlike Example One, the critical path of 18.8 ns includes three logic cell levels. The largest potential improvement involves optimizing the equations to remove a logic cell level in the critical path. Even if the modified critical path includes higher fanouts, the removal of a logic cell level will usually improve the critical path delay.

Path #	Delay	Delay Path	Constraint
+1+	15.9	IN[4] -- QB1.SE	
		2.55F IN[4] -- IIN[4]	
		4.71F IIN[4] -- QB1.DAA	
		5.76F QB1.DAA -- QB1.SE-OZ_	
		2.87F QB1.SE-OZ_ -- QB1.SE	
-2-	15.8	IN[4] -- AQ[7]	

FIGURE 6
Example Two After Equation Optimization

Optimizing the equations removes one logic cell level, at the expense of using one additional logic cell. The input pad (net **IIN[4]**) drives 4 loads and the first logic cell (net **QB1.DAA**) drives 4 loads. The result is a critical path delay of 15.9 ns and a setup time

$$t_{setup} = pads\_to\_ffs - clock\_to\_ffs = 15.9 - 5.5 = 10.4$$

of 10.4 ns.

Example Two Summary

In circuits with two or more levels of logic and low fanouts, typically the greatest improvemetn in critical path delay is achieved by optimizing the equations to remove logic cell levels. Timing-Driven Placement may be used for additional improvement.

Example Three

The circuit in this example is a more complex state machine, with four logic cell levels in the critical path. Figure 7 shows an 34.4 ns critical path with its component trails—

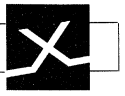
- 2.10 ns propagation delay through input pad
- 20.42ns routing & loading delay from input pad
- 2.68 ns propagation through first logic cell, plus routing & loading delay
- 3.10 ns propagation through second logic cell, plus routing & loading delay
- 3.24 ns propagation through third logic cell, plus routing & loading delay
- 2.87 ns setup time through fourth logic cell

FIGURE 7
Path Analyzer Report for Example Three

Path #	Delay	Delay Path	Constraint
+1+	34.4	IN[0] -- O[19]	
		2.10R IN[0] -- I[0]	
		20.42R I[0] -- I_2.QQ3.QD-INT7	
		2.68F I_2.QQ3.QD-INT7 -- I_2.QQ	
		3.10R I_2.QQ3.QD-INT4 -- I_2.QQ	
		3.24F I_2.QQ3.QD-INT0-FZ_ -- I_2	
		2.87F I_2.QQ3-OZ_ -- O[19]	
-2-	32.8	IN[0] -- O[20]	

Inspection reveals that the input pad (net **I[0]**) drives 30 loads. The fanouts from the logic cells are very low, and the equations have already been optimized to reduce the levels of logic cells to a minimum. The high fanout of 30 loads from the input pad offers the greatest opportunity for improving the critical path delay.

The most appealing option is selective buffering—add a single buffer to the **I[0]** net, drive the critical path directly from the input pad and drive the remaining 29 loads from the buffer. Selective buffering is typically the first choice in circuits with very high fanouts where only a small number of the loads are in the critical path. Such is not the case in this circuit. The top 20 critical paths include the **I[0]** net, so selective buffering is not applicable.



Another buffering option is uniform buffering—add buffers in a symmetrical tree to lower the overall fanouts. Uniform buffering has the appealing advantage of wide applicability, but the unappealing disadvantage of adding a logic cell level in the critical path. Experimentation may be used to determine the optimal number and arrangement of buffers. A good first approximation employs a number of buffers equal to the square root of the number of loads, in this case five buffers. The input pad will drive five loads, and each of the five buffers will drive six loads.

Uniform buffering requires minor modifications in the boolean file to add the buffer tree. The following lines are added at the top of the boolean file.

```
IA[0] = I[0]
IB[0] = I[0]
IC[0] = I[0]
ID[0] = I[0]
IE[0] = I[0]
```

All boolean equations using **I[0]** must be modified to replace **I[0]** with one of the buffered versions. Each of the five buffers (**IA[0]** through **IE[0]**) will be used six times.

Figure 8 shows the result of these modifications, a reduction in the critical path delay from 34.4 ns to 26.8 ns. The added buffers require less than three additional logic cells.

Path #	Delay	Delay Path	Constraint
+1+	26.8	IN[0] -- O[20]	
		2.10R IN[0] -- I[0]	
		3.49R I[0] -- I_2.KADD5	
		3.87R I_2.KADD5 -- I_2.QQ2.QD-I	
		4.56R I_2.QQ2.QD-INT11 -- I_2.Q(	
		5.22R I_2.QQ2.QD-INT2 -- I_2.QQ	
		4.68R I_2.QQ2.QD-INT0 -- I_2.QQ	
		2.87F I_2.QQ2-OZ_ -- O[20]	
-2-	26.3	IN[6] -- O[19]	

FIGURE 8
Example Three After Uniform Buffering

The routing & loading delay from the input pad has been reduced from 20.42 ns to 3.49 ns, but the buffer has added an additional logic cell level with a 3.87 ns delay.

A third option is paralleling—duplicate the driving logic to create multiple "spigots" without adding an additional level of logic cell delay for buffering. Paralleling inside a chip requires the duplication of the driving gate or flip-flop, which increases the fan-in to the driving logic. Such is not the case in this circuit, as the high-fanout net in question is driven from an input pad.

The equations modified to use **IA[0]** through **IE[0]** are left unchanged, but the five lines shown earlier for the buffer tree are removed. The symbol for the boolean block is modified to replace input pin **I[0]** with five input pins **IA[0]** through **IE[0]**, and these pins are wired to five individual input pads.

FIGURE 9
Example Three After Paralleling

Path #	Delay	Delay Path	Constraint
+1+	23.0	IN[10] -- O[20]	
		2.10R IN[10] -- I[10]	
		3.64R I[10] -- I_2.QQ2.QD-INT11	
		4.54R I_2.QQ2.QD-INT11 -- I_2.Q(	
		4.87R I_2.QQ2.QD-INT2 -- I_2.QQ	
		4.94R I_2.QQ2.QD-INT0 -- I_2.QQ	
		2.87F I_2.QQ2-OZ_ -- O[20]	
-2-	20.7	IN[9] -- O[19]	

Figure 9 shows the results of these modifications, a reduction in the critical path delay from 34.4 ns to 23.0 ns (Timing-Driven Placement will yield additional improvement, as each of the five input pads can be placed very close to their respective logic cells). Although no additional logic cells are used (compared to the original implementation), four extra input pads are required. Lower degrees of paralleling have the trade off of using fewer input pads versus the increase in the critical path delay.

A fourth option is the use of high-drive input pads. In this circuit, the high-fanout net is driven from an input pad. Simply replacing the **INPAD** macro with an **HD2PAD** macro requires no modifications to the boolean file.

Figure 10 shows the result of this simple substitution, a reduction in the critical path delay from 34.4 ns to 20.5 ns, and a setup time

$$t_{setup} = pads\_to\_ffs - clock\_to\_ffs = 20.5 - 5.5 = 15.0$$

of 15.0 ns.

Path #	Delay	Delay Path	Constraint
+1+	20.5	IN[0] -- O[19]	
		1.69R IN[0] -- I[0]	
		6.84R I[0] -- I_2.QQ3.QD-INT7	
		2.66F I_2.QQ3.QD-INT7 -- I_2.QQ	
		3.39R I_2.QQ3.QD-INT4 -- I_2.QQ	
		3.25F I_2.QQ3.QD-INT0-FZ_ -- I_2	
		2.87F I_2.QQ3-OZ_ -- O[19]	
-2-	19.9	IN[0] -- O[18]	

FIGURE 10 Example Three With High-Drive Pads

While the high-drive pad technique yields the best results in this circuit, it is a good idea to keep all of these (and other) techniques in mind when optimizing a given circuit.

Example Three Summary

In circuits with high fanouts, typically the greatest improvement in critical path delay is achieved by reducing fanouts by employing buffering, paralleling, and high-drive techniques. Timing-Driven Placement may be used for additional improvement.

SUMMARY

All of the analysis and optimization techniques employed in schematic designs can be employed in QuickBoolean designs. The advantages of rapid editing makes it easier to optimize and experiment with boolean files.

When optimizing any design, it is important to focus on the critical path. Attempting to optimize too many paths at once typically yields only minor improvements (this is the infamous ***too many variables*** phenomenon). Although the critical path in example three has four levels of logic cells in the critical path, most of the delay occurs in a single high-fanout net. Attacking this net yields the majority of possible improvement. After attacking the primary "delay culprit," further (albeit smaller) improvements can be made by continuing analysis and optimization. Timing-Driven Placement, for example, can be considered a final improvement step of tuning the placement to optimize the desired paths.

QuickBoolean is not intended to replace schematic capture. Many circuits are best described with a schematic and a selection of macros from the pASIC Macro Library. Other circuits are best described with boolean. The pASIC Toolkit with QuickBoolean allows the user to combine boolean and schematic entry where it is preferred. Combined with SpDE's Path Analyzer and Timing-Driven Placement, the pASIC Toolkit provides the user with all of tools necessary to produce mixed-entry designs optimized for performance.

pASIC 1 FAMILY
Quality Program

OVERVIEW

The pASIC product quality program has the goal to meet or exceed the industry's highest quality standards. The program includes product acceptance inspection in the Standard Process Flow (see the following Standard Process Flow diagram). Electrical, visual, mechanical, solderability, and hermeticity inspection monitors check on the adequacy of the process controls. Inspection data defects, should they occur, are used as feedback to the operating departments for analysis and process improvement.

Designed for testability, the pASIC 1 Family incorporates many special test modes to verify circuit performance and array programmabilityof the one time programmable devices. Test row and column patterns, specifically designed to detect array anomalies, are programmed at both the wafer sort and packaged product test stages. These test patterns and conditions represent the worst-case operating and programming conditions that will be encountered in the field. For example, the ViaLink element requires a 10- to 11-volt pulse to program. During wafer sort and final test, all devices are stressed near this voltage to eliminate products with potentially weak links.

The QL8x12 contains 80,000 ViaLink elements, including 1,336 test links. Half of the test links are programmed at wafer sort, the balance at final test. In a typical application, less than 4 percent of the links, approximately 3,200 are required to be programmed. Therefore, links representing one-third of the number which will be programmed at the customer site will have been checked on every device prior to shipment. A similar proportion of links are verified on all devices.

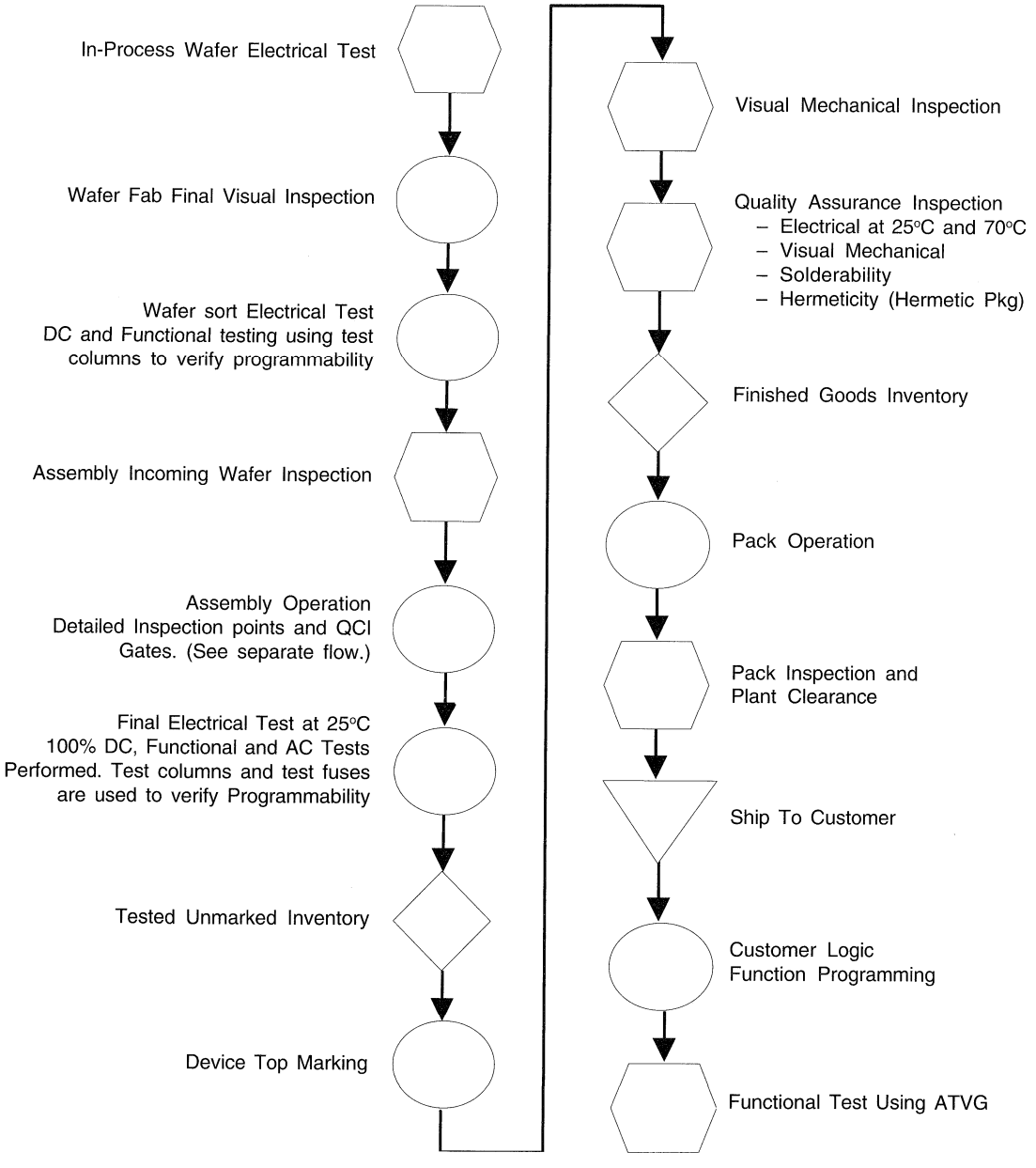
Another on-chip test feature is a silicon signature test column. The signature allows process sequence verification at each electrical test step. Devices not programmed with the proper signature for the sequence to be performed are rejected.

A serial scan path allows 100 percent functional testing of each logic cell and its interconnecting wiring channels. The channels are checked for lateral and verticle shorting under stress conditions. Dedicated input and I/O cells are fully tested to the guardband limit of data sheet D.C. specifications. A.C. performance is tested via an internal capability which chains all of the cells together. A chain of over 560 gates allows extremely accurate measurement without the traditional shortfalls encountered with high-resolution range measurements.

QUICKLOGIC

PRODUCT QUALITY ASSURANCE

Standard Process Flow



pASIC 1 FAMILY
Reliability Report

SUMMARY

The pASIC device is a highly reliable Field Programmable Gate Array. The addition of the ViaLink to a CMOS process does not measurably increase the failure rate of the pASIC devices above that of normal CMOS logic products. The following is the summary of the High Temperature Operating Life data for all the pASIC devices. The three failures were not related to the ViaLink.

Total Equivalent Device Hours at 55°C:	**210,000,000**
Total Failures:	3
Observed FITs:	14
60% Confidence FITs:	19

INTRODUCTION

The QuickLogic pASIC 1 family of very-high-speed FPGAs is built by integrating the QuickLogic ViaLink metal-to-metal antifuse programming element into a standard high-volume CMOS process. Reliability testing of pASIC devices is part of a continuous process to assure long term reliability of the product. The test consists of industry established accelerated life tests for basic CMOS devices plus additional stress tests for the ViaLink elements. The standard tests include high temperature operating life, temperature cycle, temperature-humidity-bias, pressure pot, and high temperature storage tests. The addition of two high voltage life tests stress the unprogrammed and programmed ViaLink elements beyond conventional CMOS reliability testing.

Results to date, from the evaluation of over 7000 pASIC devices from multiple wafer lots, indicate that the addition of the ViaLink element to a well-established CMOS process has no measurable effect on the reliability of the resulting product. There have been three failures, unrelated to the ViaLink, in 210 million equivalent device hours of high temperature operating life. The observed failure rate is 14 FITs, and the failure rate at a 60% confidence level is 19 FITs.

QUICKLOGIC

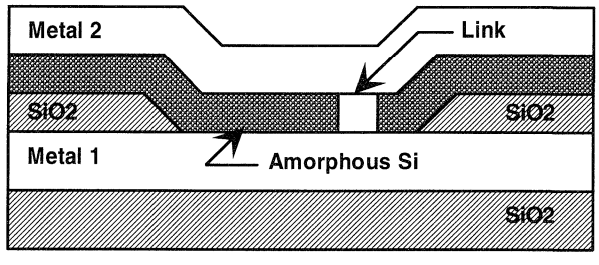
PROCESS DESCRIPTION

The 1.0 micron CMOS pASIC devices are fabricated using a standard, high volume gate array process with twin-well, single poly, and double layer metal interconnect. The base technology has been qualified to meet MIL-STD-883D. Over 1.1×10^9 equivalent device hours of operating life test have been accumulated since volume production began in 1989. The technology employs a high integrity TiW-Al+Cu-TiW metal system which offers very low contact resistance through the use of PtSi contacts, high resistance to electromigration, and freedom from stress induced opens [1]. The basic CMOS technology [2] features LDD type transistors with a gate oxide thickness of 200Å. BPSG applied over the polysilicon lines is reflowed after contact formation giving a sloped entry for metal one. The interlevel dielectric is planarized with spin-on-glass. Vias are wet/dry etched giving sloped walls for good second layer metal step coverage. Interconnect metal lines contain layers of TiW on both sides of the standard Al+Cu alloy.

QuickLogic also uses a 0.65 micron CMOS process, a high volume process with single poly and double layer metal. The base technology was developed for high speed SRAMs and Flash EPROMs. The addition of the low on-resistance and low off-capacitance ViaLink to this technology creates a very high speed programmable logic process.

The ViaLink element is located in the intermetal oxide via between the first and second layers of metal. It is created by depositing a very high resistance silicon film in a standard size metal one to metal two via. The silicon deposition is done at low temperature and causes no change to the properties of the CMOS transistors. When deposited at low temperatures silicon forms an amorphous structure which can be electrically switched from a high resistance state ($\cong$5 G-Ω) to a low resistance state ($\cong$50 Ω) for an off-to-on ratio of 10^8. QuickLogic takes advantage of this property to create the ViaLink metal-to-metal antifuse programming element. See Figure 1.

FIGURE 1 Cross Section of a ViaLink



The programming voltage of the ViaLink element varies with amorphous silicon thickness. For a desired programming voltage between 10–12 volts, the thickness of the amorphous silicon film is approximately 1000 Å. This is ideal for good process control and minimizes the capacitive coupling effect of an unprogrammed element located between the two layers of metal.

Amorphous silicon is deposited with standard semiconductor manufacturing equipment and processing techniques. In addition to antifuse elements, amorphous silicon is used in the high-volume fabrication of image sensors, decode and drive circuits for flat panel displays, and high efficiency solar cells.

FAILURE MECHANISMS IN THE pASIC DEVICE

A variety of failure mechanisms exist in CMOS integrated circuits. Since the overall failure rate is composed of various failure mechanisms, each having different temperature dependence and thus varying time-temperature relationships, it is important to understand the characteristics of each contributing failure mechanism. Table 1 lists several key failure mechanisms that have been characterized for standard CMOS devices, plus the two mechanisms for the programmed and unprogrammed ViaLink elements.

TABLE 1 Failure Mechanisms Which May be Operative in pASIC Devices

Failure Mechanism	t_{50} Dependence	Activation Energy (E_a)	Detection Tests
Short channel charge trapping (V_T and g_m shifts)	$g_m \cong \exp(-AE)$	Appox. -0.06 eV	Low temp. high voltage oper. life test
Insulator breakdown (leakage, opens)	$\exp(-\beta/E)$ value of β depends on the dielectric and may be temp. dependent.	Approx. 0.3 eV for SiO_2 and dependent on E	High voltage operating life test (HTOL)
Open metal from electrolytic corrosion	$(\%RH)^{-4.5}\exp(E_a/kT)$	0.3 to 0.6	High temp./high humidity/ bias test
Masking and assembly defects	$\exp(E_a/kT)$ (Arrhenius)	0.5 eV	High temp. storage and HTOL
Silicon defects (leakage, etc.)	Arrhenius	0.5 eV	High voltage and guard banded tests
Metal line opens from electromigration	$\frac{WT}{J^2}\exp(E_a/kT)$	Approx. 0.7 eV for A1+Cu alloys	HTOL
Stress induced open metal (operative only on non-clad metal systems)	$W^mT^p \exp(E_a/kT)$ (m and p range from 1.3 to 4.7)	0.6 to 1.4 eV (E_a difficult to reproduce)	Temp. cycling
Wire bond failure from excessive gold-aluminum interdiffusion	$1/(Dt)^{1/2}$ where $D = D_0\exp(E_a/kT)$	0.7 eV	HTOL
Parameter shifts due to contamination (such as Na)	Arrhenius	1.0 eV	High temp. bias
Plastic Chemistry of the package	Arrhenius	1.0 eV	High temp./high humidity/ bias test
Polarization in the thin film layers	Arrhenius	1.0 eV	High temp./high humidity/ bias test
Microcrack in oxides and thin films	Arrhenius	1.3 eV	HTOL/Temp Cycling
Unprogrammed ViaLink	exp(-BE)	0 eV	High V_{cc} static life test
Programmed ViaLink	exp(-PJ)	0 eV	Low Temperature Operating Life test

In Table 1, t_{50} is the mean time to failure, E is the electric field, E_a is the activation energy, k is the Boltzmann constant ($8.62X10^{-5}eV/°K$), W is the metal width, t is the metal thickness, J is current density, g_m is transconductance, V_T is the threshold voltage, A, m and p are constants, T is the absolute temperature, RH is the relative humidity, and D is the diffusion constant.

Various accelerated life tests are used to detect the possible contribution of each mechanism to the overall failure rate of the device. Failure rate data taken at elevated temperature can be translated to a lower temperature through the Arrhenius equation. This equation, in the form of an acceleration factor, A_f, can be written as,

$$A_f = \exp[-E_a/k(1/T_s - 1/T_o)] \quad (1)$$

where T_s is the stress temperature, T_o is the operating temperature of the device, E_a is the activation energy for that mechanism, and k is the Boltzmann constant.

ACCELERATED LIFE TESTS ON THE pASIC

The purpose of a life test is to predict the reliability and failure rate of a device. However, a device operating under normal operating conditions would require years of testing to determine its long term reliability. Methods of accelerating failures developed by the industry allow accurate prediction of a device life time and failure rate in a much shorter time duration. Accelerated stress tests are run at high temperature, high voltages, or a combination of both. Table 2 contains the results of the tests performed on programmed pASIC devices, where approximately 4% of ViaLink elements were programmed and the remaining ViaLink elements were left unprogrammed. These percentages are typical for a programmed, fully utilized pASIC device.

All tested devices were programmed with a proprietary reliability pattern which stresses the programmed and unprogrammed ViaLinks. A failure is defined as any change in the DC characteristic beyond the data sheet limits and any measurable change in the AC performance.

The overall reliability of the pASIC devices shown in the Table 2 is 19 FITs with a 60% confidence. Details of each test in Table 2 are given in the following sections. The reliability mechanisms specific to the ViaLink antifuse element are described in detail.

TABLE 2 Results of Accelerated Life Tests on the pASIC

Test	Process Qual. Acceptance Requirements	Test Results
HTOL, 1000 hrs, 125°C, V_{cc} = 5.5V, MIL-STD-883D, Method 1005	<100 FITs @ 55°C, E_a= 0.7eV, 60% confidence	3 failures, 14 observed FITs, 19 FITs at a 60% confidence. 2778 units from 41 lots
High temp. storage, 1000 hrs., 150°C, unbiased	LTPD = 5	0% failures 210 units from 6 lots
THB, 1000 hrs., alternately biased, 85% R.H., 85°C, JEDEC STD 22-B Method A101 or HAST 50 hrs 85% RH, 130°C, JEDEC STD 22-A110	LTPD = 5	0.51% failures 791 units from 26 lots
Temp. cycle, 1000 cycles, -65°C to 150°C, MIL-STD-883D, Method 1010	LTPD = 5	0.12% failures 854 units from 26 lots
Pressure Pot, 168 hrs., 121°C, 2.0 atm., no bias	LTPD = 5	0% failures 518 units from 26 lots
High Vcc Static Life, 1000 hrs, 25°C, V_{cc} = 7.0V, Static	< 20 FITs due to unprogrammed ViaLink element, A_f = 130	0 observed FITs. 675 units from 12 lots.
Low Temp. Operating Life, 1000 hrs, –55°C, V_{cc} = 6.0V, 8 to 15 MHz	< 20 FITs due to programmed ViaLink element, A_f = 380	0 observed FITs, 1605 units from 21 lots. 3 failures not related to the programmed ViaLink element

STANDARD CMOS TESTS AND RESULTS

High-Temperature Operating Life Test

HTOL is the life test which operates the device at a high Vcc and high temperature. This test is used to determine the long term reliability and failure rate of the device in the customer environment. The specific condition of this test is defined by MIL-STD-883D Quality Conformance Test. The devices are operated at 5.5v and 125°C for 1000 hours. The acceleration due to temperature can be calculated by using equation (1), assuming an average activation energy of 0.7 eV and an operating ambient temperature of 55°C. The observed failure rate in FITs is,

Failure Rate = (failures) * (10^9 device-hrs)/(total equiv device-hrs) (2)

The generally reported failure rate is a 60% confidence level of the observed FITs. The 60% confidence level is a statistical analysis of the actual failure rate which accounts for the observed failure rate and the sample size. For larger sample sizes, the 60% confidence level approaches the observed failure. The failure rate at this confidence level is calculated using the Poisson distribution, which is valid for a low failure occurrence in a large sample.

TABLE 3
Results of High-Temperature Operating Life Test

High-Temperature Operating Life Test
V_{cc} = 5.5V, Dynamic, f = 1 MHz, Temp. = 125°C

Package	Fab Lot	Device	Quantity	Failure @ Hours		
				168	500	1000
68 PLCC	18362	QL8X12	100	0	0	0
68 PLCC	19194	QL8X12	100	0	0	0
68 PLCC	19618	QL8X12	100	0	0	0
68 PLCC	20454	QL8X12	100	0	0	0
68 PLCC	20470	QL8X12	76	0	0	0
68 PLCC	20534	QL8X12	82	0	0	0
68 PLCC	21786	QL8X12	76	0	0	0
68 PLCC	33669	QL8X12	70	0	0	0
68 PLCC	34515	QL8X12	100	0	0	0
68 PLCC	35421	QL8X12	100	0	0	0
68 PLCC	34267	QL8X12A	100	0	0	0
68 PLCC	36129	QL8X12A	100	0	0	0
68 PLCC	40673	QL8x12A	51	0	0	0
68 PLCC	1346945	QL8X12B	100	0	0	0
68 PLCC	1351104	QL8X12B	100	0	0	0
68 PLCC	1409354	QL8X12B	100	0	0	1
84 PLCC	20762	QL12X16	100	0	0	0
84 PLCC	22164	QL12X16	100	0	0	1
84 PLCC	23001	QL12X16	100	0	0	1
84 PLCC	23093	QL12X16	36	0	0	0
84 PLCC	22988	QL12X16	32	0	0	0
84 PLCC	23091	QL12X16	38	0	0	0
84 PLCC	35284	QL12X16	100	0	0	0
84 PLCC	36935	QL12X16	100	0	0	0
84 PLCC	36936	QL12X16	98	0	0	0
84 PLCC	37448	QL12X16	65	0	0	0
84 PLCC	37449	QL12X16	74	0	0	0
84 PLCC	39068	QL12X16	54	0	0	0
84 PLCC	40670	QL12X16	50	0	0	0
84 PLCC	40672	QL12X16	50	0	0	0
84 PLCC	1415570	QL12X16B	67	0	0	0
84 PLCC	1418672	QL12X16B	73	0	0	0

Package	Fab Lot	Device	Quantity	Failure @ Hours		
				168	500	1000
84 PLCC	1328491	QL16X24B	2	0	0	0
84 PLCC	1337713	QL16X24B	22	0	0	0
84 PLCC	1337739	QL16X24B	19	0	0	0
84 PLCC	1407324	QL16X24B	20	0	0	0
84 PLCC	1407325	QL16X24B	38	0	0	0
84 PLCC	1413492	QL16X24B	6	0	0	0
84 PLCC	5-6	QL16X24B	24	0	0	0
84 PLCC	15-18	QL16X24B	46	0	0	0
84 PLCC	various	QL16X24B	9	0	0	0
TOTAL			**2778**	**0**	**0**	**3**

The acceleration factor from equation (1), for 55°C and E_a = 0.7 eV is 78. Therefore, from the results shown in Table 3, the pASIC has been operating for 210 million equivalent device hours with three failures. One was due to a gate oxide failure and the other was an Icc failure due to an improper test limit at final test. The test limit was corrected before any units were shipped to customers. The failure in the QL8X12B was due to a particle at via etch. The particles have been significantly reduced as part of an ongoing quality and yield improvement. The observed failure rate is 14 FITs and the failure rate at a 60% confidence level is 19 FITs.

There were no failures during the first 500 hours of life test. The pASIC does not have an infant mortality failure problem.

High Temperature Storage

High temperature storage test is a 150°C, 1000 hour, unbiased bake. This test accelerates failures due to mobile charge, thermal instabilities and bond ball intermetallic formation. The results in Table 4 demonstrate the stability of the programmed and unprogrammed ViaLink element and the long term shelf life of the pASIC.

