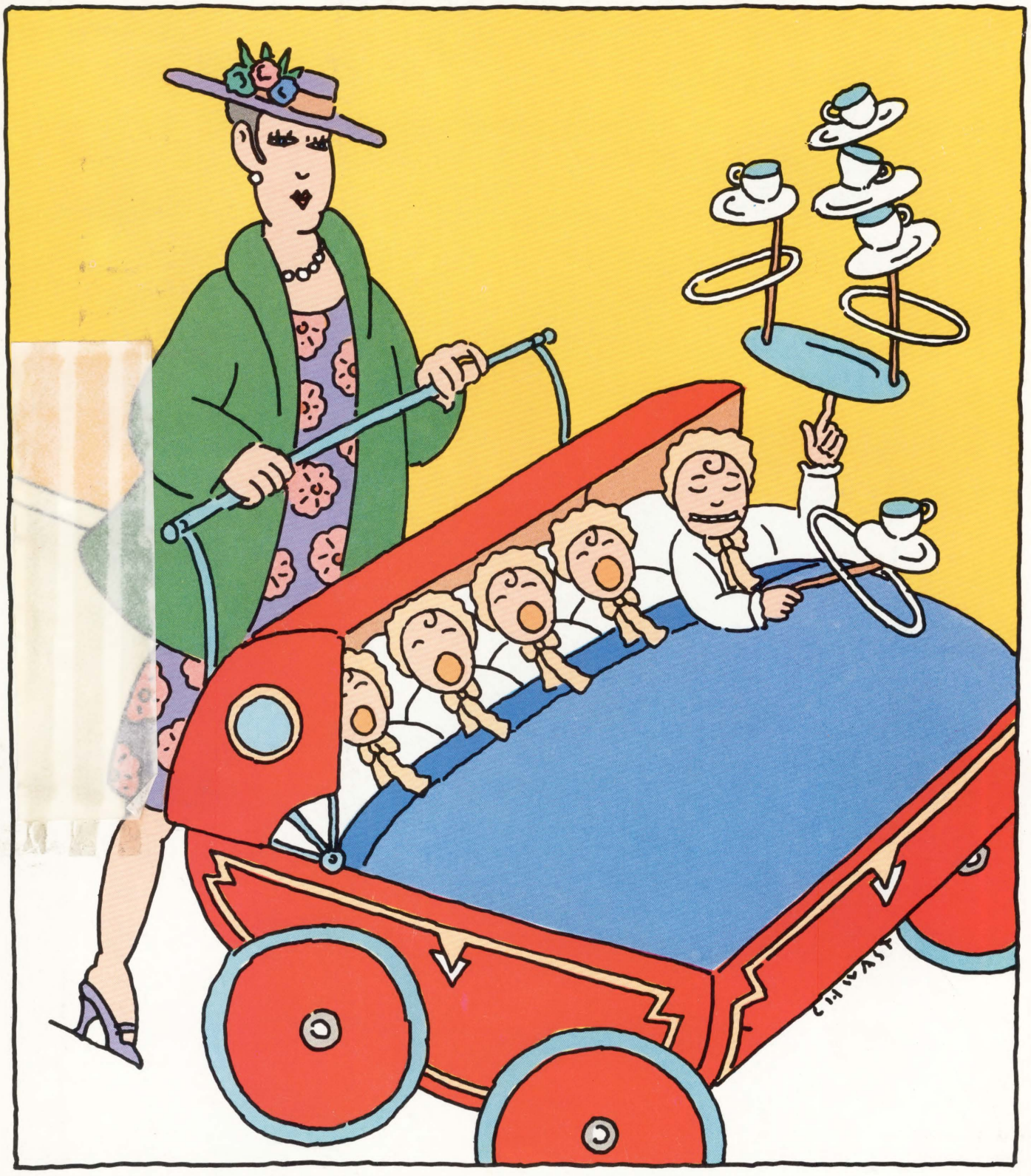
EIGHT DOLLARS A CMP PUBLICATION JUNE 1988

VLSI Systems Design

FOR DESIGNERS OF HIGH-PERFORMANCE SYSTEMS

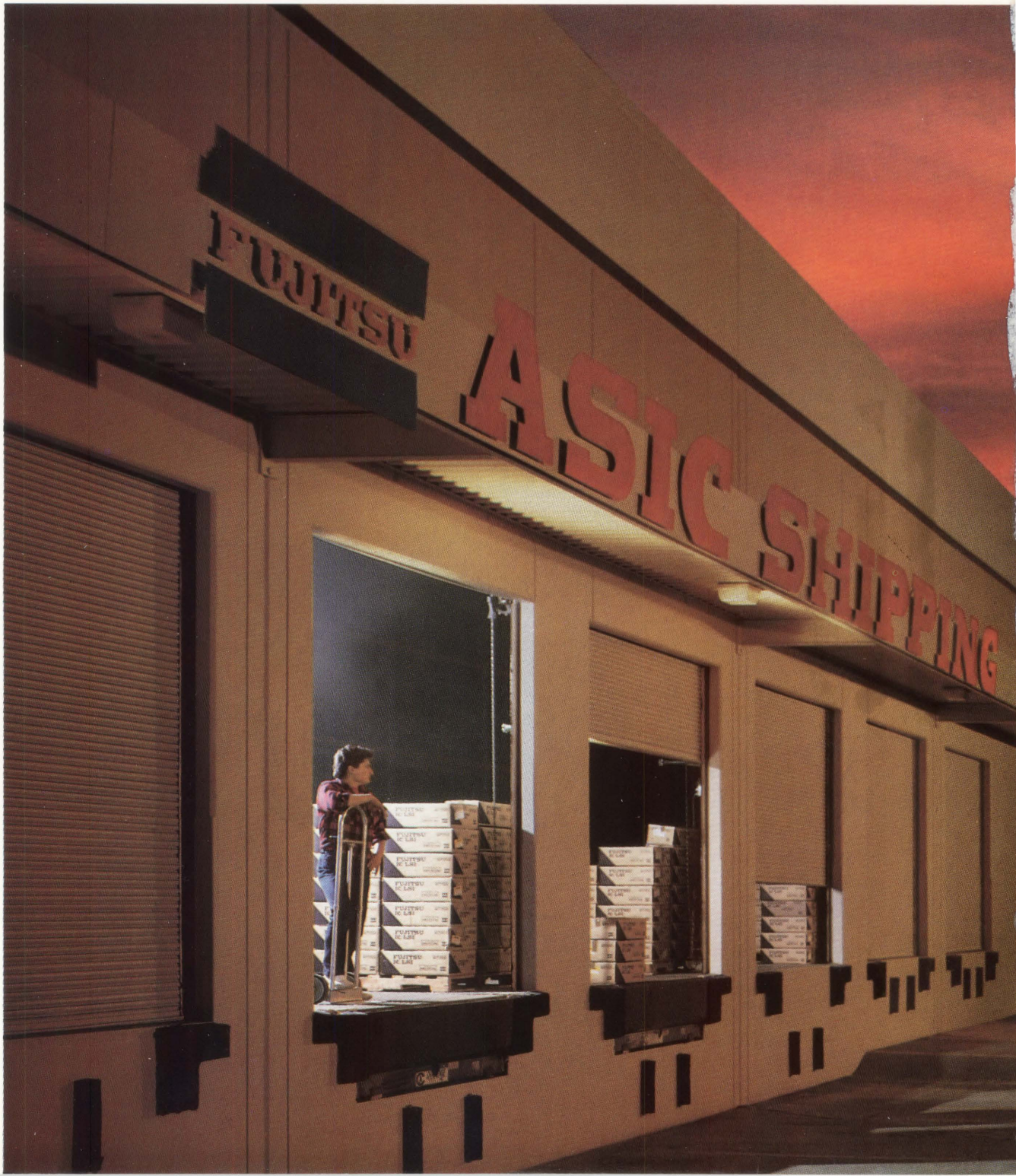


SOME RISC CHIPS ARE BORN PERFORMERS

THE POINT OF NO RETURN.

When you put a Fujitsu ASIC to work, you can rest assured it will work the way it should. And keep on doing its job for a very long time to come. In fact, when you look at our performance record over the years, you'll be hard pressed to find any field failures at all.

This is no empty promise. Product reliability has been a way of life for us for more than 15 years.



That's why we always take a conservative approach to the design process. Giving you realistic worst case specs that no production device will exceed.

Guaranteeing a minimum 90% utilization of all gates. And giving you a simulation-to-production correlation of 99%.

It's also why we control every step of the production process. From design to wafer fab to assembly and final test, including 100% AC testing at frequency. So nothing is left to chance.

To us, reliability in the field is everything. And when you remember we've taken over 8,000 ASIC devices from design through mass production, you can see that we'll give you a level of confidence no one else can offer.

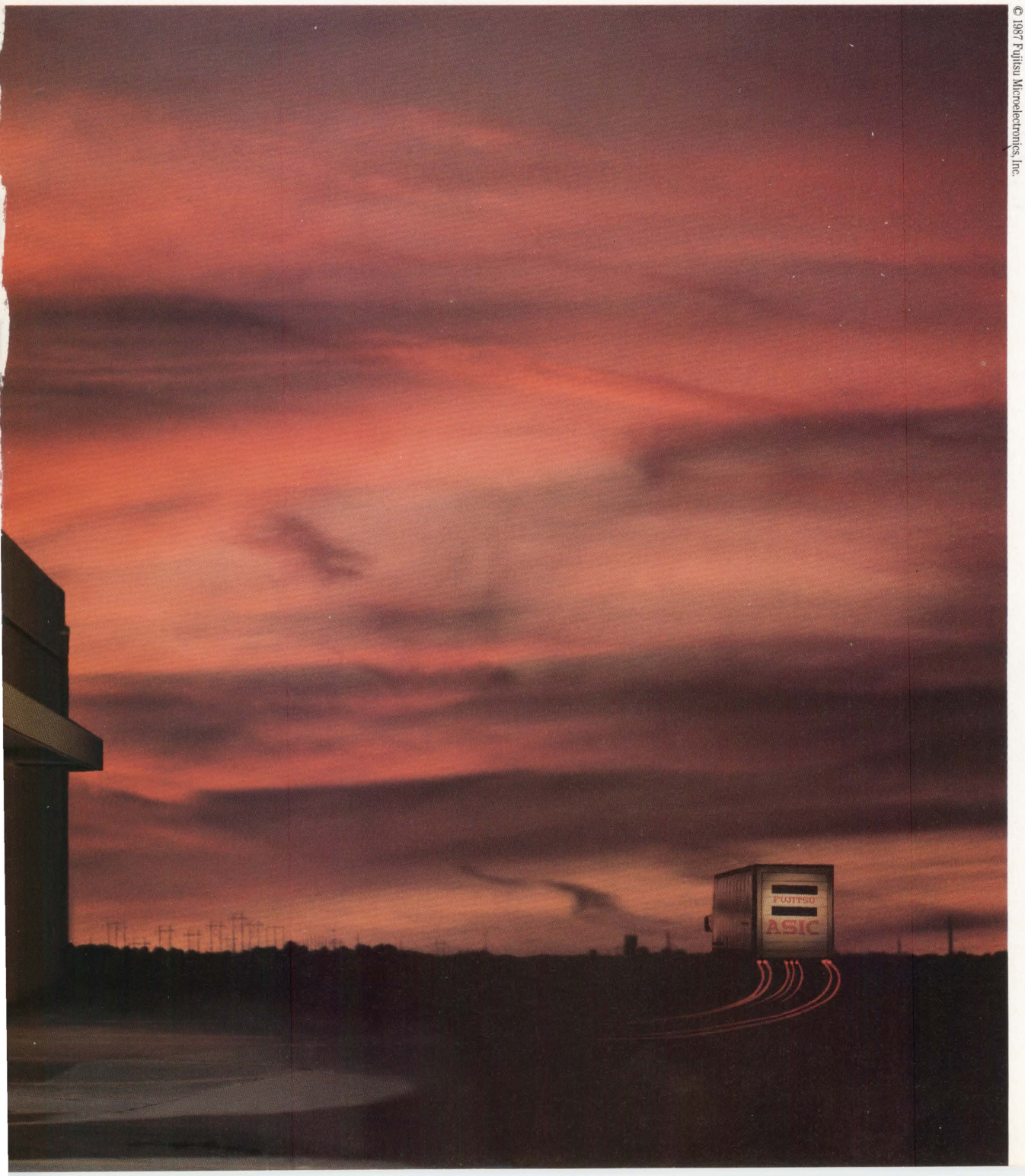
So count on parts that have longer life expectancies.

Call our Hot Line today at (800) 556-1234, Ext. 82; in California (800) 441-2345. Look into ASICs you can send out the door. Never to return again.

CIRCLE NUMBER 1

FUJITSU **FUJITSU**
MICROELECTRONICS, INC.

Technology That Works.



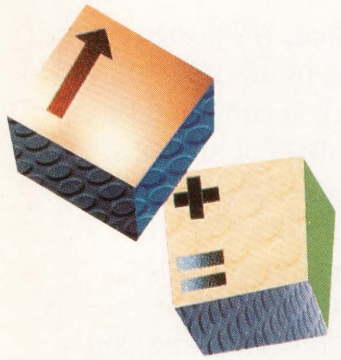
Contents

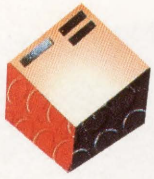
ARTICLES

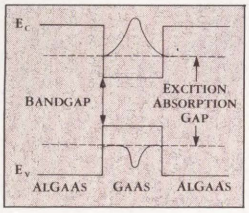
The right balance of RISC vs. CISC features can breed higher-performance RISC chips

80

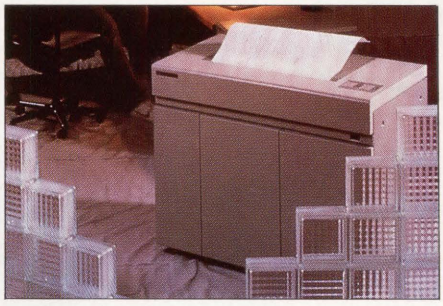
Real-time system development calls for a fine balance between hardware and software







DEPARTMENTS



COVER ILLUSTRATION BY SEYMOUR CHWAST

VLSI Systems Design

EDITOR/EDITORIAL DIRECTOR
Girish Mhatre

EXECUTIVE EDITOR
Roland C. Wittenberg

MANAGING EDITOR
Mike Robinson

SENIOR EDITOR
Bob Cushman

EDITOR EMERITUS
Roderic Beresford

EDITOR-AT-LARGE
Stan Baker

WESTERN REGIONAL EDITOR
David Smith

DIRECTORIES EDITOR
Michelle A. Losquadro

TECHNICAL ADVISERS
John A. Darringer
Jeffrey T. Deutsch
Edward J. McCluskey
Alan F. Podell
Daniel G. Schweikert
Susan L. Taylor

EDITORIAL PRODUCTION
Patricia L. Gaynor, Sr. Production Editor
Deborah Porretto, Ass't Production Editor

EDITORIAL ART
Terry Koppel, Art Director
Rosemary Intrieri, Designer
Marie D'Ippolito, Design Director

MANUFACTURING
David Getlen, Dir. of Manufacturing
Marie Myers, Production Manager
James Pizzo, Production Supervisor
Jane Mahoney, Asst. Production Supervisor
Vance Hicks, Coordinator

PUBLISHER
Norm Rosen

VLSI SYSTEMS DESIGN (ISSN 0279-2834) is published monthly with an extra issue in May by CMP Publications, Inc., 600 Community Drive, Manhasset, NY 11030. (516) 562-5000. VLSI SYSTEMS DESIGN is free to qualified subscribers. Subscriptions to others in the US: one year $60.00, two years $95.00; Canada and Mexico: one year $90.00, two years $165.00; Europe, Central and South America: one year $120.00, two years $225.00. Asia, Australia, Israel and Africa: one year $150.00, two years $285.00. Second-class postage paid at Manhasset, NY and additional mailing offices. POSTMASTER: Send address changes to VLSI SYSTEMS DESIGN, 600 Community Drive, Manhasset, NY 11030.

CMP ELECTRONICS GROUP
Kenneth D. Cron
Vice President/Group Publisher
Electronic Buyers' News
Electronic Engineering Times
VLSI Systems Design

CMP PUBLICATIONS, INC.

Michael S. Leeds, President
Pearl Turner, Vice President/Treasurer
Daniel H. Leeds, Vice President

Lilo J. Leeds, Gerard G. Leeds
Co-Chairpersons of the Board

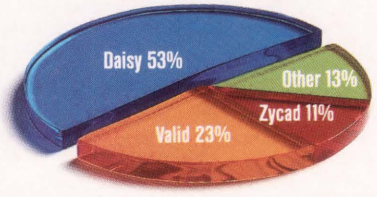
DO YOU HAVE WHAT IT TAKES TO MAKE IT BIG IN ASICS?

Not long ago, designing ASICs wasn't even part of your job. Now it's the part everyone's counting on.

That's where Daisy comes in.

Daisy CAE tools are used by more ASIC designers than any other CAE workstations.

Because from schematic creation through post-layout simulation, Daisy has what it takes to keep ASICs on time and on budget.



Simulation accelerator market share. Source: Prime Data, 1985 and 1986 unit shipments.

For example, our MegaLOGICIAN™ simulation accelerator is seamlessly integrated with the schematic, so you can locate and correct design problems interactively. That means faster debugging and more time to improve the quality of your design. Plus no other accelerator is as well supported, with more than 170 design kits supplied by 70 different vendors. So you can build productivity instead of libraries.

Which may explain why more MegaLOGICIANs are in use today than all other accelerators combined.

Speaking of combining, you

*Based on minimum 10x performance improvement compared to 32-bit workstations. Source for design kit estimates: VLSI Systems Design's Semicustom Design Guide, 1987.

can share a MegaLOGICIAN with a network of our 386-based desktop workstations, for a high-powered low cost ASIC design environment.

And that's just the beginning.

With our library of more than 4,500 system-level components, you can include your ASIC in complete "real world" system simulations to ensure that your designs will be ready for production, instead of revision.

All of which makes Daisy today's choice for no-sweat ASIC success.

But what about tomorrow?

Gate counts are on the rise. If your tools run out of steam at 5,000 gates, so could your future.

No problem.

Our ASIC design tools glide through 20,000-gate designs without even breathing hard.

In fact, new design kits already support arrays of over 100,000 gates.

So you'll never have to worry about hitting a dead-end.

But don't take our word for it, listen to what Rockwell and other industry leaders have to say. For a free copy of "Making It Big In ASICs" call Daisy at 1 (800) 556-1234, Ext. 32. In California, 1 (800) 441-2345, Ext. 32.

European Headquarters:
Paris, France (1) 45 37 00 12.
Regional Offices:
England (256) 464061;
West Germany
(89) 92-69060;
Italy (39) 637251.

daisy SYSTEMS CORP.

It Was the Best of Times, It Was the Worst of Times

Universal standards are a must for the design automation industry

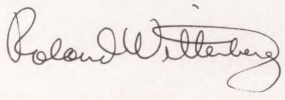
Although it was over 100 years ago when Charles Dickens began *A Tale of Two Cities* with "It was the best of times, it was the worst of times " to describe the conditions in London and Paris just before the French Revolution, the same statement describes the financial and mental state of today's electronic design automation industry.

This month the Design Automation Conference will celebrate its 25th anniversary. The design automation industry grew rapidly during these years, with the electronics segment blossoming at the start of this decade. Unfortunately, not all the players fared well. The losers included not only small vendors, but also industry giants that stumbled badly and whose efforts were eventually acquired whole, carved up into small pieces, or simply abandoned. But all is not lost: some vendors showed a remarkable ability to avoid the potholes on the road to success; some finally got their act together; and some were revived after successful mergers.

What caused the problems? Perhaps some industry leaders were like the royalty of Dickens's novel; they were too busy enjoying the best of times to realize their customers' needs and moods were changing. When many of the original CAE/CAD products were introduced, the IC designers—overwhelmed by the exploding complexity of ICs—were desperate for help, even from high-priced, user-unfriendly tools that were often late and didn't perform as promised. They did, however, help designers in their time of need.

But the tides changed; the early successes of the CAE/CAD tool vendors brought out new ventures in droves; every company went its own way, with different interfaces, operating systems, and computing platforms; and the growth of the IC design automation market faltered. The anticipated large-scale move of design automation tools into the much bigger system engineering market didn't materialize.

Nevertheless, it can be the best of times again if industry leaders look beyond their own palaces and make a serious effort to provide systems engineers with the universal standards they desparately need.



ROLAND WITTENBERG
EXECUTIVE EDITOR

There Will Still Be a Few Uses for Conventional ECL ASICs.

Cold facts: now the highest-density ECL logic array runs at a cool 1/10 the gate power of competing devices.

Raytheon's ASIC design expertise and proprietary technology make conventional ECL arrays too hot to handle. The superior performance of the new CGA70E18 and CGA40E12: the ECL logic array family with the highest density and the lowest power requirement now available.

☐ **Superior performance:** 300 pS delay and 300 μW (typical gate) power dissipation deliver the industry's lowest speed-power product: <0.1 pJ. Toggle frequency 1.2 GHz (typical).

☐ **Highest density:**
CGA70E18 — 12540 equivalent gates
CGA40E12 — 8001 equivalent gates

☐ **Lowest power:** Industry's smallest bipolar transistors result in power dissipation that is a fraction of conventional ECL at comparable propagation delays. Typical chip power dissipation of 3W to 5W.

☐ **Et cetera:** Interface TTL, ECL (10K, 10KH, 100K), ETL. Customer access to proven, fully integrated CAD system. Commercial and military operating ranges.

Call Raytheon for access to the right ECL technology. We're not blowing any smoke, and neither should your system's performance.

Raytheon Company
Semiconductor Division
350 Ellis Street
Mountain View, CA 94039-7016
(415) 966-7716

Access to the right technology

Raytheon

CIRCLE NUMBER 2

Calendar

Design Automation Conference '88

June 12–16
Anaheim Convention Center
Anaheim, Calif.

DAC '88, sponsored by the IEEE Computer Society and the Association for Computing Machinery, is devoted solely to the field of design automation. This year's conference will offer tutorials, panel discussions, and technical presentations. General session topics will include design for testability, VHDL in use, timing verification, parallel simulation, high-level synthesis, placement algorithms, layout compaction, logic synthesis and optimization, physical design verification, fault simulation, and micro-architecture synthesis. For more information, contact Pat Pistilli, MP Associates Inc., 7490 Clubhouse Road, Suite 102, Boulder, Colo. 80301. (303) 530-4333. ■

International Workshop on VLSI for Artificial Intelligence

July 20–22
University of Oxford
Oxford, England

This workshop will provide a forum where AI experts and VLSI system designers can come to discuss trends in AI applications and their computational requirements, VLSI implementations, and computer architectures. Topics to be discussed will include alternative technologies, functional-language architectures, knowledge-oriented machines, rule-based engines, Prolog and Lisp machines, and fifth-generation computers. Further information may be obtained by contacting, Dr. Jose G. Delgado-Frias or Dr. Will R. Moore, Dept. of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, England, U.K. Phone: (0865) 273188. ■

SIGGRAPH '88

August 1–5
Atlanta, Ga.