High-Temperature Storage Test
No bias, Temp. = 150°C

TABLE 4 Results of High-Temperature Storage Test

Package	Fab Lot	Device	Quantity	Failure @ Hours		
				168	500	1000
68 PLCC	18362	QL8X12	35	0	0	0
68 PLCC	19194	QL8X12	35	0	0	0
68 PLCC	19390	QL8X12	35	0	0	0
68 PLCC	34515	QL8X12	35	0	0	0
68 PLCC	35421	QL8X12	35	0	0	0
68 PLCC	35422	QL8X12	35	0	0	0
TOTAL			**210**	**0**	**0**	**0**

TABLE 5
Results of Temperature, Humidity, and Bias Test

Temp. = 85°C, 85% R.H., pins alternately biased at 5.5V

Package	Fab Lot	Device	Quantity	Failure @ Hours		
				168	500	1000
68 PLCC	19194	QL8X12	100	0	0	0
68 PLCC	19618	QL8X12	100	0	0	0
68 PLCC	19454	QL8X12	100	0	0	0
68 PLCC	34515	QL8X12	35	1	0	0
68 PLCC	35421	QL8X12	35	1	0	0
68 PLCC	35422	QL8X12	35	0	0	0
84 PLCC	20762	QL12X16	29	0	0	0
84 PLCC	23000	QL12X16	24	1	0	0
84 PLCC	23001	QL12X16	24	0	0	0
TOTAL			**482**	**3**	**0**	**0**

Temp. = 130°C, 85% R.H., pins alternately biased at 5.5V

Package	Fab Lot	Device	Quantity	Failure @ hours	
				50 @ 130°C	128 @ 140°C
100TQFP	36936	QL12X16	15	0	-
100TQFP	37447	QL12X16	13	0	-
100TQFP	37448	QL12X16	17	-	-
68 PLCC	1409354	QL8X12B	25	-	0
68 PLCC	1409353	QL8X12B	20	-	0
84 PLCC	1402176	QL16X24B	30	-	0
84 PLCC	1402177	QL16X24B	15	-	0
84 PLCC	1410389	QL16X24B	15	-	0
84 PLCC	1408334	QL16X24B	16	-	0
84 PLCC	1450263	QL16X24B	14	-	0
84 PLCC	1417657	QL16X24B	22	-	0
84 PLCC	10885	QL16X24B	2	-	0
144 TQFP	1412454	QL16X24B	14	0	-
144 TQFP	1413492	QL16X24B	15	1	-
144 TQFP	49403007	QL16X24B	16	0	-
144 TQFP	1411431	QL16X24B	15	0	-
144 TQFP	1417657	QL16X24B	45	0	-
TOTAL			**309**	**1**	**0**

Temperature Cycle Test
Air-to-air -65°C to 150°C

TABLE 6 Results of Temperature Cycle Test

Package	Fab Lot	Device	Quantity	Failure @ Cycles		
				250	500	1000
68 PLCC	18362	QL8X12	35	0	0	0
68 PLCC	19194	QL8X12	35	0	0	0
68 PLCC	19618	QL8X12	35	0	0	0
68 PLCC	34515	QL8X12	35	0	0	0
68 PLCC	35421	QL8X12	35	0	0	0
68 PLCC	35422	QL8X12	35	0	0	0
68 PLCC	39017	QL8X12A	38	0	0	0
84 PLCC	20762	QL12X16	29	0	0	0
84 PLCC	22999	QL12X16	24	0	0	0
84 PLCC	23000	QL12X16	24	0	0	0
84 PLCC	39068	QL12X16	85	0	0	1
84 PLCC	38980	QL12X16	37	0	0	0
84 PLCC	38979	QL12X16	65	0	0	0
100 TQFP	36934	QL8X12A	30	0	0	0
100 TQFP	36128	QL8X12A	30	0	0	0
100 TQFP	36129	QL8X12A	30	0	0	0
100 TQFP	37447	QL12X16	30	0	0	0
100 TQFP	37448	QL12X16	30	0	0	0
100 TQFP	36936	QL12X16	30	0	0	0
84 PLCC	1409378	QL12X16B	30	0	-	0
84 PLCC	1417657	QL16X24B	31	0	-	0
84 PLCC	1416600	QL16X24B	16	0	-	0
144 TQFP	1342851	QL16X24B	25	0	-	0
144 TQFP	1405263	QL16X24B	15	0	-	0
144 TQFP	1351123	QL16X24B	15	0	-	0
144 TQFP	1417657	QL16X24B	30	0	-	0
TOTAL			**854**	**0**	**0**	**1**

Temperature, Humidity, and Bias

The temperature, humidity, and bias test is performed under severe environmental conditions. The device is exposed to a temperature of 85°C and a relative humidity of 85% for 1000 hours, while the pins are alternately biased between 0 and 5.5 volts (JEDEC STD 22-B). An alternate temperature, humidity and bias test is HAST, a Highly Accelerated Stress Test. This test is similar to the 85°C/85% relative humidity test except that the test is done at 130°C and 85% relative humidity. The vapor pressure of at this condition is 33.5 psia. The pins are bias alternately at 5.5 volts for 50 hours (JEDEC STD 22-A110). Some parts were stressed more at a more stringent condition, 140°C for 128 hours. These two tests are effective at detecting corrosion problems, while also stressing the package and bonding wires. Table 5 shows that the pASIC had three failures from the total 482

units in 85/85 and one failure in HAST. The first failure was an Icc failure caused by a power supply surge. The second failure was a short due a particle under metal 2. The third failure was an Icc failure due to the same improper test limit which caused a failure in HTOL. The limit was corrected before units were shipped to customers. The one HAST failure in the QL16X24B was due to a metal short. The ongoing quality and yield improvement effort has reduced the defect level.

Temperature Cycle Tests

The temperature cycle test stresses the packaged part from -65°C to 150°C for 1000 cycles. The air-to-air cycling follows the MIL-STD-883D Quality Conformance Test. This test checks for any problems due to the thermal stresses. The plastic package, lead frame, silicon die, and die materials expand and contract at different rates. This mismatch can lead to cracking, peeling, or delamination of the high stress layers. The results in Table 6 show that the pASIC had one failure. The failure was due to a lifted bond.

Pressure Pot Tests

The pressure pot test is performed at 121°C at 2.0 atmospheres of saturated steam with devices in an unbiased state. This test forces moisture into the plastic package and tests for corrosion in the bonding pads and wires which are not protected by passivation. Corrosion can also occur in passivated areas where there are micro cracks or poor step coverage. The pASIC had no failures as shown in Table 8.

TABLE 7
Results of Pressure Pot Test

Pressure Pot Test
Pressure = 2.0 atm., Temp. = 121°C, no bias

Package	Fab Lot	Device	Quantity	Failures @ Hours		
				48	96	168
68 PLCC	18362	QL8X12	35	0	0	0
68 PLCC	19194	QL8X12	35	0	0	0
68 PLCC	19390	QL8X12	35	0	0	0
68 PLCC	34515	QL8X12	35	0	0	0
68 PLCC	35421	QL8X12	35	0	0	0
68 PLCC	35422	QL8X12	35	0	0	0
84 PLCC	20762	QL12X16	28	0	0	0
84 PLCC	22999	QL12X16	25	0	0	0
84 PLCC	23000	QL12X16	24	0	0	0
100TQFP	36936	QL12X16	15	-	-	0
100TQFP	37447	QL12X16	15	-	-	0
100TQFP	37448	QL12X16	15	-	-	0
68 PLCC	1346945	QL8X12B	15	-	-	0
68 PLCC	1350096	QL8X12B	15	-	-	0
68 PLCC	1350104	QL8X12B	15	-	-	0
84 PLCC	1417657	QL16X24B	30	-	-	0
84 PLCC	1416600	QL16X24B	15	-	-	0

Package	Fab Lot	Device	Quantity	Failures @ Hours		
				48	96	168
84 PLCC	1409360	QL16X24B	36	-	-	0
84 PLCC	1410389	QL16X24B	15	-	-	0
84 PLCC	1350093	QL16X24B	15	-	-	0
84 PLCC	1403209	QL16X24B	15	-	-	0
84 PLCC	1402177	QL16X24B	15	-	-	0
144 TQFP	1342851	QL16X24B	15	-	-	0
144 TQFP	1405263	QL16X24B	15	-	-	0
144 TQFP	1351123	QL16X24B	15	-	-	0
144 TQFP	1417657	QL16X24B	30	-	-	0
TOTAL			**518**	**0**	**0**	**0**

ViaLink ELEMENT RELIABILITY TESTS AND RESULTS

The ViaLink antifuse is a one time programmable device. In the unprogrammed state it has a resistance of greater than one gigaohm and capacitance of less than one femtofarad.

The application of a programming voltage across the antifuse structure, above a critical level causes the device to undergo a switching transition through a negative resistance region into a low resistance state. The magnitude of the current allowed to flow in the low resistance state, the programming current, is predetermined by design. A link of tungsten, titanium, and silicon alloy is formed between metal one and metal two during the programming process [3].

The link has a metallic-like resistivity of the order of 500 micro-ohms-cm and is responsible for the low 50 ohm resistance that is a unique characteristic of the QuickLogic ViaLink antifuse.

The link forms a permanent, bi-directional connection between two metal lines. The size of the link, and hence, the resistance, depends on the magnitude of the programming current. Figure 2 shows the relationship between programming current and programmed link resistance. Figure 3 shows the distribution of link resistance for a fixed programming current.

Fig. 2. Resistance vs 1 / Programming Current
R = 0.810 / Ip

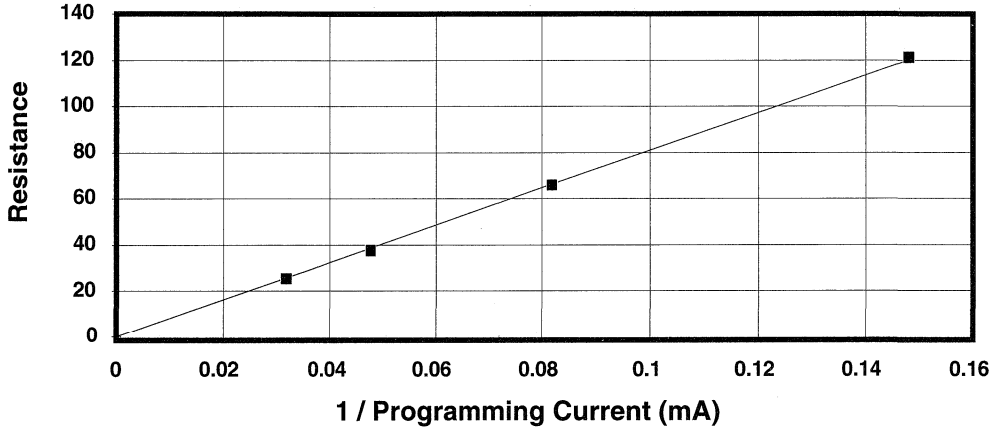
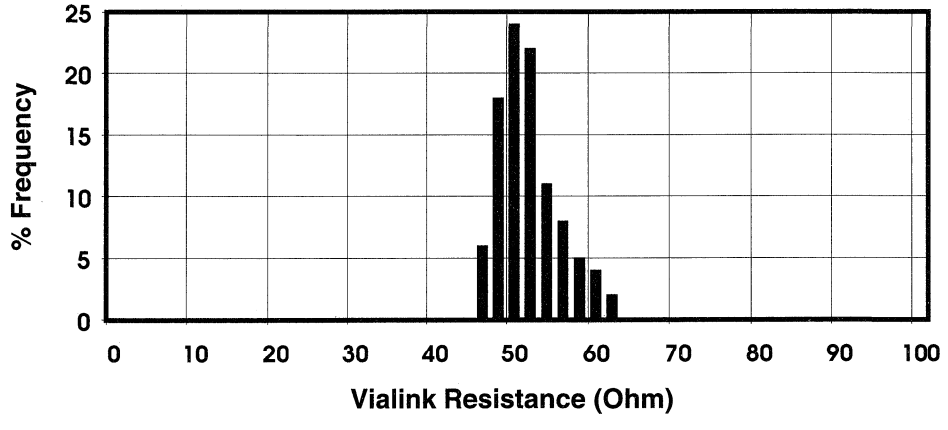


Fig. 3. Distribution of ViaLink Resistance at Ip = 15mA
Average = 52.3 Ohms Std Deviation = 3.69



Unprogrammed ViaLink Element Reliability

Reliability studies on an antifuse that can exist in two stable resistance states must focus on the ability of an unprogrammed and a programmed device under stress to remain in the desired state. In the context of standard IC testing, the antifuse should be stressed under conditions similar to those for a dielectric (in the unprogrammed state) and for a conductor (in the programmed state).

For ViaLink elements in the unprogrammed state, the tests must determine their ability to withstand applied voltages over the range of operating conditions without changing resistance or becoming programmed. Amorphous materials might be expected to show gradual changes in resistance as a result of relaxation or annealing. Reliability studies have been designed to explore these effects. An analysis of the unprogrammed reliability follows. A more detailed presentation of the unprogrammed ViaLink, a reprint of a paper presented at the 1994 International Reliability Physics Symposium, follows this section.

When a ViaLink element is stressed at high electric fields its resistance can decrease from the initial 1 G-Ω value. The reliability testing program examined the time for the resistance to reach 50 M-Ω at different stress fields. Figures 4 and 5 illustrate that because of time constraints (≅500 years), it is impossible to detect this effect at normal operating fields in systems. To keep the data consistent, all reported resistances are at 20°C unless specified.

The QL8x12 is designed to operate with resistances of the unprogrammed ViaLink element from 50 M-Ω to greater than 1 G-Ω at 20°C. Even with all unprogrammed ViaLink elements at 50 M-Ω, the QL8x12 would remain within the guaranteed speed and standby Icc specifications at all temperatures.

Figure 4 shows the time required for a ViaLink element to reach 50 M-Ω under various applied electric fields at different temperatures. The time required for the change is not accelerated by temperature over the studied range of electric fields. The activation energy, E_a, for this process is zero.

Figure 5 shows the time required for a ViaLink element to reach 50 M-Ω under various electric field stresses. A range of amorphous silicon thicknesses have been included in this chart. The data can be modeled using the equation,

$$t_{50M\Omega} = t_0 \exp(-BE) \qquad (3)$$

where the time to 50 M-Ω decreases exponentially with increasing applied electric field. The constant t_0 is $3x10^{15}$ seconds and the field acceleration factor, B, is 20 cm/MV. The model is valid for electric fields, E, below 1.5 MV/cm. Above this field, the amorphous silicon antifuse programs with a different mechanism. This mechanism is an advantage over dielectric antifuses where the unprogrammed reliability mechanism, Time Dependent Dielectric Breakdown, is the same as the programming mechanism. The electric field for 5.0 volt Vcc operation with a typical amorphous silicon thickness is 0.61 MV/cm, which extrapolates $t_{50M\Omega}$ to $1.5x10^{10}$ seconds, or 500 years. The time to 50 M-Ω for the worst case amorphous silicon thickness and operating at worst case Vcc is in excess of 30 years.

Fig. 4. Temperature Dependence of Time to 50 Megaohms
Lot 617 Wafer 8

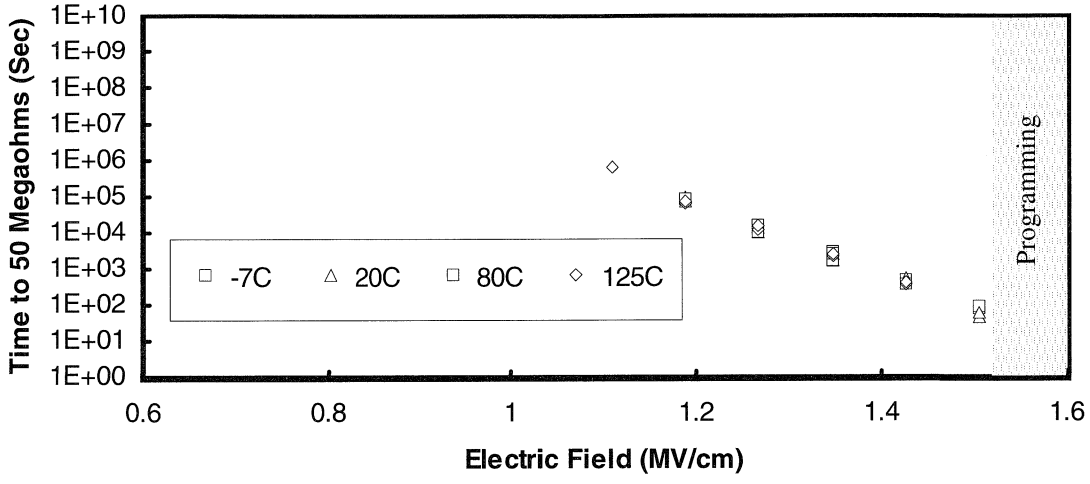
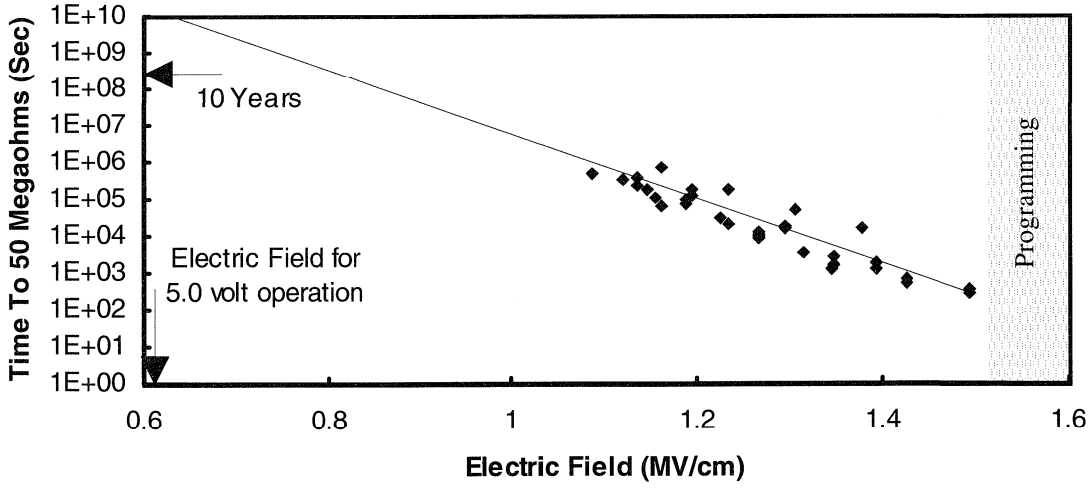


Fig. 5. Electric Field Acceleration of Unprogrammed ViaLink Element from 6 Production Lots



The high field effect is predictable, reproducible, and reversible. This effect is inherent in the amorphous silicon in the ViaLink element [4]. The pASIC device has been designed to operate where the effect is minimized and has no impact on the reliability of the pASIC device. The pASIC device lifetime extrapolations are based on the average unprogrammed ViaLink element since the effect on the increased Icc of the pASIC is the sum of the leakages through the unprogrammed ViaLink elements. The time dependent resistance of the ViaLink elements does not degrade the functionality or AC performance of the pASIC.

Accelerated Stress Tests for Unprogrammed ViaLink Elements

The high field effect is created in the packaged pASIC devices through a high Vcc static life test. This test stresses the unprogrammed ViaLink element with a Vcc = 7.0 volts for 1000 hours. Over 600 pASIC devices from twelve lots have been tested. This condition stresses over 20,000 unprogrammed ViaLink elements in each QL8x12. The failure criteria for the pASIC device for this test is the same as the previous tests, with emphasis placed on the standby Icc, which increases as the resistance of the unprogrammed ViaLink element decreases. The acceleration factor for this stress is calculated by using equation (3) to find the ratio of the $t_{50M\Omega}$ for E = 0.61 MV/cm at 5 volts and E = 0.85 MV/cm at 7 volts. This test has an acceleration factor = 130 for the unprogrammed ViaLink element. The test results in Table 8 show that no device has failed this stress in more than 220 million equivalent device hours. Four lots have 6000 hours of test which is equivalent to over 80 years of operation. The High Vcc Static test was not done on the 0.65 micron pASICs because the submicron CMOS devices are not reliable at these very high voltages. The reliability stresses of the unprogrammed ViaLink for the 0.65 micron process was done on test structures where the Via Link could be stress independently from the CMOS.

High V_{cc} Static Life Test
V_{cc} = 7.0V, static, Temp. = 25°C

TABLE 8
Results of High V_{cc} Static Life Test

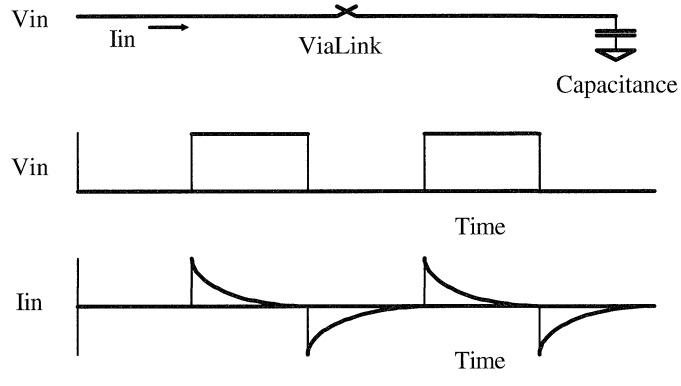
Package	Fab Lot	Device	Quantity	Total Hours	Failures	Equiv. Dev Hrs
68 PLCC	16558	QL8X12	14	3650	0	6.6E+06
68 PLCC	18362	QL8X12	38	1907	0	9.4E+06
68 PLCC	19194	QL8X12	110	1756	0	2.5E+07
68 PLCC	19618	QL8X12	101	1464	0	1.9E+07
68 PLCC	20454	QL8X12	100	2800	0	3.6E+07
68 PLCC	34515	QL8X12	37	1000	0	4.8E+06
68 PLCC	35421	QL8X12	36	1000	0	4.7E+06
68 PLCC	35422	QL8X12	33	1000	0	4.3E+06
68 PLCC	33403	QL8X12A	100	1000	0	1.3E+07
68 PLCC	34515	QL8X12A	36	6000	0	2.8E+07
68 PLCC	35421	QL8X12A	34	6000	0	2.7E+07
68 PLCC	35422	QL8X12A	36	6000	0	2.8E+07
TOTAL			**675**		0	**2.2E+08**

Programmed ViaLink Element Reliability

The reliability tests on the programmed ViaLink element must demonstrate the stability of the link resistance in the programmed state. While an increase in resistance of the programmed device may not be catastrophic, a higher resistance can affect the device operating speed. Because the programmed ViaLink element is part of the on-chip interconnect, reliability tests should be similar to those that are normally used to validate the integrity of metal interconnects.

In operation, the programmed ViaLink elements are subjected to capacitive switching current of the interconnect network. They do not experience any DC current or voltage. See Figure 6. Each switching pulse forces a capacitive charging current to flow through programmed ViaLink elements into the network on the rising edge, and an opposite, or discharging current, to flow on the falling edge. Each cycle is analogous to a read pulse for a memory device. A 10% increase in resistance was set as the read disturb criteria for the ViaLink element. The typical impedance of a network is about 500Ω with the programmed ViaLink element contributing 50Ω. A 10% increase in the ViaLink resistance will increase the network impedance by approximately 5Ω, or 1%. This increase in resistance will increase a network delay in the pASIC device by about the same proportion.

FIGURE 6 Switching of Programmed ViaLink Antifuse



Programmed ViaLink elements were stressed under severe capacitive currents. AC stresses rather than DC stresses were used to accelerate the failures for closer correlation with actual operation. Figure 7 shows that the mean number of read cycles to disturb, N_{50}, for 25°C and 250°C are identical. The lack of an observable temperature dependence indicates that the activation energy is 0, or the self heating in the antifuse is high enough to make the 225°C ambient delta insignificant. Figure 8 shows the acceleration of the read disturb at high AC current densities through the programmed ViaLink element. Thus, the number of cycles to disturb can be modeled as,

$$N_{50} = N_0 \exp(-PJ) \qquad (4)$$

Where $N_0 = 7x10^{41}$ cycles is a constant, P = 1.2 cm^2/MA is the current density acceleration factor, and J is the peak AC current density through the link.

The pASIC is designed to operate at worst case AC current density of $35x10^6$ A/cm^2. The N_{50} for this condition is $4x10^{23}$ cycles. The failure rate can be calculated using the cumulative density F(t),

$$F(t) = \Phi \ln [(N/N_{50})/\sigma] \qquad (5)$$

The failure distribution can be determined by plotting the data on a log normal probability scale versus the log of the number of cycles to failure. See Figure 7. The shape parameter, σ, is $\ln(N_{50}/N_{16}) = 2.5$. The shape parameter is relatively large because it includes the measurement tolerances of the current density.

High AC current density occurs at low frequencies where there is sufficient time for the network to be fully charged or discharged. At frequencies above 50 MHz, AC current through a ViaLink element decreases due to incomplete charging and discharging cycle. The worst case process, programming and operating conditions will produce an operating current density of $35x10^6$ A/cm^2 in the present pASIC devices. Using equation (5), the cumulative failure rate for the ViaLink element operating at $35x10^6$A/cm^2 for $1.6x10^{16}$ read cycles (equivalent to continuous operation at 50 MHz for 10 years) is less than 0.1 parts per billion. The worst case pattern in a programmed QL16x24B has less than 1200 ViaLink elements operating at the high current density. Most of the programmed ViaLink elements operate at much lower current densities. The failure rate of the programmed ViaLink element in the QL16x24B contributes less than 0.01 FITs to the overall failure rate.

Fig 7. Temperature Dependence of Accelerated Read Disturb

J is the same for both temperatures

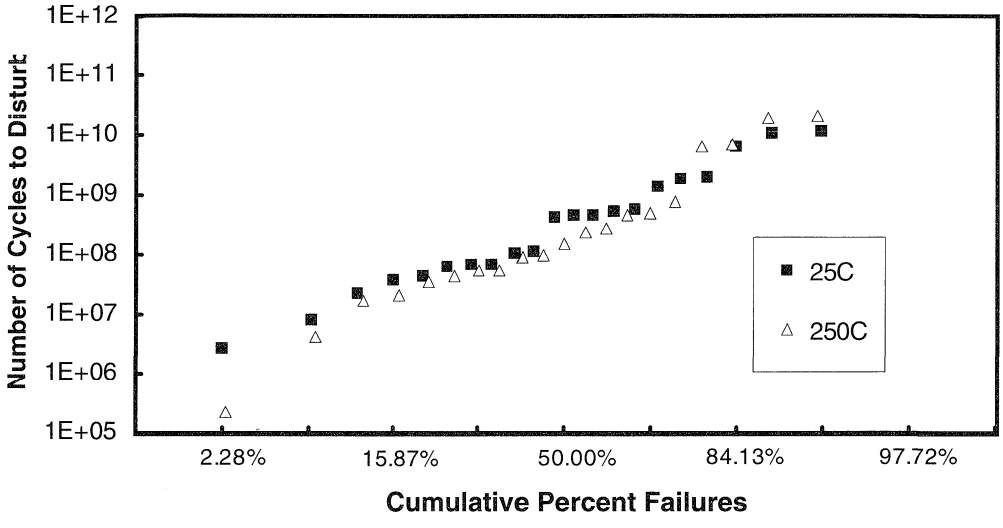
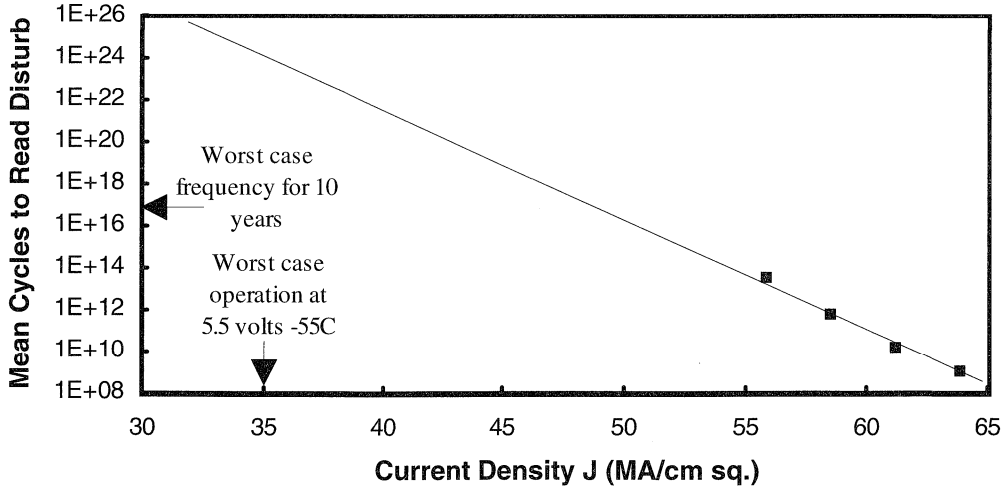


Fig. 8. Acceleration of Read Disturb for Programmed ViaLink Element



Accelerated Stress Tests for Programmed ViaLink elements

The low temperature operating life test stresses the pASIC devices with Vcc = 6.0 volts at 15 MHz for 1000 hours at -55°C. The lack of an ambient temperature dependence on read disturb allow QuickLogic to stress at any temperature without changing the deprogramming cycle dependence. Cold temperature operation was chosen because it accelerates the stress by increasing the drive current of the CMOS devices. This test stresses the programmed ViaLink elements at $40x10^6$ A/cm^2 for $5.4x10^{13}$cycles. The acceleration factor, calculated from equation (4), is 380. This test is equivalent to $2.0x10^{16}$ switching cycles, or continuous operation under worst case condition at 50 MHz for 12 years at the worst case current density of $35x10^6$ A/cm^2. Over 1000 pASIC devices from 13 lots have been stressed. The failure criteria is the same as previously described, with emphasis placed on careful monitoring of AC performance. Test results in Table 9 show that there have been no failures of the programmed ViaLink.

This Low Temperature Operating Life test stresses the programmed ViaLinks at $40x10^6$ A/cm^2. The acceleration factor for this test is 380. The results in Table 9 show no failure on programmed ViaLink elements in over 110 million equivalent device hours. There were three failures unrelated to the ViaLink element. The first was a metal one to metal two short. The other two devices shorted when a power supply surged during the test.

Low Temperature Operating Life Test
Vcc = 6.0V, Dynamic, f = 15 MHz, Temp = -55°C

TABLE 9
Results of the Low Temperature Operating Life Test

Package	Fab Lot	Device	Quantity	Failures @ Hours		
				168	500	1000
68 PLCC	34267	QL8X12A	100	0	0	0
68 PLCC	36128	QL8X12A	100	0	0	0
68 PLCC	36129	QL8X12A	150	0	0	0
68 PLCC	36934	QL8X12A	70	0	0	0
84 PLCC	20762	QL12X16	100	0	0	0
84 PLCC	22999	QL12X16	100	0	0	0
84 PLCC	23001	QL12X16	100	1	0	0
84 PLCC	36935	QL12X16	100	0	0	0
84 PLCC	36936	QL12X16	100	2	0	0
84 PLCC	38713	QL12X16	50	0	0	0
84 PLCC	1323368	QL16X24B	4	0	0	0
84 PLCC	1337713	QL16X24B	10	0	0	0
84 PLCC	1337739	QL16X24B	20	0	0	0

Package	Fab Lot	Device	Quantity	Failures @ Hours		
				168	500	1000
68 PLCC	1409353	QL8X12B	150	0	0	0
68 PLCC	1409354	QL8X12B	150	0	0	0
84 PLCC	1409390	QL12X16B	200	0	0	0
84 PLCC	1409360	QL16X24B	40	0	0	0
84 PLCC	1410389	QL16X24B	33	0	0	0
84 PLCC	404333	QL16X24B	66	0	0	0
84 PLCC	404788	QL16X24B	1	0	0	0
TOTAL			**1604**	**3**	**0**	**0**

Conclusions From Life Tests

The testing reported here establishes the reliability of the pASIC devices. No failures have been observed in 210 million equivalent device hours of high temperature operating life. The observed failure rate is 14 FITs and the failure rate with a 60% confidence is 19 FITs with no infant mortality. The acceleration factors that can lead to the degradation of the programmed and unprogrammed ViaLink elements were studied. The pASIC devices are designed to operate at voltages and currents where the failure rate of the ViaLink element does not measurably increase the failure rate of the pASIC device above that of normal CMOS products.

Up to date data on new parts can be obtained from QuickLogic.

REFERENCES

[1] Jim Nulty, et al, "A High Reliability Metallization System for a Double Metal 1.5µm CMOS Process", *Proc. Fifth IEEE VMIC*, 1988, pp. 453-459.

[2] Dipankar Pramanik, et al, "A High Reliability Triple Metal Process for High Performance Application Specific Circuits", *Proc. Eighth IEEE VMIC*, 1991, pp. 27-33.

[3] K Gordon and R Wong, "Conducting Filament of the Programmed Metal Electrode Amorphous Silicon Antifuse" *IEDM* 1993, pp27-30.

[4] F. Yonezawa, Fundamental Physics of Amorphous Semiconductors, *Proc of the Kyoto Summer Inst.*, 1981.

METASTABILITY REPORT FOR FPGAs

As system designers continue to push the upper bound of performance, understanding the metastability operation of flip-flops is important to reliability. High reliability can be achieved by good synchronous design practice or careful evaluation of device characteristics. As the speed of designs increases to 50 MHz and above, metastability becomes an important issue in determining system MTBF (mean-time-between-failure).

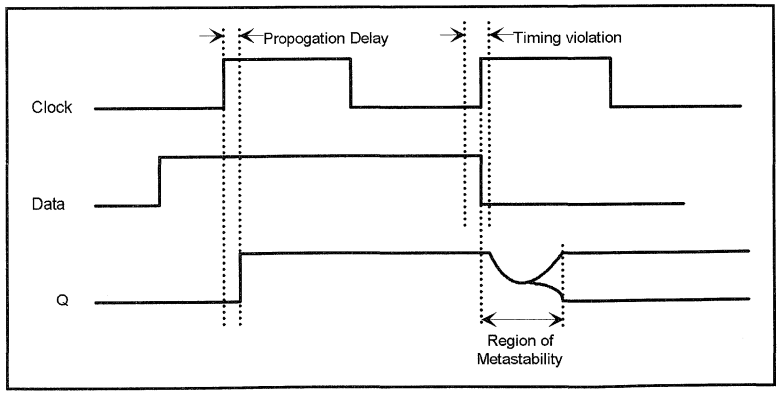
Metastability is a very unpredictable event. It can cause erratic system operation which can not be replicated on the bench. A detailed analysis of the potential conditions and mechanisms involved can help the designer avoid problems in metastability.