This 15th annual conference on computer graphics and interactive techniques is sponsored by the Association for Computing Machinery's Special Interest Group on Computer Graphics in cooperation with the IEEE Technical Committee on Computer Graphics. It will feature panel sessions, courses, and exhibitions, as well as a film and video show and an art show. Technical presentation topics will include algorithms, animation, CAD/CIM, applications, color, computational geometry, geometric modeling, graphics hardware, and graphics systems. Additional information may be obtained by contacting SIGGRAPH '88 Conference Management, Smith, Bucklin, and Associates Inc., 111 E. Wacker Dr., Suite 600, Chicago, Ill. 60601. (312) 644-6610. ■

OIS '88

September 7–9
Washington Sheraton
Washington, D.C.

The eighth annual Optical Information Systems Conference and Exhibition, sponsored by Meckler Corp., will focus on write-once and erasable optical storage systems and digital document-image automation. Sessions are planned in areas such as electronic image and document storage systems, erasable optical disk media developments, erasable optical disk drives and systems, evaluating and selecting a WORM subsystem, future trends and new developments, converging optical information technologies, and integrated systems development. Additional information about the conference may be obtained by contacting Marilyn Reed, OIS '88 Conference Manager, Meckler Corp., 11 Ferry Lane West, Westport, Conn. 06880. (800) 635-5537. ■

7th VLSI and GaAs Packaging Workshop

September 12–14
San Jose, Calif.

This workshop is being sponsored by the IEEE's Components, Hybrids, and Manufacturing Technology Society and the National Bureau of Standards. Topics that will be addressed include package interconnection options, GaAs IC packaging, die-attachment solutions for large chips, VLSI and wafer-scale package design, and VLSI package materials advancements. For additional information about the workshop, contact Paul Wesling, IEEE Council Office, 701 Welch Road, Suite 2205, Palo Alto, Calif. 94304. (415) 327-6622. ■

International Test Conference 1988

September 12–14
Sheraton Washington Hotel
Washington, D.C.

The ITC provides a major forum for an exchange of information about the testing of electronic devices, assemblies and systems. This year the conference focuses on the test techniques and equipment needed to meet the challenges of new technologies. Technical papers will be presented on such topics as analog devices, yield modeling and process diagnosis, testability analysis, education and training, application-specific devices, micro-

Continued on page 113

ILLUSTRATION BY RENEE KLEIN

At 100 MHz, the IMS Logic Master XL creates a new world standard for ASIC verification.

But speed isn't everything.

You might think it sufficient that we're introducing the highest performance ASIC verification system available anywhere.

And that our new Logic Master XL is the first system with data rates of 100 MHz.

But we didn't think that was enough.

Instead, we concentrated on the kind of features that turn superior performance into superb productivity.

Like advanced driver electronics which produce symmetrical waveform edges for measurement accuracy.

And complete per-pin control. So you can handle complex timing configurations with ease. And mixed technology devices.

Plus automatic wiring for rapid, error-free setup, and superb system skew for greater measurement confidence.

Best of all, the Logic Master XL comes from Integrated Measurement Systems, the international leader in ASIC verification.

Contact us for more information.

IMS

Integrated Measurement Systems, Inc.
Corporate Headquarters
9525 S.W. Gemini Drive
Beaverton, Oregon 97005
TEL 503 626-7117

CIRCLE NUMBER 3

Photo courtesy of NASA.

U.S. Regional Offices: Boston, MA 617-366-3676; Irvine, CA 714-768-5808; Portland, OR 503-626-7117. **International Offices:** Belgium 02-219-2453; Canada 416-890-2010; France 01-4534-7535; Hong Kong 5-701181; Japan 0423-33-8111; Sweden 753-38465; Switzerland 830-5040; Taiwan 02-501-0055; The Netherlands 01-40-582911; United Kingdom 06-284-76741; West Germany 5722-2030.

BIT Stream

Calma Unbundles IC CAD for a Valid Deal

VALID LOGIC SYSTEMS Inc. (San Jose, Calif.) has purchased Calma Co. (Milpitas, Calif.), whose GDS II design system for ICs was the granddaddy of most modern IC CAD systems, from General Electric Co. for cash and stock valued at less than $3.5 million. Valid will consolidate its IC CAD product line and the Calma group into an IC CAD division and will assume responsibility for Calma's GDS II and newer EDS III design systems. The deal became a reality when W. Douglas Hajjar, Valid's president and CEO, convinced GE executives to unbundle the IC CAD division from the mechanical division. ■

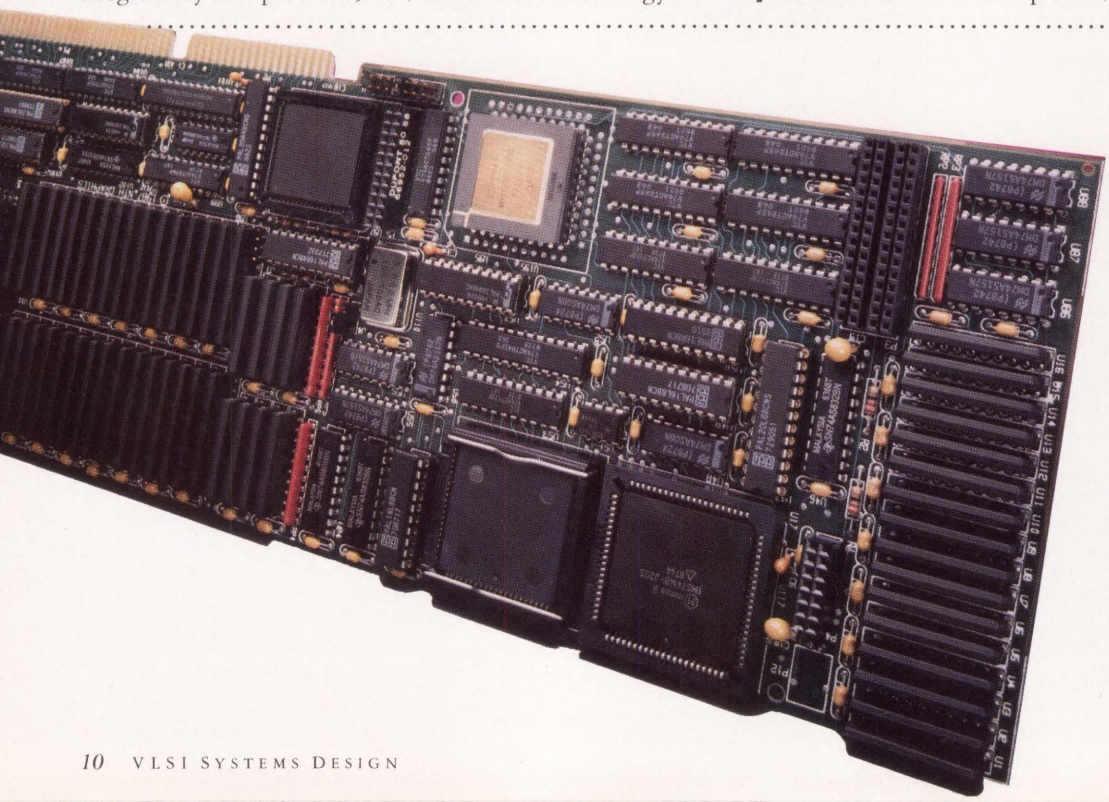
State Machine EPLD Clocks at 50 MHz

THE FIRST of a family of 28-pin EPLDs, the CY7C330 Programmable State Machine from Cypress Semiconductor Inc. (San Jose, Calif.), can implement state machines that operate with a 50-MHz clock. The new chip contains 11 dedicated inputs and 12 I/O macrocells, both with registers; four buried registers; and a logic array that provides 9 to 19 product terms for each output macrocell. Two input clock pins synchronize the arrival of input signals, and a third clock input drives the output and buried registers. The input register setup time is 5 ns, and the clock-to-output delay is 15 ns.

The new EPLD is built with Cypress's 0.8-μm dual-layer-metal CMOS technology. ■

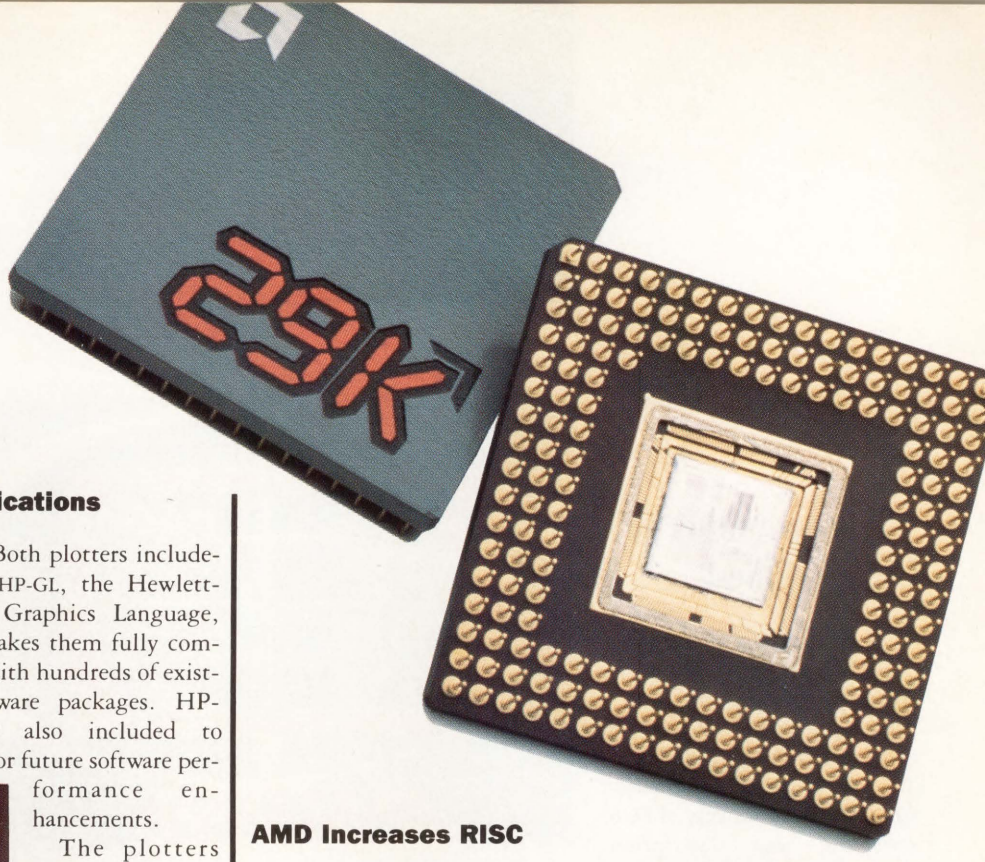
Major Computer Companies Fight for Open Unix

SEVEN GIANTS in the computer industry put their money on the line for a "truly open Unix." The chief executives of IBM, Hewlett-Packard, Digital Equipment, Apollo Computer, Siemens, the Bull Group, and Nixdorf expressed their dissatisfaction with the AT&T-Sun Microsystems handling of the development of a unified Unix operating system by making a three-year, $90 million commitment to an international effort directed toward open software standards. The efforts will be channeled through the newly formed Open Software Foundation. John L. Doyle, executive vice president of HP's Systems Technology Division, will serve as the OSF's first chairman. ■



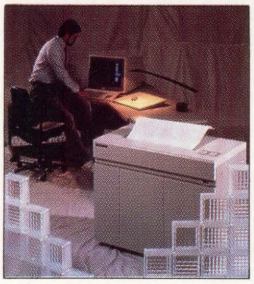
Super-3D on a PC

A NEW DISPLAY controller board from Nth Graphics (Austin, Texas) delivers 3D color graphics performance at a superworkstation level when plugged into an IBM PC AT. The Nth 3D Engine, priced from $13,000, uses Inmos Transputers, in addition to the company's proprietary graphics coprocessors, to provide the required computing power. The board, which drives color monitors with a 1,024 × 768-pixel resolution at a 60-Hz non-interlaced refresh rate, can handle a variety of complex graphics operations, such as transforming 5,000 constant-shaded, 500-pixel polygons per second. ■



HP Plotters Target CAE/CAD Applications

TWO LOW-PRICED electrostatic plotters have been introduced by Hewlett-Packard Co. The HP 7600 series, Models 240D and 240E plotters can typically produce mechanical engineering type drawings in less than one minute. HP says that the Model 240D is the only electrostatic plotter priced at less than $23,000 that features pen-plotter quality with a resolution of 406 dots per inch. The plotter creates drawings on D-sized paper or vellum.



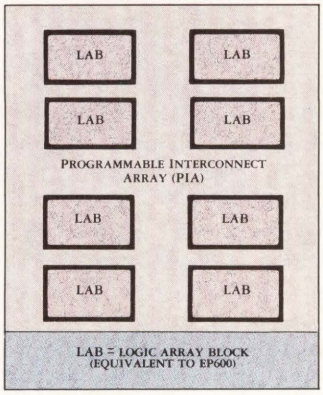
The Model 240E, which is priced at $27,500, has the same resolution as the 240D, but handles the larger E size format. Both plotters include built-in HP-GL, the Hewlett-Packard Graphics Language, which makes them fully compatible with hundreds of existing software packages. HP-GL/2 is also included to provide for future software performance enhancements.

The plotters can connect to a wide variety of host computers through their RS-232-C, IEEE-488, and Centronics parallel interfaces. The host central processing unit is offloaded by the 40-megabyte storage capacity of the built-in vector-to-raster converter, which can hold up to 3 million vectors. ■

Altera PLDs Evolve to the Max

A NEW PLD architecture developed by Altera Corp (Santa Clara, Calif.) combines logic array blocks (LABs) with a programmable interconnect array (PIA) that connects signals among blocks and I/O pads. The architecture takes form in the Max family of devices, ranging from the EPM5016 with 20 pins and 16 macrocells to the EPM5128 with 68 pins and 128 macrocells.

Each LAB contains macrocells (with an AND-OR logic array and flip-flops), "logic expanders" that expand the number of available product terms, and I/O functions that can drive either I/O pins or other macrocells to create buried resources. The combination of new architecture and 0.8-μm CMOS process technology (from second-source Cypress Semiconductor) allows the parts to operate with 40-MHz system clock frequencies. The Max family and the Max+ design system will be available in the second half of this year. ■



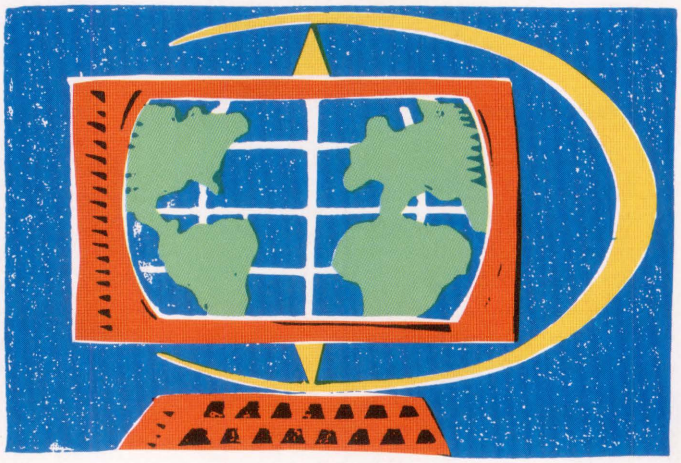
AMD Increases RISC

ADVANCED MICRO Devices Inc. (Sunnyvale, Calif.) is now sampling a 30-MHz, 20-MIPS version of its Am29000 RISC processor, six months ahead of schedule. It has also promised three support chips by year-end: the Am29027 floating-point processor; the Am29062 integrated cache unit, with cache memory and control logic; and the Am29041 data transfer controller, which implements DMA and I/O functions. Next year expect to see a 50-MHz, 35-MIPS version of the 29000. ■

VLSI Technology and Hitachi Make a Swap

WILLIAM GSAND, vice president and general manager of Hitachi America Ltd.'s Semiconductor and IC division, and Alfred J. Stein, chairman and CEO of VLSI Technology Inc. (San Jose, Calif.), shook hands on a deal that took 18 months to negotiate. The five-year agreement gives Hitachi rights to all of VLSI Technology's broad line of proprietary IC physical design tools, silicon compilers, megacells, and standard-cell libraries. In return, VLSI will obtain manufacturing know-how on Hitachi's 1-μm and smaller-geometry CMOS processes. Other provisions are second-source rights to Hitachi's manufacturing technologies and the availability of VLSI's tools to Hitachi's customers. ■

News Analysis



The Dual Nature of Logic Synthesis

The term *logic synthesis* implies the creation of a circuit structure directly from a high-level description. A crop of programs to be first shown at the Design Automation Conference comes very close to satisfying that description. From a point high in the design hierarchy, they automatically translate and optimize designs that can then be implemented in semicustom ICs.

Translation: The designer creates high-level models—register-transer-level (RTL) blocks, behavioral descriptions, or equations to represent portions of his design. The synthesizer replaces each of those blocks with lower-level components that can be implemented in the target technology. Often the high-level description must first be translated into more malleable forms. For example, a state machine may be divided into a sequential block and a logic block, both of which can be implemented directly in lower-level components.

Optimization: Once the design has been translated into lower-level components, the tools try to optimize the implementation. They use signal path delay, power consumption, and silicon area (in terms of total number of gates) to judge whether a change in the design results in a better implementation.

All tools contain some optimization routines that perform minimization using Boolean algebra. Espresso is the most prevalent algorithm for such optimization. Minimization usually reduces silicon area and power consumption because it reduces the total number of components in the implementation. The tools can optimize across entire designs, or the designer can limit them to optimize portions of designs within specified boundaries.

To minimize propagation delay, logic synthesis tools must incorporate timing analysis programs to measure the delay along signal paths. One strategy for reducing signal delays rearranges logic expressions to minimize components along a signal path.

■ Timing in Critical Paths

Another technique replaces components in critical signal paths with higher-drive alternatives. When timing is improved through fewer components, silicon area and power consumption may improve as well. More often, though, faster designs obtain their speed through the use of higher-drive components and are therefore a little larger and consume more power than slower implementations. Logic synthesis is designed to be complementary to, instead of a replacement for, existing CAE environments. The tools create netlists or schematics (or both) in the format for Mentor Graphics workstations (and some others). Designers then verify the functionality and timing of the design, as well as the placement and routing, using existing tool sets.

■ New Tools Define the Art

Three logic synthesis programs are becoming available that execute both functions of synthesis: the Design Consultant, from Trimeter Technologies Corp. (Pittsburgh, Pa.); the Design Compiler, from Synopsys Inc. (Mountain View, Calif.); and SilcSyn, from Silc Technologies Inc. (Burlington, Mass.). Another new logic synthesis tool, the DS23 design program from Xilinx Inc. (San Jose, Calif.), performs only optimization, and it produces designs solely for Xilinx's programmable gate arrays.

Trimeter's Design Consultant (Figure 1) accepts designs in five forms: register-transfer-level designs, state machines, Boolean equations, PLA truth tables, and schematic diagrams. It evaluates the design using two knowledge bases. It first performs optimization at the RTL level, making architectural and structural trade-offs that are independent of the implementation technology. Then the company's Logic Consultant, which is embedded in the Design Consultant, makes trade-offs that are specific to the implementation technology.

To the Logic Consultant, the Design Consultant adds the additional input forms and the translation of those input forms into equations, state machines, and netlists for implementation. It produces designs for gate arrays from LSI Logic, Toshiba, and Hughes. Preliminary release is planned by the beginning of the fourth quarter of this year.

The founders of Synopsys developed the Design Compiler's progenitor, Socrates, while at General Electric's Research Triangle Park facility (for a description of Socrates, see *VLSI Systems Design,* January 1988, p. 40). After five years of experience with Socrates, Synopsys has begun to encode some of the knowledge base in algorithms instead of expert rules. Algorithms, which can be considered highly structured, complex rules, execute more quickly than rules.

Some of the Design Compiler remains rules-driven, because it is easier to create and modify a knowledge base in terms of rules than it is in algorithms. Once a set of rules becomes tested and verified over time, however, the effort

Continued on page 112

Illustration by Renee Klein

ARRAY FOR BiCMOS!

180 MHz with low power.

It's cause for celebration. AMCC extends its lead as the high performance/low power semi-custom leader with three exciting, new BiCMOS logic arrays that optimize performance where today's designs need it most. In throughput (up to three times faster than 1.5µ CMOS).

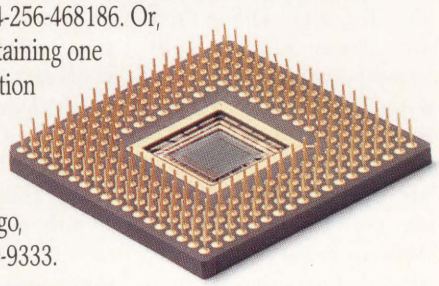
Today, system designers look at speed, power and density. For good reasons. As CMOS gate arrays become larger and faster, designers can't meet their critical paths due to fanout and interconnect delay. As Bipolar arrays become larger and faster, power consumption becomes unmanageable. So AMCC designed a BiCMOS logic array family that merges the advantages of CMOS's low power and higher densities with the high speed and drive capability of advanced Bipolar technology. Without the disadvantages of either.

Our new Q14000 BiCMOS arrays fill the speed/power/density gap between Bipolar and CMOS arrays. With high speed. Low power dissipation. And, mixed ECL/TTL I/O compatibility, (something CMOS arrays can't offer).

Q14000 SERIES			
	Q2100B	Q9100B	Q14000B†
Equivalent Gates	2160	9072	13440
Gate Delay* (ns)	.7	.7	.7
Maximum I/O Frequency (MHz)	180	180	180
Utilization	95%	95%	95%
Power Dissipation (W)	1.8	4.0	4.4
I/O	80	160	226
Temperature Range	COM, MIL	COM, MIL	COM, MIL

*(2 loads, 2 mm of metal) †Available soon

For more information on our new BiCMOS logic arrays, in the U.S., call toll free (800) 262-8830. In Europe, call AMCC (U.K.) 44-256-468186. Or, contact us about obtaining one of our useful evaluation kits. Applied MicroCircuits Corporation, 6195 Lusk Blvd., San Diego, CA 92121. (619) 450-9333.



A Better BiCMOS Array is Here.

AMCC

CIRCLE NUMBER 4

The 29000's Architect Believes in Evolution

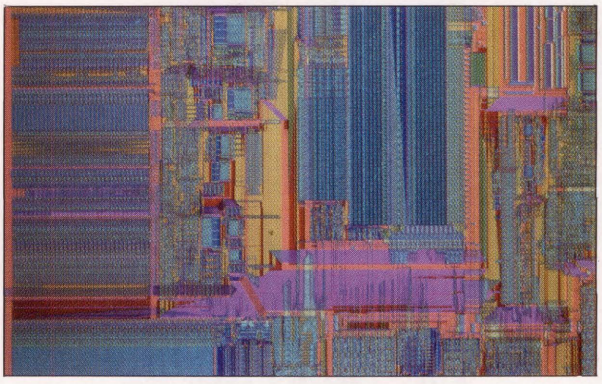
A Darwinian Approach to Microprocessor Design

A STAUNCH proponent of an evolutionary approach, Mike Johnson learned the hard way about how to design a microprocessor. A survivor of the RISC microprocessor program at IBM, he received a chance to create a new architecture at Advanced Micro Devices. The result, the Am29000 RISC processor, didn't pop out from a product specification; instead, it evolved.