WHAT IS METASTABILITY?

Metastability is a region of uncertainty exhibited by a flip-flop when the timing characteristics of setup and hold have been violated. In an ideal world, where all logic designs are synchronous and all inputs are tied to the system clock, metastability would not be a concern because all timing conditions for the flip-flops would be met. But since all systems are not synchronous and almost every design has at least one completely asynchronous signal which needs to be synchronized to the system clock, the designer needs to take into account the possibility of violating some timing specifications.

Metastable events are associated with data transitions occurring close to the active edge of the clock. Figure 1 shows an example of data changing within a timing window which will result in a timing violation. Because a timing violation has occurred, the flip-flop will exhibit erratic behavior. The erratic behavior manifests itself in the form of an extended propagation delay with an unpredictable resolution of the Q output.

FIGURE 1 Metastable Condition



QUICKLOGIC

HOW IS METASTABILITY DESCRIBED?

Metastability is typically described by four measurements of flip-flop performance — MTBF, tau, W and t_{res}. MTBF is the mean-time-between-failure of a flip-flop. Tau and W are measured characteristics of flip-flop performance which are dependant on the process technology and the internal design of the flip-flop. The value t_{res} is the resolution time allowed for a metastable event to resolve itself before the output will be sampled.

The general expression for describing MTBF is:

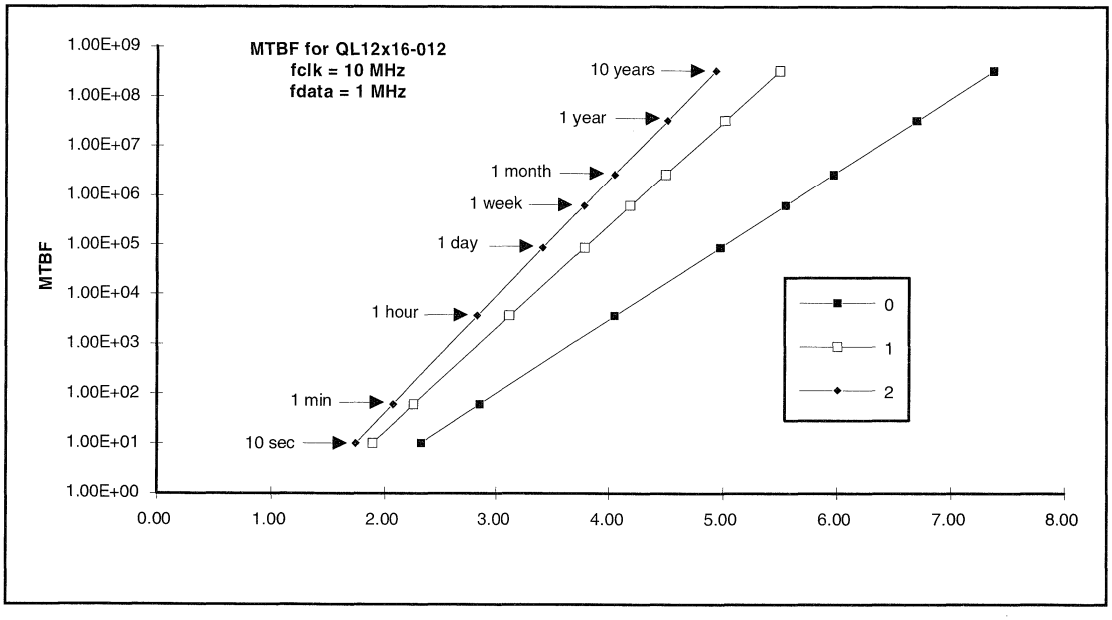
$$MTBF = \frac{e^{t_{res} / tau}}{f_{clk} f_{data} W}$$

where f_{clk} and f_{data} are the frequency of clock and data respectively. Solving the equation for t_{res} results in the following equation:

$$t_{res} = (tau)\ Ln\ [(MTBF)\ (f_{clk} f_{data})\ (W)]$$

Figure 2 shows a graph of MTBF vs t_{res} and how slight increases in t_{res} cause a significant change in the MTBF of a flip-flop.

FIGURE 2
MTBF vs t_{res} (ns)



THE MEASUREMENTS

A series of measurements were taken to evaluate the metastability hardness of QuickLogic's pASIC® 1 family of FPGA's. Figure 3 depicts the circuit used to measure metastable events. The first flip-flop (Labeled A) is the metastable event generator. Under normal operation, the output of flip-flops B and C will be different, making the output of the exclusive nor gate a logic 0. If a metastable event occurs, the outputs will be the same and cause the exclusive nor gate to become a logic 1. When this occurs, flip-flop D will record this event and transmit it through the nand gate to the signal ERRCLK. ERRCLK is a signal that clocks a ripple counter to record the number of events that occurred during the test. The multiplexor output to the A flip-flop is used as a test circuit to determine the f_{max} of the register. The signal ERR is a test point for further evaluation.

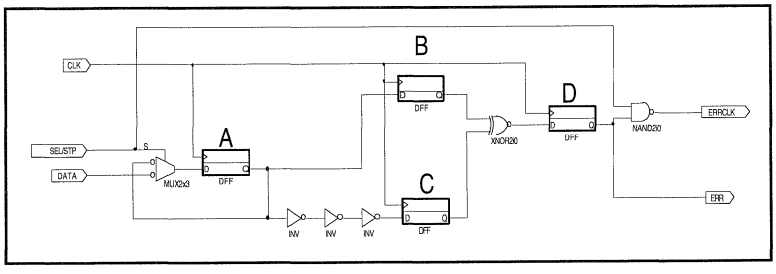


FIGURE 3
Test Circuit

The collection of data is accomplished in two stages. Stage one is to determine the f_{max} of the test circuit by setting up flip-flop A in a divide-by-two configuration. This guarantees that any failures detected by flip-flops C and D are a result of exceeding f_{max} and not of generated metastable events. Once f_{max} is established, a variety of combinations of f_{clk} and f_{data} are applied to the circuit and the number of failures are recored over a specified period of time. By maintaining a constant relationship between the product of f_{clk} and f_{data} the following equation can be used to determine the tau of the register.

$$tau = \frac{t_{res1} - t_{res2}}{\text{Ln}\,(MTBF_1) - \text{Ln}\,(MTBF_2)}$$

Tau was determined by taking measurements for several values of f_{clk} and f_{data} over many different devices. By inserting the value of tau into the MTBF equation, W was determined and the graph in Figure 2 was plotted for MTBF vs t_{res}.

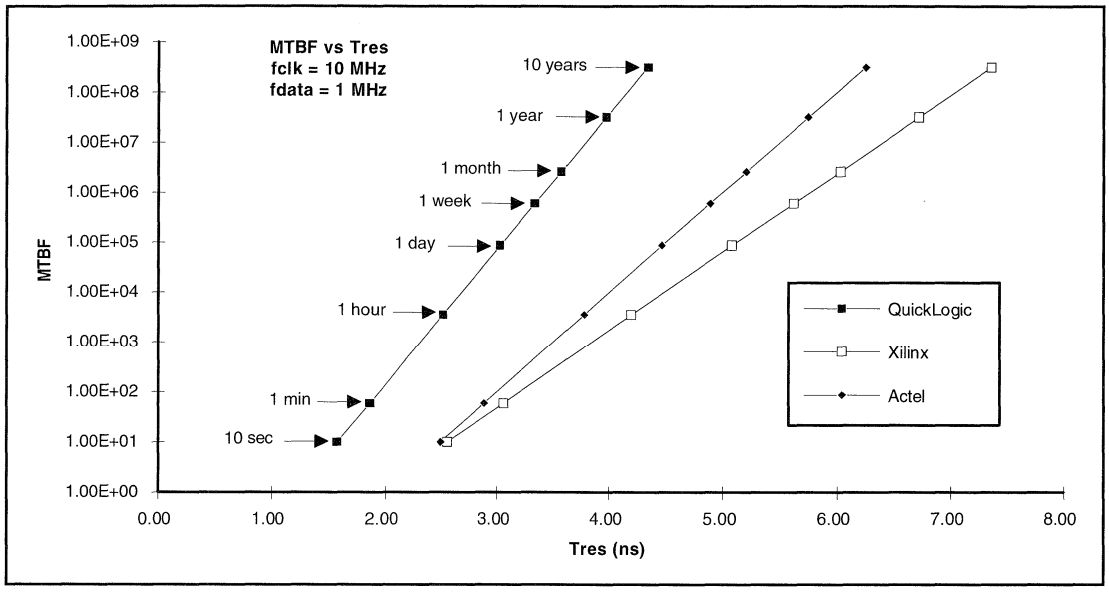
The table below gives the values of tau and W for the QuickLogic QL12X16-0,1,2:

Device	tau (sec)	W (sec)
QL12x16-0	2.91E-10	2.94E-11
QL12x16-1	2.09E-10	8.38E-11
QL12x16-2	1.85E-10	1.23E-10

CONCLUSION

Comparing the MTBF results of the QuickLogic metastability tests with other published FPGA manufacturers, Figure 4 shows that pASIC devices have a much better MTBF than other FPGAs' published data.

FIGURE 4[1] MTBF Comparisons



This very low MTBF, combined with an easy to use high-speed architecture, deterministic routing delays and fully automatic compilation tools, makes QuickLogic the best choice when designing high-performance, highly reliable FPGA-based systems.

[1]Xilinx data was taken from the 1992 Xilinx "The Programmable Gate Array Data Book" pages 6-16 through 6-17. Actel data was taken from the April 1992 Actel Data Book pages 5-1 and 5-2. Altera is not shown because from the October 1990 Altera Data Book pages 289 through 296, a MTBF of 10 seconds is equal to a t_{res} of 38.2 ns.

pASIC 1 FAMILY
Power vs Operating Frequency

pASIC 1 FAMILY POWER CALCULATIONS

Bipolar devices draw similar amounts of current regardless of frequency. CMOS devices use power in relation to the switching frequency, in addition to drawing a nominal amount of static Icc. CMOS power calculations are therefore based upon combining both static power and active power. Many of the power calculation models used for CMOS gate arrays can be applied to pASIC devices.

Static Power

The FPGA draws static current when in a quiescent state. QuickLogic FPGAs are rated at 10 mA worst case and are typically 2 mA.

Active Power

The FPGA draws active current dependent upon operating frequency. The active power calculation uses the following components:

- Macrocells
- Input Buffers
- Output Buffers
- High-Drive Buffers
- Clock Buffers

The general equation for calculating active power for each individual component is:

$$P_{comp(mw)} = (\#Components)*(C_{equiv(pf)})*(F_{ave(MHz)})*(V_{cc(volts)})^2*(10^{-3})$$

The "#Components" is the number of components being used; "C_{equiv}" is the equivalent capacitance from Figure 1 and Table 1; "F_{ave}" is the average frequency of the components and "V_{cc}" is the operating voltage of the device.

Example Power Calculation

For this example we will use a 16-bit counter which is implemented in 16 macrocells, uses one clock pad, 16 outputs and two half-column clock drivers with eight loads each. The incoming clock frequency is 33 MHz operating at 5 volts. Total device power is determined by calculating the power of the individual components.

Macrocells

From Figure 1 or Table 1 we locate C_{equiv} for an individual macrocell. Because each macrocell operates at a unique speed, the average speed (F/16) is used to determine this number. The C_{equiv} for 2 MHz is 17.62 pf/MHz-Macrocell. Plugging this into the equation we get

$$P_{macro(mW)} = (16)*(17.62)*(33/16)*(5)^2*(10^{-3}) = 14.54 \text{ mW}$$

Clock Input (CKPAD)

From Figure 1 we get a C_{equiv} for an individual clock pad operating at 33 MHz of 14.33 pf/MHz-Macrocell.

$$P_{clock\ pad(mW)} = (1)*(14.33)*(33)*(5)^2*(10^{-3}) = 11.82 \text{ mW}$$

Clock Column Buffers

Since this design has been laid out in two columns, the component number is 2 and from Figure 1 we get a C_{equiv} for an individual clock column buffer operating at 33 MHz of 3.13 pf/MHz-Macrocell.

$$P_{colbuf(mW)} = (2)*(3.13)*(33)*(5)^2*(10^{-3}) = 5.16 \text{ mW}$$

Clock Loading

From Figure 1 the C_{equiv} for a single clock load is 0.86 pf/MHz-macrocell. Since 16 macrocells are being used, the number of Components is 16.

$$P_{clock\ loading(mW)} = (16)*(0.86)*(33)*(5)^2*(10^{-3}) = 11.35 \text{ mW}$$

Output Buffers

The C_{equiv} for an individual output buffer operating at the average frequency of 2 MHz is 15.00 pf/MHz-Macrocell.

$$P_{outputs(mW)} = (16)*(15.00)*(33/16)*(5)^2*(10^{-3}) = 12.38 \text{ mW}$$

Total Device Power

By adding both the typical static power of 10 mW and the total active power, we determine that the typical total device power equals the following:

$P_{sys(mW)}$ =
(14.54mW)+(11.82mW)+(5.16mW)+(11.35mW)+(12.38mW)+(10mW)
= 65.25 mW

FIGURE 1 Ceq vs Operating Frequency

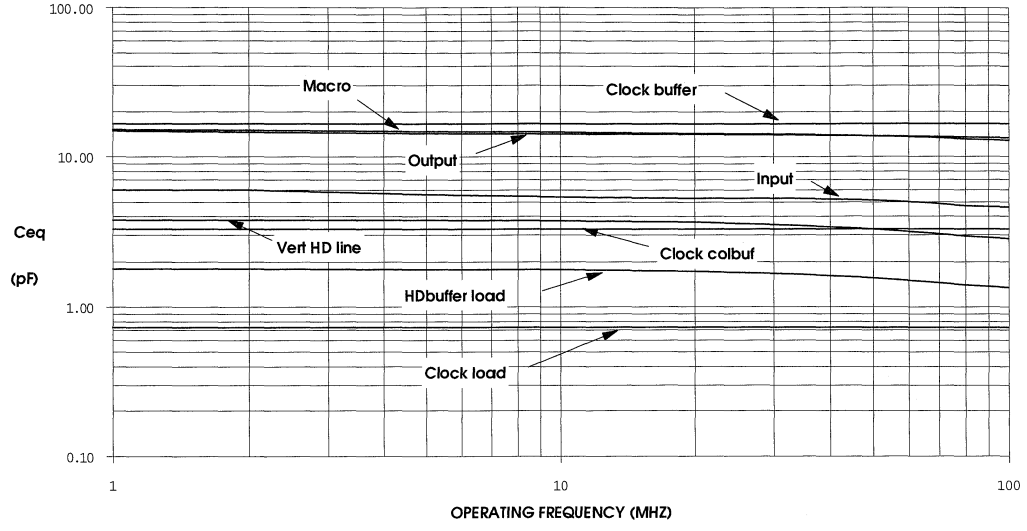


TABLE 1 Equivalent Capacitances

Speed	MHz	1	2	5	10	20	33	50	66	80	100	120
Input	(pF)	6.00	6.00	5.60	5.40	5.30	5.27	5.12	4.91	4.70	4.60	4.57
Output	(pF)	15.20	15.00	14.73	14.57	14.33	14.09	13.75	13.46	13.06	12.87	12.81
Macro	(pF)	17.63	17.62	17.60	17.57	17.50	17.41	17.30	17.19	17.10	16.97	16.84
HD buffer load	(pF)	1.79	1.79	1.78	1.77	1.72	1.65	1.55	1.47	1.39	1.34	1.30
Vert HD Line	(pF)	3.79	3.79	3.77	3.75	3.66	3.51	3.29	3.11	2.96	2.85	2.78
Clock buffer	(pF)	14.33	14.33	14.33	14.33	14.33	14.33	14.33	14.33	14.33	14.33	14.33
Clock colbuf	(pF)	3.13	3.13	3.13	3.13	3.13	3.13	3.13	3.13	3.13	3.13	3.13
Clock load	(pF)	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86

The thermal performance of a pASIC in its package is determined by many factors, including packaging materials, die size, package design and construction, etc. Thermal resistance is the measure of the ability of the package to conduct heat from the chip through the package to the external environment. This is represented in degrees C/Watt by $Theta_{JA}$, the junction to ambient thermal resistance. The lower the resistance, the easier it is for the package to dissipate heat away from the chip to the external environment. For a packaged pASIC device, heat generated near the junction of the power chip causes the junction temperature to rise above the ambient temperature. The total thermal resistance is defined as

$$Theta_{JA} = (T_J - T_A)/P$$

where T_J is the silicon junction temperature, TA is the ambient still air, and P is the power dissipated by the chip.

The junction-to-case thermal resistance is defined as

$$Theta_{JC} = (T_J - T_C)/P$$

where T_J is the silicon junction temperature, T_C is the temperature of the case (package), and P is the power dissipated by the chip. The ambient temperature is the air temperature in the system. The worst case commercial condition is 70°C, while the typical condition inside a system in an office environment is 40°C.

The thermal resistance is also a function of the environment surrounding the package. The layout and the positioning of the board can influence the thermal resistance. Significant heat can also be removed by the leads. Forced air will also reduce the thermal resistance. An air flow of 200 Linear Feet per Minute can reduce the $Theta_{JA}$ by 25%. Higher air flow rates, 500LFM, can reduce the $Theta_{JA}$ by 35%. The thermal resistance can be lowered further by attaching a heat sink to the package.

The power dissipation of the part depends on the design, operating frequency, and supply voltage. The calculation is described in the Power Vs Operating Frequency section. Although this calculation is a very good approximation, the actual power consumption at the operating frequency should be measured. The maximum power dissipation for the pASIC is defined by

$$P_{max} = (T_{Jmax} - T_{Amax})/ Theta_{JA}$$

where T_{Jmax} = 150°C for plastic packages. For systems with forced air cooling, $Theta_{JA}$ can be approximated by subtracting 25 % from the still air value for 200LFM forced air, or by subtracting 35% for 500 LFM forced air.

If the power consumption of the part is higher than the maximum allowed, ceramic packages, heat sinks, liquid cooling, or forced air cooling should be used to increase the maximum allowed power.

Thermal Resistance, Junction-to-Ambient ($Theta_{JA}$ in °C/W)

Pkg. Type	PLCC			TQFP		PQFP	CQFP		CPGA		
Pin	44	68	84	100	144	208	100	160	68	84	144
QL8X12B	45	43		50					35		
QL12X16B		40	37	47			44			33	
QL16X24B			35	45	39		42	25			30
QL24X32B					36	31					

Thermal Resistance, Junction-to-Case ($Theta_{JC}$ in °C/W)

Pkg. Type	PLCC			TQFP		PQFP	CQFP		CPGA		
Pin	44	68	84	100	144	208	100	160	68	84	144
QL8X12B	11	11		16					8.9		
QL12X16B		10	9.2	11			4.6			8.3	
QL16X24B			7.4	9.7	8.0		3.9	5.3			7.7
QL24X32B					5.6	4.5					

PASIC 1 FAMILY
Packaging Specifications

HIGHLIGHTS

For plastic packages, QuickLogic offers surface-mount packaging in PLCC (Plastic Leaded Chip Carrier) and PQFP packages. The PQFP (Plastic Quad Flatpack) comes in two categories, TQFP (Thin Quad Flatpack) and PQFP. The TQFP package has a nominal thickness of only 1.4 mm and is ideally suited for designs with height restrictions. The PQFP is a metric quad flatpack with a nominal thickness of 3.37 mm. Plastic Ball Grid Array package is being developed for the high pin count products starting at the 225 pin count.

QuickLogic offers both through-hole and surface-mount hermetic packaging. The through-hole packages are ceramic PGA packages in 68, 84 and 144 pin counts. The surface mount hermetic packages are 100 CQFP with ceramic tie bar and 160 CQFP with a molded carrier ring. The tie bar and carrier ring will maintain lead integrity and protect the leads from bending and skewing.

HANDLING OF PLASTIC SURFACE MOUNT PACKAGING

The soldering process used to attach plastic surface mount devices to PC boards exposes the plastic component to high temperatures. The processes of concern are vapor phase, IR, hot air and wave soldering. The high solder temperature vaporizes moisture absorbed in the plastic, and creates a high pressure. The package may crack or delaminate to reduce the pressure. This is popularly known as the popcorn effect. The cracks or delamination can allow contaminants to be trapped at the die area, leading to early failure of these plastic components.

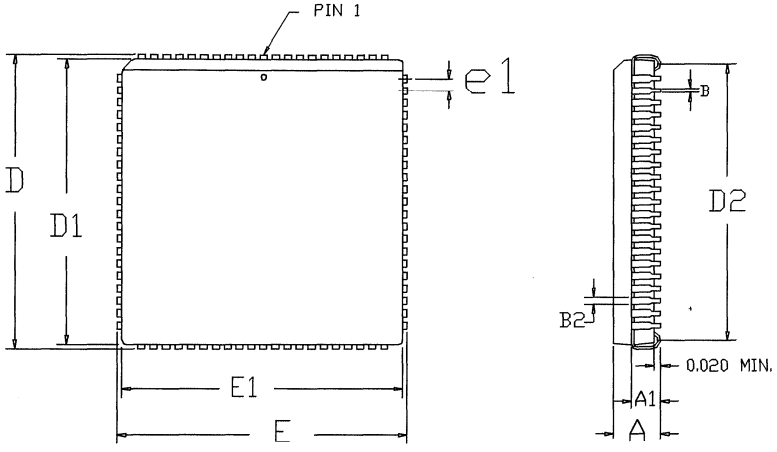
This effect can be reduced or eliminated by selecting the proper package design and materials. It can also be prevented by removing the moisture from the plastic component.

QuickLogic reliability tests show that the PLCCs are not susceptible to moisture induced cracking. This agrees with industry studies. The Plastic Quad Flat Pack components are more susceptible to moisture induced cracking. Therefore, QuickLogic recommends, in line with industry practice, that these components be baked prior to reflow soldering.

Dry bake and dry packing are available on request. Contact your QuickLogic representatives for lead-times and pricing.

QUICKLOGIC

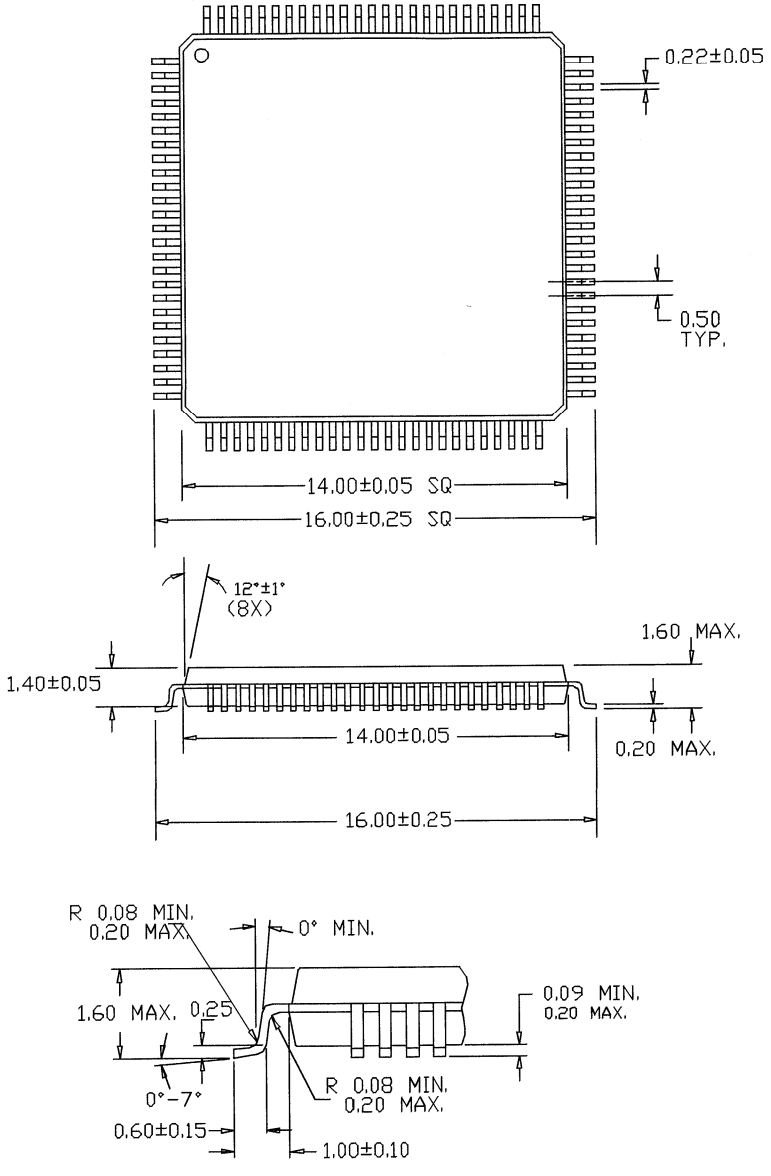
PL44 PL68, PL84
PLCC, Plastic Leaded Chip Carriers



Symbol	44-lead			68-lead			84-lead		
	min	nom	max	min	nom	max	min	nom	max
A	0.165	0.172	0.180	0.165	0.170	0.180	0.165	0.170	0.180
A1	0.100	0.101	0.120	0.095	0.099	0.118	0.095	0.099	0.118
B	0.013	0.013	0.021	0.013	0.013	0.021	0.013	0.013	0.021
B2	0.026	0.026	0.032	0.026	0.026	0.032	0.026	0.026	0.032
D	0.685	0.69	0.695	0.985	0.990	0.995	1.185	1.190	1.195
D1	0.650	0.652	0.656	0.950	0.952	0.955	1.150	1.152	1.156
D2	0.590	0.620	0.630	0.890	0.920	0.930	1.090	1.120	1.130
E	0.685	0.690	0.695	0.985	0.990	0.995	1.090	1.120	1.130
E1	0.650	0.652	0.656	0.950	0.952	0.955	1.150	1.152	1.156
e1 typ only	0.050			0.050			0.050		

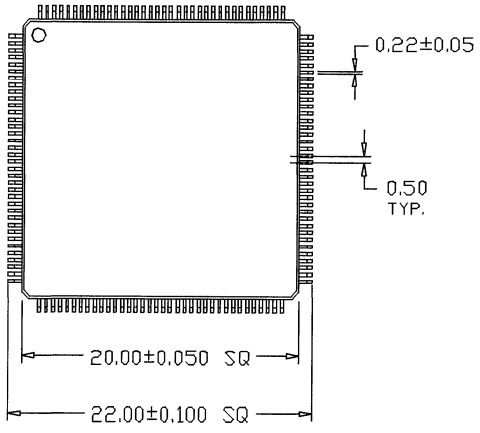
Notes:
1. All dimensions are in inches.
2. The inch is defined as 25.4 mm exactly.

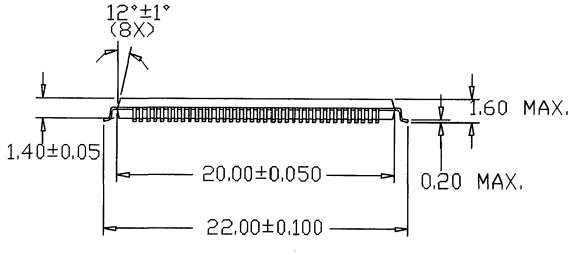
PF100
TQFP, Thin Plastic Quad Flat Pack

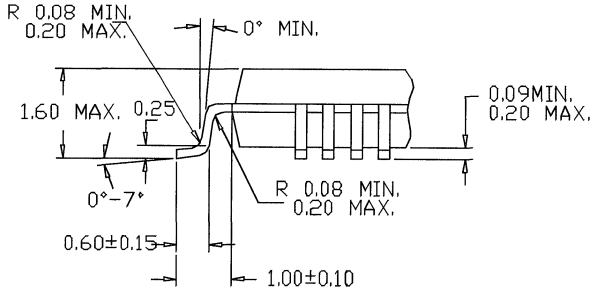


Notes:
1. All dimensions are in millimeters.

PF144
TQFP, Thin Plastic Quad Flat Pack

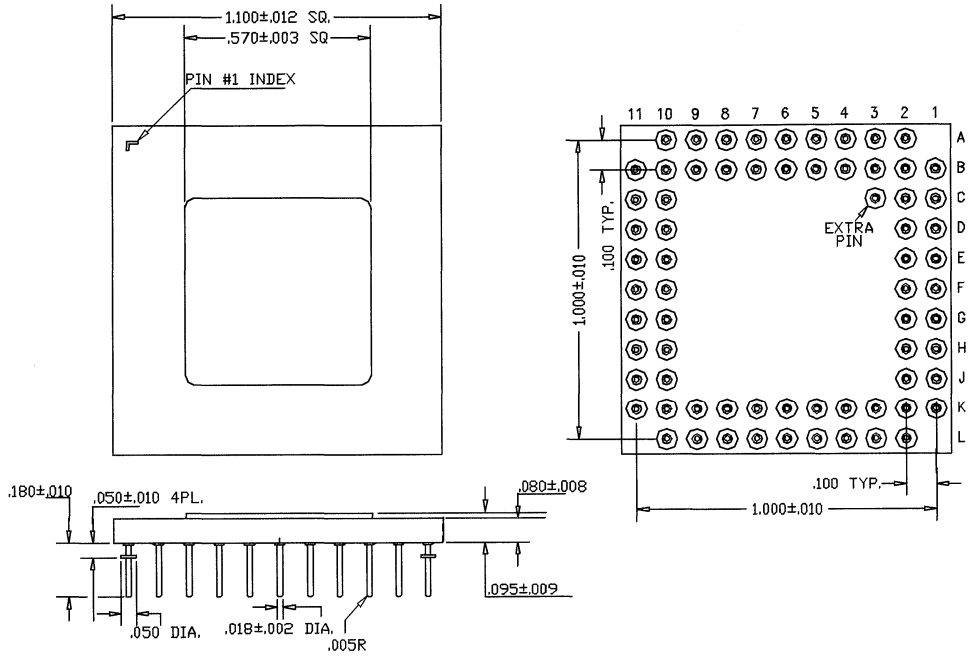






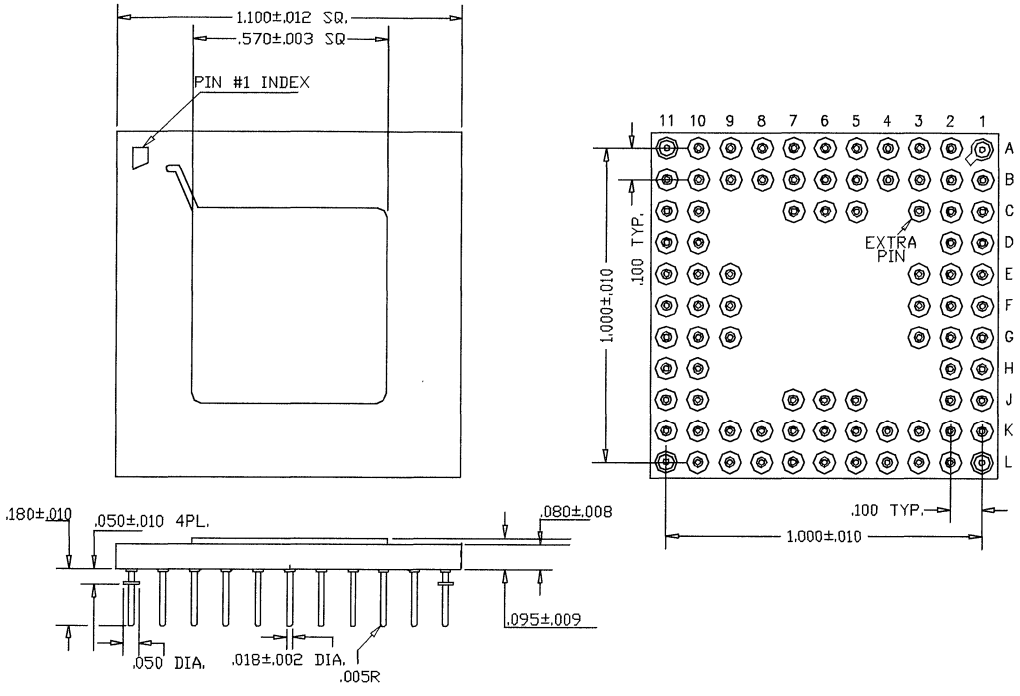
Notes:
1. All dimensions are in millimeters.

CG68
CPGA, Ceramic Pin Grid Array



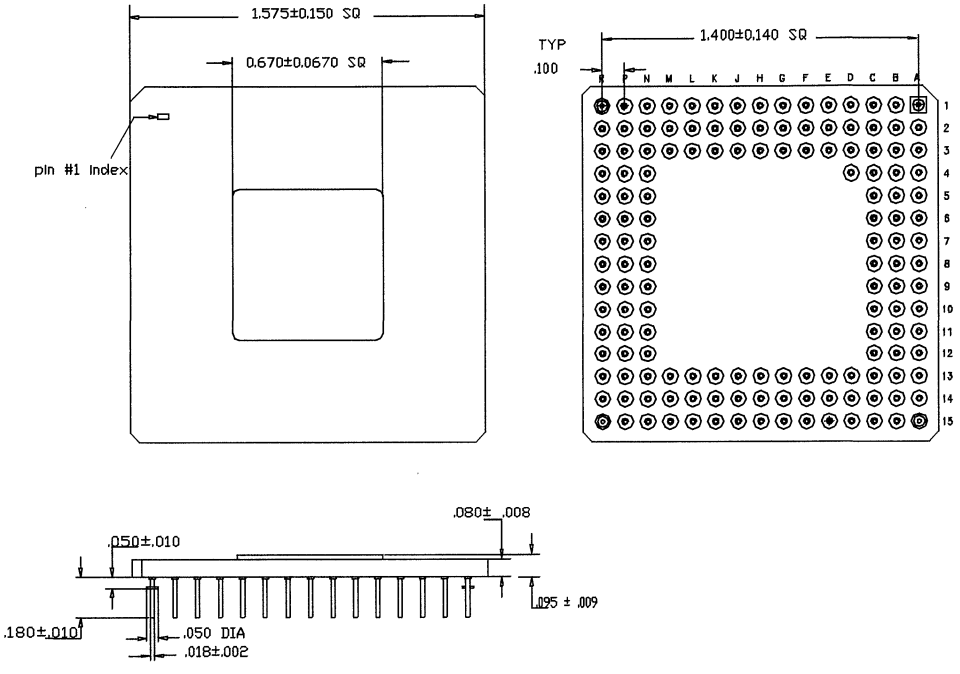
Notes:
1. All dimensions are in inches.

CG84
CPGA, Ceramic Pin Grid Array



Notes:
1. All dimensions are in inches.

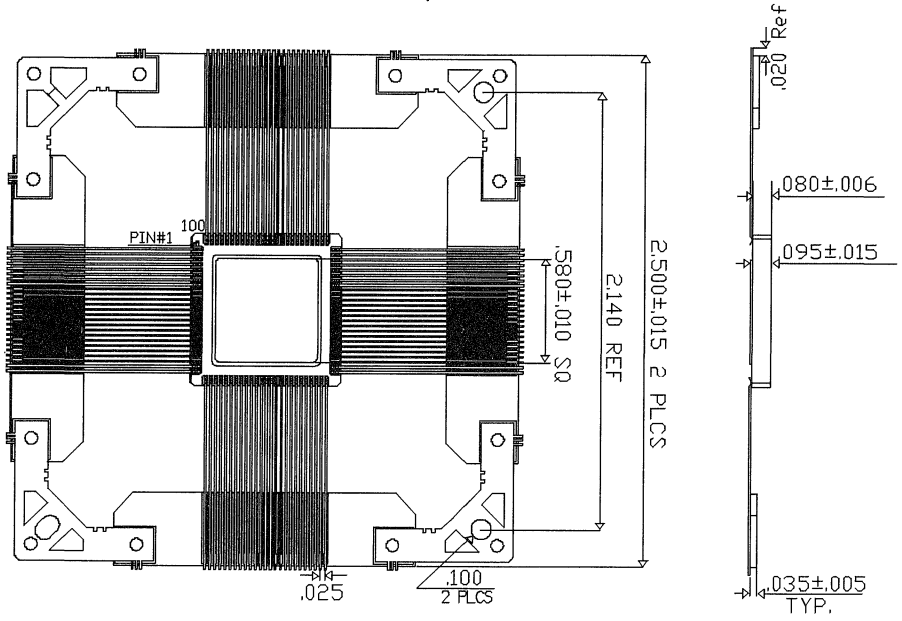
CG144
CPGA, Ceramic Pin Grid Array

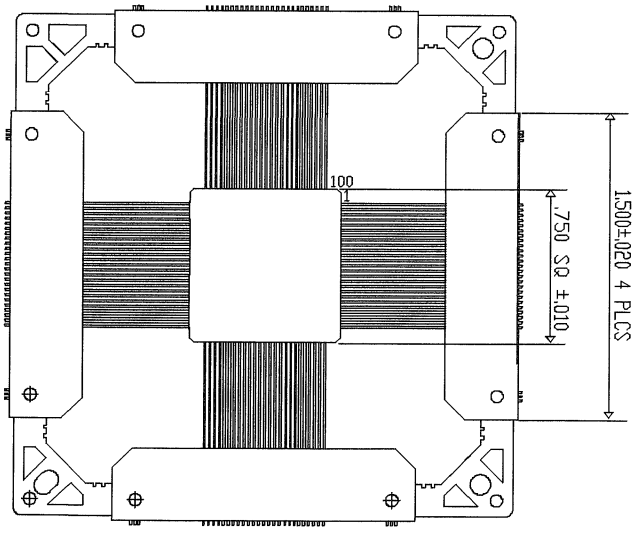


Notes:
1. All dimensions are in inches.

CF100
CQFP, Ceramic Quad Flat Pack with Ceramic Tie Bar

top view

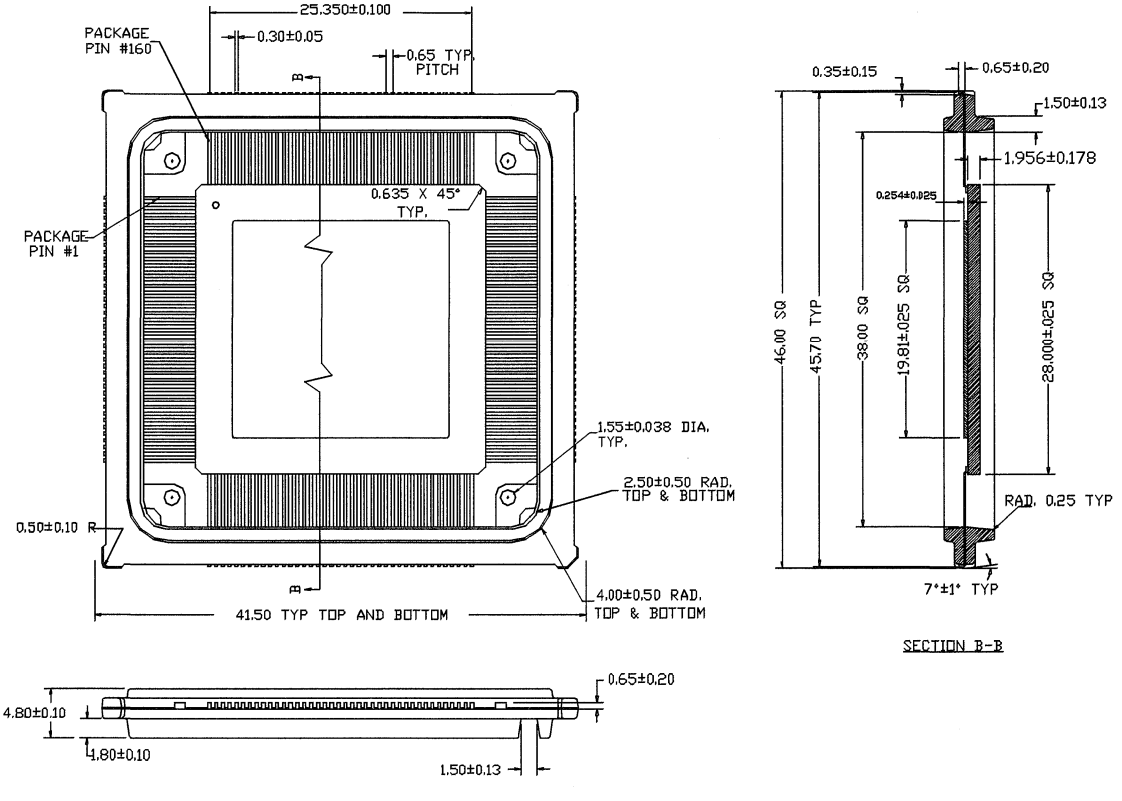




bottom view

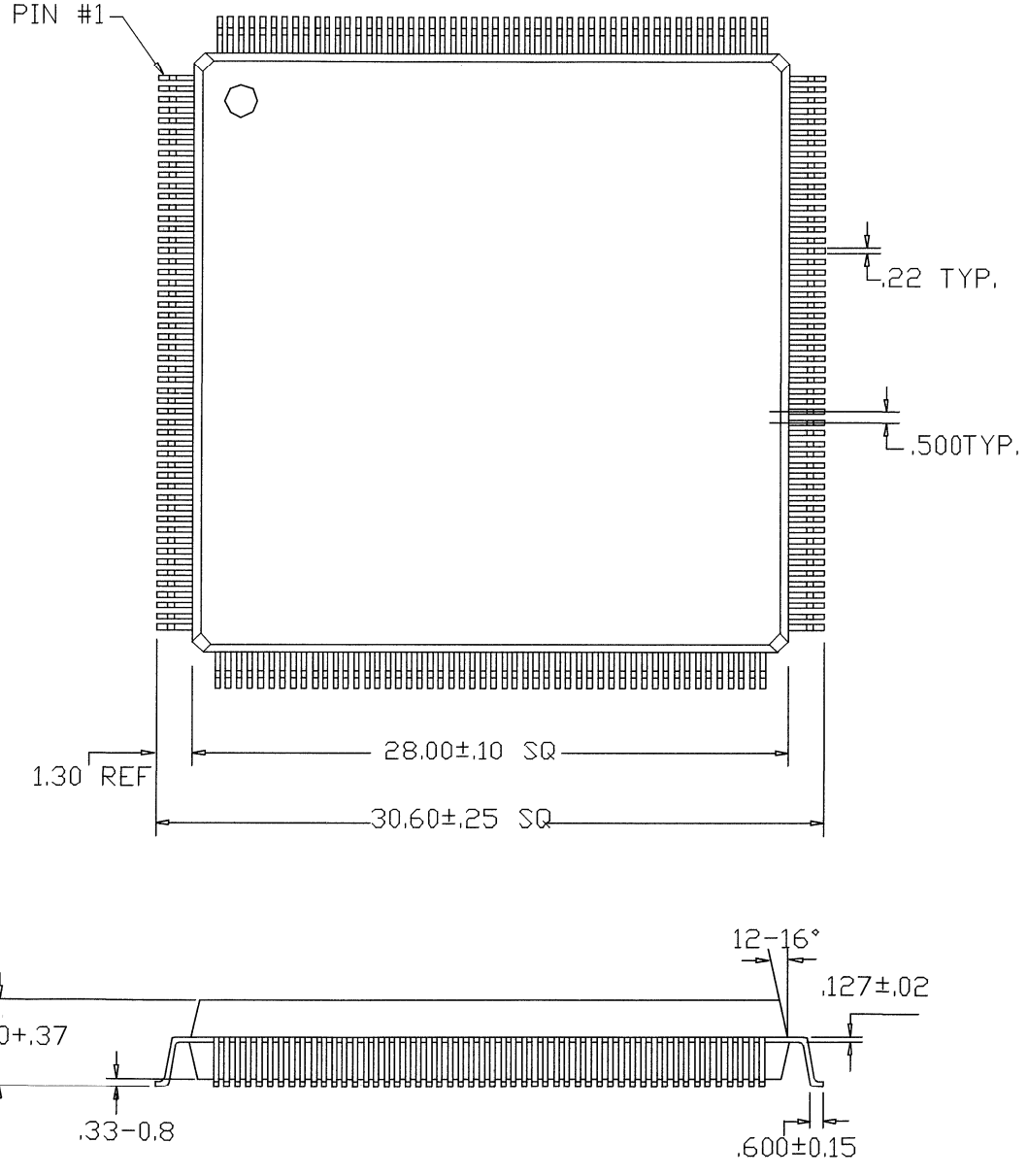
Notes:
1. All dimensions are in inches.

CF160
CQFP, Ceramic Quad Flat Pack with Molded Carrier Ring



Notes:
1. All dimensions are in millimeters.

PQ208
PQFP, Plastic Quad Flat Pack



Notes:
1. All dimensions are in millimeters.

QL8x12B

Function	68 pin PLCC	44 pin PLCC	100 pin TQFP	68 pin CPGA
GND	1	1	88	F10
I/O	2	N/C	89	F11
I/O	3	2	90	E10
I/O	4	3	91	E11
I/O	5	N/C	93	D10
I/O	6	4	94	D11
I/O	7	5	95	C10
I/O	8	N/C	97	C11
I/O	9	6	99	B11
I/O	10	7	2	B10
I/O	11	N/C	4	A10
I/O	12	N/C	6	B9
I/O	13	8	7	A9
I/O	14	N/C	8	B8
I/O	15	9	10	A8
I	16	10	11	B7
I/CLK	17	11	12	A7
VCC	18	12	13	B6
I	19	13	14	A6
I	20	14	15	B5
I/O	21	15	17	A5
I/O	22	N/C	18	B4
I/O	23	16	19	A4
I/O	24	N/C	20	B3
I/O	25	N/C	22	A3
I/O	26	17	24	A2
I/O	27	18	27	B2
I/O	28	19	29	B1
I/O	29	N/C	31	C2
I/O	30	20	32	C1
I/O	31	21	33	D2
I/O	32	N/C	34	D1
I/O	33	N/C	36	E2
I/O	34	22	37	E1

Function	68 pin PLCC	44 pin PLCC	100 pin TQFP	68 pin CPGA
GND	35	23	38	F2
I/O	36	N/C	39	F1
I/O	37	24	40	G2
I/O	38	25	41	G1
I/O	39	N/C	43	H2
I/O	40	26	44	H1
I/O	41	27	45	J2
I/O	42	N/C	47	J1
I/O	43	28	49	K1
I/O	44	29	52	K2
I/O	45	N/C	54	L2
I/O	46	30	56	K3
I/O	47	N/C	57	L3
I/O	48	31	58	K4
I/O	49	N/C	60	L4
I	50	32	61	K5
I/CLK	51	33	62	L5
VCC	52	34	63	K6
I	53	35	64	L6
I	54	36	65	K7
I/O	55	37	67	L7
I/O	56	38	68	K8
I/O	57	N/C	69	L8
I/O	58	N/C	70	K9
I/O	59	N/C	72	L9
I/O	60	39	74	L10
I/O	61	40	77	K10
I/O	62	41	79	K11
I/O	63	N/C	81	J10
I/O	64	42	82	J11
I/O	65	43	83	H10
I/O	66	N/C	84	H11
I/O	67	N/C	86	G10
I/O	68	44	87	G11

QUICKLOGIC

QL12x16B

Function	84 pin CPGA	68 pin PLCC	84 pin PLCC	100 pin TQFP
I/O				1
I/O	B10	10	12	2
I/O	B9	11	13	3
I/O	A10		14	4
I/O	A9	12	15	5
I/O	B8		16	6
I/O	A8	13	17	7
I/O	A7	14	18	8
GND	C7		19	9
I/O	A6	15	20	10
I	B7	16	21	11
I/CLK	C6	17	22	12
VCC		18		13
I	B6	19	23	14
I	B5	20	24	15
VCC	C5		25	16
I/O	A5	21	26	17
I/O	A4	22	27	18
I/O	B4	23	28	19
I/O				20
I/O	A3	24	29	21
I/O	A2		30	22
I/O	B3	25	31	23
I/O	A1	26	32	24
I/O				25
I/O				26
I/O	B2	27	33	27
I/O	C2	28	34	28
I/O	B1	29	35	29
I/O	C1	30	36	30
I/O				31
I/O	D2	31	37	32
I/O	D1	32	38	33
I/O	E1		39	34
GND	E3		40	35
I/O	E2	33	41	36
I/O	F1	34	42	37
GND		35		38
I/O	F2	36	43	39
I/O	F3	37	44	40
I/O	G1	38	45	41
VCC	G3		46	42
I/O	G2	39	47	43
I/O	H1		48	44
I/O	H2	40	49	45
I/O	J1	41	50	46
I/O	K1		51	47
I/O	J2	42	52	48
I/O	L1	43	53	49
I/O				50

Function	84 pin CPGA	68 pin PLCC	84 pin PLCC	100 pin TQFP
I/O				51
I/O	K2	44	54	52
I/O	K3	45	55	53
I/O	L2		56	54
I/O	L3	46	57	55
I/O	K4		58	56
I/O	L4	47	59	57
I/O	L5	48	60	58
GND	J5		61	59
I/O	L6	49	62	60
I	K5	50	63	61
I/CLK	J6	51	64	62
VCC		52		63
I	K6	53	65	64
I	K7	54	66	65
VCC	J7		67	66
I/O	L7	55	68	67
I/O	L8	56	69	68
I/O	K8	57	70	69
I/O				70
I/O	L9	58	71	71
I/O	L10		72	72
I/O	K9	59	73	73
I/O	L11	60	74	74
I/O				75
I/O				76
I/O	K10	61	75	77
I/O	J10	62	76	78
I/O	K11	63	77	79
I/O	J11	64	78	80
I/O				81
I/O	H10	65	79	82
I/O	H11	66	80	83
I/O	G11		81	84
GND	G9		82	85
I/O	G10	67	83	86
I/O	F11	68	84	87
GND		1		88
I/O	F10	2	1	89
I/O	F9	3	2	90
I/O	E11	4	3	91
VCC	E9		4	92
I/O	E10	5	5	93
I/O	D11		6	94
I/O	D10	6	7	95
I/O	C11	7	8	96
I/O	B11		9	97
I/O	C10	8	10	98
I/O	A11	9	11	99
I/O				100

QL16X24B

	CPGA	PLCC	T/CQFP	TQFP	CQFP		CPGA	PLCC	T/CQFP	TQFP	CQFP
Fnct	144	84	100	144	160	Fnct	144	84	100	144	160
I/O	A2			1	1	I/O	B15			37	41
I/O	B3	12	1	2	2	I/O	C14	33	26	38	42
I/O	C4		2	3	3	I/O	D13	34	27	39	43
I/O	A3	13	3	4	4	I/O	C15	35	28	40	44
I/O	B4			5	5	I/O	D14			41	45
nc					6	nc					46
I/O	A4	14	4	6	7	VCC	E13			42	47
VCC	C3			7	8	I/O	D15	36	29	43	48
I/O	B5	15	5	8	9	I/O	E14		30	44	49
I/O	A5	16	6	9	10	I/O	E15	37	31	45	50
I/O	C6			10	11	I/O	F13	38	32	46	51
I/O	B6	17	7	11	12	I/O	F14			47	52
I/O	A6			12	13	I/O	F15	39	33	48	53
I/O	A7	18	8	13	14	I/O	G15		34	49	54
nc					15	nc					55
I/O	B7			14	16	GND	C13	40	35	50	56
GND	C5	19	9	15	17	I/O	G14	41	36	51	57
I/O	A8	20	10	16	18	I/O	H15	42	37	52	58
I	B8	21	11	17	19	I/O	H14			53	59
I/CLK	C8	22	12	18	20	GND	G13		38	54	60
VCC	C7		13	19	21	I/O	H13	43	39	55	61
I	A9	23	14	20	22	I/O	J15	44	40	56	62
I	B9	24	15	21	23	I/O	J14	45	41	57	63
VCC	C11	25	16	22	24	VCC	J13	46	42	58	64
I/O	A10	26	17	23	25	I/O	K15	47	43	59	65
nc					26	nc					66
I/O	A11			24	27	I/O	L15			60	67
I/O	B10	27	18	25	28	I/O	K14	48	44	61	68
I/O	A12			26	29	I/O	M15		45	62	69
I/O	B11	28	19	27	30	I/O	L14	49	46	63	70
I/O	C10		20	28	31	I/O	K13	50	47	64	71
I/O	A13	29	21	29	32	I/O	N15			65	72
GND	C9			30	33	GND	L13			66	73
I/O	B12			31	34	I/O	M14	51	48	67	74
nc					35	nc					75
I/O	A14	30	22	32	36	I/O	P15			68	76
I/O	B13			33	37	I/O	N14	52	49	69	77
I/O	C12	31	23	34	38	I/O	M13			70	78
I/O	A15		24	35	39	I/O	R15	53	50	71	79
I/O	B14	32	25	36	40	nc	P14			72	80

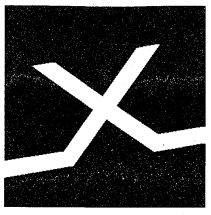
	CPGA	PLCC	T/CQFP	TQFP	CQFP		CPGA	PLCC	T/CQFP	TQFP	CQFP
Fnct	144	84	100	144	160	Fnct	144	84	100	144	160
I/O	R14			73	81	I/O	P1			109	121
I/O	P13	54	51	74	82	I/O	N2	75	76	110	122
I/O	N12			75	83	I/O	M3	76	77	111	123
I/O	R13	55	52	76	84	I/O	N1			112	124
I/O	P12			77	85	I/O	M2	77	78	113	125
nc					86	nc					126
I/O	R12	56	53	78	87	VCC	L3			114	127
VCC	N13			79	88	I/O	M1	78	79	115	128
I/O	P11	57	54	80	89	I/O	L2		80	116	129
I/O	R11	58	55	81	90	I/O	L1	79	81	117	130
I/O	N10		56	82	91	I/O	K3	80	82	118	131
I/O	P10	59	57	83	92	I/O	K2			119	132
I/O	R10			84	93	I/O	K1	81	83	120	133
I/O	R9	60	58	85	94	I/O	J1		84	121	134
nc					95	nc					135
I/O	P9			86	96	GND	N3	82	85	122	136
GND	N11	61	59	87	97	I/O	J2	83	86	123	137
I/O	R8	62	60	88	98	I/O	H1			124	138
I	P8	63	61	89	99	I/O	H2	84	87	125	139
I/CLK	N8	64	62	90	100	GND	J3		88	126	140
VCC	N9		63	91	101	I/O	H3	1	89	127	141
I	R7	65	64	92	102	I/O	G1	2	90	128	142
I	P7	66	65	93	103	I/O	G2	3	91	129	143
VCC	N5	67	66	94	104	VCC	G3	4	92	130	144
I/O	R6		67	95	105	I/O	F1	5	93	131	145
nc					106	nc					146
I/O	R5	68	68	96	107	I/O	E1			132	147
I/O	P6			97	108	I/O	F2	6	94	133	148
I/O	R4	69	69	98	109	I/O	D1			134	149
I/O	P5	70	70	99	110	I/O	E2	7	95	135	150
I/O	N6		71	100	111	I/O	F3	8	96	136	151
I/O	R3	71	72	101	112	I/O	C1			137	152
GND	N7			102	113	GND	E3			138	153
I/O	P4			103	114	I/O	D2	9	97	139	154
nc					115	nc					155
I/O	R2	72	73	104	116	I/O	B1		98	140	156
I/O	P3			105	117	I/O	C2	10	99	141	157
I/O	N4	73	74	106	118	I/O	D3			142	158
I/O	R1			107	119	I/O	A1	11	100	143	159
I/O	P2	74	75	108	120	nc	B2			144	160

QL24x32B

FUNC	208 PQFP	144 TQFP	FUNC	208 PQFP	144 TQFP	FUNC	208 PQFP	144 TQFP	FUNC	208 PQFP	144 TQFP
I/O	1	NC	I/O	53	37	I/O	105	NC	I/O	157	109
I/O	2	1	I/O	54	NC	I/O	106	73	I/O	158	NC
I/O	3	2	I/O	55	38	I/O	107	NC	I/O	159	110
I/O	4	3	I/O	56	39	I/O	108	74	I/O	160	111
I/O	5	NC	I/O	57	40	I/O	109	75	I/O	161	112
I/O	6	4	I/O	58	NC	I/O	110	76	I/O	162	113
I/O	7	5	GND	59	NC	I/O	111	NC	GND	163	NC
I/O	8	NC	I/O	60	41	I/O	112	77	I/O	164	NC
I/O	9	6	VCC	61	42	I/O	113	78	VCC	165	114
VCC	10	7	I/O	62	43	VCC	114	79	I/O	166	115
I/O	11	NC	I/O	63	NC	I/O	115	80	I/O	167	116
GND	12	NC	I/O	64	44	GND	116	NC	I/O	168	NC
I/O	13	8	I/O	65	45	I/O	117	NC	I/O	169	117
I/O	14	NC	I/O	66	NC	I/O	118	81	I/O	170	118
I/O	15	9	I/O	67	46	I/O	119	NC	I/O	171	119
I/O	16	NC	I/O	68	47	I/O	120	82	I/O	172	120
I/O	17	10	I/O	69	48	I/O	121	NC	I/O	173	NC
I/O	18	11	I/O	70	NC	I/O	122	83	I/O	174	NC
I/O	19	12	I/O	71	49	I/O	123	84	I/O	175	121
I/O	20	13	I/O	72	NC	I/O	124	85	I/O	176	NC
I/O	21	NC	GND	73	50	I/O	125	NC	GND	177	122
I/O	22	14	I/O	74	51	I/O	126	86	I/O	178	123
GND	23	15	I/O	75	52	GND	127	87	I/O	179	124
I/O	24	16	I/O	76	53	I/O	128	88	I/O	180	NC
I	25	17	I/O	77	NC	I	129	89	I/O	181	125
I/CLK	26	18	GND	78	54	I/CLK	130	90	GND	182	126
VCC	27	19	I/O	79	55	VCC	131	91	I/O	183	127
I	28	20	I/O	80	56	I	132	92	I/O	184	128
I	29	21	I/O	81	57	I	133	93	I/O	185	129
VCC	30	22	I/O	82	NC	VCC	134	94	I/O	186	NC
I/O	31	NC	VCC	83	58	I/O	135	95	VCC	187	130
I/O	32	23	I/O	84	59	I/O	136	NC	I/O	188	131
I/O	33	NC	I/O	85	60	I/O	137	96	I/O	189	132
I/O	34	24	I/O	86	61	I/O	138	NC	I/O	190	NC
I/O	35	NC	I/O	87	NC	I/O	139	97	I/O	191	133
I/O	36	25	I/O	88	62	I/O	140	NC	I/O	192	134
I/O	37	26	I/O	89	63	I/O	141	98	I/O	193	NC
I/O	38	27	I/O	90	NC	I/O	142	99	I/O	194	135
I/O	39	28	I/O	91	NC	I/O	143	NC	I/O	195	136
I/O	40	NC	I/O	92	64	I/O	144	100	I/O	196	nc
VCC	41	NC	I/O	93	NC	VCC	145	NC	I/O	197	137
I/O	42	29	I/O	94	65	I/O	146	101	I/O	198	NC
GND	43	30	GND	95	66	GND	147	102	GND	199	138
I/O	44	31	I/O	96	67	I/O	148	103	I/O	200	139
I/O	45	NC	VCC	97	NC	I/O	149	104	VCC	201	NC
I/O	46	32	I/O	98	NC	I/O	150	105	I/O	202	140
I/O	47	33	I/O	99	68	I/O	151	106	I/O	203	NC
I/O	48	34	I/O	100	69	I/O	152	NC	I/O	204	141
I/O	49	NC	I/O	101	NC	I/O	153	107	I/O	205	142
I/O	50	35	I/O	102	70	I/O	154	NC	I/O	206	NC
I/O	51	36	I/O	103	71	I/O	155	108	I/O	207	143
I/O	52	NC	I/O	104	72	I/O	156	NC	I/O	208	144

Paper & Article Reprints Summary

QUICKLOGIC



CONDUCTING FILAMENT OF THE PROGRAMMED METAL ELECTRODE AMORPHOUS SILICON ANTIFUSE

by Richard J. Wong and Kathryn E. Gordon

Presented at the
International Electron Devices Meeting
December 1993

Conducting Filament of the Programmed Metal Electrode Amorphous Silicon Antifuse

Kathryn E. Gordon and Richard J. Wong

QuickLogic Corporation
2933 Bunker Hill Lane, Santa Clara, CA 95054
Phone: (408) 987-2000

Abstract

Antifuses in PROM and FPGA applications have used silicon and/or polycrystalline silicon electrodes (1,2,3). Metal electrode antifuses have the lowest resistance and lowest capacitance among programmable interconnect structures (4). The ViaLink, a metal electrode amorphous silicon antifuse, has been used as a programmable interconnect device for a FPGA (5). This paper describes for the first time, the composition, structure, electrical characteristics, and temperature dependence of the conducting filament in the programmed TiW electrode amorphous silicon antifuse.

Introduction

The ViaLink, a metal electrode amorphous silicon antifuse, was developed for a high speed Field Programmable Gate Array (FPGA) application. A metal electrode amorphous silicon antifuse is the best choice to meet the low capacitance and low resistance requirements for high speed applications. High gate densities push the need for manufacturable and scalable technologies. The thickness of the programmable antifuse layer must be tightly controlled since it affects the antifuse programming voltage. For equivalent programming voltages, an amorphous silicon layer is more than an order of magnitude thicker than an oxide/nitride/oxide (ONO) or nitride/oxide (NO) composite dielectric layer (6). The thicker amorphous silicon layer is easier to manufacture and easier to scale than the thinner layers required for ONO or NO dielectric antifuses.

Antifuse Technology

The antifuse is added into a standard 1.0 um CMOS process. The lower TiW (10 wt% Ti) electrode is deposited and patterned on top of the intermetal dielectric. A 1.2 um via etched through 3000A of oxide defines the antifuse area. PECVD amorphous silicon is deposited at 400°C into the vias. The PECVD process creates cusping and thinning of the amorphous silicon at the via side wall. Oxide is deposited and etched back to create spacers at the side wall. The spacers isolate the thinnest and most variable amorphous silicon area from the top electrode for improved Vp, programming voltage, control. Figure 1 shows the Vp as a function of the deposited silicon thickness. A 10% change in deposited thickness produces a 5% change in Vp. This allows for a tight Vp control with a large process window. Figure 2 shows the Vp distribution for 15 lots.

The oxide spacers force the programmed filament to form away from the via side wall, so that the filament is surrounded by silicon for better thermal stress symmetry. Without spacers, the filament would form near the via side wall, with part of its surface contacting the oxide and the remainder contacting the silicon. The asymmetric thermal stresses around this filament are likely to cause failures. Table 1 summarizes the thermal constants of the materials in and around the filament.

TABLE I
Thermal Constants of Materials in the Antifuse Structure

Material	Thermal Conductivity (W/cm°C)	Linear Coefficient of Expansion (ppm/°C)
Tungsten	1.82	4.5
Silicon	1.41	2.5
Tungsten Silicide	0.42	6 - 8
Silicon Dioxide	0.014	0.2

Programmed Antifuse Characteristics

The antifuse is programmed by applying a voltage to the top electrode while the bottom electrode is grounded. The conducting programmed filament is formed by moving electrode material in the electron flow direction into the amorphous silicon. Figure 3 is a SEM photograph of an antifuse programmed with a negative pulse. Metal has moved from the top electrode to form the filament. Figure 4 is a SEM photograph of an antifuse programmed with a positive pulse. The metal has moved from the bottom electrode to form the filament. Figure 5 is a cross sectional TEM photograph of a positive pulse programmed antifuse. The filament is 0.14 microns wide near the bottom electrode and 0.09 microns wide at the top. Preliminary analysis indicates that the filament consists of polycrystalline tungsten silicide. Near the bottom electrode the filament contains higher amounts of Ti in the form of titanium silicide or Ti-W silicide. Figure 6 is a cross sectional TEM showing that the amorphous silicon adjacent to the filament has recrystalized into polycrystalline silicon. The polycrystalline silicon extends 0.2 microns beyond the filament. The programming mechanism generates enough heat to move the electrode material into the amorphous silicon, react the metal to form a silicide, and recrystalize the adjacent silicon. Experiments are underway to further understand this programming mechanism.