Chance, a key factor in evolution, steered Mike into processor design to begin with. For example, as a master's candidate at Arizona State, he had specialized in digital control and signal processing. This expertise suited him well for developing magnetic-card typewriters at his new employer, IBM. However, on leaving college, his car broke down, forcing him to scrap his travel plans. Showing up early at IBM, Mike was placed in the next available slot: microprocessor design. "It was a total accident," he says.

Mike was the first of a group whose charter was to design IBM's next generation of microprocessors. The Research Office-Products Mini Processor (ROMP) project faltered because it was started too early to benefit from technology trends and too late to begin from scratch.

"ROMP suffered a lot from being the bridge between the old and the new," Mike says. For example, the designers toiled to design an interleaved memory interface that subsequent advances in DRAM technology would make not only unnecessary but also cumbersome.



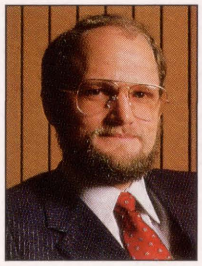
Also, ROMP was originally a 24-bit architecture, because, he says, IBM "didn't have the guts to go directly from 16 to 32 bits." A year after its inception, the architecture did expand to 32 bits; in addition the designers then had to retrofit virtual memory onto it. The experience in trying to tune the stubborn ROMP architecture taught him "what you don't do" in creating a microprocessor.

When Mike joined the team developing the 29000, he used a model for the project that could be pulled from the notebooks of Charles Darwin. He explains: "Biological things progress by changing randomly, with many intermediate forms dying off. Processors can change randomly too, and intermediate steps don't need to live. I can have an intermediate stage that is this long and this wide and has pink polka dots on it—it doesn't matter. But an intermediate stage gives ideas that continue the evolution."

As a result, the 29000 started as an infinitely large chip from which the designers subtracted functions, instead of being built up piece by piece. "It's creating the chip instead of designing it," Mike says.

In addition to evolution, another philosophy emerged from his ROMP days: Don't design all functions in hardware or the processor can get in the way of the final system design. "Saying 'let me do that in hardware' doesn't always make things faster," he says.

'PROCESSORS CAN CHANGE RANDOMLY TOO,



AND INTERMEDIATE STEPS DON'T NEED TO LIVE'

System-level technology, such as software design, changes faster than designers can implement functions in hardware. Processor designs should implement a set of flexible primitive functions that execute as fast as possible. The system designer then plugs them together, using the most recent ideas in system design.

As an illustration of both philosophies, Mike points to the 29000's large register set. It began as 32 registers and progressed through numerous intermediate stages, including a 64-register file with cache backup to a 256-register file. It was finally pared down to 192 registers. No restrictions are placed on the use of the registers, so the system designer can use them to best advantage in the type of system he is building.

This RISC concept of implementing only primitive functions in hardware was difficult to accomplish because of human nature. "We had to slap each other's hands to keep features out of the design," Mike explains.

His hands, as well as the rest of him, are now at Stanford University, where he's investigating parallel processing at the instruction level that will result in microprocessors that can execute more than one instruction each clock cycle. So Mike Johnson, it seems, continues to be closely tied to the evolution of microprocessors. ■

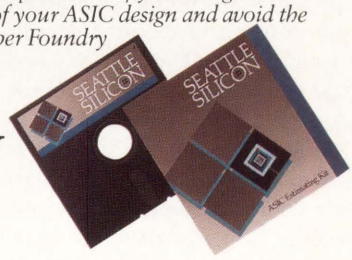
—DAVID SMITH

Feel trapped by one ASIC supplier? Reach for the Super Foundry.

Now you can design complex ASICs to meet your needs, not to fit one supplier's limited capabilities. The Super Foundry™ offers the combined resources of the world's premier CMOS manufacturers. Our process independent design tools and production services give you maximum freedom and control—from concept through delivery.

FREE. ASIC Estimating Kit.

What will it take to do your ASIC? With our free ASIC Estimating Kit, you can analyze design-process trade-offs and explore the performance, cost and scheduling implications of your design in a variety of processes. Take control of your ASIC design and avoid the single-supplier prison. Call the Super Foundry for your free ASIC Estimating Kit: ***1-800-FOR-VLSI ext. 200.***



THE SUPER FOUNDRY

SEATTLE SILICON

3075-112th Ave NE., Bellevue, WA 98004, (206) 828-4422.

CIRCLE NUMBER 5

System-level simulation hinges on accurate models

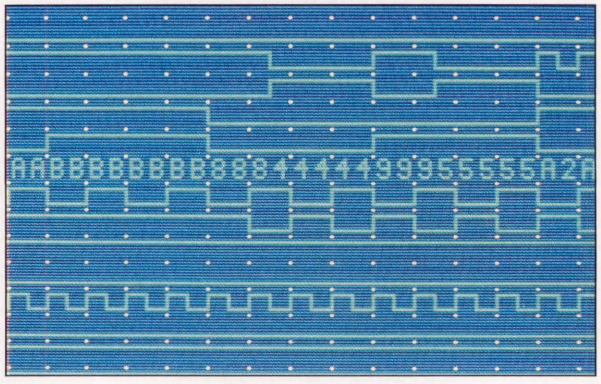
How Accurate Must Beavioral Models Be?

WILLIAM W. LATTIN, LOGIC AUTOMATION INC., BEAVERTON, ORE.

AS WE approach the 1990s, many factors are driving companies to find faster, more efficient methods of design verification. Burgeoning system complexity and shrinking design cycles are the most imperative factors. In addition, more and more systems are incorporating ASICs. Although it's true that 99% of the time an engineer can make an ASIC meet its design specification on the first pass, only 50% go into production; the other 50% don't function correctly in their systems. This situation is prompting the need for logical simulation of ASICs within a complete board design, and as a result, system-level simulation—the logic simulation of board- and system-level products—is becoming very popular.

The efficacy of system-level simulation hinges on the availability and accuracy of simulation models. Understandably, users of simulators want the behavior of device models to match the behavior of the devices exactly. To realize this state, some users argue, models should be certified by the IC vendors themselves. Unfortunately, it is both a mathematical and a physical impossibility that 4,000 to 5,000 lines of code can represent a silicon device exactly, replicating its unspecified deviant behavior as well as its specified good behavior. Therefore we must ask what "accuracy" means.

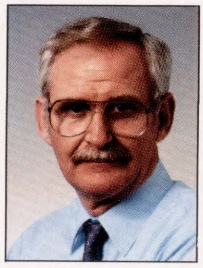
Most semiconductor companies will not certify the accuracy of even their own data sheets. It isn't reasonable, then, for them to assure users that 4,000 to 5,000 lines of code will faithfully represent all possible responses of a quarter of a million transistors interconnected on a chip. What *is* possible is to submit the model to the same characterization and test suites that the semiconductor vendor runs on the device. This approach verifies the model's behavior for the *specified* behavior of the chip. It does not model the *unspecified* or deviant behavior of the chip.



Developers of models must work very closely with semiconductor companies and use the same test vectors on the model of the chip that are used on the device itself. At Logic Automation, for example, we have relationships with most leading semiconductor vendors to exchange proprietary test vectors and technical information. These relationships help ensure the highest possible accuracy of our behavioral models.

Most importantly, a behavioral model that is verified in the same manner as the device it models allows a designer to determine whether a device is being used correctly in a system design. The model performs to the device's specified behavior, which is what a designer needs. The system designer doesn't need the model to re-create the device's unspecified behavior—he doesn't use that behavior when he defines the functionality of his system. So the model, despite not matching 100% of the device's behavior, should meet the designer's needs completely.

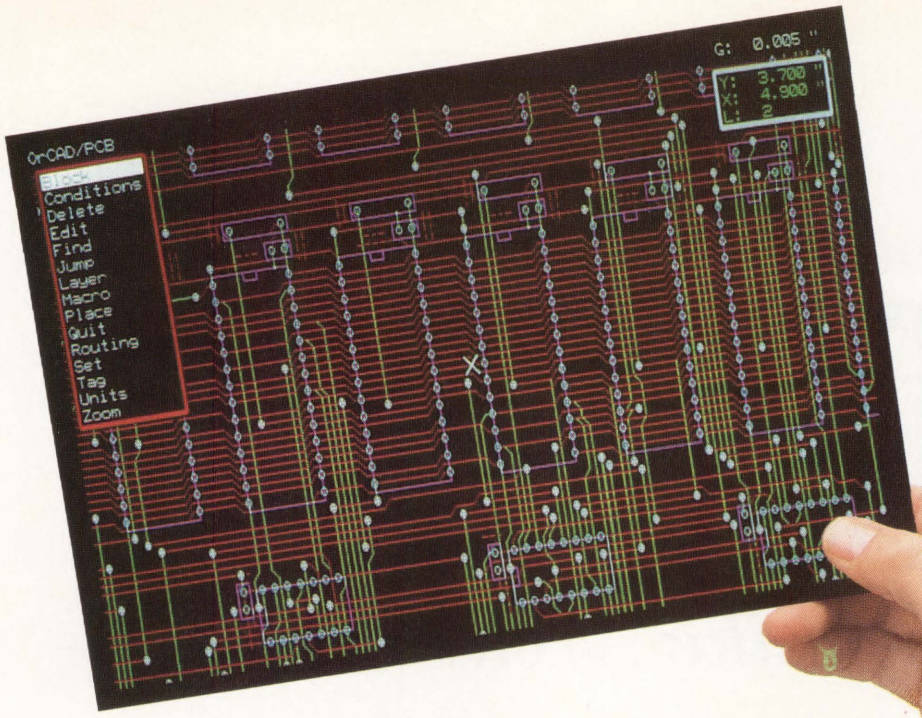
'THE SYSTEM DESIGNER DOESN'T NEED THE MODEL TO RE-CREATE THE DEVICE'S UNSPECIFIED BEHAVIOR'



We are often asked, "Do these 4,000 to 5,000 lines of code exactly represent the chip?" My response is, "No, they don't; in fact, the software model does more than the chip does." The reason is that our software models actually issue an error message when it is being used improperly—a feat that no chip can do. Therefore, I think this issue is similar to an argument that arose in the old days regarding the accuracy of testing the chips themselves. If you think of a complex microprocessor with 300,000 transistors, a complete test would contain $2^{300,000}$ vectors that set each transistor into an on and off state. We all know that that is not only impractical but impossible. The argument about the accuracy of software models for simulation is analogous.

The real value of behavioral models does not arise from having them represent every transistor in the chip with 100% accuracy. Their value stems from the fact that behavioral models allow designers to use system simulation to control the complexity and reduce the time of their design projects. ■

WILLIAM W. LATTIN *is the president and chief executive officer of Logic Automation. He has been involved with simulation and modeling for over 20 years, playing a key role in the introduction of design automation at both Motorola and Intel. Prior to Logic Automation, he was the vice president and general manager of Intel's Systems Group.*



PCB LAYOUT ON YOUR PC. FOR ONLY $1495. OrCAD PCB COMES TO LIFE.

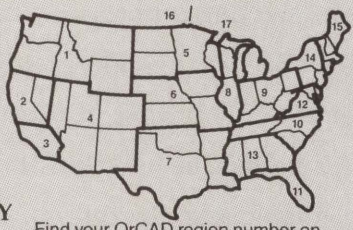
With nearly 10,000 systems already in use, electrical engineers like the no-nonsense capability of OrCAD/SDT schematic capture. Well, now there's even more to like. Introducing OrCAD/PCB. A fast, easy-to-use PCB layout package that runs on your PC. And costs only $1,495.

A NO-NONSENSE DESIGN TOOL FOR NO-WAIT DESIGNERS.

OrCAD/PCB pumps up your productivity. Pop-up menus appear in a flash. Macros speed repetitive operations. Bit-mapped graphics make redraws, zooms and pans instantaneous. And, since you know SDT, you have no new commands to learn; PCB's interface is exactly the same.

POWER AND FLEXIBILITY MINUS CLUTTER.

OrCAD/PCB produces structured, elegant PCB layouts. It auto-routes boards up to 32″x32″ with up to 8,000 track segments and 16 layers, taking into account manually routed connections. And, it adapts to your design with features like:



Find your OrCAD region number on this map; then locate your sales and support representative on the list below.

- Selectable track and via widths from 0.001″ to 0.255″.
- Square, rectangular, round, elliptical, and SMD pads.
- Grid bases of 100, 50, 25, 10 and 5 mils. Or, go off grid to 1 mil.
- Support for digital and analog components and surface-mount devices.
- Ratsnest and force vector placement.
- Definition of board edge, forbidden zones, and copper zones.

GIVE OUR DEMO DISK A SPIN.

If you need affordable, no-nonsense PCB layout, call for a free OrCAD/PCB demo disk. Not an OrCAD user? Call for the SDT demo disk while you're at it (OrCAD/SDT costs only $495). Then, watch your PCB layouts take shape.

OrCAD sales and support representatives.

1. WA, OR, MT, ID, AK
   Seltech, Inc.
   206-746-7970
2. N. CA, Reno NV
   Elcor Associates, Inc.
   408-980-8868
3. S. CA
   Advanced Digital Group
   714-897-0319
4. Las Vegas NV, UT, AZ, NM, CO
   Tusar Corporation
   602-998-3688
5. ND, SD, MN, W. WI
   Comstrand, Inc.
   612-788-9234
6. NE, KS, IA, MO
   Walker Engineering, Inc.
   913-888-0089
7. TX, OK, AR, LA
   Abcor, Inc.
   713-486-9251
8. MI, E. WI, IL
   Cad Design Systems, Inc.
   312-882-0114
9. IN, OH, KY, WV, W. PA
   Frank J. Campisano, Inc.
   513-574-7111
10. TN, NC, SC
    Tingen Technical Sales
    919-878-4440
11. FL
    High Tech Support
    813-920-7564
12. DE, VA, MC, DC
    MGM Visuals
    703-352-3919
13. MS, AL, GA
    Electro-Cadd
    404-446-7523
14. E. PA, NJ, NY
    Beta Lambda, Inc.
    201-446-1100
15. CT, RI, MA, VT, NH, ME
    DGA Associates, Inc.
    617-935-3001
16. BC, AB, SK, MB
    Interworld Electronics & Computer Industries, Ltd.
    604-984-4171
17. ON, PQ
    Electralert, Ltd.
    416-475-6730

OR8801

OrCAD™ Systems Corporation

1049 S.W. Baseline St., Suite 500
Hillsboro, OR 97123 USA
503-640-5007

CIRCLE NUMBER 6

80960's

RISC-Like Architecture

PHILIP BRIDE AND TONY BAKER, INTEL CORP., HILLSBORO, ORE.

MODULAR APPROACH PRODUCES EMBEDDED CONTROLLERS FOR SPECIFIC APPLICATIONS

Designers of microprocessor-based system perform a balancing act, juggling performance, cost, power consumption, and size. Because the priorities of embedded systems differ significantly from those of reprogrammable systems, achieving this balance in an embedded application requires different microprocessor capabilities.

The classic requirements for embedded systems are low cost, small size, low power consumption, and predictable performance. In contrast, reprogrammable systems balance these same concerns but generally put performance in first place.

To illustrate the difference, compare the design considerations for an engineering workstation and a desktop laser printer. The primary competitive advantage for the workstation is high performance. The printer must achieve high performance as well, but price, size, and power consumption are equally important. In other words, embedded systems have all the problems of reprogrammable systems, plus a few more.

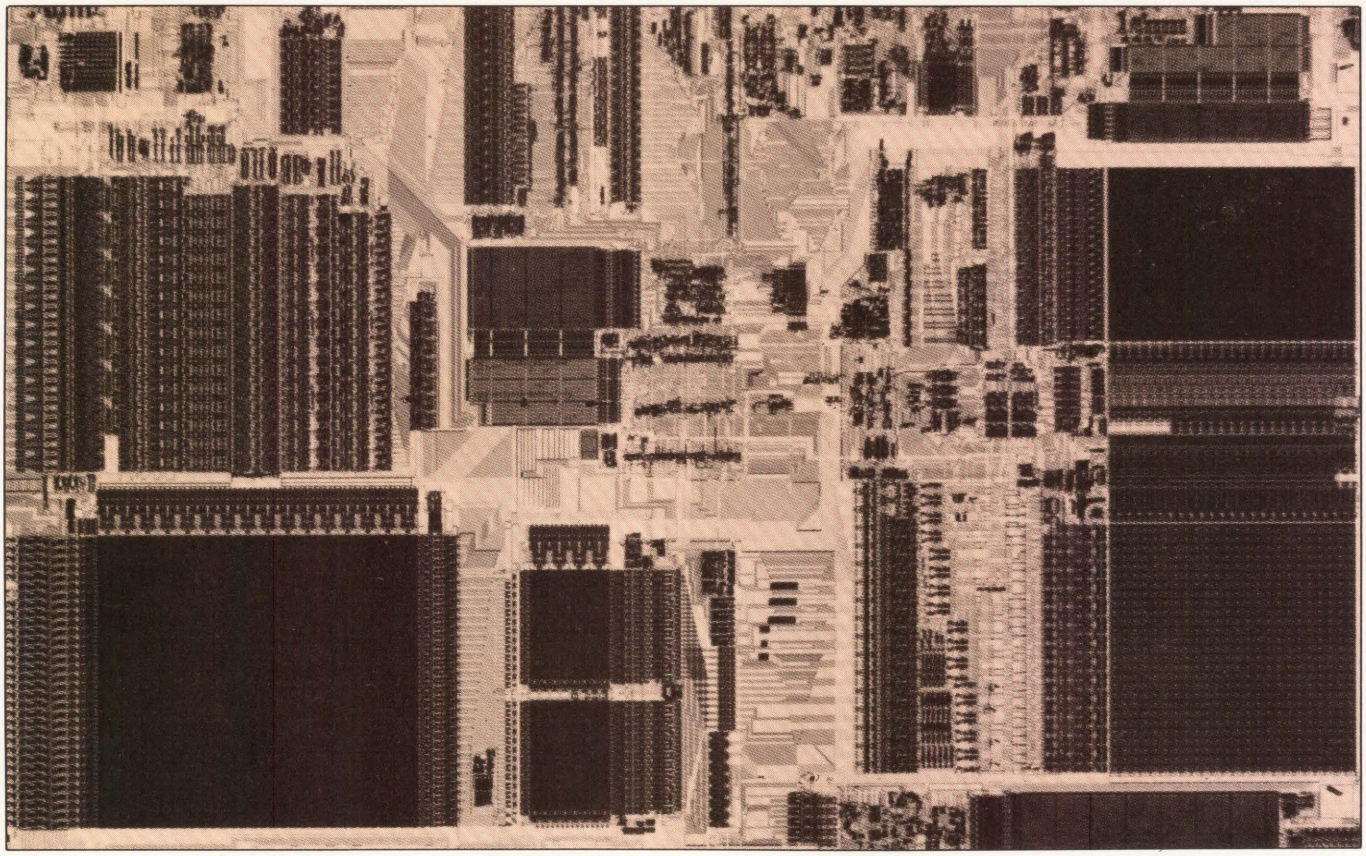
In simpler designs, embedded controllers can balance cost and size with lower performance. That compromise is not applicable, however, to more complex products that require significant processing power. Advanced embedded designs in machine control, robotics, process control, and avionics need lots of processing horsepower to handle complex, real-time interrupts and high-speed floating-point calculations.

To achieve these higher performance levels, some embedded system designers have turned to 32-bit architectures, but available general-purpose processors were not designed to meet embedded controller needs. Recognizing this need for a high-performance embedded processor, Intel has developed the 80960 processor family, based on a high-performance core architecture that limits system size and cost, integrates a wide variety of special capabilities, and provides an upward growth path for future development.

■ RISC-LIKE

The architecture of any processor is an abstract definition of the computer, its instruction set, its register model, and the functions needed for control. In developing the 80960 architecture, the designers viewed RISC as a set of design techniques available to architects, instead of as predefined architectural constructs. We took a pragmatic approach, selectively applying RISC techniques when they were appropriate and modifying them when necessary to improve functionality or ease of use. The resulting architecture uses pipelined structures and highly parallel instruction execution in the engine that form the core of its computation and data processing capabilities. Around the engine are modular features that extend the functionality of the core in embedded systems (Figure 1).

Because the 80960 was built from the ground up specifically for embedded applications, each architectural decision was subjected to very specific design tests. For



example, many RISC theorists assume that systems will be programmed only in a high-level language (HLL). Because of this assumption, designers may omit safety mechanisms such as silicon interlocks within pipelines and depend on the software compiler to manage pipeline execution. As a consequence, the RISC processor may be difficult to program in assembly language.

Designers of embedded systems, however, require assembly-language programming capabilities for certain time-critical operations. The 80960 solves these problems with a careful balance of hardware and software functions. It includes silicon interlocks to managing its pipeline invisibly to the programmer. This capability makes it practical to program the processor in assembly language—essential for such tasks as building I/O control routines.

The processor includes other features, like the register cache, that are effective without compilation. On the other hand, it contains some functions, such as register scoreboarding, which operate more efficiently with software from an optimizing compiler.

The 80960 architecture addresses the special needs of embedded systems by controlling system costs, size, and power consumption at every stage of the development process. For example, operations for bit-field manipulation, debugging, and self-testing simplify system development and add functionality without forcing the designer to compromise performance. Integration of special features like floating-point processing and interrupt control directly onto the processor eliminates the cost and board space of additional peripheral devices and coprocessors. In this way the architecture minimizes overall chip count and simplifies system design to decrease space and power needs.

The 80960 uses many RISC design techniques: a simple load and store design; large, general-purpose register sets; a small set of instructions that are aligned by the word—32 bits wide—and that execute in one cycle; optimizing compilers that operate on complex HLL code; and simple hardware that makes efficient use of silicon. These characteristics have been modified to meet the practical requirements of embedded computing, and the modular structure of the architecture allows different versions of the architecture to be produced quickly.

In the 80960 family architecture, operations work on data in on-chip registers, with only load and store instructions accessing memory. This attribute simplifies both the instruction set and the decoding process. A load/store architecture is well suited to embedded applications, because simple instruction decoding lets the compiler optimize source code more effectively. In addition, the decoupling of external memory operations from internal processing aids performance. The 80960KB implementation (see "80960 Implementations," p. 24, and the table) further reduces the number of memory accesses by providing a 512-byte, direct-mapped instruction cache.

All instructions are 32 bits long and aligned on 32-bit boundaries, a detail that eliminates the need for an instruction alignment stage. The instruction set has a three-operand format with space for two source operands and one destination operand. That means it is possible to do an operation without writing over the source operand in the course of executing the instruction.

EXTENSIBLE ARCHITECTURE USES PIPELINES AND PARALELL INSTRUCTION EXECUTION

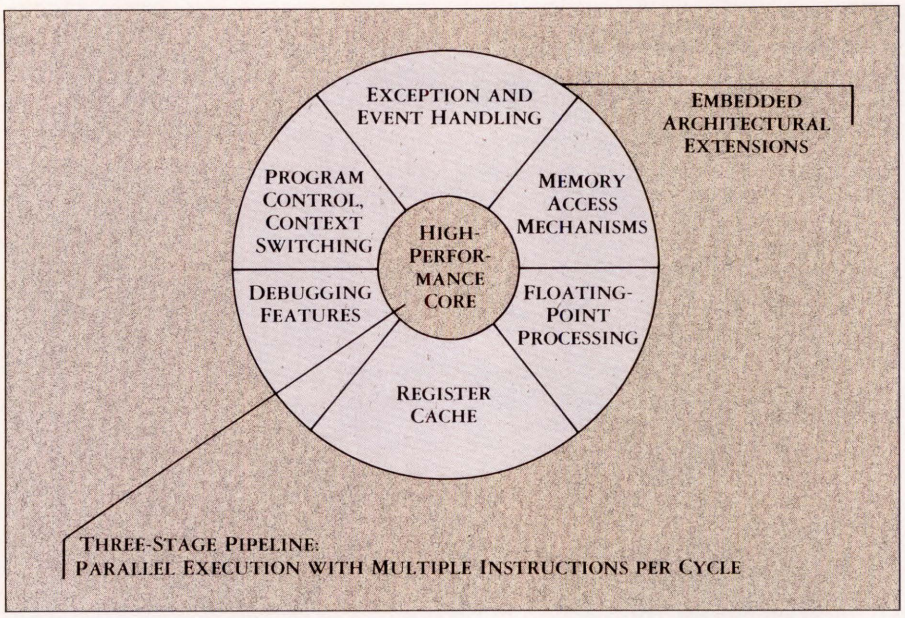


Figure 1. The modular architecture of the 80960 family allows it to be implemented in a variety of ways.

The 80960 instruction set provides an extensive set of operations to the programmer. It supports logical operations in addition to integer operations such as add, subtract, multiply, and divide. It also provides a broad range of data instructions to deal with individual bits within fields and within registers and with Boolean operations. More than 50 of these instructions can be executed in a single clock cycle. Multicycle instructions, such as sine or cosine, are also available to make programming more practical. The instruction set supports the ability of future 80960 family members to add functional blocks like DMA controllers, timers, and interrupt controllers.

The 80960 architecture includes large register files that serve to simplify task switching and execution. In this architecture, local registers are unique to each procedure, whereas global registers retain their values across procedure boundaries. Global registers serve as a common block for passing information from one procedure to another, so that procedure calls using this mechanism do not need to access external memory. For example, more than 90% of all C routines pass fewer than six parameters, but the 80960KB has 16 global 32-bit registers that can be used to pass as many as 12 words of parameters. Most HLL routines can execute wholly out of the 'KB's 16 local registers, with 13 words of local variables available per procedure and excess global registers available for scratchpad use. The other local registers are reserved for such chores as task linkage and tracing return instructions.

Although the architecture specifies 16 local and 16 global registers, it does not define how many local register sets its "register cache" will hold. That characteristic is defined within each implementation. All parts will have these 32 registers, but an implementation may extend this number to add features such as the 80960KB's 64-register cache and four 80-bit floating-point registers.

The pipeline structure of current 80960 implementations (Figure 2) is organized in three functional stages: fetching, decoding, and execution. The fetch stage includes bus control logic (BCL) and an instruction fetch unit (IFU). The decoding stage corresponds with the instruction decoder (ID) and the micro-instruction sequencer (MIS). The execution stage consists of the integer execution unit (IEU) and, in the 'KB and 'MC versions, a floating-point unit. An autonomous interrupt controller operates next to the pipeline. Future implementations of the 80960 will be similar to, but more elaborate than, the version described here.

■ The Fetch Stage

In the fetch stage, the bus control logic is the interface between the 80960 and the external 32-bit L-bus that connects it to the system memory and I/O modules. This multiplexed bus can transfer data at a maximum sustained rate of 53 MB/s when the processor is operating at 20 MHz with no wait states. The BCL accepts memory requests from other units within the 80960 on a first-come, first-served basis and executes them. It attempts to maximize bus access efficiency through buffering and burst accesses.

The BCL buffers memory references in a three-deep FIFO. Once a memory access is initiated, it will be completed before any other access in the FIFO is initiated. The BCL queuing mechanism, coupled with other 80960 features such as scoreboarding, enables other modules in the 80960 pipeline to continue operation without waiting for memory bus requests to be completed. Most memory reference instructions are executed with little or no delay in the instruction pipeline.

Bus throughput is improved by the use of burst memory access, which allows a multiword access to and from the L-bus with only one address cycle. Any address cycle can be followed by as many as four data words and a recovery cycle. The BCL controls the burst access, automatically maintaining 16-byte boundaries within memory accesses.

Burst memory access improves memory bandwidth substantially. For example, if this bus were limited to single-word access, it would be capable of only 27 MB/s, one half the burst bandwidth of 53 MB/s. Burst access also allows the 80960-based system to utilize the nibble and static-column modes in DRAMs. By limiting bursts to four words, under control of the BCL, the bus also simplifies external memory-control logic.

The hardware portion of the interrupt controller is located within the BCL, although interrupt control logic is located in a separate module. When an external source requests the 80960 to handle another chore, it causes an implicit procedure call to the interrupt controller. Each of the four interrupt pins has an associated 8-bit interrupt vector. Programmable registers allow the designer to multiplex two or more interrupt pins to service a larger number of interrupts. Using the interrupt vector, the interrupt controller signals the micro-instruction sequencer that an interrupt has occurred and the MIS starts a microprogram to service the interrupt. The current state of the machine is saved on an interrupt stack in main memory. If the interrupt occurs during an instruction that requires many machine cycles, the instruction state is also saved in the interrupt stack and the execution of the instruction is suspended.

A 4-bit register records and prioritizes further interrupts that occur after the initial interrupt and before the interrupt controller's microprogram can finish its task and return to the interrupted task. Upon returning, the interrupt controller examines the 4-bit register and if need be begins executing the next-highest priority interrupt.

The last function of the BCL is arbitration of control of the three major internal buses in the 80960: the bus interface (BI) bus, the data bus, and the microinstruction (μI) bus. The BI and data buses are

each 32 bits wide, and the μI bus is 42 bits wide. The 80960 also employs a series of other, smaller buses, including the X bus, the microinstruction address (μADDR) bus, and an internal bus within the microinstruction sequencer. The BI, X, and μADDR buses all are within the same data path. The BI bus connects the BCL and the instruction fetch unit. The X bus extends from the IFU to the instruction decoder, and the μADDR bus continues from the instruction decoder to the microinstruction sequencer.

The next module in the fetch stage is the instruction fetch unit, which acts as an intelligent buffer for the instruction decoder. It includes the 512-byte, direct-mapped instruction cache, which holds instructions from the bus control logic. If an instruction in a new block of instructions must be initiated immediately, it is put into the cache and sent immediately to the instruction decoder for decoding instead of being put into the cache and then read back out and decoded.

The effective-address ALU within the instruction fetch unit performs address calculations under the control of the instruction decoder. Address calculation is executed during decoding (as opposed to during the execution phase), saving two to three clock cycles when instruction execution occurs in parallel with address calculations. The address calculation can be done in parallel with an integer execution or a floating-point operation.

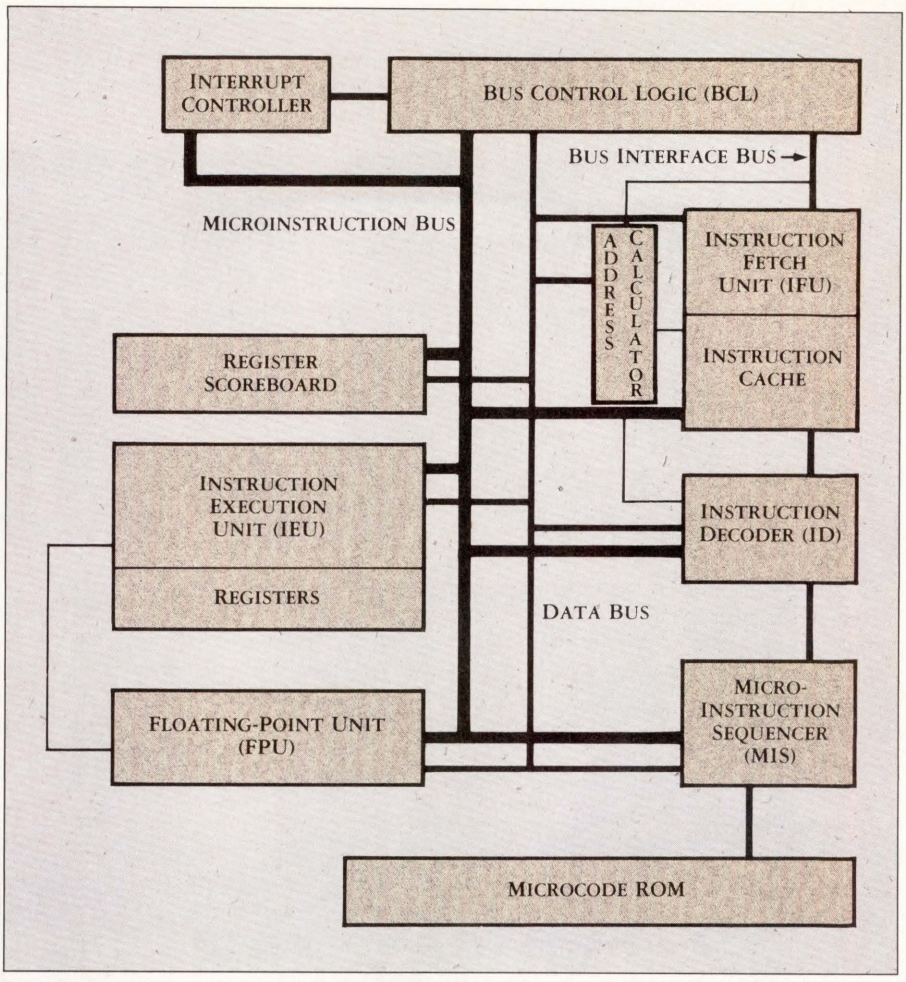


Figure 2. The multiple autonomous units and bus structure of the 80960KB implement the part's three-stage pipeline operation.

```
ld xyz, r6          # r6 <-- load data from address xyz
addi g4, g6, g7     # add: g7 <-- g4 + g6
addi g9, g10, g11   #g11 <-- g9 + g10
ld abc, r8          #r8 <-- load data from address abc
and g0, 0xffff, g1  # g1 <-- g0 AND 0xffff
addi r6, r8, r7     # r7 <-- r6 + r8
```

Figure 3. Register scoreboarding permits parallel execution of load and register-driven instructions. Here, four additional instructions can be executed during the load operation, because they don't make use of register 6.

■ DECODING STAGE

The instruction decoder decodes the instructions it receives from the instruction fetch unit and routes them to the appropriate execution units. It manages all tracing mechanisms and illegal op-code faults. Upon detection of a trace event or an illegal op-code, it issues a command to the microinstruction sequencer to start a flow for that fault. Instructions are decoded based on how they are to be executed: simple instructions; floating-point and branch instructions; complex instructions; and load and store instructions.

Simple instructions—logical, compare, shift, integer add, integer subtract, and ordinal add and subtract—require little decoding. The ID decodes these instructions and passes them to the instruction execution unit, where they are executed, usually in a single clock cycle.

Floating-point and branch instructions do not execute in one cycle. Floating-point instructions are executed by the floating-point unit and may require interaction among the FPU, the instruction decoder, and the microinstruction sequencer. The instruction decoder executes branch instructions directly. If the branches are unconditional, no interaction with the processor's other execution units is required. On conditional branch instructions, the ID uses a condition code scoreboard (described later) to streamline the branching process.

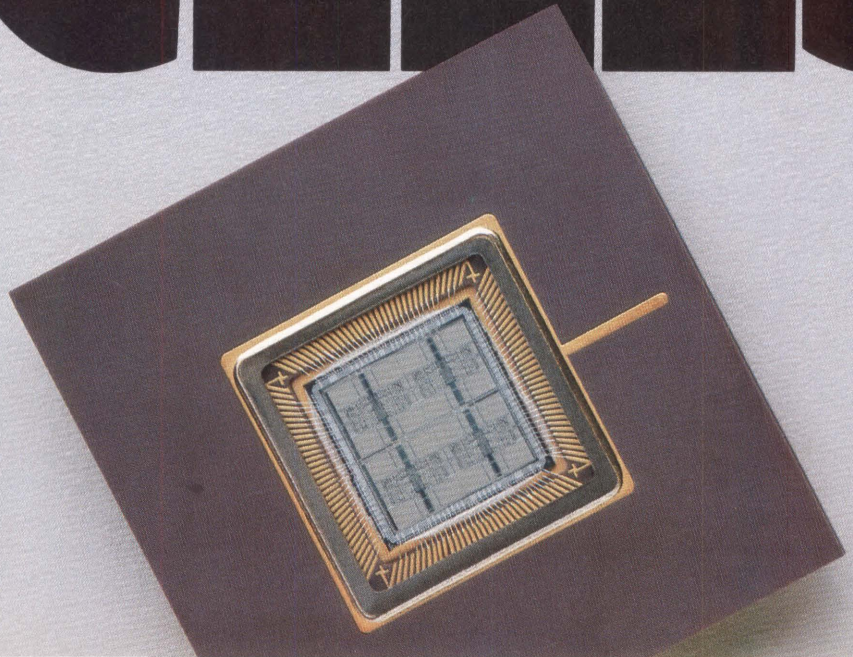
Pipelined operation can make the branching appear to occur in zero clock cycles. For example, the branch instruction ("b") shown below will execute in zero cycles, because the branch time is overlapped completely by the execution time of the floating-point instruction ("sinr"):

```
        sinr        g0, g1
        b           some—location
some—location:
        mov         g1, g2
```

Complex instruction execution requires the use of microcode within the microinstruction sequencer. The instruction decoder decodes complex instructions and forwards them to the sequencer, which then sends the equivalent microcode to the appropriate execution units.

Load and store instructions request data to be read from or written into memory.

BUILD KILLER CELLS.



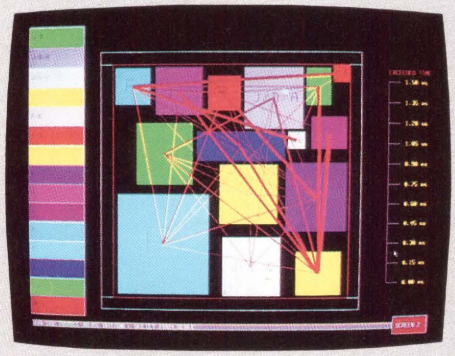
This is the stuff reputations are built on.

A cell design with 100,000 gates. More than 1000 library elements. 2MB of memory. And sub-micron HCMOS technology.

You can search high and low, but there's only one place in the world you can find a chip of such staggering complexity.

LSI Logic.

Designs like this are precisely why we're the #1 domestic ASIC supplier with more than 4,000 working cell and array-based designs in the field.



High complexity Cell-Based ASICs require advanced tools like the MDE chip floorplanner to optimize delays and verify performance prior to layout.

And why our Modular Design Environment (MDE)™ software is the best foundation for building killer cells.

MDE is the industry's most advanced design software for ASICs. It arms you with the capability to build today's most sophisticated cell-based designs. Easily. And with the smallest possible die size.

Besides MDE, LSI Logic delivers more than 400 LSI and VLSI building blocks, including 16 and 32-bit microprocessor cells and industry-standard processor and peripheral functions; the longest list of SSI and MSI functions; and memory compilers to develop exactly the RAM or ROM your chip requires.

And if your application calls for lower densities, that's covered, too. Because LSI Logic has more cost-effective cell-based solutions than you can imagine.

No matter what kind of Cell-Based ASIC you build, LSI Logic will deliver a fully-tested prototype in as little as 4 weeks.

All in the production quantities you want, thanks to our advanced worldwide wafer fabrication, assembly, and test facilities.

So find out more about LSI Logic's Cell-Based ASICs by calling the sales office nearest you.

After all, we can help you make a killing.

LSI LOGIC

LSI Logic Sales Offices and Design Resource Centers: Scottsdale, AZ 602-951-4560, Milpitas, CA 408-433-8000, San Jose, CA 408-248-5100, Irvine, CA 714-553-5600, Sherman Oaks, CA 818-906-0333, Denver, CO 303-756-8800, Altamonte Springs, FL 305-339-2242, Boca Raton, FL 305-395-6200, Norcross, GA 404-448-4898, Chicago, IL 312-773-0111, Bethesda, MD 301-897-5800, Waltham, MA 617-890-0161, Ann Arbor, MI 313-769-0175, Minneapolis, MN 612-921-8300, Bridgewater, NJ 201-722-7522, Poughkeepsie, NY 914-454-6593, Raleigh, NC 919-783-8833, Worthington, OH 614-438-2644, Beaverton, OR 503-644-6697, Trevose, PA 215-638-3010, Austin, TX 512-338-2140, Dallas, TX 214-788-2966, Bellevue, WA 206-822-4384, Calgary, Alta 403-262-9292, Paris, France 33-1-46212525, Israel 972-3-403741/6, Milan, Italy 39-39-651575, Tokyo, Japan 81-3-589-2711, Seoul, Korea 82-2-785-1693, Nidau/Biel, Switzerland 032-515441, Bracknell, United Kingdom 44-344-426544, Munich, West Germany 49-89-926903-0.

CIRCLE NUMBER 7

80960 Implementations

The 80960 architecture provides the framework for a complete family of products. It is currently implemented in three versions: the 80960KB, which integrates a floating-point unit and an interrupt controller (see Figure 2 and the table, main text); the 80960KA, which is socket-compatible with the 'KB but does not include the floating-point unit; and the 80960MC, a military-qualified version of the 'KB with memory management and instructions that simplify the use of Ada multitasking capabilities. All three are built using 1.5-μm two-layer-metal CHMOSIII technology. The silicon technology produces chips with low power requirements—less than 2 W in the 80960KB. The 80960KB is housed in a 132-pin PGA, although the processor actually requires only 78 pins. The 132-pin PGA was chosen because its cavity size accommodates the current die size of 390 × 390 mils.

The low pin count of the 80960KB is made possible by the use of a multiplexed bus, which reduces the number of pads necessary on the die. The lower bus pin count translates into easier signal routing, and the reduced number of drivers means higher reliability and lower power consumption. The lower bus pin count, coupled with a high ratio of power pins to bus pins, results in lower levels of switching noise, higher reliability, and superior performance in noisy environments.

The 80960MC adds multiprocessing and fault-tolerant computing to its basic embedded capabilities. The chip works with the 82965 bus extension unit (BXU) to support error detection, reporting, and automatic system reconfiguration. The BXU connects the processor with Intel's proprietary Advanced Processor (AP) silicon bus, allowing designers to replicate subsystems and achieve any desired level of multiprocessing or fault tolerance.

Fault-tolerant systems based on the 80960MC use functional redundancy checking (FRC) processor pairs that run simultaneously in lockstep. The processors can be started and stopped synchronously using an interagent communication (IAC) message. In a quad modular redundancy (QMR) system, two FRC pairs are joined to form a QMR. One pair (the primary) is actively issuing and responding to AP bus accesses, while the other (the shadow) is in lockstep with the primary but not active on the bus. This system can toggle between the primary and shadow pair and can run nonstop when a single fault occurs. The hardware can reconfigure itself, with the shadow assuming the bus activity of the primary, in real time. Because configuration control resides entirely in BXU programmable registers, the same processor pairs can also be configured for multiprocessing.

We expect the next standard version to be tailored specifically for data control. Its capabilities would be optimized for applications such as protocol handling and telecommunications. This data controller will operate at three to four times the performance level of the current 80960KB and is expected to be announced in early 1989. ■

The instruction decoder sends these instructions directly to the bus control logic, which executes them. The instruction decoder is responsible for converting addressing information encoded in load, store, branch, and call instructions into effective memory addresses. If during decoding an operand reduction (address calculation) is required, the instruction decoder issues all the commands for the calculation to the instruction fetch unit.

■ Execution Stage

The instruction execution unit contains the ALU, register sets, and the mechanisms for register and condition-code scoreboarding. In a single clock cycle, it can perform addition and subtraction of integers and ordinals, moves between registers, logical operations, bit operations, shifts, rotations, or a comparison. It can also work with integer literals in the range of 0 to +31, which are encoded in the instruction format. The IEU receives its control either from the microinstruction sequencer or the instruction decoder via the μI bus.

Operands for these operations are fetched from internal registers or from external memory, or they are literals. The instruction execution unit contains a simple sequencer used for aligning and accessing multiple-word operands. For floating-point operations, the IEU passes data to and from the floating-point unit across the data bus. Any register can be used for a floating-point operation.

Whenever two operands go through the ALU and their result or a constant is used in the next operation, the IEU saves that operand internally, so it need not be fetched again from the register. Most integer operations are two cycles long without the bypass. With the bypass, an operation for which the compiler is optimized, they require one cycle.

The instruction execution unit contains 16 global registers and 16 local registers, plus 32 scratchpad registers used for internal microprograms. It is responsible for maintaining the local register cache and the associated allocation logic.

In parallel with the instruction execution unit, the microinstruction sequencer executes instructions that require microcode. Processor microcode is stored in a 3K × 42-bit microcode ROM, which is accessed through the MIS. When the instruction decoder receives a complex instruction that requires microcode to be executed, it sends the starting microaddress of the flow to the MIS, which steps through the microprogram.

Microcode is used for complex instruction execution, exceptions such as interrupt and fault handling, local register cache manipulations, and the power-up and self-testing performed during processor initialization. The microinstruction sequencer also initializes the FPU for floating-point instructions.

■ Register Scoreboarding

The register scoreboarding mechanism keeps track of outstanding memory accesses and enables subsequent instructions to execute while outstanding accesses are being completed. The resulting concurrent instruction execution improves performance by 20% to 50% in tight software loops and by 10% in large C programs.

When a LOAD instruction begins, the 80960 sets a scoreboard bit on the target register to indicate that a particular register or group of registers is being used in a load operation. If the instructions that follow do not use registers in that group, the processor is able to execute those instructions before execution of the prior instruction is complete. The 80960 ensures that these additional instructions do not make reference to the target register by transparently checking the scoreboard

bit. After the target register is loaded with data, the processor resets the bit.

A common application of this feature is to execute one or more fast instructions—taking one to three clock cycles—concurrently with load instructions. A load instruction typically takes three to nine clock cycles, depending on the use of burst access. For example, the instructions in Figure 3 load a group of local registers while performing some other operations on data in global registers. The two "addi" instructions following the first load, as well as the instruction following the second load, are executed in parallel with the load instructions.