The programmed resistance, Rf, is a constant, Cr, divided by the programming current, Rf=Cr/Ipp (4). The constant can be reduced by using multiple pulse programming to anneal the formed filament. Figure 7 shows that for the same total programming time, a single pulse programming produces a Cr = 1.1, while multiple pulse programming produces a Cr = 0.7. The I-V characteristic of the programmed antifuse is shown in Figure 8. The resistance characteristics are linear and reproducible at currents below its programming current. Figure 9 shows the resistance of various programmed antifuses measured over temperature. The programmed resistance has no measurable temperature dependence from -175°C to 300°C.

Reliability

The programmed and unprogrammed antifuse reliability have been characterized (7). The programmed antifuse must remain in its low resistance state during operation of the circuit. The antifuse is used to connect CMOS drivers to interconnect networks. The antifuse experiences only AC current and no DC current. The number of cycles for read disturb, a 10% increase in the programmed antifuse resistance, is exponentially dependent on the peak AC current density (7). Figure 10 shows the mean number of cycles to disturb for 25°C and 250°C are similar. The lack of an observable temperature dependence indicates that the activation energy of the disturb mechanism is zero, or that the self heating in the antifuse is high enough to make the 225°C ambient delta insignificant.

Summary

A metal electrode amorphous silicon antifuse has been presented. The programmed resistance has a narrow distribution and a low Cr constant. The IV characteristics of the programmed antifuse is linear and has no temperature dependence. SEM and TEM photographs show that the conducting filament is formed by moving the TiW electrode material into the amorphous silicon and reacting it with the silicon. The reliability of the programmed antifuse is exponentially proportional to the peak AC current density through the filament. The read disturb mechanism of the programmed antifuse does not have an ambient temperature dependence.

Acknowledgements

We wish to thank Dr. David Su at Philips Material Analysis Group, Professor Moshe Eizenberg at Techion - Israel Institute of Technology and Jay Glanville at FAST Labs for the TEM preparation, analysis and discussions. We thank ICE for SEM reproduction. We wish to thank Doug Hamilton at Charles Evans & Assc. for excellent SEM work. We also wish to thank Ron Neale for his lively discussions.

References

(1) E Hamdy et al, "Dielectric based antifuse for logic and memory ICs," *IEDM* 1988, pp.786-789.

(2) N Sato, T Nawata, and K Wada, "A new programmable cell utilizing insulator breakdown," *IEDM* 1985 pp.639-642.

(3) V Malhotra, J Mahan, D Ellsworth, "An electrothermal model of memory switching in vertical polycrystalline silicon structures," *IEEE Transactions on Electron Devices*, Vol. 35, No. 9, pp.1514-1523, 1988.

(4) C Hu, "Interconnect devices for field programmable gate array," *IEDM* 1992, pp.591-594.

(5) J Birkner et al, " A very high-speed field programmable gate array using metal-to-metal antifuse programmable elements," *Microelectronics Journal*, Vol. 23, No.7, pp.561-568.

(6) D Liu, K Chen, H Tigelaar, J Paterson, S Chen, "Scaled dielectric antifuse structure for FPGA applications," *IEEE Electron Device Letters*, Vol. 12, No. 4, pp.151-153, 1991.

(7) R Wong and K Gordon, "Evaluating the reliability of the QuickLogic antifuse," *Electronic Engineering*, June 1992 pp.49-56.

FIGURE 1
The programming voltage as a function of the deposited amorphous silicon thickness.

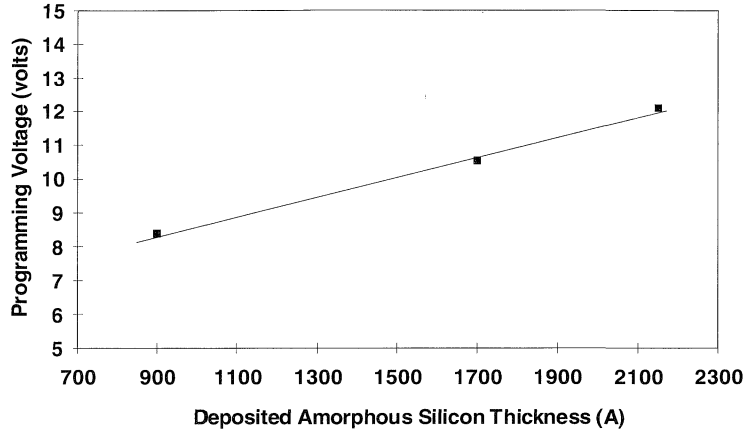
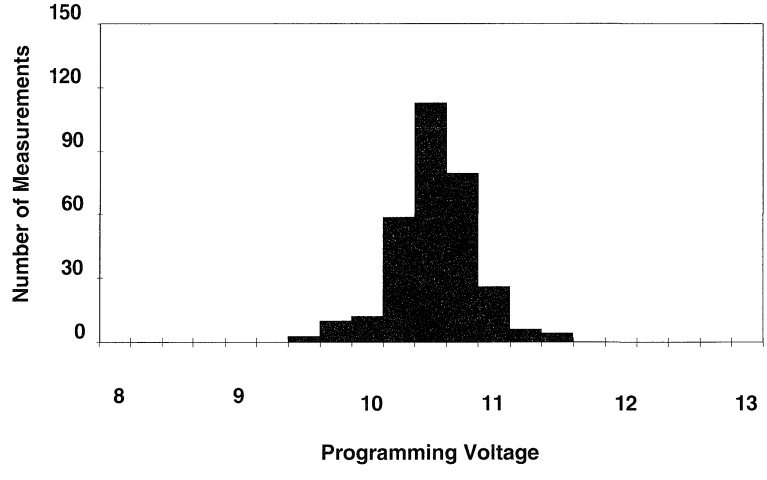


FIGURE 2
The programming voltage distribution for 15 lots.



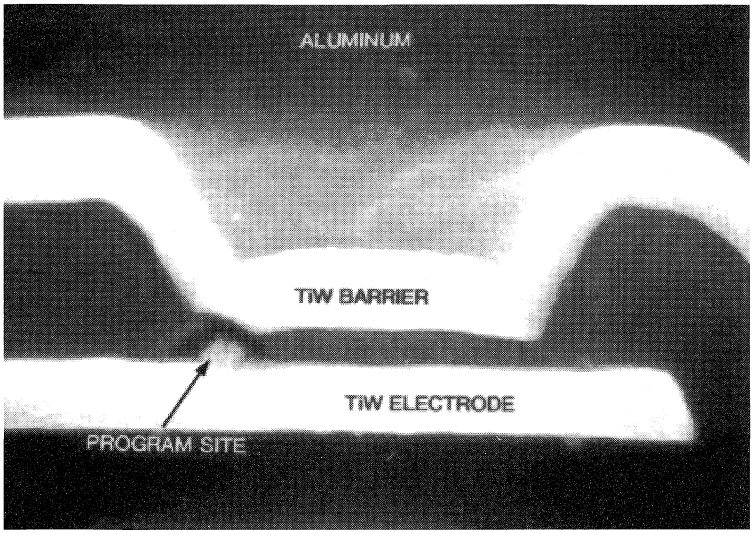


FIGURE 3
SEM photograph of an antifuse programmed with a negative pulse. The top electrode material has moved into the amorphous silicon.

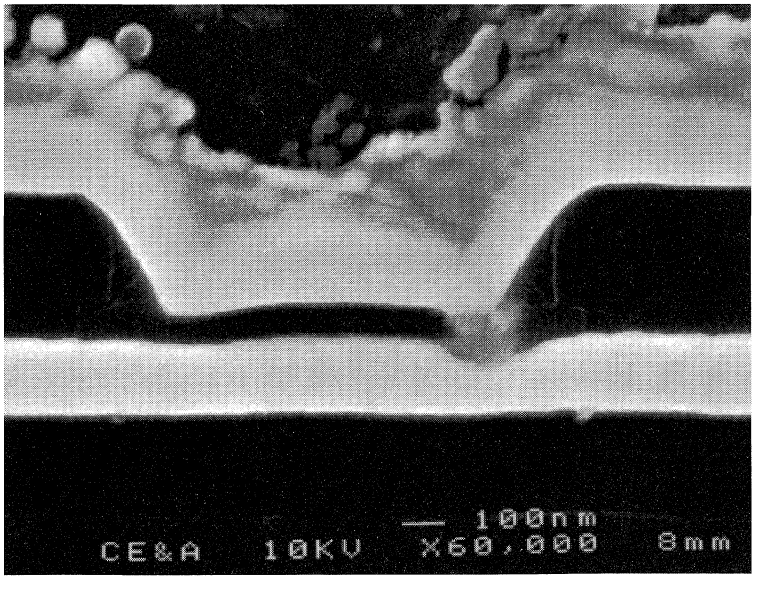


FIGURE 4
SEM photograph of an antifuse programmed with a positive pulse. The bottom electrode material has moved into the amorphous silicon.

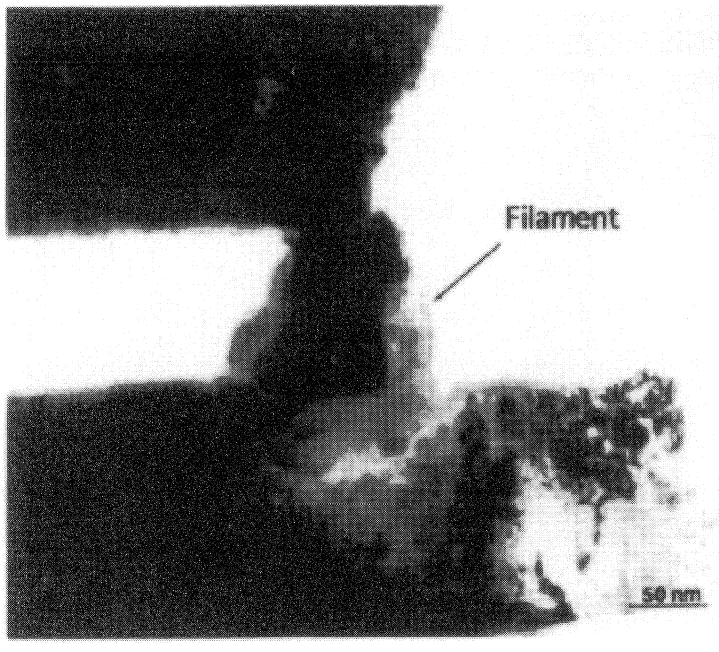


FIGURE 5
A dark field cross sectional TEM photograph of positive pulse programmed antifuse.

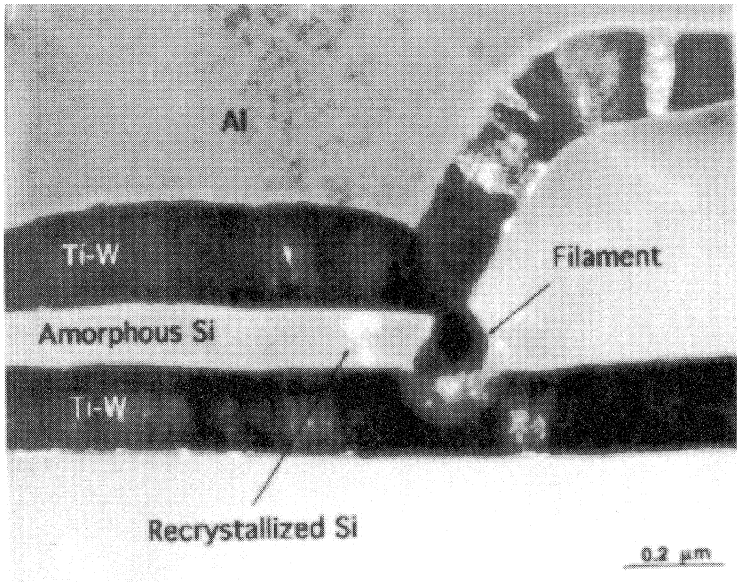


FIGURE 6
A dark field cross sectional TEM of the programmed antifuse showing the amorphous silicon around the filament has recrystalized into polysilicon.

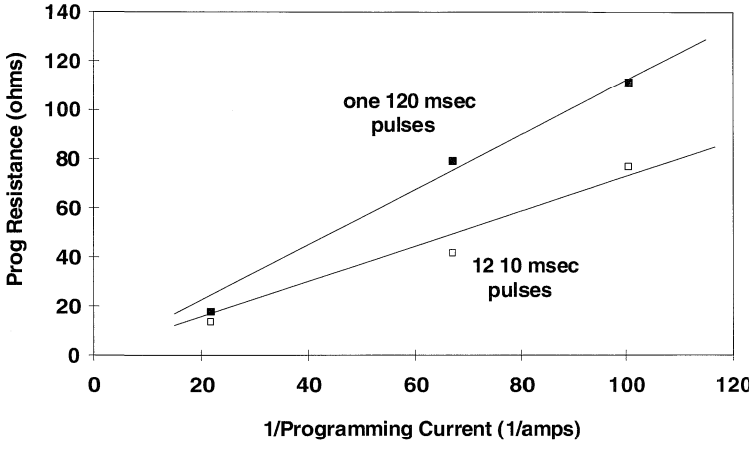


FIGURE 7
Programming constants, C=Rf x Ipp, Cr=1.1 for programming with a single 120 millisecond pulse, and Cr=0.7 for programming with twelve 10 millisecond pulses.

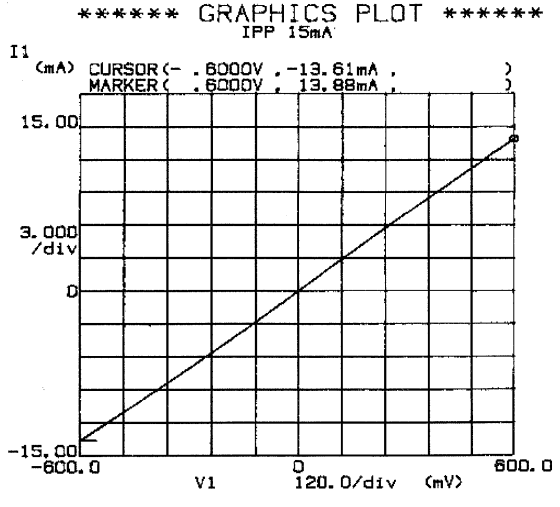


FIGURE 8
I-V characteristics an antifuse programmed with 15mA.

FIGURE 9
The resistances of various programmed antifuses over temperature.

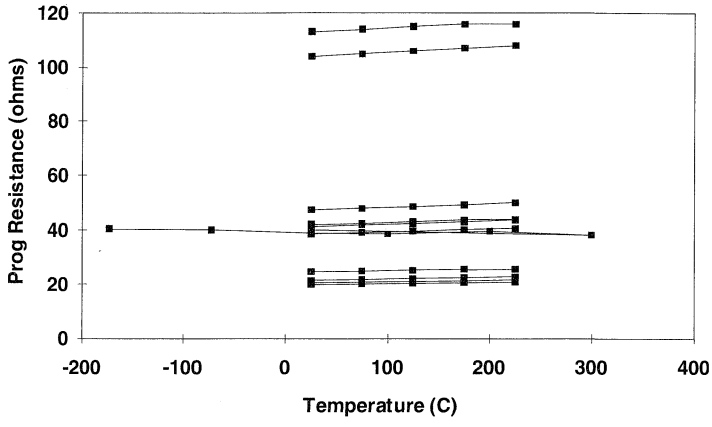
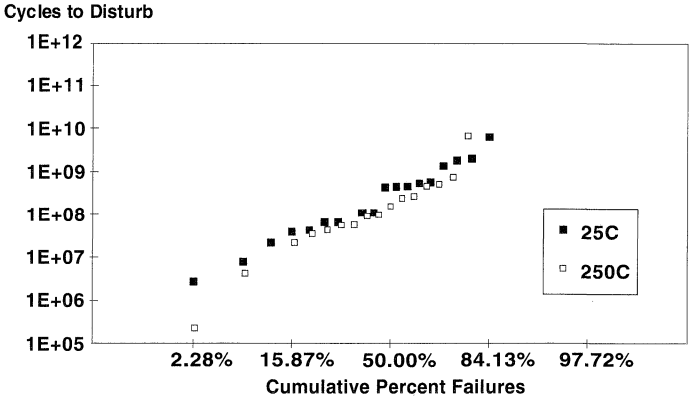


FIGURE 10
The number of cycles for read disturb, a 10% increase in the programmed resistance, for 25C and 250C. The stress current densities for the two groups were identical.



RELIABILITY MECHANISM OF THE UNPROGRAMMED AMORPHOUS SILICON ANTIFUSE

by Richard J. Wong and Kathryn E. Gordon

Presented at the
International Reliability and Physics Symposium
April 1994

QUICKLOGIC

Abstract

The electrical properties of the unprogrammed metal electrode amorphous silicon antifuse have been characterized. A model is proposed for the reliability mechanism. During a voltage stress, the leakage current through the antifuse creates localized states which increase the leakage current from 1 nA to tens of nA. The effect eventually saturates and can be annealed out. The amorphous silicon antifuse does not have a catastrophic failure mechanism such as the Time Dependent Dielectric Breakdown found in dielectric antifuses. The increase in the amorphous silicon antifuse leakage current is predictable and reproducible. The increase does not effect the reliability of the Field Programmable Gate array which uses this antifuse as a programmable interconnect. The FPGA product has been stressed for 200 million equivalent device hours with a 7.0 volt static burn in with no failures.

Introduction

Field Programmable Gate Arrays use various programmable interconnect elements, such as SRAMs, EPROMs, EEPROMs, or antifuses, to connect logic blocks. For high speed applications, the on state must have a low resistance and the off state must have a low capacitance. For high density, the programmable element must be small. The metal electrode antifuse has lowest on-resistance, lowest off-capacitance and smallest size among these technologies [1,2].

Antifuses have used ONO [3], NO [4], oxide [5], polysilicon [6], and amorphous silicon [7] as the programmable material. In the unprogrammed state, dielectric antifuses such as NO, ONO and oxide are characterized by a reliability mechanism described as Time Dependent Dielectric Breakdown [8]. The time for the dielectric antifuse to breakdown is a function of the applied electric field. The product fails when any one of the hundreds of thousand dielectric antifuses on a product ruptures.

The ViaLink, a metal electrode amorphous silicon antifuse, is the programmable interconnect device in the pASIC, a FPGA developed by QuickLogic [9]. This antifuse uses amorphous silicon instead of a dielectric as the programmable material. This paper presents the characterization data and reliability model of the amorphous silicon antifuse. The electrical characteristics of the unprogrammed amorphous silicon antifuse in three different states will be described. The initial characteristics will be compared with the characteristics after a voltage stress and after a voltage stress and anneal. Finally, a conduction model and reliability mechanism will be proposed.

Amorphous Silicon Antifuse Structure

The antifuse is added into a standard 1.0 micron double level metal CMOS process after first metal. The process temperatures for the antifuse module are less than 400°C, which allows this module to be inserted after aluminum metalization.

Figure 1 shows a cross sectional drawing of the antifuse structure. The lower TiW electrode is deposited and patterned on top of the intermetal dielectric. A 3000Å oxide is deposited. A 1.2 um via defines the antifuse area. The low aspect ratio via allows PECVD amorphous silicon to be deposited on the via bottom and sidewall with good uniformity. Oxide is deposited and etched back to create spacers at the sidewall. The spacers isolate the thinnest and most variable area of the amorphous silicon from the top TiW electrode to improve the control of the programming voltage, which is a function of the amorphous silicon thickness.

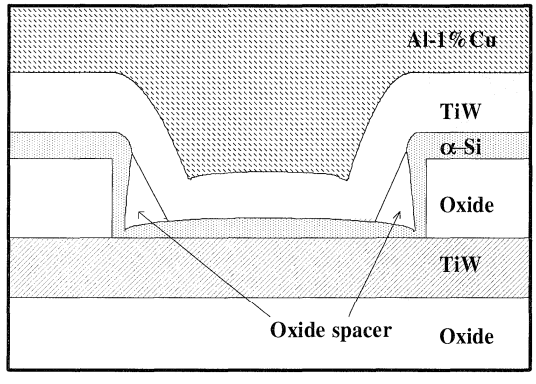


Figure 1. Cross section drawing of the metal electrode amorphous silicon antifuse.

Electrical Characteristics

Figure 2 shows the unprogrammed antifuse initial and post stress I-V characteristics. The bottom electrode is grounded and the top electrode is swept with a positive and negative voltage. The initial antifuse leakage current has an exponential dependence on the voltage which is symmetric from 5.5 volts. The initial leakage current at 5.5 volts is 1 nA.

The leakage current through the antifuse can increase after a high electric field stress. The post stress I-V characteristics

in Figure 2 are after a constant voltage stress of 9.0 volts for 240,000 seconds. The entire I-V characteristic has increased and remained symmetric. The post stress leakage current has a lower exponential dependence on the voltage. The leakage current after this stress is 100 nA at 5.5 volts.

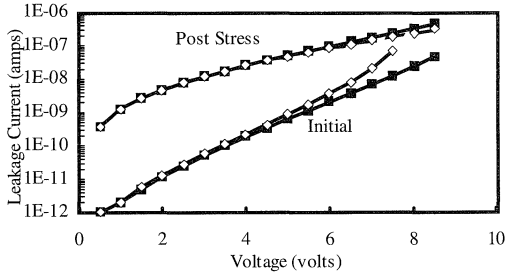


Figure 2. Unprogrammed amorphous silicon antifuse I-V characteristics at 20°C. Filledsymbols are + voltage and open symbols are - voltage.

Figure 3 shows that the leakage current at 5.5 volts increases until it saturates. A 100 nA leakage current at 5.5 volts was chosen as an arbitrary characterization point. The time scale was chosen such that the leakage current at 5.5 volts was 100 nA after stressing for one time unit. The leakage current increases linearly with time until it begins to saturate.

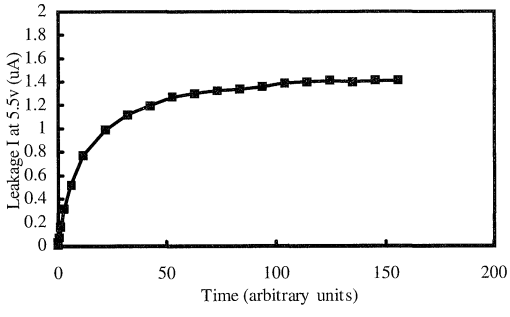


Figure 3. The saturation of the leakage current at 5.5 volts during a high voltage stress.

Since the leakage current increases linearly with time from the initial 1 nA to about 500 nA, the time calculation for leakage currents between 1 and 500 nA is straight forward. The leakage current saturated at 1.4 uA for this accelerated stress. This data shows that the amorphous silicon antifuse does not fail catastrophically, even when stressed well beyond the time for 100 nA of leakage at 5.5 volts.

Figure 4 shows the activation energy of the initial and post stress leakage current for the antifuse. During the stress the leakage mechanism changes. The activation energy in the initial state is 0.4 eV at low biases, 1 to 4 volts, and decreases to 0.25 eV at higher bias. After stress, the activation energy is 0.1 eV and is less dependent on the bias. The lower activation minimizes the increase in the post stress high temperature leakage. The initial leakage current is 1 nA at 20°C and 50 nA at 125°C. When the 20°C post stress leakage current is 100 nA, the 125°C leakage current is only 300 nA.

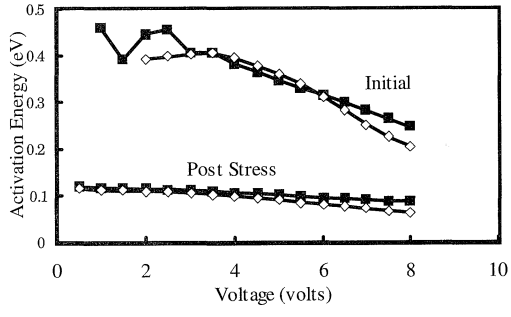


Figure 4. The leakage current activation energy as a function of the voltage. Filled symbols are + voltage and open symbols are - voltage.

Since the initial leakage current at 5.5 volts is about 1 nA, the rate of the initial leakage current increase can be examined as the time for the leakage current to reach 100 nA as a function of the stress electric field [10]. Figure 5 shows that the increase in leakage current for the individual antifuse is almost the same as the average increase of an array of one thousand antifuses. The leakage current effect of 1000 unprogrammed antifuses can be approximated as the sum of the leakage currents of 1000 average antifuses. The distribution within an array is such that the array behaves close to the average.

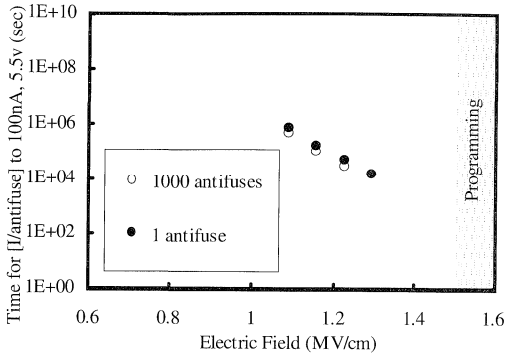


Figure 5. The accelerated electric field stress on the individual antifuse and a 1000 array of antifuses. The data has been normalized to leakage per antifuse.

Figure 6 shows the electric field acceleration data for 6 production lots from two fabs. The degradation effect is controllable and reproducible. Each individual lot shows the same trend with electric field. The variation in the data is most likely due to the measurement accuracy of the amorphous silicon thickness.

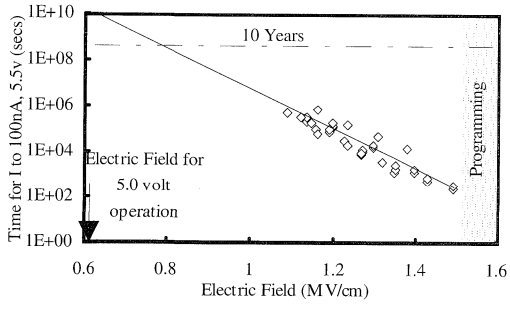


Figure 6. Accelerated stress on individual antifuses from 6 lots from 2 wafer fabs.

Figure 7 shows that the time for the antifuse leakage current to reach 100 nA at 5.5 volts as a function of the stress electric field is not temperature dependent. The stress temperatures were varied but the leakage currents were measured at 20°C. This model for degradation acceleration as a function of the electric field allows for an accurate prediction of the antifuse degradation in the product, since unprogrammed antifuses in the product are stressed with an electric field.

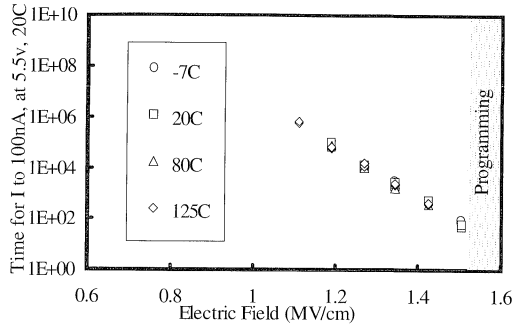


Figure 7. The accelerated stress on the antifuse at several temperatures. The stresses were done at various temperatures but the leakage current was measured at 20°C.

The data in Figures 5, 6, and 7 cannot be used to extrapolate to higher electric fields. Near 1.5 MV/cm, the amorphous silicon antifuse programs with a different mechanism which occurs in less than 1 nanosecond and results in a permanent low resistance link. The programmed I-V characteristics are linear and do not have a temperature dependence [11].

The lack of an observable temperature dependence in Figure 7 occurs when the antifuse degradation is modeled as a function of the electric field. The electric field stress at 125°C produces an initial stress current which is 37 times greater than the initial stress current at -7°C. If the antifuse degradation is modeled as a function of the stress current, the degradation would be reduced at higher temperatures.

To determine if an annealing mechanism exists, two antifuses were stressed at 20°C to the same leakage level at 5.5 volts. The antifuses were baked without bias and the leakage currents were measured. Figure 8 shows the post stress leakage current mechanism being annealed out at 80°C and 125°C. The activation energy of the anneal is approximately 0.8 eV.

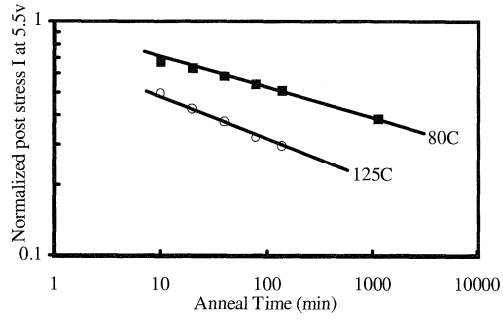


Figure 8. Annealing of the post stress leakage mechanism at 80°C and 125°C. The leakage current at 5.5 volts was normalized for the post stress values.

A stressed antifuse with a leakage current of 100 nA at 5.5 volts can be annealed to near its initial state. Figure 9 shows that the I-V characteristics of an antifuse after a stress and anneal is similar to its initial I-V characteristics. The initial leakage current at 5.5 volts was 0.3 nA and the post stress leakage current was 100 nA. The 225°C bake for 24 hours dropped the leakage current at 5.5 volts to 1 nA.

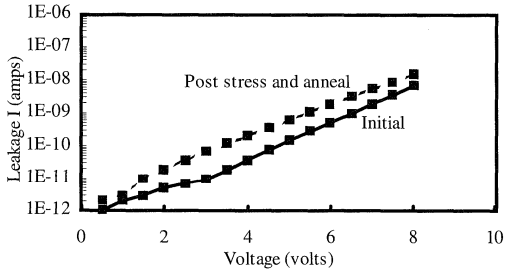


Figure 9. The initial I-V characteristics of an antifuse and the I-V characteristics after stressing and annealing. The stress condition had raised the leakage current at 5.5 volts to 100 nA. The anneal was 225°C for 24 hours.

The post stress anneal also returns the activation energy of the leakage current to near its initial value. Figure 10 shows that the activation energy after stress and anneal is similar to the initial activation energy. The stress and anneal conditions are the same as in Figure 9.

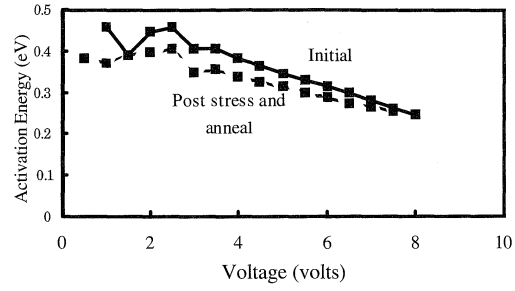


Figure 10. The post stress and anneal leakage current activation energy as a function of the voltage. The 225°C 24 hr anneal returns the activation energies to near the initial values.

Conduction Model

The conduction mechanism in the amorphous silicon is different from that in single crystalline silicon. The forbidden gap of the electron energy states is essentially closed by localized states as shown in Figure 11 [12]. The extended states are similar to those in crystalline silicon and exist on a macroscopic scale, but do not have sharply defined regions at the conduction and valence band edges. Instead, the density of the states falls off rapidly. The tail regions of the extended states are in the band gap, but the density is very low.

The localized states, such as dangling bonds, exist on the inter-atomic scale. These localized states have energy levels throughout the forbidden energy gap, which essentially closes the gap. The charge carriers in the gap, hop between localized states with very low mobility. In the conduction and valence bands, the carriers move with much higher mobility in the extended states. Thus, the band gap in the amorphous silicon is due to the low mobility and not the absence of states within the gap. Conduction in amorphous silicon can be phonon-assisted excitation of carriers across the mobility gap or carrier hopping from one localized state to the next.