Another situation where scoreboarding can be useful for procedure optimization is when floating-point instructions are being executed. Floating-point operations are executed by a separate unit from the IEU. As a result, non-floating-point instructions can often be executed concurrently with floating-point instructions, providing that they do not use the same registers and do not use the ALU.

■ Floating-Point Unit

The floating-point unit performs all the floating-point computations for the processor, as well as the integer multiplication and division operations. It manages the four 80-bit floating-point registers (mentioned earlier) used for extended-precision floating-point calculations. It is fully compatible with IEEE-754 Rev. 10, including all extensions for transcendental, exponential, and logarithmic operations.

Capable of 32-bit (single), 64-bit (double), and 80-bit (extended) precision, the FPU passes data to and from the rest of the chip via the data bus. It shares the resources of the processor, using the global and local registers (in addition to its own registers) as operands for floating-point operations and the microinstruction sequencer for initialization. All floating-point operations are initiated by microinstructions received on the microinstruction bus.

There are three main logical blocks within the floating-point unit: the mantissa block, the exponent block, and the control block. The mantissa block consists of a shifter, an adder, three working registers (two operand and one accumulator), and a 44-word ROM that is 68 bits wide. The exponent block comprises two temporary registers, an adder, and a four-word, 16-bit-wide ROM. The control block directs the FPU, receiving instructions from the instruction decoder or the microinstruction sequencer, as well as control signals. It controls sequencing, data movement from registers, and bus usage internal to the FPU.

To perform integer multiplications and several floating-point calculations, the FPU contains a 32-bit integer Booth multiplier. This multiplier performs integer multiplication operation in a variable amount of time, depending on the number of significant bits.

■ The Programming Environment

The 80960 architecture defines a completely flat address space: no translations are necessary to calculate addresses. Direct one-for-one address mapping, without segmentation, provides a straightforward programming environment and eliminates the need for special address control registers.

The 80960 architecture provides the multiple addressing modes typical of embedded applications, plus, as noted, a separate ALU for address calculation. Addressing modes are built into the instruction stream and managed by the part itself to make programming easier.

The 80960's procedure call mechanism uses a register cache to reduce memory accesses. At every procedure call, the processor caches the contents of the previous procedure's registers on chip and allocates a new set of registers, often eliminating the need for stack accesses. The 80960KB has a four-deep cache that can make up to four procedure calls before it has to do a stack access, and the processor automatically handles stack accesses.

Embedded applications frequently involve the development of custom hardware in parallel with custom software. The debugging facilities built into the 80960 are designed to make system integration easier. The processor offers two different break instructions—conditional and unconditional. In addition, the design includes a breakpoint register that stores two addresses for designated breaks.

The 80960's tracing facility looks for specific classes of instructions like subroutines, branch conditions, supervisor calls, and single-stepping operation. When any of those user-specified events occurs, it transfers control of the processor to a user-specified subroutine, and the processor provides information on the exact nature of the event to the subroutine. ■

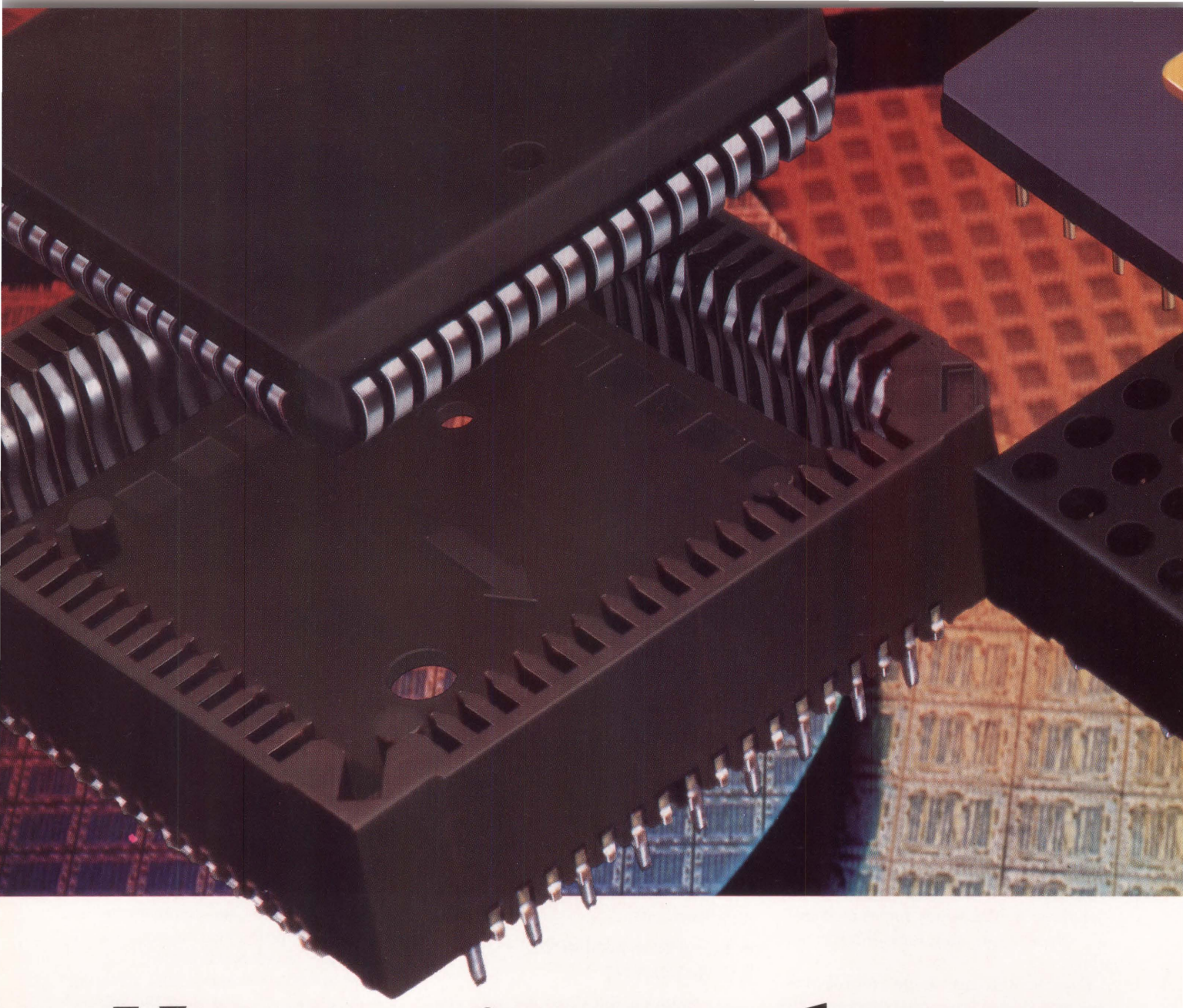
The 80960KB at a Glance

Performance	20 MIPS burst execution at 20 MHz
Registers	7.5 MIPS sustained execution at 20 MHz 16 32-bit global 16 32-bit local 4 80-bit floating-point Register scoreboarding Condition code scoreboarding
Clock speeds	16, 20, 25 MHz
Instruction cache	512 bytes, direct-mapped
Register cache	4 sets of local registers
Address space	4 GB, linear
Bus	32-bit multiplexed Up to 16 bytes per burst 53.3 MB/s in the burst mode
FPU	IEEE-754–compatible 4 million Whetstones/s at 20 MHz
Interrupt controller	32 priority levels 248 vectors Supports 8259A

About the Authors

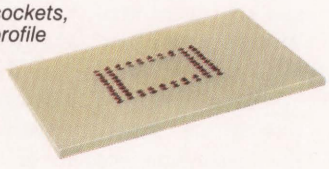
Philip Bride *is a hardware applications engineer in the 80960 applications group. He joined Intel in 1984 as a member of the VLSI design group that developed the 80960 and was a circuit design engineer for the architecture's instruction decoder. He graduated from the University of Portland with a BSEE in 1984.*

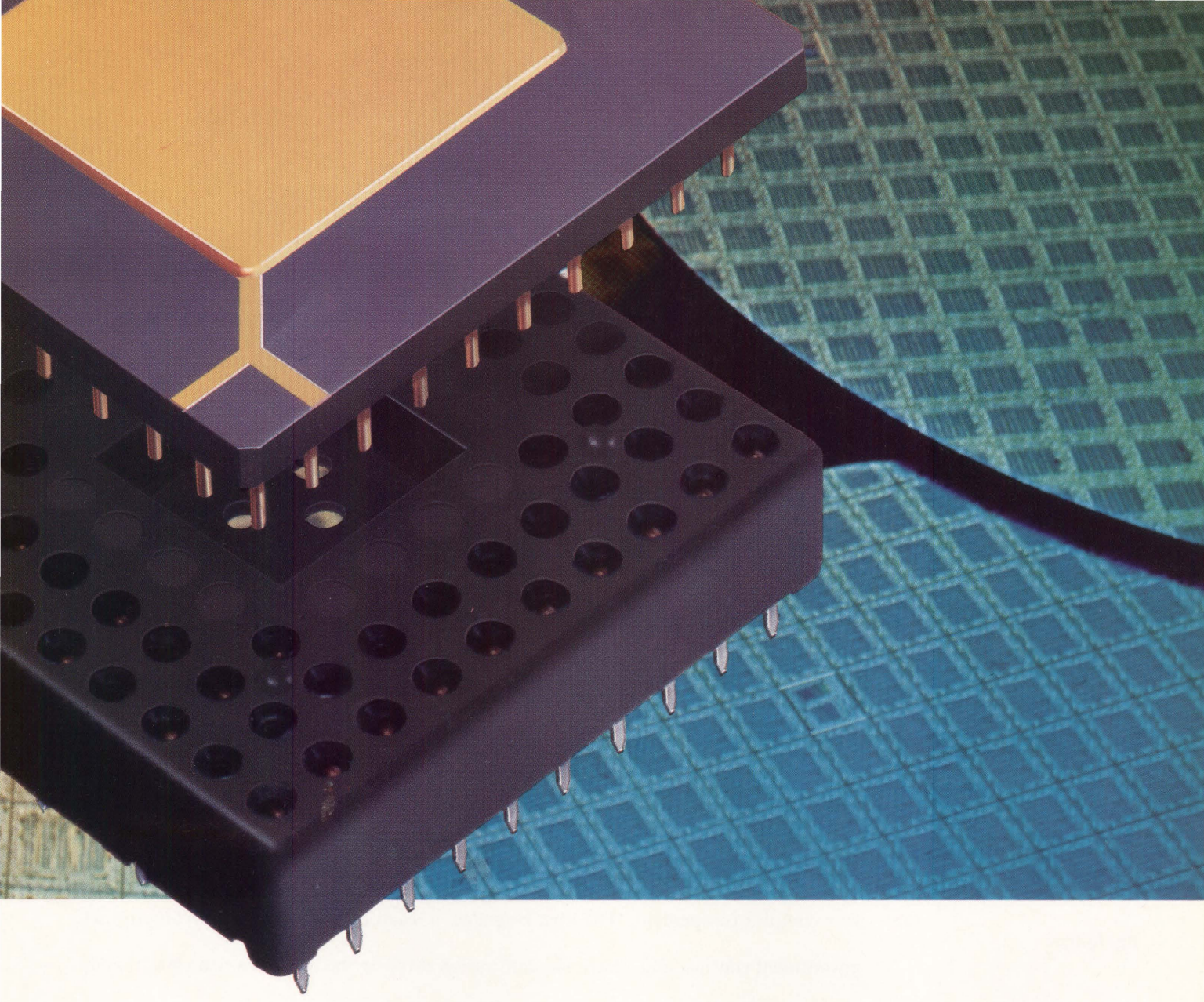
Tony Baker *is an applications engineer for the 80960 project. He has been at Intel for seven years, serving also as a software engineer and as a field applications engineer. He received his BSEE in 1981 from the University of Portland.*



How we've made sure the chip's locked in, but you're not.

ZIF and LIF PGA sockets, plus our minimal profile spring sockets in PGA footprint.





Whichever direction you take in microprocessor technology, AMP makes sure you have the socketing options you need to make it pay off.

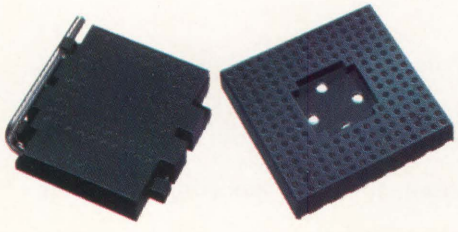
Our high pressure tin sockets for plastic leaded packages come with an exclusive Positive Lock retention system that keeps chips secure during handling and shipping. Ceramic chip carrier sockets feature duplex plated contacts and snap-on covers that accommodate heat sinks. Both are available in standard and surface-mount versions.

AMP offers you more: high-speed, surface-mount sockets on .020″ centers. Gold-plated plastic carriers and sockets. For pin grid arrays, sockets in ZIF and LIF styles, plus custom VHSIC capability.

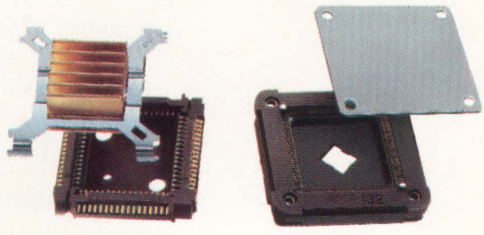
Choose your technology. AMP makes it easy to implement, with full socketing support.

Call 1-800-522-6752 for more information on AMP sockets. AMP Incorporated, Harrisburg, PA 17105-3608.

AMP Interconnecting ideas



Low-height sockets in all standard sizes (JEDEC A, B, D), with duplex-plated contacts for sure performance. High-speed, surface-mount sockets feature 0.5pF, 1.4nH contact characteristics.



CIRCLE NUMBER 8

BEHAVIORAL DESCRIPTIONS IN V·H·D·L

With Process Statements, VHDL Gives Designers the Facilities of a General-Purpose Programming Language

DAVID L. BARTON, INTERMETRICS INC., BETHESDA, MD.

The VHSIC Hardware Description Language (VHDL) was designed at the request of the VHSIC Program Office to provide a notation capable of the design and description of very complex components. The final language, a result of efforts under both an original government contract and a later standardization drive by the IEEE, is a rich collection of constructs for various levels of hardware description.

This article considers in depth one specific problem domain within the overall field of hardware design and description: behavioral descriptions and the mechanisms for describing relationships between behaviors. All of the examples and the information in this article reflect the 1076/B version of VHDL, as published in the May 1987 Language Reference Manual. This version was the subject of the standardization ballot in the IEEE, and VHDL was accepted as a standard in December 1987 as IEEE-STD-1076-1987. As a normal part of the standardization process, however, some changes have been made in response to comments.

As noted in a recent article (Barton, 1988), elementary VHDL descriptions are composed of signal assignment statements and component instantiations (see "VHDL Fundamentals," p. 31, for a review of these concepts). When modeling extremely complex circuits behaviorally, signal assignment and component instantiation state-

ILLUSTRATION BY ANTHONY RUSSO

```
entity full_adder is
  port (x, y, cin: in BIT; sum, cout: out BIT);
end full_adder;

architecture algorithmic of full_adder is

begin

  process (x, y, cin)
    variable s: BIT VECTOR (1 to 3) := x & y & cin;
    variable num: INTEGER range 0 to 3 := 0;
  begin
    for i in 1 to 3 loop
      if s(i) = '1' then
    num := num + 1;
      end if;
    end loop;

    case num is
      when 0 => cout <= '0'; sum <= '0';
      when 1 => cout <= '0'; sum <= '1';
      when 2 => cout <= '1'; sum <= '0';
      when 3 => cout <= '1'; sum <= '1';
    end case;
  end process;

end algorithmic;
```

Figure 1. An example showing the use of a process statement. In this version of a full adder, a 'for' loop counts the number of ones on the three input lines.

```
entity and_gate is
  port (a, b: in BIT; c: out BIT);
end and_gate;

architecture process_statement of and_gate is
begin
  process
  begin
    if a = '1' and b = '1' then
      c <= '1';
      wait on a, b;
    elsif a = '0' then
      c <= '0';
      wait on a;
    elsif b = '0' then
      c <= '0';
      wait on b;
    end if;
  end process;
end process_statement;
```

Figure 2. A process statement implementing an AND gate. Because the controlling input conditions are considered first, only three cases need be checked.

ments can be cumbersome. A general method of describing behavior is needed. The center of this method is the *process*. Indeed, all units of action in VHDL are processes; a signal assignment statement is just a special case of a process. With the VHDL process statement, the facilities of a general-purpose programming language are available to the hardware designer.

The process statement encloses the equivalent of a general software program. It may make use of most of the general-purpose structures found in Ada, Pascal, or any other programming language.

Writing a process statement is like writing a program that describes the behavior of a portion of a system. An example is given in Figure 1. The process statement shown executes once at the beginning of simulation, then once each time any of the ports "x," "y," or "cin" change. The list in parentheses after the key word *process* lists these signals; such a list is called a sensitivity list and the process is said to be sensitive to those three signals. The algorithm sums the number of input bits that are set equal to 1 and sets (through signal assignment statements) the output ports appropriately.

The statements inside a process statement are executed sequentially, as opposed to those signal assignment statements outside a process statement, which execute concurrently with one another. The sequential statements available to the writer of a process statement are signal assignment (executed sequentially), variable assignment, procedure, and function call, "if," "case," "loop," "exit," "next," "return," and "null."

The function of the signal assignment statement inside a process statement is very similar to that outside a process statement; changes to the target signal are reflected by the simulation cycle. If two signal assignment statements appear in a process statement, they are executed in the order in which they are encountered. Each signal assignment statement affects only the value that the process containing the signal assignment statement contributes to the final value of the signal. That final value is determined during the simulation cycle as for concurrent assignments, according to bus resolution functions described below. The other statements in the list are similar to their Ada counterparts (as well as equivalent statements in other languages, like Pascal) and will not be covered here.

A concurrent signal assignment statement is equivalent to a process statement containing a signal assignment statement; a conditional signal assignment statement is equivalent to a process statement containing an "if" statement whose branches are the alternatives of the conditional signal assignment statement; and a selected signal assignment statement is equivalent to a process statement containing a "case" statement whose alternatives are the same as the alternatives of the selected signal assignment statement.

■ WAIT STATEMENT

Some descriptions do not lend themselves to a "straight-through" execution. In particular, a process might want to wait for a specific event or change before proceeding. Being able to specify such a wait would allow greater flexibility in writing complex behaviors. For this purpose, a special statement called the *wait* statement may appear in a process statement. This statement specifies that the process should suspend execution until a simulation cycle when the specified conditions are met.

Consider the process statement shown in Figure 2, which implements an AND gate. This process waits for both "a" and "b" to change only if both are high; if one is low, it waits for that signal to change (since the other signal changing does not affect the output value). This specification is more efficient than a process that executes each time either "a" or "b" changes.

Note that no sensitivity list appears after the key word *process;* instead, the sensitivity list is attached to each wait statement. A process containing a wait statement must not have a sensitivity list following the key word *process;* the presence of such a key word is equivalent to the single statement "wait on ‹sensitivity list›;" as the last statement of the process statement.

A process may also wait until a condition is true ("wait until ‹condition›") or wait for a given period of time ("wait for ‹time›"). Any combination of these options is legal. The time expression is a maximum time, and the process will resume execution when either the condition is true or a signal in the sensitivity list changes.

The wait statement gives the designer freedom of expression in writing high-level behavioral models. A sequence containing "write address bus, wait for return, read data, and continue" is extremely cumbersome without this facility.

VHDL Fundamentals

VHDL is a simulation-oriented language for the design and description of complex systems. Several different "views," or ways of decomposing a complex description into parts, are inherent in the definition of VHDL. This section briefly reviews two common problem domains: register-transfer descriptions and structural decomposition.

A register-transfer description of hardware consists of a series of Boolean logic expressions. Each operator represents a gate or a series of gates in a hardware realization. In the simulation cycle, assignment statements are treated as executing concurrently; that is, the order of the statements in the description is immaterial. VHDL supports simple register-transfer statements, as well as timed, conditional, selected, and guarded assignment statements.

Unlike many notations, VHDL requires a separate signal declaration statement. Each expression in a series of signal assignment statements may be followed by the key word *after*, followed by a time (timed assignment statement). The conditional signal assignment statement allows the use of conditional expressions to filter the expressions that appear in the actual signal assignment statement. The selected signal assignment statement is like the case statement of many high-level languages, enabling the results of the expression to be selected by a series of conditions based on a single value. A *guard* expression is associated with a block of assignment statements, implementing synchronous circuit descriptions.

The basic unit of design description is called a design entity. Any one entity may be reused many times within the overall description. Different architectures for a design entity, corresponding to alternative physical realizations of a given function, may be used in different portions of the description. So-called "generic formals," which appear in design entity declarations, take on different values in the different uses of the entity. Thus a design entity is analogous to a subroutine.

After dividing a large hardware description into parts represented by design entities, those entities must be implemented using specific hardware. Thus components are introduced as the basic unit of design implementation. The language is organized to permit components to be declared and instantiated within architectures. Then, a configuration specification binds component instances to the design entities and architectures that describe the desired parts. ■

With the wait statement, its expression is little more than the previous sentence in this paragraph.

■ FUNCTIONS, PROCEDURES, AND PACKAGES

Given the ability to create complex detailed algorithmic descriptions of behavior, the ability to decompose these descriptions (as hardware descriptions may be decomposed) is necessary. As with hardware decomposition, this ability involves the encapsulation of parts of the algorithm in different textual sections of the description.

The answer to software behavior encapsulation has been solved in other languages, including Ada. Ada includes *functions*, *procedures*, and *packages* to encapsulate behavior in different textual units. These are dealt with in other texts, particularly those concerning Ada. We confine ourselves here to a short discussion of packages.

Packages are used in much the same way as an "include" file. Definitions in a package are made available for use with a *use* statement. This statement names a package and makes all or part of the items in that package visible. For example, the statement "**use** resolution_functions.wired_and, system_types.**all**;" makes the function "wired_and" and everything in the package "system_types" available as though they had been declared where the use statement occurs. A use statement may appear virtually any place a declaration may appear and may make available either all of the declarations in a package specification or only selected declarations in a package specification.

Thus a package defines a specific function. Packages may contain signal and constant declarations, subprogram declarations, and various other declarations (types and the like). Packages may not declare variables and objects that change value, other than signals. Signals can be declared in packages. Such signals can be used by any design unit and fulfill the purpose of global resources (such as an overall system clock).

Signals may be used as parameters, with some very stringent restrictions. The functions defined in packages are free of side effects—they may not change objects they do not declare. A second restriction is that any operations on signals must be encapsulated in the process statement that operates upon them. It must be immediately obvious, from the textual context, which process is executing an assignment statement.

The use statement is a convenient shorthand. Any declaration in a package may be referred to, using an expanded name, which specifies the package containing the declaration.

■ BUS RESOLUTION FUNCTIONS

Depending on the technology and design techniques used, multisource signals exhibit different behavior. This behavior must be expressed and reflected in the hardware description. In VHDL, a signal that is driven by different processes (that is, one that appears on the left-hand side of a signal assignment statement in two or more processes) is called a bus. This usage is different from the normal definition of a bus as a collection of related signals or wires.

Given this terminology, the behavior of a bus is defined by a function that is specified by the declaration of the signal. This function is called whenever one of the processes changes the value that it is contributing to the entire value of the signal (called the driving value). All of the processes contribute their own portion of the value as elements in the array that is the single input parameter of the function; the returned value is the value of the signal. Consider the following example:

```
signal s: wired_and BIT;
s <= '0', '1' after 5ns;
s <= '0', '1' after 10ns;
```

Each signal assignment statement defines a process; "s" has two processes (called drivers) contributing to its value. The final value of "s" is determined by the bus resolution function "wired_and." The bus resolution function is called three times, at 0 ns, 5 ns, and 10 ns, with inputs (0,0), (0,1), and (1,1) respectively. (See also the

Advanced Features

There are many advanced features available in VHDL for attaching information to various portions of the design. Once the behavior of the hardware has been described by the mechanisms already given, there remains that portion of the design that is not directly associated with behavior. This portion might include assertions describing relationships between signals, user-specific information about components or wires, and how conversion among types is handled.

■ ASSERTIONS

The behavior of a circuit does not completely describe that circuit. In particular, it is useful to be able to specify relationships between signal values and ports in such a way that these relationships are subject to confirmation. To that end, VHDL includes an assertion statement. It may occur either inside a process statement (a sequential context) or outside a process statement (a concurrent context). It specifies a relationship or Boolean condition and reports an error if this condition is ever false. Conditions too complex to be expressed in a single condition require a process statement, which can declare variables and the like.

■ USER-DEFINED ATTRIBUTES

No matter how complete a design language, there is always additional information that is part of a design. VHDL addresses specification, behavior, and logic design, while largely ignoring physical design. For example, the user may need to specify the location of a particular chip on a printed-circuit board. "Attributes" are the means within VHDL of recording and using this kind of information.

The values of user-defined attributes do not change and are accessed in a similar fashion to predefined attributes. Consider the following package that sets up an attribute for recording board locations:

```
package attribute_declarations is
  type board_location is
    record
      x, y: integer;
    end record;
  attribute position: board_location;
end attribute_declarations;
```

The statement with the key word *attribute* is called an attribute declaration. It marks the fact that the identifier "position" is an attribute and gives the form of that attribute (a record with two integer components).

Then, within an architecture description, the user could provide statements of the form:

```
attribute position of X1:label is (2,2)
```

which are attribute specifications. These statements actually specify the value of the attribute for the listed objects (the label in the statement). In this case, component label "X1" has an attribute value (2,2) associated with it. An attribute can also be attached to component declarations, procedures, architectures, entities, and other portions of the language. Attributes may also provide information to tools processing VHDL descriptions (such as automatic layout tools) or receive information produced by these tools (in a back-annotation process).

```
package conversions is
  function int_to_bits (int_value:INTEGER) return BIT VECTOR;
  function bits_to_int (bit_value:BIT VECTOR) return INTEGER;
end conversions;

use conversions;

architecture structural_int of int_adder is
    signal cout:BIT;
    component four_bit_adder
    port (x, y: in BIT ARRAY; cin: in BIT;
      sum: out INTEGER; cout: out BIT);
    end component;
begin
    adder:four_bit_adder
      port map (x => int_to_bits (addend1),
        y => int_to_bits (addend2),
        bits_to_int (sum) => sum,
        cout => cout);
    assert cout = '0'
      report "Overflow occurred."
      severity ERROR;
end structural_int;
```

Example showing the use of type conversions. Code for the conversion routines themselves is not shown.

■ TYPE CONVERSION FUNCTIONS

The ability to express information in different types, is extremely powerful. At times in the design, a designer will need to connect a port to a signal of a different type. Some mechanism of making this connection is necessary. Functions may be used to convert values from one type into another. Such functions may be attached to port maps in order to accomplish this conversion.

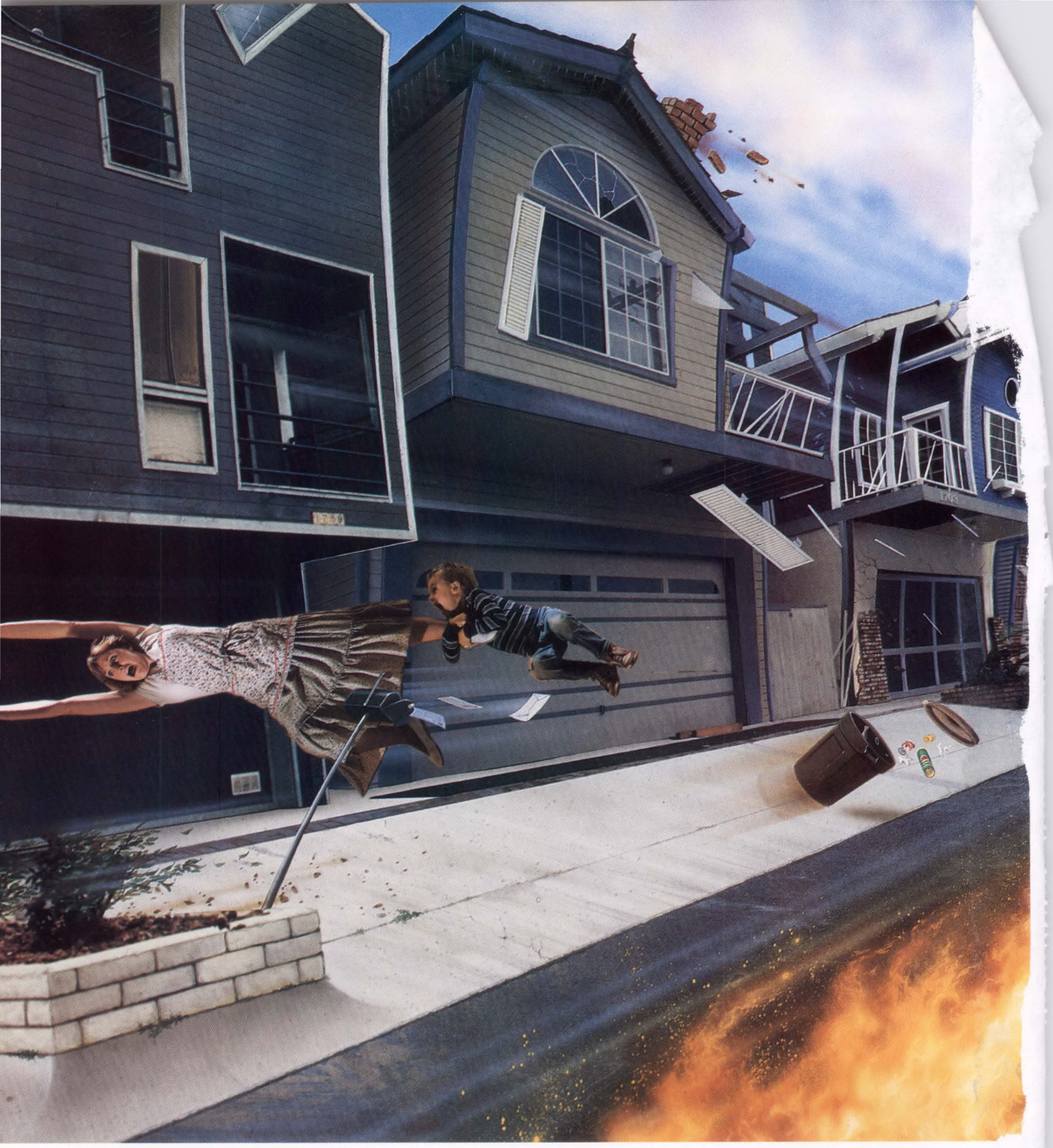
Consider the example shown in the figure. The type conversion functions are defined in the package "conversions" and used in the port map aspect of the component instantiation statement with label "adder." (Note that the package body—the code that actually implements the conversion routines—is not shown in the figure.)

We now see a form of port association that we have not seen before, called named association. The formal ports are named explicitly, and the actuals associated with the formals are designated by use of the "=>" symbol. It is not possible to "skip" a parameter with positional association, but it is possible in named association, since order is not important.

The position of the type conversion function depends on the mode of the port. Type conversions for ports of mode *in* appear on the actual name; type conversions for ports of mode *out* appear on the formal name; type conversions for ports of mode *inout* appear on both names.

Type conversion functions have much the same role in the simulation cycle as the bus resolution function. Thus the determination of the value of a signal may require the evaluation of both the type conversion function and the bus resolution function. ■

WITH THE WRONG ASIC DESIGN TOOLS, YOU COULD BLOW THE WHOLE PROJECT.



WITH VLSI'S ASIC TOOLS, YOU CAN

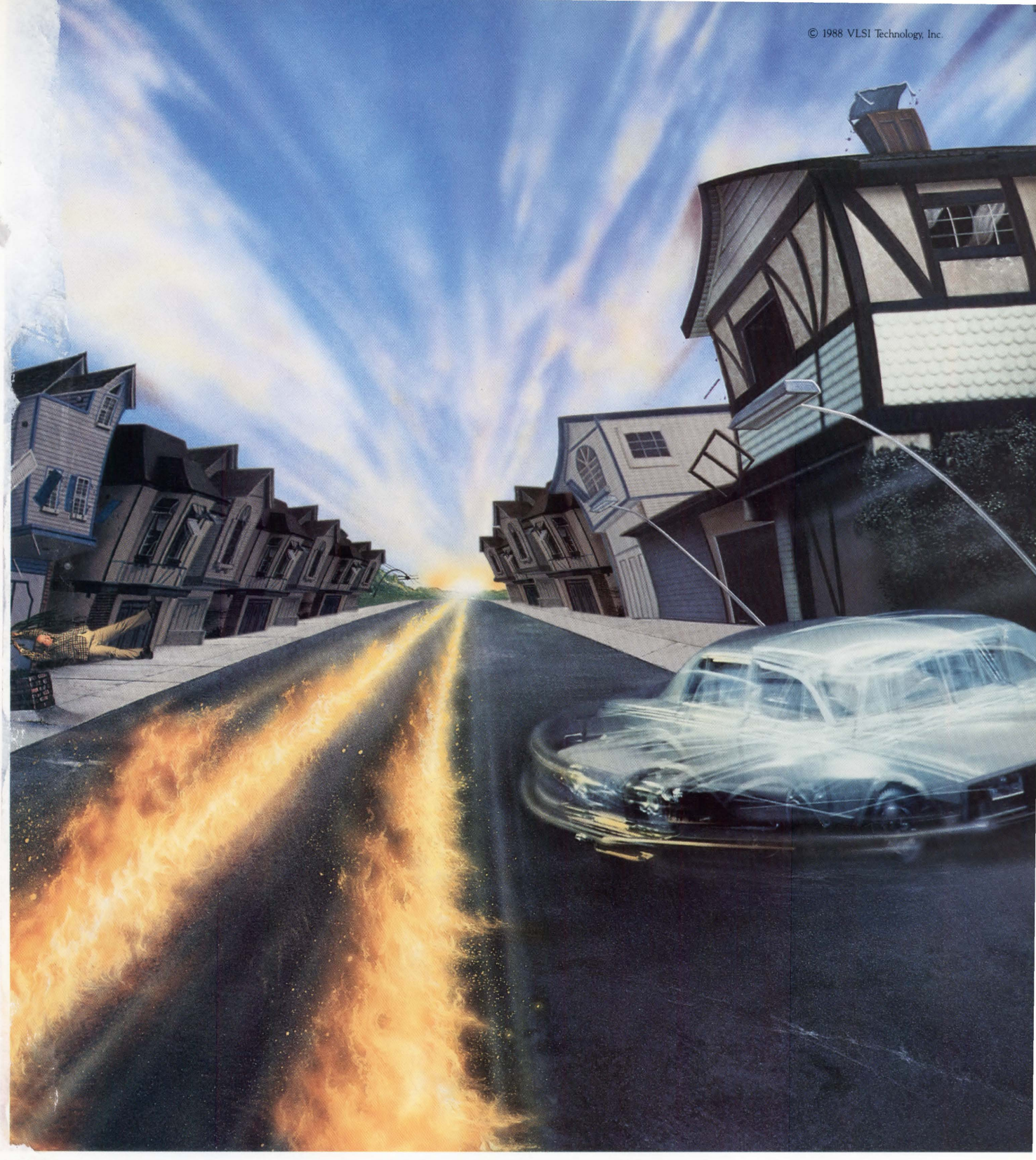
VLSI Technology's software is the fastest, surest way to create successful ASIC chips.

Our tools accelerate each and every step of the design process.

And our years of experience insure the success of your finished ASIC chip.

EVERYTHING'S FASTER WHEN YO HAVE AN ASSISTANT.

Our Design Assistant™ tool partitions yo chip and estimates chip size, power consun



BLOW RIGHT BY THE COMPETITION.

tion, and packaging possibilities. In short, it tells you the best silicon solution to your problem.

Simply enter your design in block diagrams and global interconnect forms. Design Assistant shows you ways to implement it.

Gate array, standard cell or cell-based.

You can run dozens of "what if" configurations in a few hours. In no time, you'll have the crucial information you used to wait forever for vendors to supply.

STREAMLINED DESIGN, COMPLIMENTS OF OUR COMPILERS.

Our logic and memory compilers create multipliers, PLAs, or super fast SRAM memory blocks as fast as 8ns.

And we've got a couple of "expert" compilers for cell-based or gate array designs that think exactly the way you do.

When you design a datapath, you think of it as a linear schematic, right?

Well, our Datapath Compiler just happens to use schematics as input. Complex multi-bit datapaths practically pop right out of it.

When you design state machine and other logic blocks, you think in terms of equations. Our State Machine Compiler does, too.

And it even optimizes your equations.

GET IT TOGETHER. FAST.

Ready to place and route your cell-based design? Just call up our Chip Compiler.

It provides floor-planning and auto-routing. It works with our compilers and standard cell libraries to produce an optimized layout.

It even pours standard cells into the gaps between blocks to make sure you use the minimum chip area required.

How fast does it work? Glad you asked.

A company came to us with a layout that took them three months. We did it in two days.

TO BE FAST, YOU HAVE TO BE FLEXIBLE.

Only our Portable Library lets you choose gate array or cell-based, 1.5μ or 2μ, *after* you've completed your logic design.

It can allow you to do that because your library always remains stable. No matter what process you use.

Process obsolescence is now obsolete.

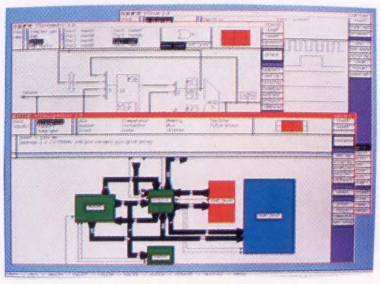
IF YOU NEED TO GET THERE FAST, TAKE THE EXPRESS.

You can buy VLSI's tools in six configurations. From our quick, easy-to-use Logic Express™ to the powerful Design Express.™

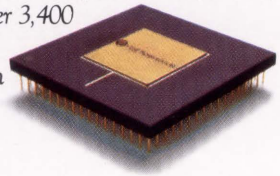
If you'd like to find out how quickly you can design successful ASIC chips, give us a call at (800) 872-6753.

Because when you're driving to market this fast, it's good to have insurance.

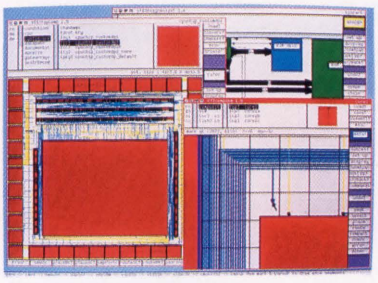
CONCEPT EXPRESS™:



The Concept Express Design System's highly productive logic tools and silicon compilers were used to develop this very-large-scale ASIC. It incorporates a 2901 datapath, RAM, ROM, and over 3,400 gates of random logic.

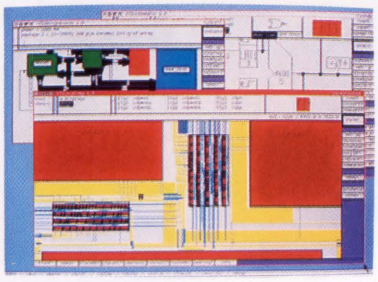


DESIGN EXPRESS™:

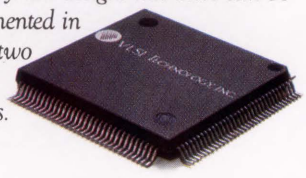


This highly-integrated design combines control logic, a register file, a refresh counter, and five peripheral chips onto a die size of 275x315 mils. The logic design, layout, and verification were completed in only 12 weeks.

SILICON EXPRESS™:



This design integrates all the peripheral chips for an AT computer with six megacells and control logic. Using the Silicon Express Design System, logic and physical designs like these can be implemented in under two man months.



VLSI TECHNOLOGY, INC.

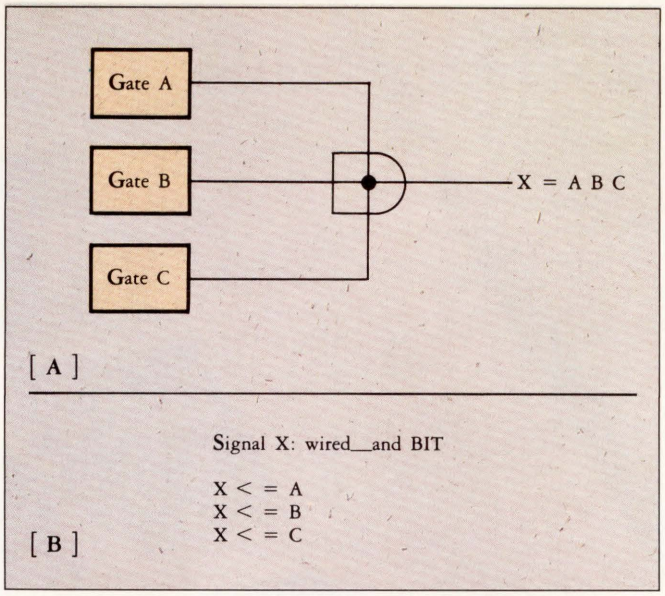


Figure 3. An example of a bus resolution function: the wired-AND circuit topology (A) is implemented with the "wired__and" function call (B).

example in Figure 3.)

Different signals may have different bus resolution functions. There need be no single behavior throughout a hardware description; different behaviors may be used, just as different techniques may be used for different multisource signals in real designs.

The bus resolution function is a major part of the flexibility of VHDL. The behavior of signals, which may represent wires, is specified by the user. Actually, most users will not have to write bus resolution functions; they will typically be provided in standard packages by the user's support organization.

■ TYPES

All the examples thus far have been given in terms of bits and arrays of bits. More complex methods of description are needed to represent unknown conditions and high-impedance lines. Moreover, describing quantities such as voltage and current requires some more descriptive typing mechanism than simple integers. VHDL includes an extensive typing mechanism. Bits, integers, and floating-point numbers are represented as basic types. Arrays and records may be formed of these types, much as in Ada. In addition, there are several features in VHDL that are unique to the hardware domain.

VHDL allows the definition of enumeration types, just as in Ada. An example is:

```
type multi_valued_logic
  is ('0', '1', 'x', 'z');
```

Signals, variables, and constants could all be declared to be of type "multi_valued_logic."

Another typing mechanism is the physical type. A physical type may assign units of measurement to different values of that type. While acting as integers computationally, physical types allow numbers to be expressed in terms of the appropriate units.

The type mechanism in VHDL also provides the ability to specify a direction to an array or a range. Directions in arrays are useful for reflecting those cases in which the pins in an address or data bus are really in reverse order.

The typing mechanism in VHDL is powerful and therefore complex. Most designers will use a collection of types that have been placed in a package. These will be made available with a use statement.

Normal expressions may operate upon objects of user-defined types. All the infix operators ("+," "**xor**," and the like) may be overloaded such that their effects upon values of user-defined types are well-defined. Such overloaded function definitions will usually be provided with the same package in which the types are defined. The normal user of VHDL will not have to write such functions. Most Ada texts cover the subject of operator overloading thoroughly, and the reader is referred to any one of them for more information on this subject.

■ CONCLUSION

VHDL is an extremely rich language, with a variety of language features for many different types of situations. Specific features are designed to facilitate the description of hardware behavior by means of Boolean expressions, structural description, and algorithmic definition. There are also features that permit extra information to be attached to various portions of the hardware design, in the form of user-defined attributes and assertions (see "Advanced Features," p. 32).

The presence of packages and design entities allows a given installation to establish standard design practices. A given installation may use a set of standard library units that define types, bus resolution and type conversion functions, standard design units corresponding to available hardware components, and other portions of a design. The majority of designers on a project will usually use a small portion of the language features, while other technicians may write and assemble these standard library units.

The language features governing design decomposition allow a large team to work on portions of a complex design in isolation. Different parts of the design may be connected just as would any other component or set of components. Type conversion functions make it possible to resolve any type differences between the various ports in the design. The configuration allows the final measure of control, appropriately connecting unconnected ports, resolving unresolved generic values, and selecting the actual architectures to be used for the design.

The richness of VHDL entails some complexity. This complexity can be controlled by selecting design practices within an installation that limit the number of language features that need to be learned by the majority of designers. In this manner the complexity of the actual design, as well as the complexity of learning a rich and powerful hardware description language, can be controlled. VHDL can indeed be an asset in the hardware design process, regardless of the size of the design. ■

ACKNOWLEDGMENTS

The original suggestion for the overall format of this article came from Dr. Al Gilman. Craig Brown, Rachael Rusting, Al Gilman, and Roger Lipsett all have reviewed earlier versions of this article and made valuable suggestions. The author gratefully acknowledges his debt to these people.

REFERENCES

BARTON, D.L. 1988. "A First Course in VHDL," *VLSI System Design's User's Guide to Design Automation.*

VHDL Language Reference Manual, Draft Standard 1076/B. May 1987. IEEE Computer Society Publications Department, Los Angeles, Calif.

ABOUT THE AUTHOR

DAVID L. BARTON *is currently a principal language consultant at Intermetrics. He is also currently enrolled in the PhD program at the University of Maryland. Barton is the designer of PLCB, a programming language for use with a shared memory multiprocessor. His current research interests include the formal specification and verification of distributed programs. He received the MS degree in computer science from the University of Maryland in 1984.*

Our ASIC line-up is only half the story.

When it comes to delivery of high-quality, reliable ASICs, S-MOS wrote the book.

We did it in collaboration with our manufacturing affiliate, Seiko Epson. With 18 years of CMOS experience, Seiko Epson is one of the world's most advanced CMOS IC manufacturers.