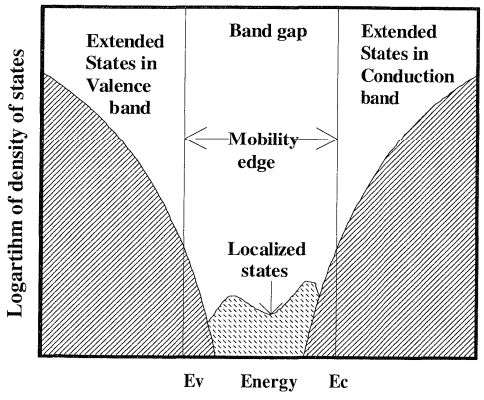


Figure 11. The density of states for amorphous silicon as a function of energy.

The amorphous silicon antifuse has been annealed with hydrogen during the fab process. The silane decomposition during the PECVD amorphous silicon deposition releases hydrogen, and the nitride passivation acts as a hydrogen source. The hydrogen passivates the dangling silicon bonds, leaving the initial amorphous silicon with a very low density of localized states [13]. Thus, the initial conduction mechanism is predominately phonon-assisted excitation of carriers across the gap. The free carriers move in the extended states [14]. The observed activation energy of the initial conduction is 0.4 eV.

During conduction, the carriers can be accelerated by the high electric field. These high energy carriers can break weak silicon-silicon and silicon-hydrogen bonds [15]. The broken bonds become localized states within the mobility gap. As more localized states are created, the hopping conduction mechanism becomes predominate [16]. Carriers hop from one localized state to the next. Carriers are still accelerated by the electric field, and more bonds continue to be broken. The hopping conduction increases as the localized state density increases. The process saturates when most of the weak bonds have been broken. The observed effect is that during a high field stress, the leakage current increases linearly with time until it saturates. The activation energy for the post stress conduction is 0.1 eV.

The dangling bonds in amorphous silicon can be annealed at elevated temperatures [17]. When most of the dangling bonds have formed silicon-silicon or silicon-hydrogen bonds, the density of localized states decreases to near the initial values. The I-V characteristic and conduction mechanism, as indicated by the activation energy, also return to the initial state.

The bake reversible effect in the amorphous silicon antifuse is different from the reversible conductivity effect observed by Staebler and Wronski [18]. Both effects eventually saturate and can be annealed out. The effects differ in the cause and result of the degradation. The antifuse is stressed under high electric fields while the Staebler Wronski Effect occurs during illumination. The most significant difference is that the antifuse degradation increases conductivity while the SWE reduces the conductivity.

Unprogrammed Antifuse Reliability

The unprogrammed antifuse must remain in its high resistance state for the lifetime of the product, typically 5.0 volts for 10 years across all temperatures. The leakage current through each antifuse contributes only to the standby Icc of the pASIC product and does not effect the functionality or AC performance of the product. Since the effect on the product is the sum of the individual antifuses, the pASIC is designed for the behavior of the average amorphous silicon antifuse. This is an advantage over dielectric based antifuses, where several sigmas of margin must be designed in to avoid the first catastrophic TDDB event.

Since the leakage current changes linearly with time, and the rate of change is a function of the stress electric field, the product can be designed to meet the standby Icc specification. The design engineer determines the amount of the standby Icc which is allowed for the antifuses and divides that by the number of biased unprogrammed antifuses. This determines the maximum leakage through an antifuse. Using Figure 6 and the linear time dependence of leakage, the maximum electric field for ten years can be calculated. Table 1 shows the calculations for various pASIC products. The operating electric fields have at least 20% margin over the maximum allowed fields.

Product Reliability

To further ensure the reliability of the pASIC, 608 units from various fabs and products have been burned in at 7.0 volts, static, and room temperature. The static condition stresses the same antifuses for the entire burn in. The 7.0 volt stress

accelerates the test by applying a high field across the antifuse. The room temperature condition avoids the annealing at high temperatures. There have been no failures of any pASIC after 200 million equivalent device hours of this test. All parts remained within all specification limits. Some pASIC devices with thousands of amorphous silicon antifuses have shown a small, less than 1 mA, increase in the standby Icc. This behavior was predicted by the reliability model.

The 7.0 volt static burn in test is in addition to the standard High Temperature Operating Life test where the devices are dynamically stressed at 125°C, 5.5 volts. Over 1800 parts have completed 170 million equivalent device hours of HTOL without an antifuse failure.

Conclusion

The electrical characteristics and reliability of the unprogrammed amorphous silicon antifuse have been characterized. The leakage current through the antifuse will increase during a high electric field stress. The degradation is accelerated by electric field and does not have an observable temperature dependence. The degradation is controllable and predictable within an array and on wafers from different fabs. The degradation eventually saturates and does not lead to a catastrophic failure of the amorphous silicon antifuse. The leakage current can be returned to the initial value by annealing.

A model has been presented for the reversible conduction mechanism. The leakage current in the antifuse creates localized states in the amorphous silicon by breaking weak silicon bonds. As the localized state density increases, the dominate conduction process changes from carrier conduction in the extended states, to carrier hopping among localized states. When most of the weak bonds are broken, the degradation process saturates. The localized states are annealed at high temperatures and the conduction returns to the initial mechanism.

The product has been designed to accommodate the small leakage current through the antifuse. There have been no antifuse failures in 200 million equivalent device hours of the 7.0 volt static stress of the pASIC devices.

REFERENCES

[1] C Hu, "Interconnect devices for field programmable gate array," IEDM 1992, pp591-594.

[2] S Chiang et al, "Antifuse comparison for field programmable gate arrays," IEDM 1992, pp611-614.

[3] E Hamdy et al, "Dielectric based antifuse for logic and memory IC's," IEDM 1988, pp786-789.

[4] D Liu, K Chen, H Tigelaar, J Paterson, and S Chen, "Scaled dielectric antifuse structure for FPGA applications," IEEE Elect Dev Let, Vol 12, No. 4, Apr 1991, pp151-153.

[5] N Sato, T Nawata, and K Wada, "A new programmable cell utilizing insulator breakdown," IEDM 1985, pp639-642.

[6] V Malhotra, J Mahan, and D Ellsworth, "An electrothermal model of memory switching in vertical polycrystalline structures," IEEE Trans on Elec Dev, Vol 35, No. 9, Sept 1988, pp1514-1523.

[7] Y Shacham-Diamand et al, "A novel ion implanted amorphous silicon programmable element," IEDM 1987 pp194-197.

[8] S Chiang et al, "Oxide-nitride-oxide antifuse reliability," IRPS, 1990, pp186-192.

[9] J Birkner et al, "A very-high-speed field programmable gate array using metal to metal programmable elements," Microelectronics Journal, Vol 23, No.7, pp561-568.

[10] R Wong and K Gordon, "Evaluating the reliability of the QuickLogic antifuse," Electronic Engineering, June 1992, pp49-56.

[11] K Gordon and R Wong, "Conducting filament of the programmed metal electrode amorphous silicon antifuse" IEDM 1993 pp27-30.

[12] A Madan, "Amorphous silicon: from promise to practice," IEEE Spectrum, Sept 1986, pp38-43.

[13] MH Brodsky, "The effect of hydrogen and other additives on the electronic properties of amorphous silicon," Fundamental Physics of Amorphous Semiconductors, Proc of the Kyoto Summer Inst, Sept 1980, pp56-71.

[14] P Chaudhuri, "Conductivity of doped GD and sputtered a-Si:H, temperature dependence," Properties of Amorphous Silicon, New York, INSPEC, 1989, pp 200-201.

[15] R Street and M Hack, "Defect creation by forward bias in amorphous silicon P-I-N diodes," Mat.Res.Soc.Symp Proc. 1991, pp135-140.

[16] H Overhof and P Thomas, Electronic transport in hydrogenated amorphous semiconductors, Berlin, Springer-Verlag, 1989, pp29-35.

[17] ZE Smith, "Defect equilibrium in undoped a-Si:H," Properties of Amorphous Silicon, New York, INSPEC, 1989, pp 518-521.

[18] D Staebler and C Wronski, "Reversible conductivity changes in discharge-produced amorphous-Si," Applied Physics Letters, Vol 31, No.4, Aug 1977, pp292-294.

Table 1. Maximum electric field for 10 year operation

Product	Bias unprogrammed antifuses	Standby Icc for antifuse leakage	Max antifuse leakage at 5.5v, 125°C	Max antifuse leakage at 5.5v, 20°C	Max electric field for 10 yr life	Operating electric field
QL8x12	20,000	8mA	400nA	140nA	0.82MV/cm	0.61MV/cm
QL12x16	40,000	8mA	200nA	71nA	0.79MV/cm	0.61MV/cm
QL16x24	80,000	8mA	100nA	36nA	0.76MV/cm	0.61MV/cm
QL24x32	160,000	8mA	50nA	18nA	0.73MV/cm	0.61MV/cm

The PREP Benchmarks for FPGAs

BACKGROUND

The Programmable Electronics Performance Corporation (PREP) is a cooperative effort between a group of established companies in the Field Programmable Gate Array (FPGA) and Electronic Programmable Logic Device (EPLD) market. The benchmarks evaluate both capacity and performance, and represent a variety of circuit types from highly sequential to highly combinational. The PREP benchmarks will assist users in evaluating different programmable devices in different application areas.

METHODOLOGY

The discussion below is intended to serve as an introduction only. The methodology and rules for the benchmarks are spelled out completely in the PREP documentation. PREP's address is provided at the end of this paper.

Step-and-Repeat

One of PREP's fundamental goals is to exercise each device under conditions of realistic utilization, in order to fully involve the interconnect in the testing. A 16-bit counter may behave much differently in a sparsely utilized device than in densely utilized device. Ideally, each device would be tested under fully utilized conditions. Unfortunately, a benchmark large enough to fill device Y might not fit into device Z at all.

A technique called step-and-repeat evolved in order to measure both capacity and performance under conditions of realistic utilization. Each benchmark specifies a functional unit. For purposes of illustration, consider the hypothetical benchmark shown in Figure 1. The functional unit is specified as a black box implementing a 16-bit register with clear, load enable, clock, 16-bit input and 16-bit output. The implementation of this black box is left to the vendor. Each benchmark includes a description, truth table, and functional vectors that fully specify the behavior of the black box from its inputs and outputs.

This functional unit is replicated until the device is full. Functional units are connected by global signals and daisy-chained signals. Global signals attach to all functional units while daisy-chained signals attach between functional units. As shown in Figure 2, the CLR, LOAD, and CLK are global signals, while the input and output busses are daisy-chained signals.

QUICKLOGIC

Capacity Measurements

Capacity measurements include a count of the number of functional units implemented and the percentage of the logic resources utilized by the functional units. The latter measurement is architecture-specific; it is the percentage of the total logic cells/blocks/modules/macrocells used by the functional units.

Performance Measurements

Performance measurements are slightly more complicated and include two sets of measurements. The internal Fmax is a measurement of the on-chip performance between functional units. Using a path analysis tool, Fmax is determined for each of the functional units. The worst, best, and mean of these Fmax measurements are reported. As the functional units have identical netlists, the spread between the worst and best provides some feel for variations introduced by placement and routing.

The external Fmax is a measurement of the chip-to-chip performance. The outputs of the last functional unit is driven off-chip and connected to the inputs of the first functional unit. This measurement thereby includes a set of output and input pad drivers.

BENCHMARKS

The PREP Benchmark Suite #1 contains nine benchmarks. These benchmarks represent a cross-section of common applications for programmable logic. Some of the benchmarks favor FPGA-style architectures while others favor EPLD- or PAL-style architectures. Some of the benchmarks are highly sequential and others are highly combinational. Some of the benchmarks are fairly large and others are fairly small; all of the benchmarks, however, should be able to be implemented in a "1000 gate" device. The nine descriptions below are intended only to provide a flavor for the benchmarks. Detailed specifications are available from PREP Corporation.

Datapath

The **Datapath** benchmark contains 4-to-1 multiplexers, an 8-bit register, and an 8-bit shift register. While the functionality is simple, the routing is difficult. The global signals include a 24-bit bus that, as discussed earlier, must be routed to each of the functional units. There is little logic between flip-flops, and fanout is low.

Timer/Counter

The **Timer/Counter** benchmark contains 2-to-1 multiplexers, two 8-bit registers, an 8-bit loadable up-counter, and an 8-bit equality comparator. Such circuitry is common in microprocessor support logic such as DMA controllers. There is a moderate amount of logic between flip-flops, and fanout is moderate.

Small State Machine

The **Small State Machine** benchmark is a Mealy state machine with 8 states, 12 transitiors, 8 input bits and 8 registered output bits. While the state machine implements no immediately apparent function, its style is indicative of many PAL applications. Wide fan-in circuitry is required to decode the

input bits and the state bits. There is a moderate amount of logic between flip-flops and fanout is moderate.

Large State Machine

The **Large State Machine** benchmark is a Moore state machine with 16 states, 40 transitions, 8 input bits, and 8 output bits. The state machine implements a function less immediately apparent and considerably more intricate than the small state machine. Complex circuitry is required to decode the input bits and the transition conditions. There is a moderate amount of logic between flip-flops and fanout is high.

Arithmetic

The **Arithmetic** benchmark is a multiplier-accumulator. It contains a 4-by-4 multiplier and an 8-bit accumulator. Arithmetic circuits present different challenges from the other benchmarks as they are highly combinational. There is a large amount of logic between flip-flops and fanout is moderate.

Accumulator

The **Accumulator** benchmark is a 16-bit accumulator. There is a large amount of logic between flip-flops and fanout is moderate.

16-bit Counter

The **16-bit Counter** benchmark is a 16-bit loadable up-counter with count enable. There is little logic between flip-flops and fanout is low.

16-bit Pre-Scaled Counter

The **16-bit Pre-Scaled Counter** benchmark is a 16-bit loadable counter with count enable. This benchmark allows 'LS163-style pre-scaled design where N-cycles are allowed after a load before counting commences. Such a counter is commonly used as a frequency divider. There is little logic between flip-flops and fanout is low.

Memory Map

The **Memory Map** benchmark implements an address decoder with 16 input bits. The outputs decode 8 nonoverlapping address ranges of different shapes and sizes. Such circuitry is indicative of many PAL applications. There is a moderate amount of logic between flip-flops and fanout is low.

REPORTING

Participating vendors must report results from all nine of the above benchmarks. Results using fully automatic place and route must be reported; optionally, results using manual placement or routing may be reported. PREP results include the capacity measurements and performance measurements discussed above, Optionally, two versions of each benchmark may be reported, one optimized for capacity and the other optimized for performance. Each version must report both capacity and performance measurements so the tradeoffs can analyzed. The astute observer will recognize that as many as four sets of results may be reported for each benchmark.

The PREP benchmarks provide a breadth and depth of data previously not available; breadth from the variety of circuit types and depth from the number of devices included. Users will be able to examine and analyze a particular device for strengths and weaknesses (e.g., a device may be very good at state machines but not good at arithmetic logic). Users will be able to compare a number of devices in a particular circuit area (i.e., which devices are good at implementing fast counters). In summary, users will be able to analyze a rich and valid set of data and reach their own conclusions.

As mentioned, this paper is intended to serve as an introduction to the PREP benchmarks. Complete and authoritative information is available from:

Programmable Electronics Performance Corporation
504 Nino Avenue
Los Gatos, CA 95032

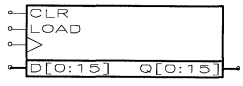


Figure 1. Hypothetical Functional Unit

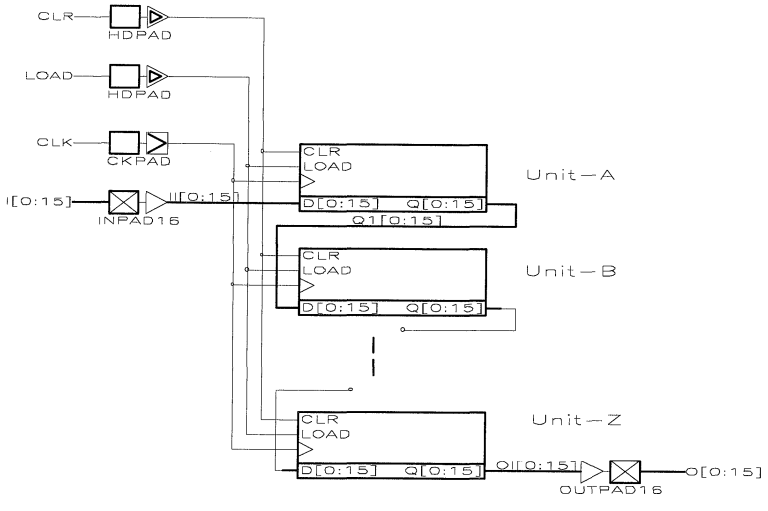


Figure 2. Interconnected Functional Units

Universal Computing
San Diego, CA

QuickLogic FPGAs help systems company to beat the competition to market with 40-MHz Intel i860-based small form factor multiprocessor system

Design Problem

Universal Computing designs and manufactures high-performance systems for a wide range of vertical market customers. One of Universal's designs is a multiprocessing VME board for signal processing applications in defense, medical imaging, seismology, image processing, molecular modeling and 3-D visualization. In addition to a motherboard supporting a VME bus and high-speed external interface, the system provides four "supercards," each containing a 40-MHz i860 XR microprocessor, up to 4 MB of high-speed DRAM, and high-speed interprocessor communications logic.

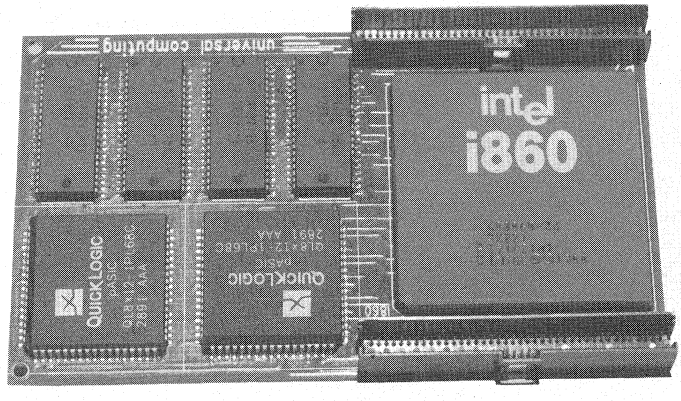


FIGURE 1
The MPi860 offers up to 320 MFLOPS of floating point performance in a single 6U VME slot

QUICKLOGIC

According to sales and marketing director John Thomas, several other companies offered similar products in the large 9-U VME form factor. However, the 9-U standard is considered by some potential customers to be unwieldy. The large form factor can create vibration problems, has loose power standards, and often suffers from inadequate cooling. While the 9-U product is a low-volume opportunity by nature, more compact 6-U products are far more popular and available. Thomas wanted Universal to become the first company on the market to offer such a powerful multiprocessing system in the 6-U form factor.

...to meet VME power specifications, PALs or EPLDs were out of the question.

The major challenge for Universal was redesigning each of the four supercards to create four highly compact daughtercards. In addition, while 9-U systems may use as much as 175W, Universal's 6-U system would need to draw less than 45W to meet VME power specifications. As a result, high-speed PALs or EPLDs were out of the question. A masked gate array could have been used, but the time needed to design and manufacture the part would shorten the period during which Universal would be the sole supplier of 6-U boards.

Design Solution

Universal recognized FPGAs as the solution. The company had used FPGAs from other vendors in several previous designs and considered them for the new design, along with the new pASIC FPGAs from QuickLogic. According to Thomas, "Even without programming logic into the other FPGAs, getting on and off the daughtercards took 45-50 ns. The only way to make the design work would be to drop the performance of the system from 40 MHz to as low as 25 MHz. With QuickLogic, the I/O delay fell to 9 ns, and we could preserve all of the high performance of the 9-U design."

Design Process

Universal used two QL8x12 FPGAs to provide the high-speed memory and bus interface logic required by each of the 40-MHz i860 XR microprocessors. Using the pASIC Toolkit development system, Universal's designers were able to create their logic quickly using high-level schematics. The design tools automatically routed the schematics into optimized gate-level designs that made efficient use of the FPGA logic cells and preserved the high performance needed. By cross probing between the schematic entry, timing simulator and physical viewer, the designers were able to easily debug and optimize the design to support the full potential of the i860 processors and the high-speed memories.

QuickLogic FPGAs allow developer of powerful real-time video processing systems to complete custom logic design in one weekend

Datacube, Inc.
Dearborn, MA

Design Problem

Datacube, Inc. develops extremely powerful computers designed to manipulate complex video data in real time. One Datacube product called the MaxVideo 20 provides 3,500 MIPS for compute-intensive video processing applications such as real-time correction of the geometric image distortion caused by a wide-angle lens. Datacube is a sophisticated user of all kinds of programmable logic, using 80 FPGAs and EPLDs in the MaxVideo 20 product to help integrate the functionality of 14 standard image processing boards onto one double-VME board.

R&D vice president J. Dunn was responsible for developing a new system for real-time pipeline processing of video data. A custom logic design was needed for data formatting and routing, and a critical path existed where 12 memory modules output 96 video data signals at 40 MHz. Here, custom logic was needed to format the data as 192 signals at 20 MHz and then route the data to other system components according to various board and operating mode specifications.

Design Solution

A masked gate array would solve Dunn's custom logic requirements, but time-to-market was a consideration, and the one masked gate array used on the MaxVideo 20 product proved to be the longest-lead component in the system. Dunn decided instead to evaluate all of the programmable logic alternatives in packages ranging from 28 pins to 160 pins, looking for the right high-speed partition for his 24-bit data path. Fast PALs or EPLDs might meet the 40 MHz data rates, but according to Dunn, "EPLDs use substantial power in the process — from 150 mA on up." FPGAs posed the ideal solution, but in Dunn's experience, "The speed criteria eliminated most of the choices. We have the Xilinx tools and the Actel tools in-house, but I didn't believe that I could make either of the architectures go fast enough." Dunn instead chose the 1,000-usable-gate QL8x12 from QuickLogic because the device easily met the 40 MHz data rate and low power requirements. "The 68-pin package was just right," Dunn added.

"We have the Xilinx and Actel tools in-house, but I didn't believe I could make either go fast enough."

Design Process

Dunn designed his logic on the pASIC Toolkit, which is designed to be easy to learn and use, even for first-time customers. According to Dunn, "I did the entire logic design for four QuickLogic parts over a single weekend." Added Dunn, "QuickLogic is a great solution for people needing high speed and low power. I hear other vendors trying to build excitement for their next-generation products. QuickLogic is already there with devices that I can use today."

Performance Controls
Horsham, PA

QuickLogic FPGAs boost speed of servo control system to 40 MHz while integrating 25 TTL devices into a single component

Design Problem

Performance Controls is a manufacturer of high-performance motion control systems. Digital design engineer Brian Fenstermacher was assigned to create a higher-speed implementation of an existing servo control system. The prior system logic implementation used about 25 TTL logic components, including LS-TTL, AS-TTL and F-TTL parts. Board space was an issue in the previous design and would be a problem in the redesign as well. More importantly, Fenstermacher predicted that the speed requirements of the redesign exceeded the capabilities of discrete logic parts. A new approach was inevitable.

Design Solution

Fenstermacher decided to try to integrate the functionality of the TTL devices into a single, high-speed FPGA. He considered all of the available FPGA solutions, looking for the one that could best meet the high-speed logic requirements of the new product specification. Fenstermacher decided to use the first member of QuickLogic's pASIC 1 family of FPGAs, the 1,000-usable-gate QL8x12.

Design Process

According to Fenstermacher, QuickLogic's Microsoft Windows-based pASIC Toolkit was easy to learn and more intuitive than competing FPGA development systems. He mastered the design tools in days, using schematic capture for design entry. Fenstermacher took advantage of cross probing between the schematic capture, simulation and physical viewer tools to quickly optimize and debug the design. The simulator provided precise results, and the programmed QL8x12 performed exactly as predicted. Within two weeks of receiving the QuickLogic tools, the pASIC FPGAs were programmed and ready for system installation. Fenstermacher was successful at integrating the functionality of two dozen MSI and LSI parts into the QL8x12 while comfortably meeting the 40 MHz speed requirements of the redesign.

...the programmed QL8x12 performed exactly as predicted.

According to Fenstermacher,"QuickLogic offers a great way to implement very fast, complex state machines in an easy-to-use part."

pASIC 1 FAMILY
FPGA User Applications Profiles

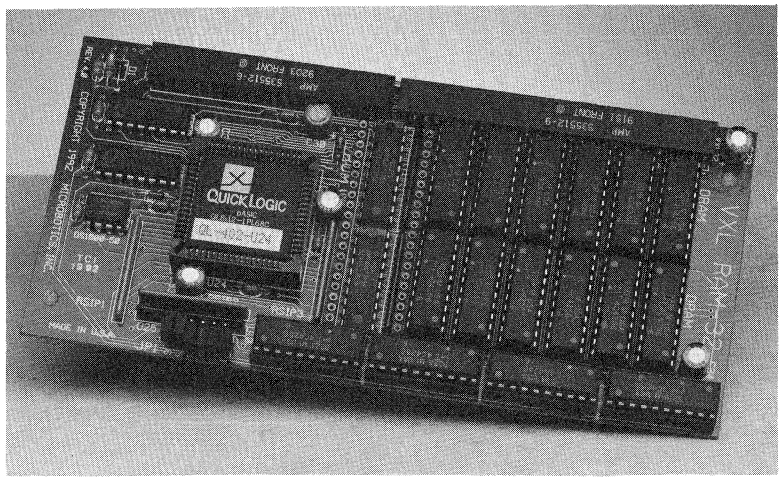
The most aggressive users of new technologies are frequently fast growing entrepreneurial companies. QuickLogic's pASIC 1 Family of very high-speed FPGAs has been no exception. The customers profiled here are typical of those who exploit the high-speed, low-power, small-size and fast time-to-market advantages offered by these devices to gain a critical competitive advantage. These comments were gathered as part of a market research study on future FPGA architectures. Applications described are:

- **Low-cost Graphics Accelerator boards – MicroBotics**
- **S-Bus Dynamic Signal Acquisition board – National Instruments**
- **Cache and Disk Controller for Microchannel bus – Cumulus**
- **50 MHz Microprocessor Development Systems – Cogent**
- **R4000 processor-based VME board – Mizar**
- **1000-processor Parallel Computer – Universal Computing**

Low cost Graphics Accelerator Boards: *Microbotics Inc.*

MicroBotics Inc. of Richardson, Texas designs and manufactures multifunction boards for the Commodore Amiga personal computer. MicroBotics' customers use these products to accelerate the basic functions of their Amiga machines, which have a heavy following in graphics and multimedia applications. The majority are shipped to countries outside the United States. A typical example is the VXL RAM 32, high-speed 32-bit RAM card for a 50 MHz 68030-based Amiga color graphics accelerator board, shown in the photo below.

MicroBotics' 32-bit, 50 MHz RAM Card



QUICKLOGIC

Key engineering requirements to satisfy this market include very fast design time and high performance in a restricted board size. MicroBotics finds programmable ASIC solutions essential to meet these demands.

QuickLogic Delivered on Its' Claims

MicroBotics uses FPGAs for a variety of system logic functions including high-speed I/O control and general-purpose glue logic. Recently MicroBotics started to design a new board based on an EPLD family which was advertised as offering very high speed. In practice the EPLDs fell significantly short of their performance and density claims. This resulted in a severe delay in the introduction of the MicroBotics board. Jerry Robinson, President and head of engineering development for MicroBotics, said: "We designed the family into our systems based on the vendor's representations. The parts didn't work according to their specifications. As a result, we were unable to ship our product."

MicroBotics quickly investigated other suppliers. According to Jerry they found one other device which appeared on paper to meet the speed, but had very poor CAE tool support. Xilinx SRAM devices were not only too slow, but also had "severe clock skew of 9 ns across the part, which is unusable."

QuickLogic's pASIC 1 FPGA family was finally selected because it "worked as billed" according to Jerry. The easy design entry and interactive development process accelerated MicroBotics overall time to market. The simulation and physical viewing tools were particularly useful in debugging the design. In fact the tools were so easy to understand that Jerry mentioned "I learned the software in a night". The device speed met it's claims and fit within MicroBotics' design goals. Finally, QuickLogic's FPGAs had very good power characteristics: "at 50 MHz, QuickLogic consumes 20 mA instead of the slower EPLD's power consumption of 180 mA."

QuickLogic's Tools Yielded Six-Week Time-To-Market

The QuickLogic tool set yielded rapid design cycle times for MicroBotics. The schematic capture system was "just great" according to Jerry. The simulation tools proved especially useful as the results were accurate to "zero nanoseconds, or so, at the circuit board level." Further, at the device level, they found that the FPGA metastability is "very good," making the design process simpler and faster.

Finally in the place and route area, MicroBotics is pleased with the ability of QuickLogic's architecture to fix a signal at the pin level and be able to route the device in that fashion. This allows board design to proceed in parallel with FPGA layout. Commenting on other technologies, Jerry mentioned that "many won't route at all."

Summary

Overall, MicroBotics is pleased with the QuickLogic solution. Jerry said that: "QuickLogic does a really good job. Fundamentally the company meets its claims." This is unique relative to MicroBotics previous experience with programmable logic vendors.

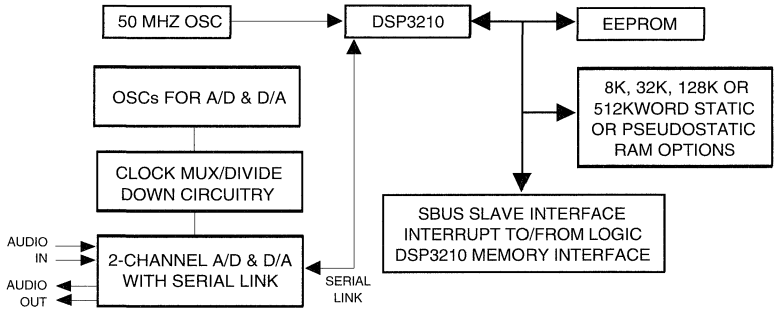
S-Bus Dynamic Signal Acquisition Board: *National Instruments*

National Instruments of Austin, Texas is a fast growing leader in test and measurement technology. Through a variety of hardware and software products, the firm takes standard off-the-shelf personal computers and turns them into powerful laboratory instruments. Its' Windows and Macintosh based applications software, called LabWindows and LabVIEW are becoming industry standards for ease of use, power, and completeness. National Instruments' current product family includes GPIB controller boards, data acquisition cards, audio boards, and application software to provide a complete and fully integrated solution.

Keith Winkler, the development engineer for a new data acquisition board that uses QuickLogic FPGAs, has been with the company for nearly two years. Prior to joining National Instruments he was an electrical engineering student at Rice University, and worked with the company as a part of the co-op engineering program. During his time as a co-op student, he learned FPGA design using Xilinx devices.

The design goals for the new product were to implement a dynamic signal acquisition board for the S-Bus. The board would provide real-time data acquisition and analysis with intelligence. As the S-Bus board is roughly 3 by 6 inches in size, and one-half of the board area was necessary for the analog subsection, the board space constraints were severe. Keith's implementation used the remaining top half of the board real estate for the digital control section and the bottom of the card for memory. The processor design is very high performance, requiring clocking at between 50 and 66 MHz. It uses a high-speed processor, transceivers, two FPGAs, a PAL, and memory. It must synchronize with two asynchronous clocks running in the system.

National Board Block Diagram



Keith's decision to use QuickLogic FPGAs was based on speed, routing, gate density and component costs. Keith commented: "QuickLogic FPGAs have good metastability characteristics, they offer high speed and are dense." He compared them favorably to his past experience with other devices. "SRAM-based products have many problems with routing, and have long, unpredictable interconnect delays. Other FPGAs have on and off chip delays two to

three times slower than QuickLogic." He also considered using PLDs, but wasn't comfortable with the board-level interconnect and timing delay issues.

National Instruments choice of QuickLogic can be summarized by Keith's comment: "If the board was going to be done, it could only be done using QuickLogic."

National Instruments found that QuickLogic's fully integrated Windows-based tools are easy to learn and use. The user interface, schematic capture, and simulation tools all contributed to design time efficiency. Keith commented: "The navigator function was very convenient" further he said "the simulator was very accurate and helped significantly in the debug cycle." The automatic tools also worked well enough that Keith "didn't have much tweaking to do." In his experience other FPGAs took many iterations before the pin-outs become stable enough to commit the PC board layout. They also required much hand tweaking to achieve the speed objectives.

Conclusion

In addition to National Instruments' favorable impression of the speed, density, and cost-effectiveness of the QuickLogic components, the productivity and efficiency of the tool set significantly reduced the design cycle time relative to other solutions. Finally, Keith said: "the support from the company has been phenomenal."

Cache and Disk Controller for Microchannel Bus: *Cumulus*

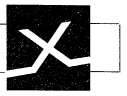
Cumulus Corporation develops personal computer products with a focus on Microchannel-based architectures. The company has design centers in Cleveland Ohio, and in San Jose and San Diego, California.

Vitek Zaba joined Cumulus six and a half years ago. Prior to joining Cumulus he spent several years with Burroughs Advanced System Design group. His initial exposure to FPGAs was through QuickLogic. Most of his semi-custom development work in the past has used gate arrays.

Very High Performance Essential

The Cumulus design using the QuickLogic technology is a high speed cache and disk controller for the Microchannel bus. The design uses two QuickLogic devices, one for addressing, and the other for control of the main CPU and the Microchannel bus. Very high bandwidth is required for the design because Cumulus uses a streaming technique for data into the drive system. A throughput of 160 MBytes per second, coupled with a drive array of up to eight hard disks, each of which having a 16-MByte cache, yields an overall throughput of twenty to thirty times the speed of a dedicated hard disk. In addition, the drive array can be configured to use one drive for storing parity data on the other data elements in the array.

The overall design uses two QuickLogic FPGAs, an 80186 control processor, a gate array, and a handful of other devices.



The original cache and disk controller design was developed a couple of years ago using discrete components. Over twenty parts were required to implement the design. When Vitek was assigned to implement the function in an FPGA, he started by examining all devices currently available. In the end, the single most important factor in his design decision was performance. Vitek indicated: "There were no alternatives available for a design at this speed. QuickLogic was chosen almost by default."

QuickLogic Selected Almost by Default

Using the QuickLogic pASIC Toolkit suite of CAE tools, Vitek completed the FPGA implementation in approximately four weeks. He stated: "The QuickLogic experience was very good and the devices worked well for high-speed designs."

---

50 MHz Microprocessor Development Systems: *Cogent*

Cogent Engineering of Marlboro, Massachusetts is a two year old company specializing in development tools for embedded control microprocessors. Cogent has designed software development and hardware evaluation boards for Motorola's EC family of processors. It is currently extending these products to different processor architectures through a common platform with interchangeable processor modules.

PAL Replacement Then Much More

The initial design goals for Cogent's system were to replace an existing PAL-based system with a higher-functionality solution with more features. Continuing to use PALs for the future system wasn't possible given the complexity of the new design. While they would have been fast enough, PALs didn't have the gate density necessary for the added functionality without growing the board real estate to an unacceptable level. They also consume vast amounts of power.

As Cogent's design clocks at 50 MHz, high performance was essential. Further, since Cogent markets their development board solutions as "custom-capable," the ability to make rapid changes in response to customer requests is very important. This eliminated the choice of high-density PAL and EPLD solutions because of the inability to re-route designs once I/O pins are locked down.

Cogent made a strategic decision to move to an FPGA-based architecture because it believed that increasingly FPGAs will offer all the speed of PAL solutions while offering higher density, greater flexibility, and substantially lower power.

Most Appropriate Solution for Cogent's Needs

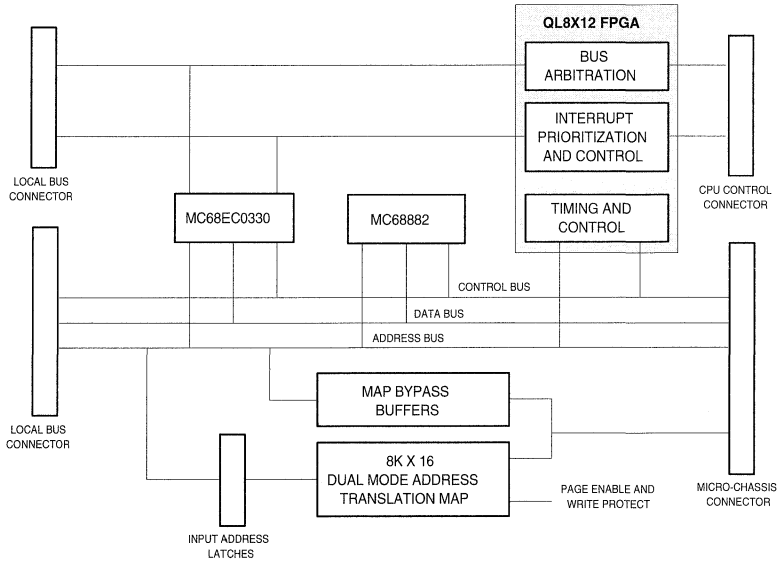
In determining the best solution to their design problem, Mike Kelley, founder of Cogent looked at a range of design alternatives, including tools and devices from Xilinx, Altera, Atmel, and Lattice. Mike's decision to use QuickLogic was based on speed, cost, and the longevity of the technology and architecture. For example, "Lattice pLSI was the only other solution that might have met the speed goals, but poor tools and high power ruled them out" according to Mike.

Speaking to the basic QuickLogic antifuse, FPGA architecture and tools, Mike said: "Among all the alternatives, QuickLogic's Vialink antifuse technology is the only one that doesn't need a fundamental technology revolution to continue to grow in density and speed." Given that Cogent wanted to commit to a family that would grow as their design challenges expanded, "QuickLogic was the preferred solution." According to Mike, QuickLogic provided the optimal mix of speed, design time, and cost effectiveness that no other supplier could match.

Cost Effective CAE Tools

Cogent's analysis of the development tool alternatives indicated that the CAE tool set provided by QuickLogic offered far more features, such as a fully integrated Windows environment, physical viewer, path analyzer and timing-driven placement, at the same time that it was less expensive than alternatives. While Mike felt that "the tools should be free "because we're using QuickLogic's chips," he agreed that the benefits of short design time outweighed the tool costs given the end result.

Cogent MC68EC0330 CPU Module Block Diagram



Most of Cogent's designs have taken less than three weeks from start to finish. Typical functions integrated were Bus Arbitration logic, Interrupt Priority and Control, Bus Timing and Memory Address Decoding. A typical FPGA project requires one week for basic design implementation and a second week for tweaking and in-system modifications.

R4000 Processor-based VME Board: *Mizar*

Mizar Inc. of Carrollton, Texas is a leader in the development, manufacturing, and marketing of VME board solutions for industrial, multiprocessor architectures for real time, process control, and embedded applications.

John Fike, the engineer who developed a solution using QuickLogic's FPGAs has been with Mizar for three years. Prior to Mizar he worked extensively in embedded control, data acquisition, and satellite communications design.

High Speed and Limited Real Estate Design Challenge

The project using QuickLogic technology is a MIPS R4000 processor-based VME board for a specific customer application. The FPGA is used to implement a write buffer interface to the R4000. According to John Fike, "The customer requested an architecture scalable in speed to 75MHz in the near term with a spin-off design to 100 MHz for follow-on MIPS process improvements." Mizar's strategy was to use QuickLogic as a protoype and preproduction vehicle anticipating a conversion in production to a gate array implementation. The regular and orthogonal QuickLogic FPGA architecture and low impedance ViaLink interconnect provided "a good simulation vehicle for ASICs."

Historically, Mizar had examined a variety of FPGA solutions, and had specific experience with Actel devices. John said: "None of the Actel families could achieve the speed goals of the new system." John chose QuickLogic for reasons of speed and logic density. There was no alternative to run at the speeds required with the logic density to fit within the VME board constraints. Furthermore, the availability of devices in the TQFP (Thin Quad Flat Package) gave Mizar a way to implement a high-speed design with very limited real estate.

The only other alternative considered for implementing the design was PALs, but with the many board interconnect delays between multiple packages they "weren't even fast enough, even if we had the space."

Tools Provided Short Design Cycle

Mizar's decision to use QuickLogic in its R4000 design was made because QuickLogic provided the only solution at the speed and density required by the design. However, the tools used to develop the solution were a pleasant surprise. They were simple to use, and met Mizar's project design time needs. Mizar's complete design cycle for the system was about three months even though it was the first experience with the pASIC architecture and tools. After the project was completed, John commented that: "The toolkit is very good. It's easy to learn, works seamlessly, and the simulation files are easy to use. It's also very convenient that the programmer works off the RS-232 port."

1000-Processor Parallel Computer: *Universal Computing*

Universal Computing of San Diego, California was founded in 1980 to provide system integration services. Customers included General Dynamics, Hughes, and SAIC. In addition, Universal Computing developed standard processor boards to fill gaps in their customers' computing needs. They currently manufacture a complete patch panel with I/O functions. Most recently, they have developed a high-performance multiprocessor board which will be sold to the United States Navy and others.

Gene Jones, Director of Hardware Engineering for Universal Computing, has been with the firm for nearly seven years He earned an Electrical Engineering Degree from the University of California at San Diego.

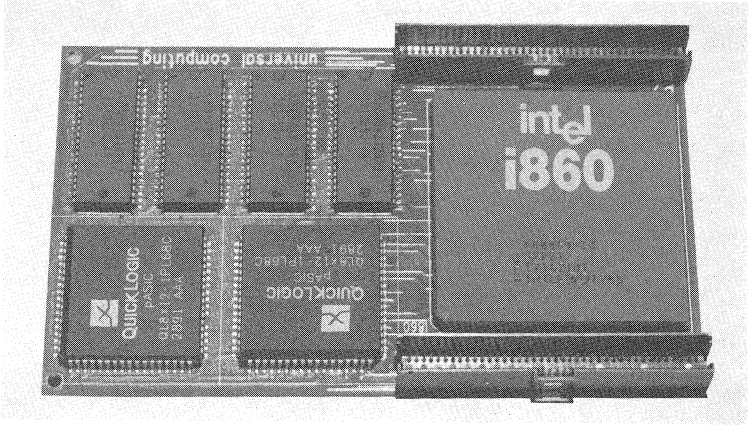
Ocean Mapping Through use of a Parallel System

Universal has a contract from the United States Navy for a very high-performance supercomputer to be used for an ocean mapping application. It is based on Universal's Quad-Processor™ technology. Universal has completed a 32-processor demonstration system, and is in the process of implementing a 200 processor machine. Eventually, the system will require up to 1,000 processors. Each processor uses an Intel 40 MHz i860 device, which provides 80 MFLOPS of floating point performance. The architecture of the system is effectively a massively parallel, loosely coupled multiprocessor implementation. Universal was selected by the Navy for this project because "all existing approaches were too expensive" according to Gene.

Universal's key design goal was to make the basic processor module small enough to fit onto a standard 6-U size VME form factor card. This small size solves a variety of interconnect, packaging, and thermal problems typically associated with large computer designs.

The solution to this challenge was twofold: first, a reduction in the number of control and "glue" logic I.C. packages required to support the processor to memory interface, and second, the use of an innovative memory interface technique to reduce the number of memory chips per processor. By using fewer memory chips Universal eliminated the need for processor-to-memory buffers, thereby further reducing board real estate, power, and cost.

Each processor daughterboard contains two QuickLogic FPGAs for the processor-to-memory interface, an Intel i860, and four memory packages. Each Universal Quad-Processor module consists of four independent processors. Control logic in two more FPGAs interfaces the module to the VME backplane. The net result is that each Universal VME board contains ten QuickLogic devices.



Universal DB-860 Processor Board

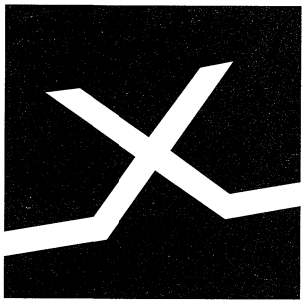
Prior to this design Universal had used Xilinx FPGAs whenever possible. However, the higher performance demands of this project determined that QuickLogic FPGAs were the only solution for the high-speed functions of the memory controller. During the design process "We lost confidence in our ability to predict performance and utilization with the SRAM technology, and are now using only QuickLogic in the design" said Gene.

QuickLogic: The Most Appropriate Design Solution

The reasons for changing to a QuickLogic only approach were several. "It took us too long to make even minimal progress, and the support we received was poor." Further, "it took us days to get the part to route, versus only minutes with QuickLogic." With QuickLogic, Universal could predict the input/output times and the gate utilization prior to routing: "we got to a 98% utilization level with QuickLogic running at 40 MHz, with no problems."

The Design Tools: A Real Advantage

According to Gene, "the QuickLogic design environment is a real plus." As all of the QuickLogic tools are fully integrated into the Windows environment, the interaction between different tools is simple and seamless. "With other tools, when a change is made you have to get out of one environment and into another, thus taking time." Complete design and debug cycle times for most of the devices were as short as two weeks.



QUICKLOGIC

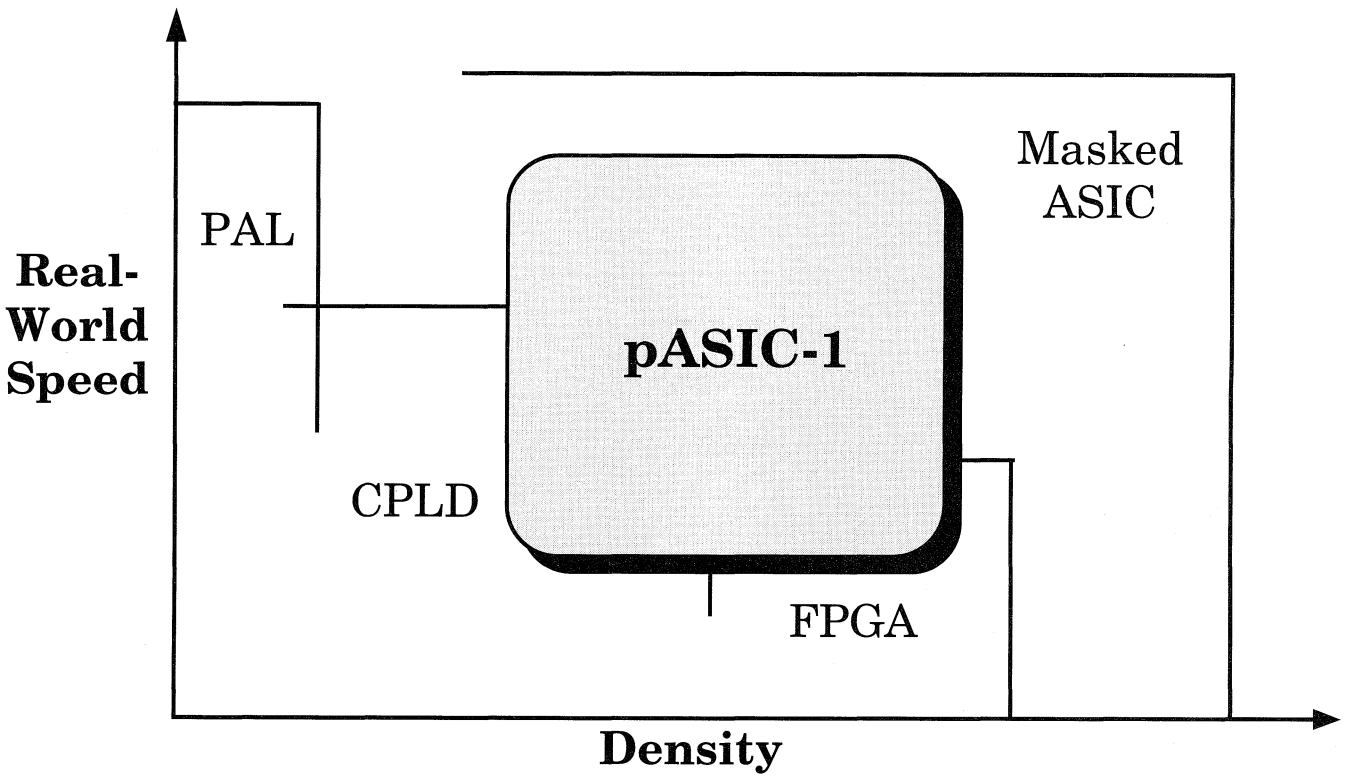
- Technology
- Architecture
- Design Tools

Revolutionizing the FPGA Industry

pASIC-1 *Wild*Cat Series

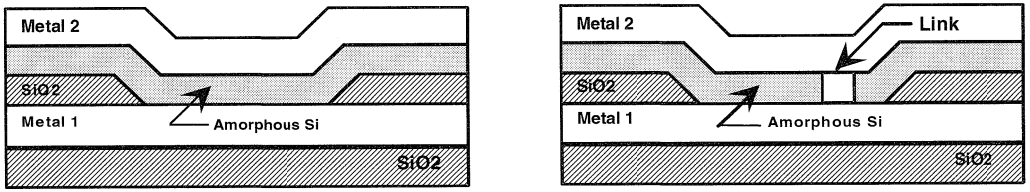
- Fastest in-system speeds: 60 -100 MHz
- 100% usable densities: 1K to 8K gates
- Maintain pin-out throughout design cycles
- Timing predictability & consistency
- 3.3 Volt operation supported
- 8K device fully compliant to PCI 2.0
- Sophisticated, high productivity tools

Speed vs. Density



Interconnect Performance

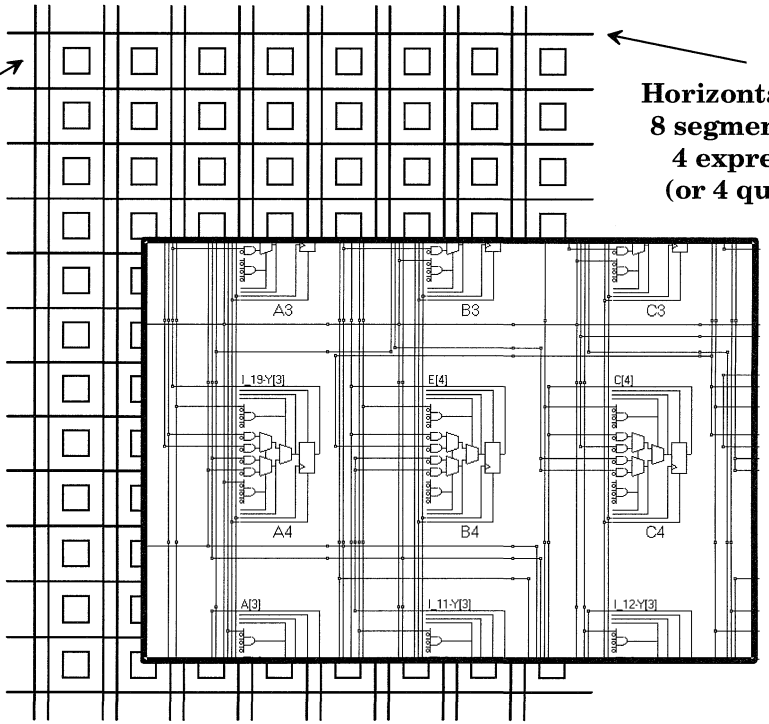
ViaLink



	SRAM	EPROM	PLICE	ViaLink
resistance	1K ohm	1K ohm	500 ohm	20-50 ohm
capacitance	~50 fF	~15 fF	< 5 fF	< 1 fF
element size	very large	large	medium	small

- ViaLink offers total security of design code
- Transistors not allocated for Interconnect Resources
- One mask added to standard 0.65 micron CMOS process

Fast and Regular Routing



Vertical Channel:
16 segmented wires,
4 express wires
(+4 quad wires)

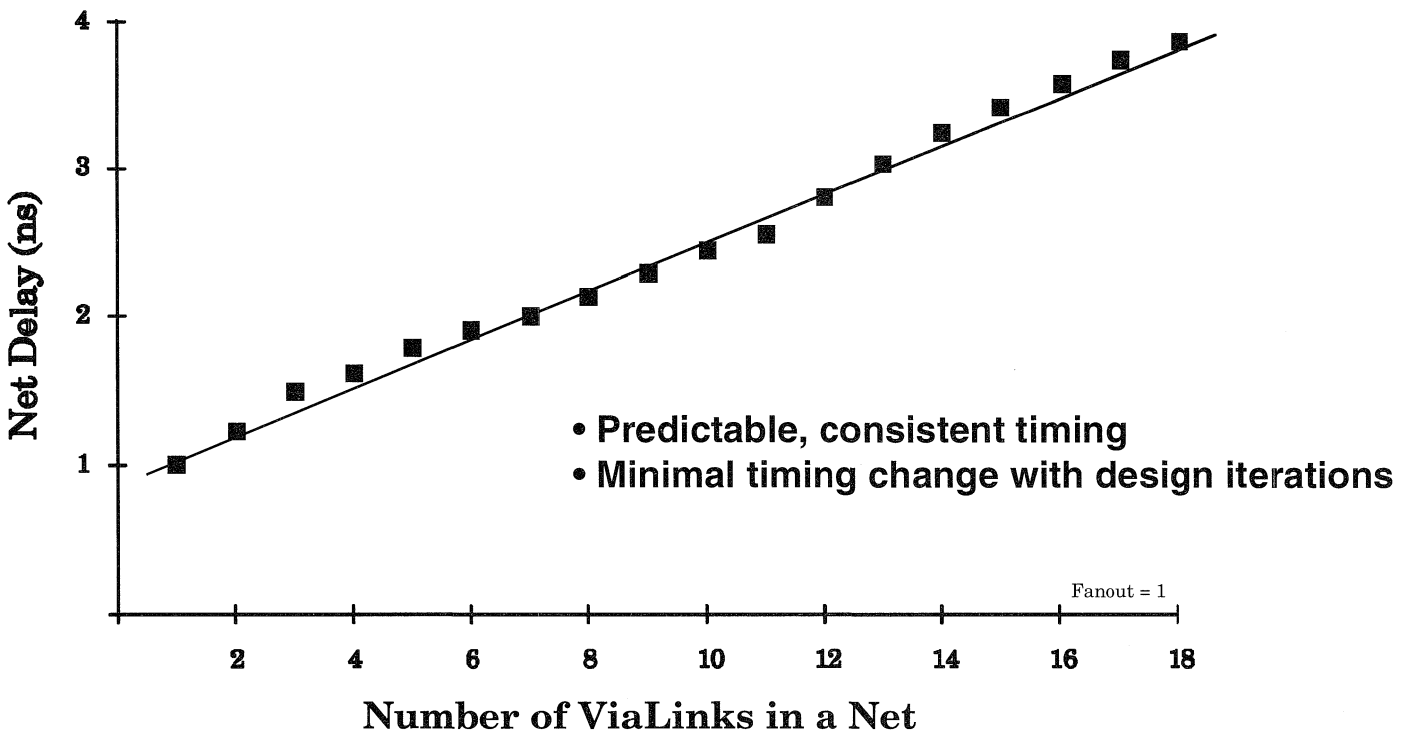
Horizontal Channel:
8 segmented wires,
4 express wires,
(or 4 quad wires)

Two Internally Buffered Clock Networks

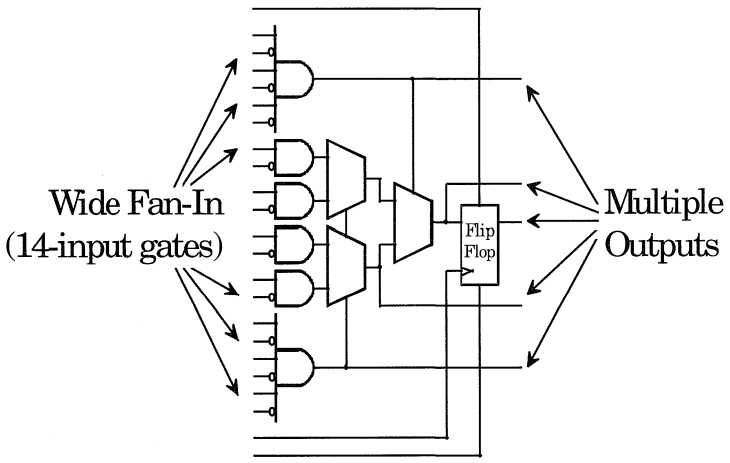
Utilizing 100% of I/O *and* 100% of Flip-Flops uses approx. 5% of available ViaLinks!!

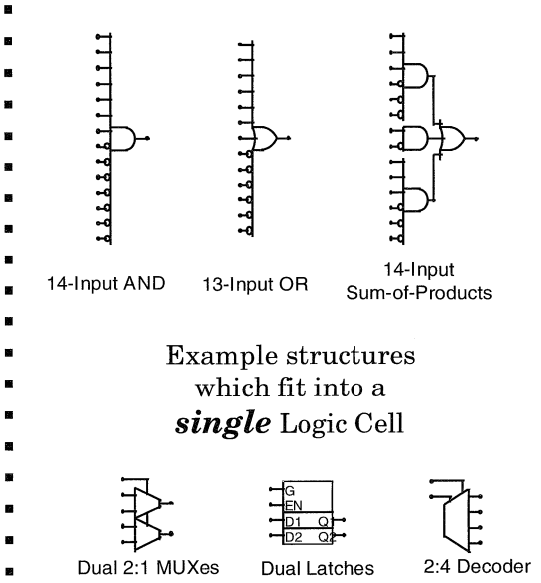
* Quad wires on 4K & 8K usable gate devices

Net Size vs. Delay



High-Speed Logic Cell





- High-performance *and* high-flexibility
- Wide fan-in functions without sacrificing density
- Multiple outputs allow multiple paths thru cell

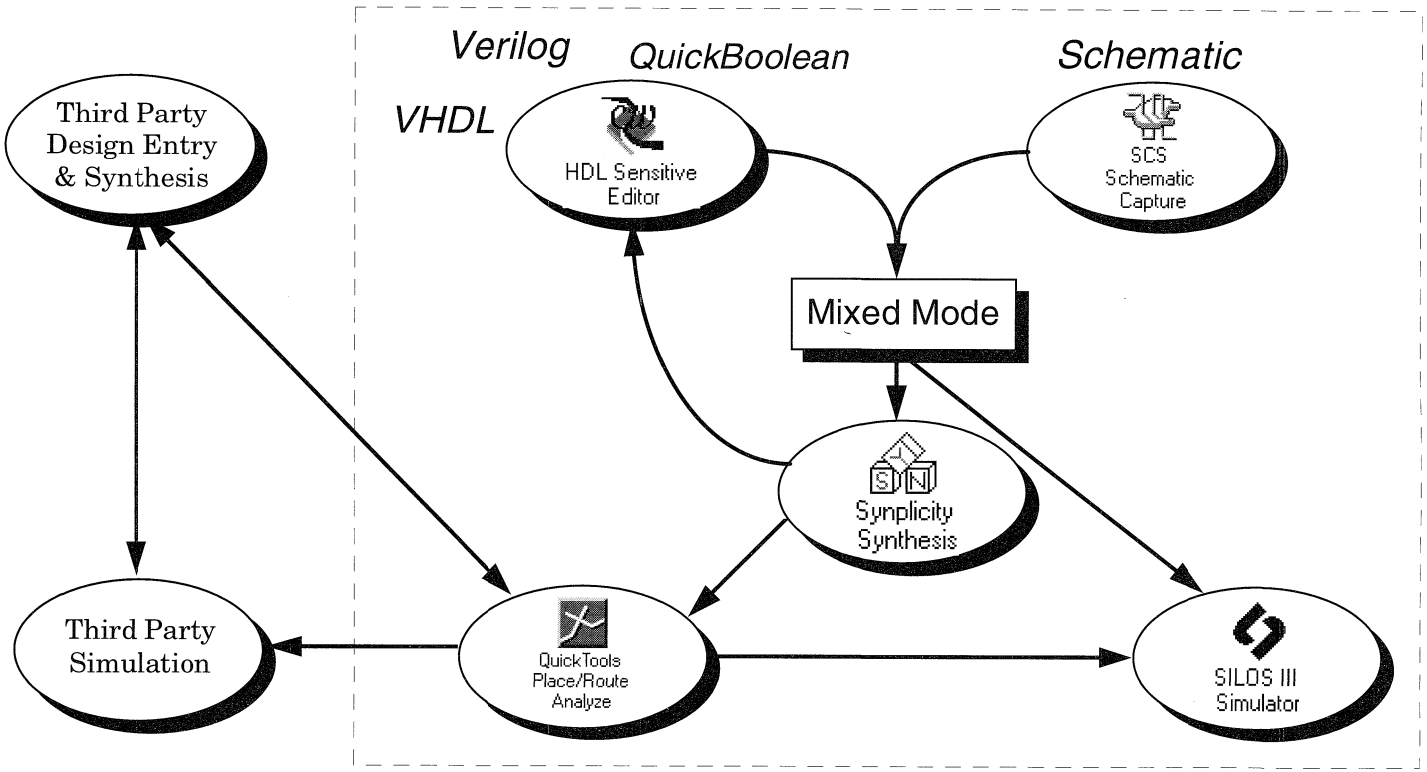
Automatic Place and Route

- 100% automatic place and route
  - Even on designs with 100% utilization of FF & I/O
  - Designs can always route
  - Very fast run-times (3 min for QL8x12B on '486 PC)
- Maintain pin-out throughout design cycle
  - Can fix I/O for board layout!
  - Manually place some or all I/O pads
  - Manually place some or all flip-flops
- Timing-driven placement
  - Optimizes chosen paths to specified goals
  - Optimal results *without* manual place & route

QuickWorks Productivity

- Mixed mode entry: Schematic & HDL
  - Syntax-sensitive editor for fast HDL design entry/readability
- Synthesis comparable to schematics in minutes
- Powerful design evaluation via graphical cross-probing between analytical tool windows
- High performance Verilog simulator
- Third party support via industry standards
- Affordable Windows PC tool suite

Design Flows



QuickWorks Tool Suite

Design Entry Capabilities

- Language sensitive entry
  - HDL Language templates guide command field entries
  - Automatic indenting / color coding
  - Real-time Syntax checking
  - Automatic Verilog test bench generation
- Windows schematic entry
  - Data I/O's Synario Capture System (SCS)
  - Library of 500 QuickLogic design macros
  - Mixed mode entry within a schematic of Verilog,VHDL, QuickBoolean or Open ABEL
  - "Cross-probes" to device physical views and simulator waveforms

Synthesis that *Works* for FPGAs

- Best synthesis in benchmark trials
  - Comparable to schematic-based designs
    - Comparable speed performance and logic cell area utilization
  - Up to 300 times faster compilation than major vendors
    - 1,000 gate 8x12B design in less than one minute
    - 4,000 gate 16x24B design in less than five minutes
  - Fully integrated into QuickWorks tool suite
    - Accepts text based inputs mixed with schematics
    - Industry Standard LPM generation
- Synthesis by Synplicity
  - Founded by architect of Mentor Graphic's AutoLogic
  - VHDL & Open ABEL support in 2Q95