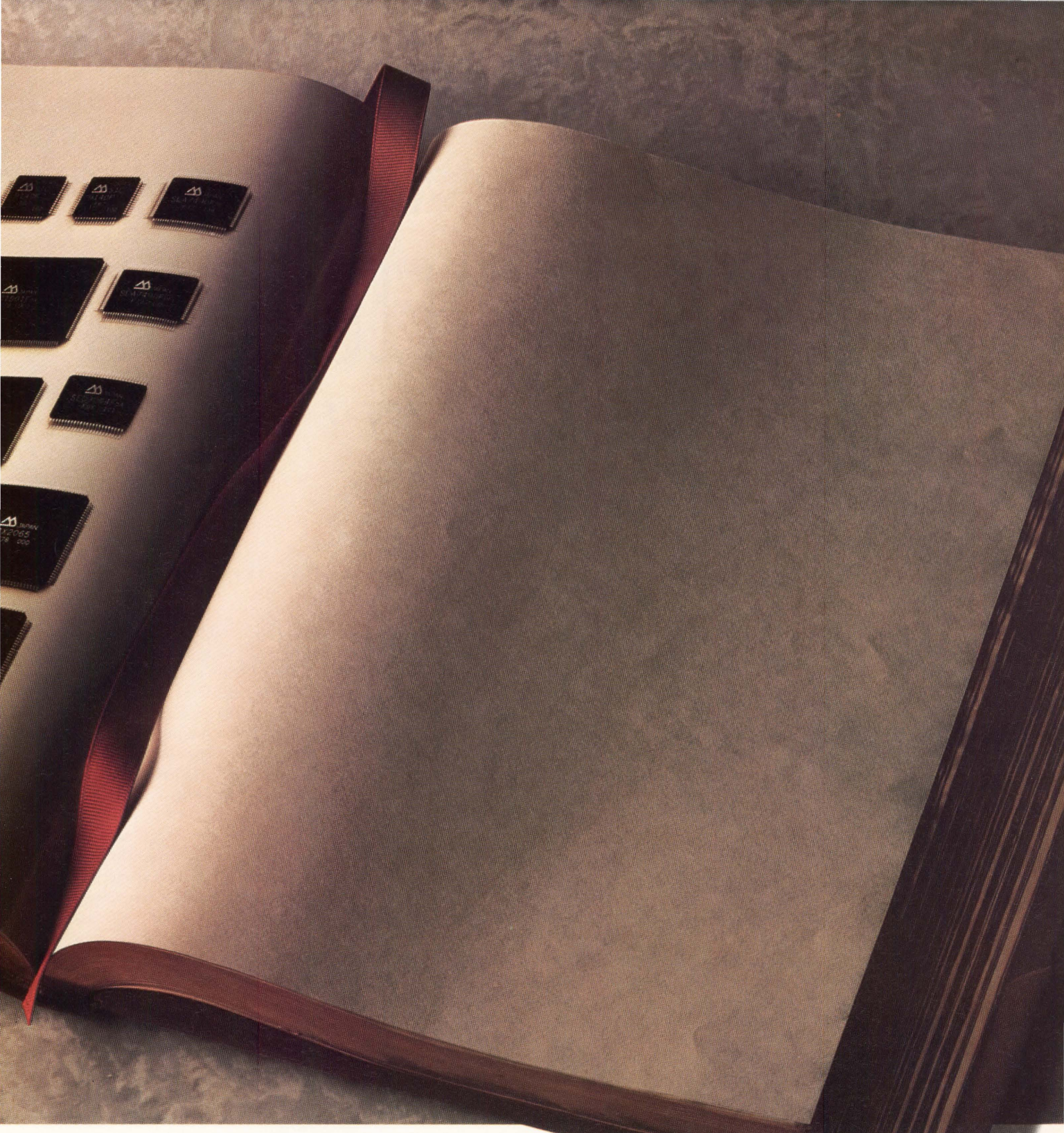
Through Seiko Epson's high-yield manufacturing technology, we ship millions of ASIC units a month, and with a reject rate of less than .0001%. That's our quality story.

Now we've added a new chapter on design. At our advanced R&D design facility, engineers from S-MOS and Seiko Epson are developing new software to simplify circuit design, simulation and the creation of new megacells for our extensive cell library.

Of course, you can still take advantage of our established design tools because S-MOS supports such workstations as Daisy, Mentor, Calma and PC-based systems using FutureNet, OrCAD and ViewLogic.

Our proprietary LADS simulator will speed up the design process. The S-MOS engineering team will support you from concept to production.

If you are looking for an ASIC program that can make your designs best sellers, call us. (408) 922-0200.



CMOS Gate Arrays	CMOS Standard Cells	Compiled Cell Custom
Up to 38,550 available gates ▪ SLA8000 (800 ps)*, 1.2µ drawn, 1.0µ Leff. ▪ SLA7000 (1.0 ns)*, 1.5µ drawn, 1.2µ Leff. ▪ SLA6000 (1.8 ns)*, 2.0µ drawn, 1.5µ Leff. ▪ SLA700B High Drive Output ▪ SLA100L Low Voltage	Complexities to 16K gates** ▪ SSC1000 (1.4 ns)*, 1.8µ drawn, 1.4µ Leff. ▪ Fully migratable from S-MOS gate arrays ▪ RAM and ROM blocks available	*The* Alternative to Full Custom ▪ 1.8µ CMOS process ▪ Can utilize dissimilar cell geometries ▪ 3-button approach to custom design ▪ Currently over 300 fully characterized cells ▪ Fast 14-week implementation time ▪ Timing-driven TANCELL® place-and-route software

TANCELL is a registered trademark of Tangent Systems.
*Typical propagation delay of 2-input NAND gate driving 2 internal loads with 1mm of interconnect.
**Maximum gate utilization depends on amount of interconnect used.

S-MOS SYSTEMS

S-MOS SYSTEMS, INC.
2460 North First Street
San Jose, CA 95131-1002

CIRCLE NUMBER 9

BOOSTING ASIC VERIFICATION Capabilities

NORBERT R. LAENGRICH, HILEVEL TECHNOLOGY INC., IRVINE, CALIF.

■ VIRTUAL MEMORY TACKLES MEGABIT VECTOR SETS ■

Comprehensive design verification of new, complex VLSI devices has been limited by the inability of test systems to deal with a number of issues. These issues include extended test vector sets, data rates, and timing accuracy.

For example, as devices with equivalent gate counts exceeding 100,000 become commonplace, the need for design verification systems that can easily accommodate larger vector sets becomes more critical. In addition, as the devices move to smaller fabrication geometries, they can support higher and higher data rates with increasingly finer timing resolutions.

A VLSI verification system that addresses these problems is the new Topaz-V design verification system. It tackles the problem of large vector sets with a new architecture called "virtual vector memory." The new architecture supports extended vector sets without the high cost of the pin-level SRAM required by traditional architectures.

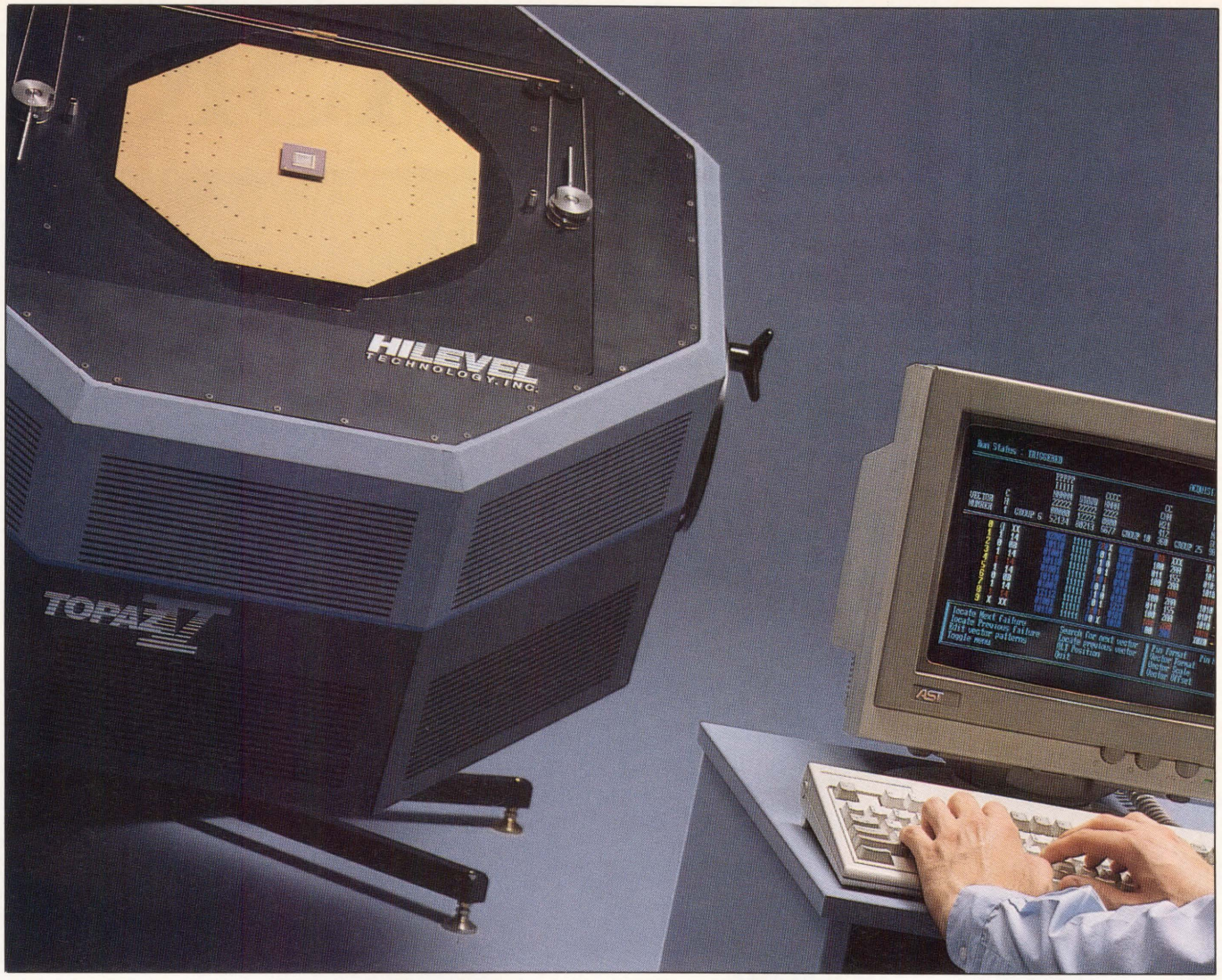
■ DESIGN VERIFICATION REQUIRES EXPANDED VECTORS

Whereas the need for an increased number of test vectors during production test is primarily a function of device complexity, test vector requirements for design verification are driven by three interactive forces. These forces combine to create design verification vector sets that may be double or triple the length of those that will eventually be used in production testing.

First, of course, is device complexity. ASIC and other VSLI devices using 0.5- to 1.5-μm geometries may contain 50,000, 100,000, or more equivalent gates. During the next few years the complexity of these custom and semicustom devices will continue to grow, and the growing complexity requires verification and test systems that can support higher pin counts and expanded test vector depth. The 1K, 4K, or even 16K vector memory depth of yesterday's design verification systems will not suffice.

The second factor affecting the vector requirements of design verification is the growing use of software programs linking CAE to testing. For higher-volume production testing, the cost versus the return of optimizing vector sets for throughput can often be justified. However, at the design verification stage, the key element is time to market. A fast, easy conversion of simulation results to test vectors is therefore a must. Since simulation is designed to provide high coverage of possible design errors, the resulting vectors are usually in excess of those required for functional testing of a proven design. Consequently, more vectors will be created when simulation vectors are converted into test vectors for use during design verification.

Third, the purpose of design verification is not only to prove the design, but also to characterize the new device. For proper characterization, a larger vector set is required than would be required to test a known good design. For example during production testing, only valid pin states need be tested for functionality. However, during design verification, invalid states may need to be checked to ensure safety, and such, should external system faults cause inva-



lid states on the device pins.

These three factors combine to demand that design verification systems be capable of executing extended vector sets without time-consuming downloading cycles.

Topaz-V is the first design verification system to allow extended vector sets to be executed without breaking the vectors into pages and waiting for the time-consuming downloading between each page of vectors.

■ Current Limitations

Other verification systems typically offer 4K or 16K of vector memory behind each pin. Therefore, after each 4K (or 16K) group of vectors is executed, the system must be reloaded with the next page. In addition, downloading from the disk memory in a host computer is often through RS-232 or IEEE-488 interfaces. Tables 1 and 2 give some typical download times using these two interfaces at common transfer rates. The values listed make no allowance for system overhead and assume that each bit is transferred in ASCII code. When system overhead, host computer overhead, and disk access time are added, large vector files can extend the time for a single test into hours or even days.

Although these tables show the total time to be identical regardless of the tester's memory size, in actual practice it will increase for the smaller memory sizes because of the need to add system and computer overhead time for each page downloaded. As a result, a larger memory size will offer shorter total test times. Unfortunately, increasing the depth of the active SRAM behind each pin can significantly increase system price. Hilevel offers models that have a 64K vector memory option and use a proprietary high-speed 8-bit parallel interface bus for customers requiring large vector sets. However, even with these improvements, it still requires several minutes to run a single pass of 1 million vectors.

■ Virtual Vector Memory

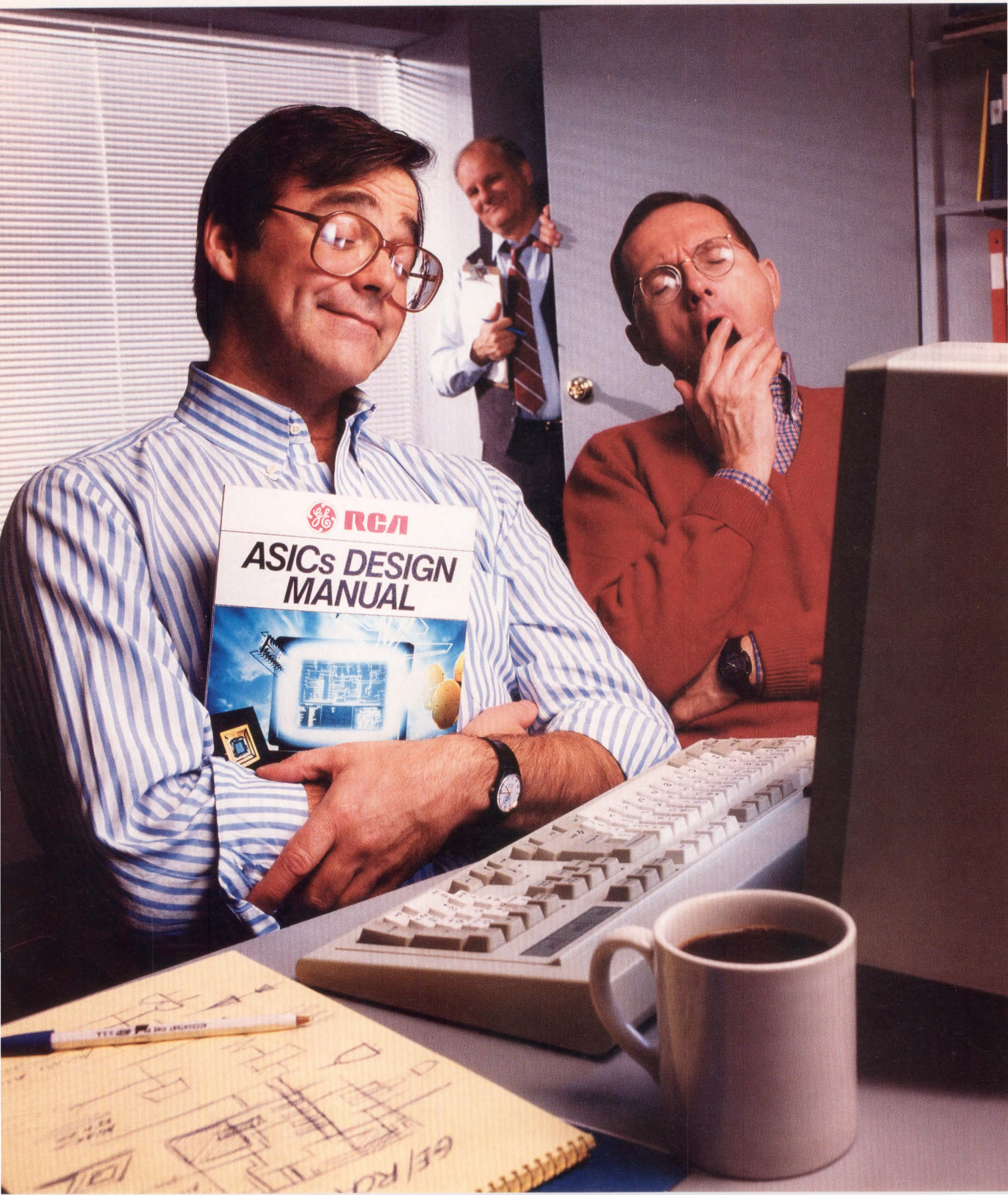
The virtual vector memory architecture allows large vector sets to be executed without the need for extended memory behind each pin. This capability is especially important with the per-pin architecture of the Topaz-V, since each pin features 4 bits of control memory per vector.

The virtual memory consists of four elements: interleaved RAM, data compression and expansion, a 32-bit proprietary bus, and pin-level SRAM. Figure 1 shows how these elements tie together with other portions of the verification system.

In operation, the vector file is moved from the hard disk into the interleaved RAM. Options allow

PAGINATION OF VECTORS IS ELIMINATED

Our ASICs



are boring.

They're easy to design. They're ready on time. And first-time success is virtually 100%.

You've heard all about the excitement of ASICs.

They improve performance, lower costs and make many new designs possible.

But, unfortunately, you've probably also heard about one big potential problem: while many ASICs pass the tests specified by the designer, they don't always work in the real world. And that causes excitement you can do without.

How to get first-time success.

It starts with our Design Simulation Software. It's been rated the best in the industry by the people who should know—designers who have used it. Within three days, you can be up to speed, working at any of the major workstations in the industry, creating and revising your ASIC with ease.

The standard cell advantage.

You'll really appreciate the power of our standard cells, which allow you to integrate a whole system, including macros, memories, logic and peripherals, onto a single chip.

We have cells with effective gate length as small as 1.5μ (.9μ coming soon). And double-level metal for higher-density chips that can handle higher clock speeds.

You can choose from a wide range of Supercells, including the leading-edge RS20C51 core micro, RAMs, analog functions, bit-slice processors, HC/HCT logic, Advanced CMOS Logic, and high-voltage cells.

If they aren't enough, we can even generate Supercells to your specs.

And we're also in the forefront of silicon compiler technology. So we can offer you the ability to create designs that are heavily BUS-structured, with your ROMs, RAMs, PLAs and ALUs compiled right into the design.

We also bring you the resources of some very powerful partners, thanks to our alternate-source agreements with VLSI on standard cells; WSI on macrocells and EPROMs; and a joint-development agreement with Siemens and Toshiba on the Advancell® library of small-geometry cells.

Gate arrays, too.

If gate arrays are better for your design, you'll be able to choose from our full line up to 50,000 gates, with effective gate length as small as 1.2μ and sub 1 ns gate delays.

These gate arrays use "continuous gate" technology for up to 75% utilization. They are an alternate source to VLSI Technology arrays.

We also alternate source the LSI Logic 5000 series.

And we have a unique capability in high-rel ASICs, including SOS. Our outstanding production facilities here in the U.S. produce high-quality ASICs in high volume at very low costs.

It almost sounds exciting for something so boring, doesn't it?

For more information, call toll-free today 800-443-7364, ext. 25. Or contact your local GE Solid State sales office or distributor.

In Europe, call: Brussels, (02) 246-21-11; Paris, (1) 39-46-57-99; London, (276) 68-59-11; Milano, (2) 82-291; Munich, (089) 63813-0; Stockholm (08) 793-9500.

USA

GE Solid State

GE/RCA/Intersil Semiconductors
Three great brands. One leading-edge company.

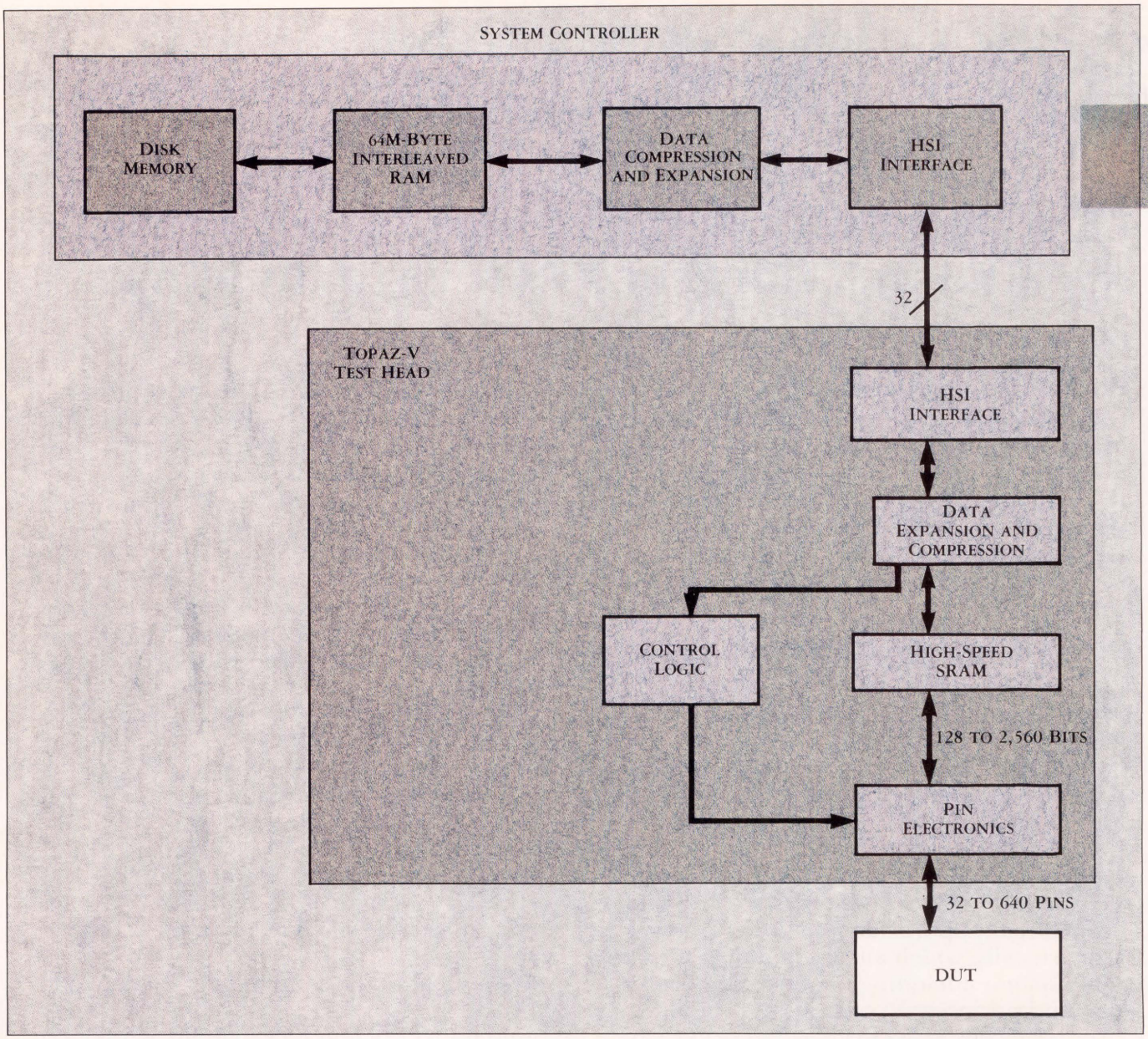


Figure 1. The virtual vector memory architecture allows the running of large vector sets (up to 64 megabytes) in seconds without extended page breaks.