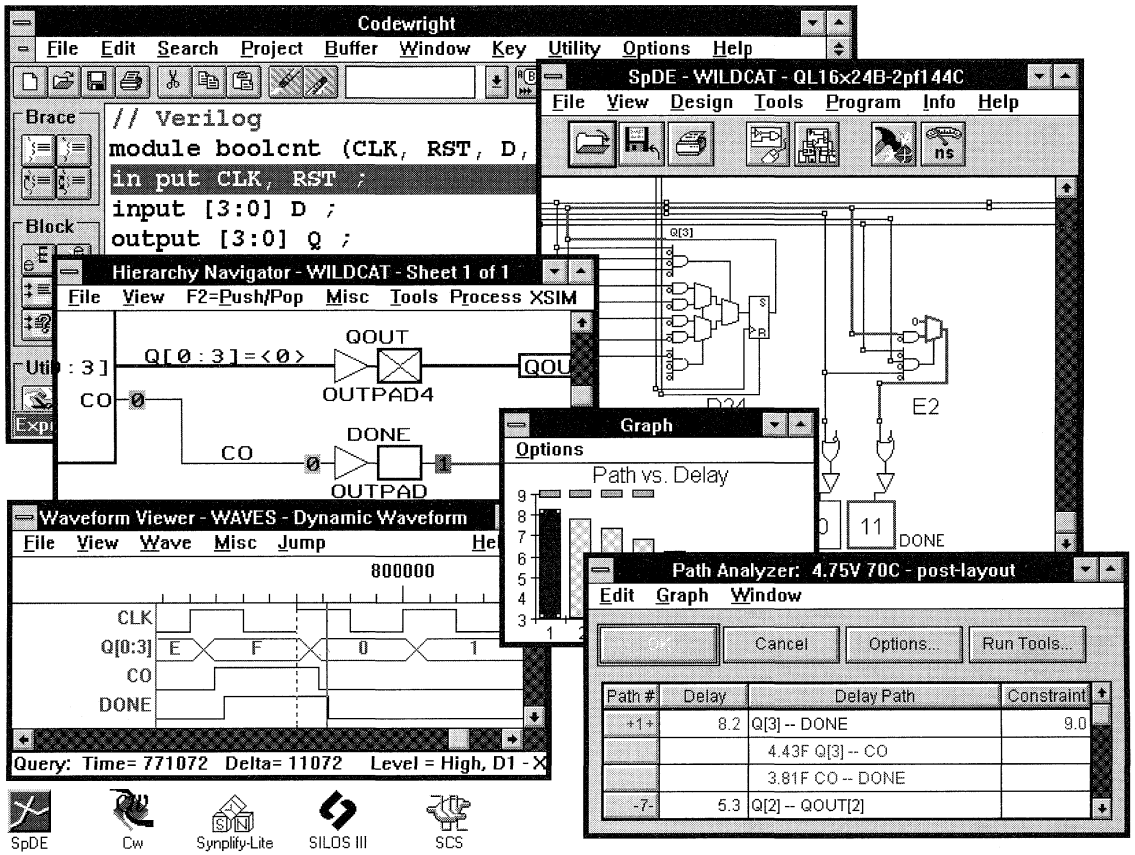
QuickTools w/SpDE

- QuickTools Bundle for third party tools:
  - SpDE Toolkit
    - Fast optimization, place & route
    - Guaranteed accurate delay modeling
    - Physical views of placed / routed design
    - Integrated path analysis tool
    - Timing driven placement for tuning critical timing paths
    - Supports all QuickLogic devices
  - QuickUtilities
    - Logic Re-Optimizer typically improves logic density by approx. 12%
    - Automatic optimal buffer generation for high fanout paths
    - Third party simulator support for Intergraph, Silos III, SusieCAD, QuickSim
    - LPM Reader (Q2/95)
  - Available for PC, Sun, and HP platforms

Verilog Simulation

- Simucad's SILOS III Verilog HDL simulator
  - Fully compliant to Verilog OVI standard
- Fast behavioral / gate level simulation
  - Input: Verilog test bench or graphical waveform
  - Output: Text-based or graphical waveform
- Interactive simulation results viewing
  - 1's and 0's on schematic correlate with timing waveform cursor position
- Source level HDL debug in Q2/95

QuickWorks Tool Suite



Tightly Integrated "Cross-Probing"

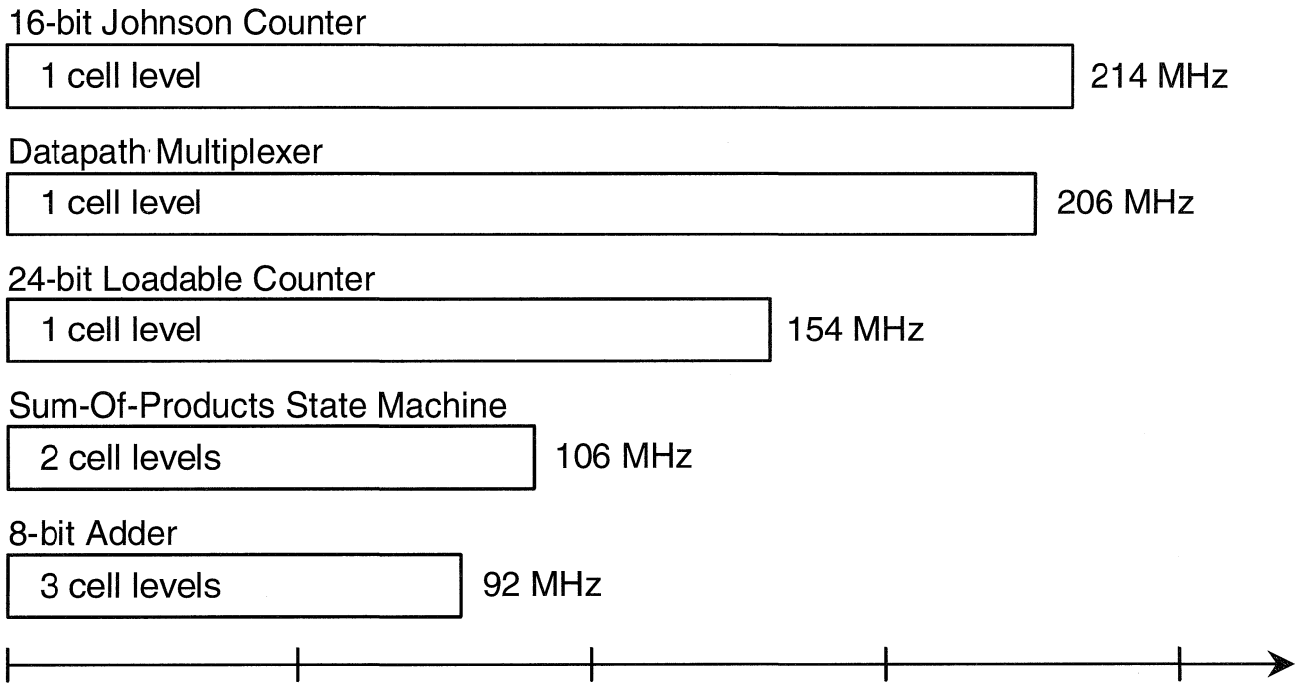
Development Alternatives

Tools Vendor	Schematic Capture	HDL/Language Entry			Simulation & Verification	Platform
		VHDL	Verilog	Other		
ViewLogic	✓	✓		✓	✓	PC, SUN
Synopsys		✓	✓		'95	SUN, HP
Cadence	✓	'95	'95		✓	SUN
Data I/O	✓			✓	✓	PC
Logic Modeling					✓	SUN, HP
Innovative Synthesis (ASYL+)		✓	✓	✓		PC, SUN
Intergraph	✓	'95	'95		✓	NT, SUN
Exemplar		✓	✓	✓		PC, SUN
Mentor Graphics	'95	'95		'95	'95	HP
Aldec (SusieCAD)	'95				'95	PC
Logical Devices (CUPL)				✓		PC
Synplicity		'95	✓	'95		PC
Flynn Systems (ATVG)					'95	PC
Silicon Automation					✓	PC
SimuCAD (SILOS III)					✓	PC

✓ = Supported Now

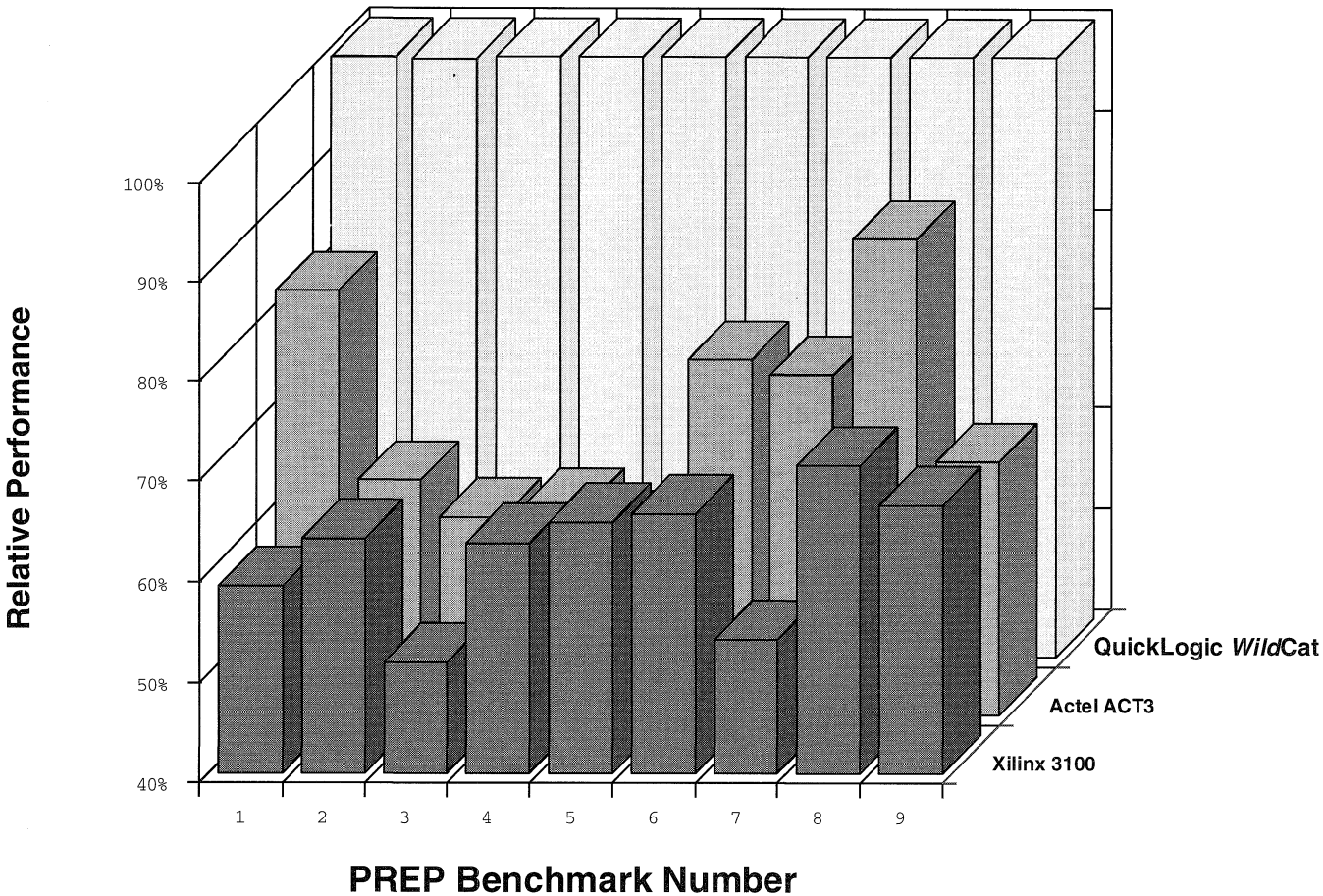
'95 = Support Planned for introduction during 1995

Real-World Speed



- Worst-Case Commercial; fully automatic place & route

PREP® Benchmarks



Application Examples

- Graphics processing : 60 MHz +
- DRAM controllers : 66 MHz +
- Video and imaging : 72 MHz +
- DSP support logic : 80 MHz +
- Data acquisition : 100 MHz +

pASIC-1 *Wild*Cat Devices

		QL 8x12B	QL 12x16B	QL 16x24B	QL 24x32B
Logic Cells		96	192	384	768
Usable Gates		1,000	2,000	4,000	8,000
Total I/O Pins		64	88	122	180
Pgmbl I/O Pins		56	80	114	172
CK / HD Pins		2 / 6	2 / 6	2 / 6	2 / 6
Pkgs	PLCC	44, 68	68, 84	84	
	TQFP	100	100	100, 144	144
	PQFP				208
	CQFP	68	100	100, 160	160
	CPGA	68	84	144	225

- Speed grades : -X (swift), -0 (fast), -1 (faster), -2 (fastest)
- Commercial, Industrial, Military and 883D

"Check-out Your Design" Eval Kit

- Everything for a complete design - $99
  - Includes all elements of the Quick*Works* toolkit
  - Schematic Capture package
  - HDL-sensitive language entry
  - Verilog synthesis (VHDL & Open ABEL mid-'95)
  - Auto Place & Route, Path Timing analyzer
  - Verilog / Waveform Simulation
  - Supports *all* QuickLogic devices
- Check-out your design in a *Wild*Cat FPGA for fit and speed
  - 30-day software license and support
  - Programming support *not* included

Summary

- Technology
- Architecture
- Design Tools

Revolutionizing the FPGA Industry by solving the problems of traditional FPGAs

Sales Representatives/Distributors

Direct Sales Offices

NORTHWEST
Corporate
2933 Bunker Hill Lane
Santa Clara, CA 95054
(408) 987-2000
FAX: (408) 987-2012

NORTHEAST
1 Batchelder Rd.
Marblehead, MA 01945
(617) 631-4279
FAX: (617) 631-4279

SOUTHWEST
516 23rd Street
Manhattan Bch, CA 90266
(310) 546-5589
FAX: (310) 546-1870

MIDWEST/SOUTHEAST
2613 Hampshire Dr.
Garland, TX 75040
(214) 414-6908
FAX: (214) 414-8374

QUICKLOGIC

Sales Representatives

ALABAMA
Rep, Inc.
11535 Gilleland Rd.
P.O. Box 4889
Huntsville, AL 35815
(205) 881-9270
FAX: (205) 882-6692

ARIZONA
Altek Components, Inc.
4525 S. Lakeshore Dr, Ste 1
Tempe, AZ 85282
(602) 345-9905
FAX: (602) 345-9906

Altek Components, Inc.
5005 Calle de las Chacras
Tucson, AZ 85718
(602) 529-0443
FAX: (602) 299-4707

ARKANSAS
Nova Marketing
8421 East 61st St.
Suite P, #139
Tulsa, OK 74133
(918) 660-5105
FAX:(918) 357-1091

CALIFORNIA
Exis
2841 Junction, Suite 202
San Jose, CA 95134
(408) 456-1650
FAX:(408) 433-0450

First Rep of So. California
143 Triunfo Canyon, Suite 222
Westlake Village, CA 91361
(805) 373-0887
FAX: (805) 495-1317

Harvey King, Inc.
9520 Padgett St., Suite 101
San Diego, CA 92126
(619) 695-9300
FAX: (619) 695-9515

COLORADO
Seale & Associates
7535 E. Hampton Ave, #100
Denver, CO 80231
(303) 695-1980
FAX: (303) 695-1984

CONNECTICUT
DatCom, Inc.
One Evergreen Ave.
Hamden, CT 06518
(203) 288-7005
FAX: (203) 281-4233

DELAWARE
CMS Marketing
527 Plymouth Rd, Ste 420
Plymouth Meeting, PA 19462
(215) 834-6840
FAX: (215) 834-6848

DISTRICT OF COLUMBIA
Tri-Mark, Inc.
1410 Crain Hwy. N.W.
Glen Burnie, MD 21061
(410) 761-6000
FAX: (410) 761-6006

FLORIDA
Electramark Florida, Inc.
2910 W. Waters Ave.
Tampa, FL 33614
(813) 915-1177
FAX: (813) 915-1188

Electramark Florida, Inc.
401 Whooping Loop, Suite 1565
Altamonte Springs, FL 32701
(407) 830-0844
FAX: (407) 830-0847

Electramark Florida, Inc.
621 N.W. 53rd St., Suite 240
Boca Raton, FL 33487
(407) 998-8820
FAX: (407) 998-8821

GEORGIA
Rep, Inc.
1944 Northlake Pkwy., Suite 1
Tucker, GA 30084
(404) 938-4358
FAX: (404) 938-0194

HAWAII
Exis
2841 Junction, Suite 202
San Jose, CA 95134
(408) 456-1650
FAX: (408) 433-0450

ILLINOIS
KMA Sales Co.
1040 S. Arlington Hgts. Rd.
Suite 105
Arlington Heights, IL 60005
(708) 398-5300
FAX: (708) 398-5708

INDIANA
Arete Sales, Inc.
2250 Lake Ave., Suite 255
Ft. Wayne, IN 46805
(219) 423-1478
FAX: (219) 420-1440

Arete Sales, Inc.
9243 N. County Rd., 200 West
Lizton, IN 46149
(317) 994-5084
FAX: (317) 994-5088

IOWA
Technical Sales Associates, Inc.
7117 Wilton Dr. N.E.
Cedar Rapids, IA 52402
(319) 373-0686
FAX: (319) 373-0548

IDAHO
Western Technical Sales
9444 Fairview Ave., Suite C
Boise, ID 83704
(208) 376-8700
FAX: (208) 376-8706

KANSAS
Technical Sales Associates, Inc.
601 N. Mur-Len Rd., Suite 19
Olathe, KS 66062
(913) 829-2800
FAX: (913) 829-9549

KENTUCKY
Arete Sales, Inc.
2250 Lake Ave., Suite 255
Ft. Wayne, IN 46805
(219) 423-1478
FAX: (219) 420-1440

LOUISIANA
Nova Marketing, Inc.
8350 Meadow Rd, Ste 174
Dallas, TX 75231
(214) 265-4600
FAX: (214) 265-4668

MAINE
Integrated Technology, Inc.
182 Main St.
Salem, NH 03079
(603) 898-6333
FAX: (603) 898-6895

MARYLAND
Tri-Mark, Inc.
1410 Crain Highway N.W.
Glen Burnie, MD 21061
(410) 761-6000
FAX: (410) 761-6006

MASSACHUSETTS
Integrated Technology, Inc.
182 Main St.
Salem, NH 03079
(603) 898-6333
FAX: (603) 898-6895

MINNESOTA
Electronic Component Sales
4974 Lincoln Dr.
Minneapolis, MN 55436
(612) 933-2594
FAX: (612) 933-9052

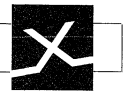
MISSISSIPPI
Rep, Inc.
11535 Gilleland Rd.
P.O. Box 4889
Huntsville, AL 35815
(205) 881-9270
FAX: (205) 882-6692

MISSOURI
Technical Sales Associates, Inc.
325 Paul Ave., Suite 100
St. Louis, MO 63135
(314) 521-2044
FAX: (314) 521-1359

NEBRASKA
Technical Sales Associates, Inc.
601 N. Mur-Len Rd., Suite 19
Olathe, KS 66062
(913) 829-2800
FAX: (913) 829-9549

NEVADA
Altek Components, Inc.
4635 S. Lakeshore Dr, Ste 122
Tempe, AZ 85282
(602) 345-4575
FAX: (602) 345-4100

Exis
2841 Junction, Suite 202
San Jose, CA 95134
(408) 456-1650
FAX: (408) 433-0450



NEW HAMPSHIRE
Integrated Technology, Inc.
182 Main St.
Salem, NH 03079
(603) 898-6333
FAX: (603) 898-6895

NEW JERSEY
MetroLogic Corporation
271 Rte 46 West, Ste D202
Fairfield, NJ 07006
(201) 575-5585
FAX: (201) 575-8023

CMS Marketing
527 Plymouth Rd, Ste 420
Plymouth Meeting, PA 19462
(215) 834-6840
FAX: (215) 834-6848

NEW MEXICO
Altek Components, Inc.
5005 Calle de las Chacras
Tucson, AZ 85718
(602) 529-0443
FAX: (602) 299-4707

NEW YORK
MetroLogic Corporation
271 Rte 46 West, Ste D202
Fairfield, NJ 07006
(201) 575-5585
FAX: (201) 575-8023

Quality Components, Inc.
116 Fayette Street
Manlius, NY 13104
(315) 682-8885
FAX: (315) 682-2277

Quality Components, Inc.
3343 Harlem Road
Buffalo, NY 14225
(716) 837-5430
FAX: (716) 837-0662

NORTH CAROLINA
Rep, Inc.
2500 Gateway Centre Blvd.
Suite 400
Morrisville, NC 27560
(919) 469-9997
FAX: (919) 481-3879

NORTH DAKOTA
Electronic Component Sales
4974 Lincoln Dr.
Minneapolis, MN 55436
(612) 933-2594
FAX: (612) 933-9052

OHIO
Norm Case Associates
21010 Center Ridge Rd.
Rocky River, OH 44116
(216) 333-0400
FAX: (216) 333-0403

OKLAHOMA
Nova Marketing, Inc.
8421 East 61st St.
Suite P, #139
Tulsa, OK 74133
(918) 660-5105
FAX: (918) 357-1091

OREGON
Western Technical Sales, Inc.
3720 S.W. 141st St., Suite 200
Beaverton, OR 97005
(503) 644-8860
FAX: (503) 644-8200

PENNSYLVANIA
CMS Marketing
527 Plymouth Rd, Ste 420
Plymouth Meeting, PA 19462
(215) 834-6840
FAX: (215) 834-6848

RHODE ISLAND
Integrated Technology, Inc.
182 Main St.
Salem, NH 03079
(603) 898-6333
FAX: (603) 898-6895

SOUTH CAROLINA
Rep, Inc.
2500 Gateway Centre Blvd.
Suite 400
Morrisville, NC 27560
(919) 469-9997
FAX: (919) 481-3879

SOUTH DAKOTA
Electronic Component Sales
4974 Lincoln Dr.
Minneapolis, MN 55436
(612) 933-2594
FAX: (612) 933-9052

TENNESSEE
Rep, Inc.
1908 Branner Ave.
P.O. Box 490
Jefferson City, TN 37760
(615) 475-4105
FAX: (615) 475-6340

TEXAS
Nova Marketing, Inc.
8350 Meadow Rd, Suite 174
Dallas, TX 75231
(214) 265-4600
FAX: (214) 265-4668

Nova Marketing, Inc.
10701 Corporate Dr., Suite 140
Stafford, TX 77477
(713) 240-6082
FAX: (713) 240-6094

Nova Marketing, Inc.
8310 N. Capitol of Texas Highway
Suite 182
Austin, TX 78731
(512) 343-2321
FAX: (512) 343-2487

UTAH
Seale & Associates
7535 E. Hampton Ave, #100
Denver, CO 80231
(303) 695-1980
FAX: (303) 695-1984

VERMONT
Integrated Technology, Inc.
182 Main St.
Salem, NH 03079
(603) 898-6333
FAX: (603) 898-6895

VIRGINIA
Tri-Mark, Inc.
1410 Crain Highway N.W.
Glen Burnie, MD 21061
(410) 761-6000
FAX: (410) 671-6006

WASHINGTON
Western Technical Sales, Inc.
13400 Northrup Way, Suite 14
Bellevue, WA 98005
(206) 641-3900
FAX: (206) 641-5829

Western Technical Sales, Inc.
E. 12015 Main St., Suite1
Spokane, WA 99206
(509) 922-7600
FAX: (509) 922-7603

WEST VIRGINIA
Norm Case Associates
2010 Center Ridge Rd.
Rocky River, OH 44116
(216) 333-0400
FAX: (216) 333-0403

WISCONSIN
KMA Sales Co.
2433 N. Mayfair Rd., Suite 202
Milwaukee, WI 53226
(414) 259-1771
FAX: (414) 259-0246

WYOMING
Seale & Associates
7535 E. Hampton Ave, #100
Denver, CO 80231
(303) 695-1980
FAX: (303) 695-1984

Domestic Distributors

Axis Components
6815 Flanders Dr.
San Diego, CA 92121
(619) 677-7950
FAX: (619) 677-7960

Additional location:
Sunnyvale, CA
(408) 522-9595

Bell Microproducts
1941 Ringwood Ave.
San Jose, CA 95131
(408) 451-9400
FAX: (408) 451-1600

Additional locations:
Irvine, CA
(714) 470-2900

Redmond, WA
(206) 861-5710

Richardson, TX
(214) 783-4191

Austin, TX
(512) 258-0725

Edina, MN
(612) 933-3236

Future Electronics
41 Main St.
Bolton, MA 01740
(508) 779-3000
FAX: (508) 779-5143

Additional locations:
Huntsville, AL
(205) 830-2322

Phoenix, AZ
(602) 968-7140

San Jose, CA
(408) 434-1122

Los Angeles, CA
(818) 865-0040

San Diego, CA
(619) 625-2800

Irvine, CA
(714) 250-4141
(714) 453-1515

Roseville, CA
(916) 783-7877

Lakewood, CO
(303) 232-2008

Cheshire, CT
(203) 250-0083

Altamonte Springs, FL
(407) 865-7900

Deerfield Beach, FL
(305) 426-4043

Largo, FL
(813) 530-1222

Norcross, GA
(404) 441-7676

Hoffman Estates, IL
(708) 882-1255

Indianapolis, IN
(317) 469-0447

Overland Park, KS
(913) 649-1531

Columbia, MD
(410) 290-0600

Bolton, MA
(508) 779-3000

Livonia, MI
(313) 261-5270

Grand Rapids, MI
(616) 698-6800

Eden Prairie, MN
(612) 944-2200

St. Louis, MO
(314) 469-6805

Albuquerque, NM
(505) 764-9470

Parsippany, NJ
(201) 299-0400

Rochester, NY
(716) 387-9550

Syracuse, NY
(315) 451-2371

Hauppauge, NY
(516) 234-4000

Raleigh, NC
(919) 790-7111

Concord, NC
(704) 455-9030

Mayfield Heights, OH
(216) 449-6996

Beavercreek, OH
(513) 426-0090

Beaverton, OR
(503) 645-9454

Marlton, NJ
(609) 596-4080

Florida, Puerto Rico
(800) 305-2343

Richardson, TX
(214) 437-2437

Houston, TX
(713) 785-1155

Austin, TX
(512) 502-0991

El Paso, TX
(915) 595-1000

Salt Lake City, UT
(801) 467-4448

Bothell, WA
(206) 489-3400

Brookfield, WI
(414) 879-0244

Vantage Components

A subsidary of
Bell Microproducts
1941 Ringwood Ave.
San Jose, CA 95131
(408) 451-9400
FAX: (408) 451-1600

Additional locations:
Billerica, MA
(508) 667-2400

Smithtown, NY
(516) 543-2000

Clifton, NJ
(201) 777-4100

Columbia, MD
(410) 720-5100

Washington, DC
(301) 621-8555

Amler, PA
(215) 540-4148

Altamonte springs, FL
(407) 682-1199

Deerfield Beach, FL
(305) 429-1001

Western Micro Technology

12900 Saratoga Ave.
Saratoga, CA 95070
(408) 725-1660
FAX: (408) 725-2958

Additional locations:
Burlington, MA
(617) 273-2800

Irvine, CA
(714) 450-0300

Agoura Hills, CA
(800) 634-2248

Marlton, NJ
(609) 596-7775

Beaverton, OR
(800) 634-2248

San Diego, CA
(619) 453-8430

Saratoga, CA
(408) 725-1660

Bellevue, WA
(206) 453-1699

INTERNATIONAL REPRESENTATIVES/DISTRIBUTORS

QuickLogic European Sales Office
Behringstrasse 10
D-82152 Planegg, Germany
+49-89-899 143-28
FAX: +49-89-857 7716

BELGIUM

Sonetech/Arcobel
Gulberg 33
5674 TE Nuenen
The Netherlands
+31-40-837075
FAX: +31-40-832300

Limburg Stirum 243, B-2
B-1780 Wemmel, Belgium
+32-2-4600707
FAX: +32-2-4601200

CANADA

Electronic Sales Professionals
5936 3rd Line Rd. North, RR#3
North Grower, Ontario KOA 2TO
(613) 489-3379
FAX: (613) 489-2778

60 Wilderness Dr.
Scarborough, Ontario M1V 3P6
(416) 321-9693
FAX: (416) 321-9794

27 Anne Court
Brampton, Ontario L6T 1K2
(416) 458-1103
FAX: (416) 458-4469

2102 Place Etienne-Brule
Montreal, Quebec H2B 1Z2
(514) 388-6596
FAX: (514) 388-8402

Future Electronics
237 Hymus Blvd.
Point Claire, Quebec H9R 5C7
(514) 694-7710
FAX: (514) 695-3707

Additional locations:
Quebec, Quebec
(418) 877-6666

Ottowa, Ontario
(613) 820-8313

Mississauga, Ontario
(905) 612-9200

Winnipeg, Manitoba
(204) 944-1446

Calgary, Alberta
(403) 250-5550

Edmonton, Alberta
(403) 438-2858

Vancouver, BC
(604) 294-1166

DENMARK

Mer-EL
Ved Klaedebo 18
2970 Hoersholm, Denmark
+45-42-571000
FAX: +45-42-572299

ENGLAND

Abacus Electronics
Abacus House, Bone Lane
Newbury, Berkshire
RG14 5SF, UK
+44-635-36222
FAX: +44-635-38670

FRANCE

MISIL Technologies
2, rue de la Couture
Silic 301
94588 Rungis Cedex, France
+33-1-45600021
FAX: +33-1-45600186

GERMANY

Scantec Mikroelektronik GMBH
Behringstrasse 10
D-82152 Planegg, Germany
+49-89-8598021
FAX: +49-89-8576574

Tannenbergstrasse 103
7312 Kircheim/Teck, Germany
+49-70-2183094
FAX: +49-70-2182568

Topas Elektronik GMBH
Striehlstrasse 18
30159 Hannover, Germany
+49-5-11131217
FAX: +49-5-11131216

Max Weber Strasse 16
D-25451 Quickborn, Germany
+49-410673097
FAX: +49-410673378

HONG KONG

Karven Electronics Ltd.
Rm. 807 Block A, MP Industrial Centre, 18 Ka Yip St.
Chai Wan, Hong Kong
+852-8983088
FAX: +852-8983312

INDIA

Silicon Automation Systems
3008, 12th "B" Main, 8th Cross
HAL 2nd Stage, Indiranagar
Bangalore 560-008, India
+91-80-5281229
FAX: +91-80-5284396

ISRAEL

Aviv Electronics Ltd.
4, Hayetzira St.
Ra'anana 43100, Israel
+972-9-983232
FAX: +972-9-916510

ITALY

Claitron S.p.A
Viale Fulvio Testi 280
20126 Milano, Italy
+39-2-661491
FAX: +39-2-66105666

JAPAN

Nippon QuickLogic k.k.
2-36-12 Yoga
Setagaya-ku, Tokyo 158
+81-3-57165930
FAX: +81-3-57165931

KOREA

CiNERGi Technologies & Devices
6th Floor Buyong Building
287-5 Yangjae-Dong, Seochu-Gu
Seoul 137-130, Korea
+822-5715781
FAX: +822-5715783

Seodu Logic, Inc.
Daeseong Bldg.
647-5 Yeoksam-Dong,
Gangnam-Gu, Seoul 135-080
Korea
+82-2-5636433
FAX: +82-2-5624657

NORWAY

Hans H. Schive
Undelestadlia 27
1371 Asker, Norway
+47-66-900900
FAX: +47-66-904484

SINGAPORE

CiNERGi Technologies & Devices
No. 57A Science Park Dr.
The Faraday, Singapore Science
Park 0511, Singapore
+65-7788211
FAX: +65-7731601

SPAIN

Unitronics, S.A.
Plaza Espana, 18-Pl. 9
28008 Madrid, Spain
+341-5425204
FAX: +341-5484228

SWEDEN

Martinsson Elektronik AB
Instrumentvagen 16
S-12609 Hagersten, Sweden
+46-8-7440300
FAX: +46-8-7443403

SWITZERLAND

Memotec AG
Gaswerkstrasse 32
CH-4901 Langenthal
Switzerland
+41-63-281122
FAX: +41-63-223506

TAIWAN

CiNERGi Technologies & Devices
12F-1 No:163, Keelung Rd. Sec.1
Taipei, Taiwan R.O.C.
+886-2-761-8555
FAX: +886-2-7665258

QuickLogic Corporation
2933 Bunker Hill Lane
Santa Clara, CA 95054
Phone: (408) 987-2000
Fax: (408) 987-2012
email: info@qlogic.com

Printed in U.S.A.

1025ESGD&A