TABLE 1. TIME TO DOWNLOAD ASCII-CODED VECTORS

	TEST MEMORY SIZE		
INTERFACES AND SPEED	**4K × 200 PINS**	**8K × 200 PINS**	**16K × 200 PINS**
RS-232, 9600 BAUD	1.4 MINUTES	2.8 MINUTES	5.6 MINUTES
RS-232, 19.2k BAUD	42 SECONDS	1.4 MINUTES	2.8 MINUTES
IEEE-488, 50k BYTES/s	16 SECONDS	32 SECONDS	1.1 MINUTES
IEEE-488, 100k BYTES/s	8 SECONDS	16 SECONDS	32 SECONDS

NOTE: Assumes no system or computer overhead and one memory column behind each pin.

TABLE 2. NUMBER OF PAGES AND TOTAL DOWNLOAD TIME REQUIRED FOR 1 MILLION VECTORS × 200 PINS

	TESTER MEMORY SIZE (BYTES)
	4K 8K 16K
NUMBER OF PAGES	**244 122 61**
DOWNLOAD TIME	
■ RS-232, 9600 BAUD	5.7 HOURS
■ RS-232, 19.2k BAUD	2.9 HOURS
■ IEEE-488, 50k BYTES/s	1.1 HOURS
■ IEEE-488, 100k BYTES/s	33 MINUTES

the user to select 16 or 64 megabytes of interleaved RAM with an effective access time of 16 ns. This memory can also be edited and modified on line.

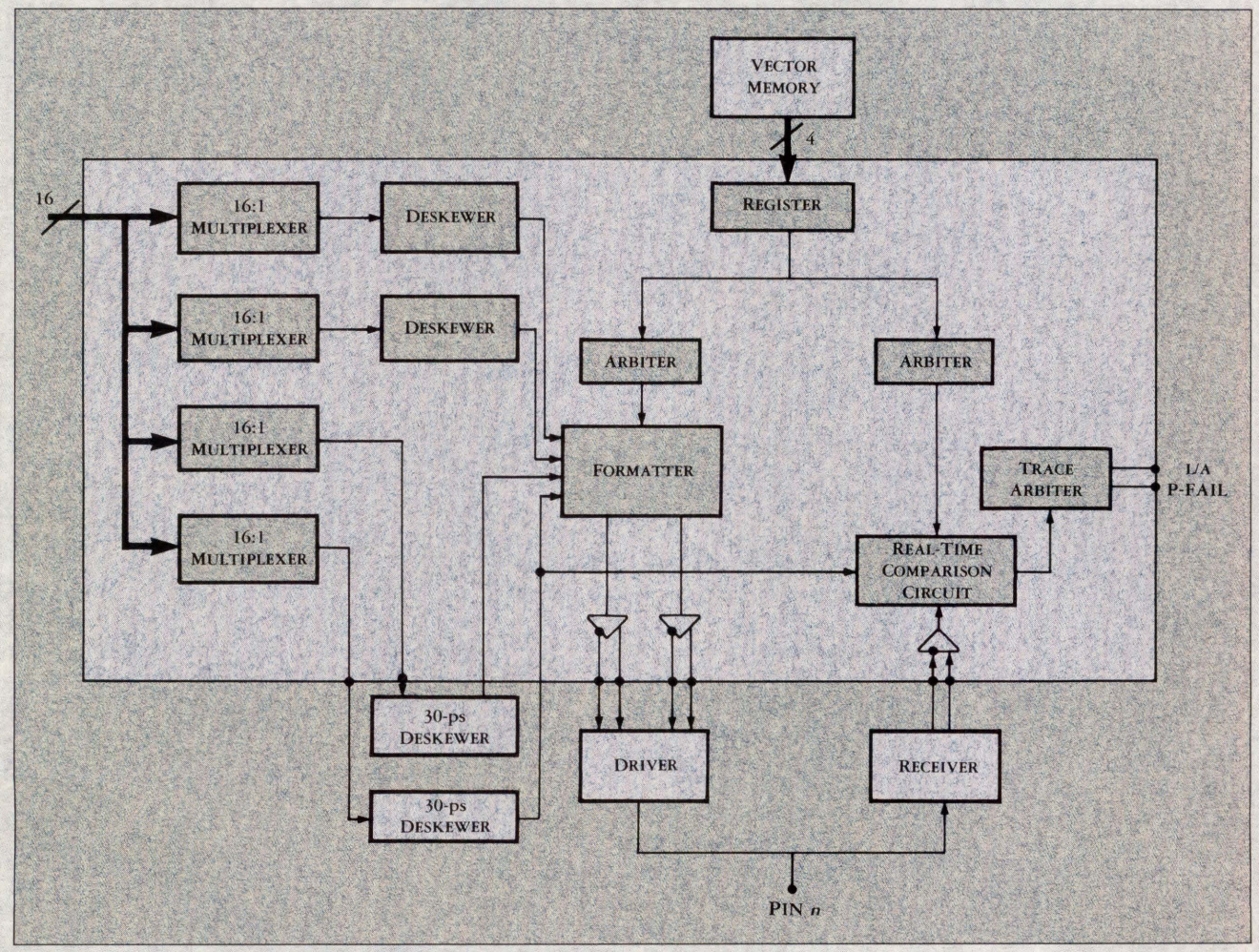
When a test is executed, the system moves the first block of data through the data compression circuitry and across the 32-bit high-speed bus to the test head, where it then passes through the data expansion circuitry and into the static RAM that feeds the pin electronics. The system is available with either 4K or 16K of SRAM,

Extensive Use of ASICs Enhances Virtual Vector Memory Architecture

The Topaz-V makes extensive use of the very devices it was designed to test and evaluate—ASICs. These custom and semicustom chips provide features and drive capabilities previously unavailable in design verification systems.

To guarantee speed yet keep the power consumption and heat dissipation low, the company's design engineers chose the new low-power Raytheon CGA40E1Z ECL gate arrays to provide the full logical control of the pin electronics. Each gate array, which contains 8,000 equivalent gates, provides four sets of pin logic (see the figure) and consumes only 4 W. Compared with the ECL gate arrays used in the company's other verification systems, this figure represents a 6:1 reduction in power. Each one of the four sets of pin logic controls the stimulus, expected response, I/O direction, and inhibit function for control stroke masking—for a separate pin. In the "failure data only" mode, only the Fail output for each pin (when the actual output is different from the expected response) is uploaded for storage and analysis. However, the actual output signals also can be fed directly to the verifier's logic analysis system for device debugging and evaluation.

Another ECL gate array is combined with additional external surface-mount circuitry to create the 100-ps delay generators for timing control within the new verification system. These 16 delay generators resolve 100 ps over a range of 0 to 10 ms. The monotonic characteristics of the delay generators allows the system to deliver an overall accuracy of ± 1.75 ns.



Four sets of the pin control logic are included in each Raytheon CGA40E1Z ECL gate array.

A third ASIC design implements the pin driver electronics. This design uses AT&T's linear bipolar arrays, which feature transistors with an f_T in the gigahertz range. The drivers also deliver a 1.5-V/ns slew rate and a 50-mA drive capability. The fast rise and fall times are essential to the testing of high-speed ASICs, such as those built with HCMOS or ECL technologies.

A verification system must be able to characterize the individual pin delays to differential values established by the characteristic gate delay of the device. The use of the three ASIC designs enables the Topaz-V to characterize emerging submicron-geometry devices with gate delays of 100 to 500 ps and clock rates greater than 100 MHz. ■

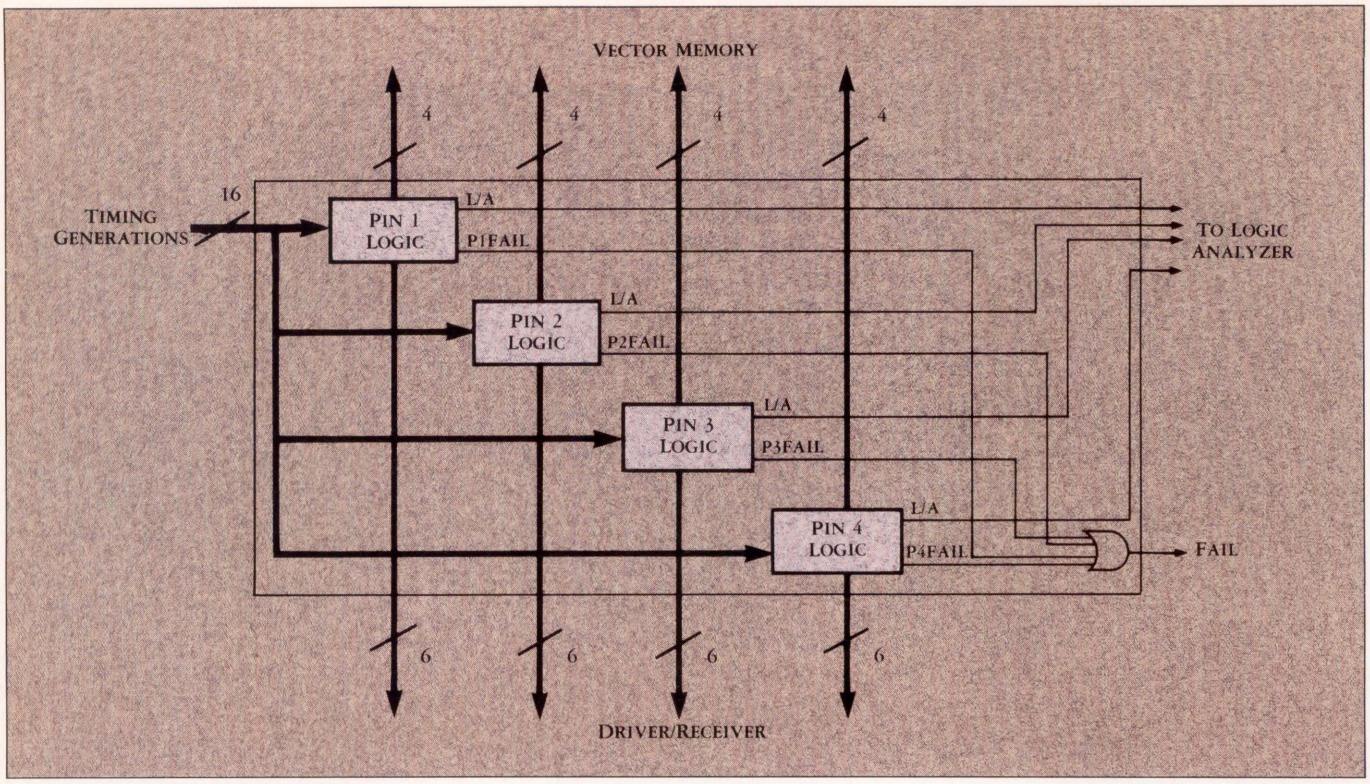


Figure 2. Each ECL gate array delivers full logic control of four I/O pins.

TABLE 3. TOTAL TEST TIME FOR 500K × 200-PIN VECTORS

ARCHITECTURE	TEST TIME
TOPAZ-V	<5 SECONDS
TRADITIONAL (WITH 16K MEMORY AND HIGH-SPEED BUS)	>35 MINUTES

although 4K is sufficient for most applications.

The combination of interleaved memory, data compression/expansion, and a high-speed 32-bit bus allows an equivalent transfer rate of 0.5 gigabits per second. The high-speed design enables a full 16K per pin of vectors for 320 pins to be moved from the interleaved RAM to the pin-level SRAM in 50 ns. This capability makes it possible to execute very large vector sets without extended test times. Table 3 shows how this new architecture compares with the more traditional approaches.

In addition to allowing the system to run large vectors sets, the virtual vector memory architecture also reduces the download time when the engineer wishes to examine the device performance with multiple vector sets. Download time for smaller vector sets takes only milliseconds, instead of minutes, giving the designer greater freedom to verify the device performance under a variety of operating conditions—without eating into his valuable design time.

■ UPLOADING ALSO ENHANCED

Another important factor that can affect total test time is the requirement to save the test results. Previous systems have addressed this issue by allowing the engineer to collect and upload only selected information to the host computer rather than the complete response from the device under test (DUT). Although this capability reduces the amount of data to be saved, the upload time is still limited by the same factors that were discussed previously.

The virtual vector memory architecture can also be used to accelerate the uploading of test results. When the new memory is operated in the "failure data only" mode, the time required for the saving of test results is so small that it is virtually transparent to the engineer.

The Topaz-V has also incorporated many other technologies, such as ECL gate arrays for the pin electronics (Figure 2), to provide the high data rates and accuracy required for verifying today's and tomorrow's complex ASICs. In addition to extensive use of custom chips (see "Extensive Use of ASICs Enhances Virtual Vector Memory Architecture," p. 41), the system features many automatic calibration capabilities. A four-stage calibration procedure corrects for driver-to-driver, receiver-to-receiver, driver-to-receiver, and external system skew. For example, the external skew of cabling, the socket, and so on, is corrected using the system's built-in time-domain reflectometer capability.

The system is also enhanced with a variety of other features, including loads that are individually programmable for each pin; a 110-MHz test rate, a 220-MHz clock generation rate, a dc parametric measurement unit, shmoo plot software, CAE link software, full interactive screen and keyboard control, Sentry link software, and the C programming language.

■ SUMMARY

ASIC designs will continue to become faster and more complex. Devices with 100-MHz data rates, 300 pins or more, subnanosecond gate delays, and over 100,000 equivalent gates will require design verification systems capable of easily and quickly providing hundreds of thousands of test vectors. The new virtual vector memory architecture meets that need now. ■

ABOUT THE AUTHOR

NORBERT LAENGRICH, *vice president of marketing, has been with Hilevel Technology since 1984. Previously, he was with Racal-Dana Instruments for 10 years. Mr. Laengrich was also with Data Technology and Raytheon, with district sales management responsibilities.*

What's waiting for you at the end of the rainbow?

CADENCE. For a full spectr

ECAD and SDA have joined forces to offer you an integrated family of IC design solutions from a single source – CADENCE.

Out of the merger of ECAD and SDA, a clear leader has emerged from the storm of IC CAD vendors. CADENCE. A company with the products, support and R&D resources to provide the integrated, high-performance design solutions you need today.

What you can expect from CADENCE.

... A merging of the best IC design technologies — SDA's Design Framework™ combined with high-performance tools, like ECAD's Dracula™ and Symbad.™ The combination increases your productivity and gets your designs to market faster than ever before.

... A uniting of two world-class software development groups to

um of IC design solutions.

create the largest and best R&D team of its kind — all dedicated to solving your IC design challenges.

...A future as bright as the rainbow itself. Two great design technologies, two world-class R&D teams and the worldwide sales and support resources to back them up. A total commitment to the success of your next IC design.

Look to the end of the rainbow... and discover CADENCE.

Send now for our new capabilities brochure and find out how to make your design dreams come true. CADENCE, 555 River Oaks Parkway, San Jose, CA 95134. 408/943-1234.

CADENCE
Leadership by Design

See us at DAC Booth #210

CIRCLE NUMBER 10

A DESIGN AUTOMATION Environment

KYLE GOLDMAN AND TED STOUT, EDA SYSTEMS INC., SANTA CLARA, CALIF.

SIMPLIFYING THE USE OF HETEROGENEOUS DESIGN TOOLS

DESIGN automation tools for electronics applications have made dramatic advances over the last decade. For example, logic simulators, have graduated from the domain of running in the batch mode on mainframe computers to sophisticated and powerful interactive tools on PC-based systems. At the same time, the electronic CAE industry has grown from essentially three suppliers (Daisy, Mentor, and Valid) selling "turnkey" systems to over a hundred companies with specific, powerful tools focused on automating individual design tasks. Examples of such tools are simulators like Verilog, Cadat, and Silos; schematic editors like Viewlogic's and Orcad's; and layout tools like Caeco's and Ecad's Dracula.

While providing new features, capabilities, and performance levels all these changes have made the life of design engineers and engineering managers somewhat difficult. Not only do they have to worry about gate count, capacitive loading, and race conditions, but also they have to keep track of all the data files representing the design, the current version of libraries and tools, and the different invocation sequences, configuration files, and so on, for each tool. Several market researchers report that these data, project, and process management tasks add more than 30% overhead to the engineering design process.

Further complicating this already complex task is the fact that CAD engineers have to worry about evolving existing design automation environments. Integrating a new simulator can take as long as six months, and being productive with one can take even longer.

In addition to all this, designs are getting bigger and more complex. Consequently, design automation environments and the data generated by design automation tools are becoming more difficult to manage.

■ ENTER THE FRAMEWORK

To tackle such problems, EDA Systems introduced the concept of an open CAD framework (Brouwers and Gray, 1987). The framework consists of the Electronic Design Management System (EDMS), which organizes, manages, tracks, and analyzes the engineering design process; and Developer Toolkits, a CAD tool development system that provides a consistent user interface and data storage and retrieval routines. This article focuses on the application of EDMS to a variety of design tools in a heterogeneous environment.

To use EDMS with design tools, the tools have to be "attached" to EDMS, through a process called encapsulation. Encapsulation consists of a layer of software surrounding the application tool that allows tracking of the design data within EDMS. Part of the encapsulation process includes customizing the user interface graphics—menus and icons—and determining how the design data will

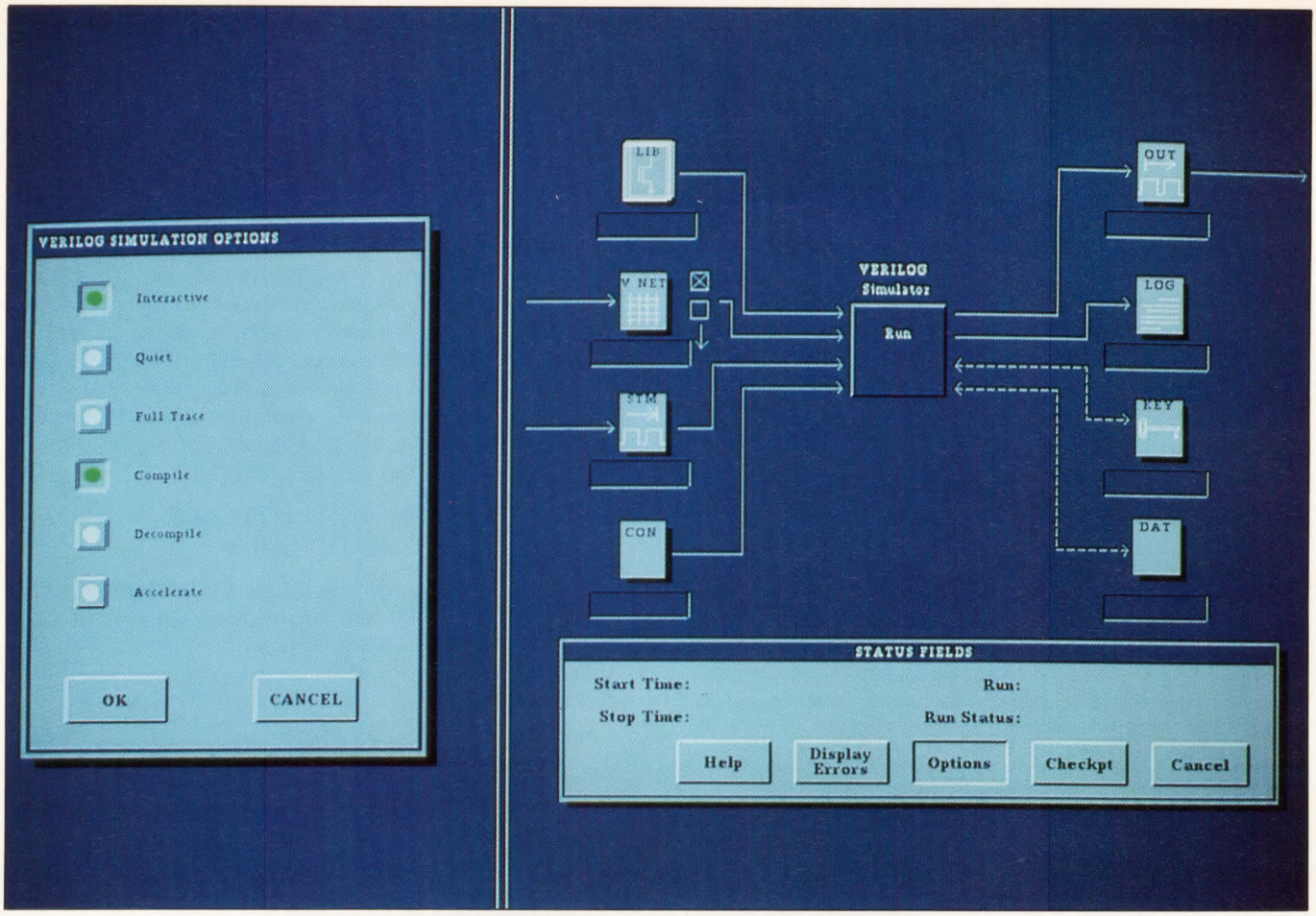


Figure 1. The tool template, in this case for Verilog, has "slots" into which design manager objects are added.

graphically appear to the end user.

■ Creating a Design Environment

As an example, let's consider an IC design project using several encapsulated DA tools. The tools were a schematic editor from Caeco, the Verilog logic simulator from Gateway Design Automation, and the HSPICE circuit simulator from Meta-Software. Also included were the appropriate netlisters for the tools mentioned above. The work was performed on Domain 3000 and 4000 series workstations from Apollo Computer.

Before a single icon was created or a single line of code was written to customize EDMS, several weeks were spent determining the project organization and how to represent the data files created by all the tools. Once the logical design was complete, the actual setup and coding time was simple.

We decided that all data for the design project would be organized under a project folder icon. Other icons were created to represent schematic, netlist, and simulation data. This set of icons gives the end user a graphical representation of the design data rather than requiring constant browsing of the file system to locate the desired files.

Through EDMS it is possible to track which database a set of data is derived from. For example, a simulation netlist may be derived from a schematic database. The "derivation" relationship can be used to relate a version of the netlist to the appropriate version of the schematic. At any given time, it is possible to determine if the netlister has been run on the current schematic design, eliminating mistakes such as running the netlister again for no reason.

We proceeded to implement the logical design of our environment in parallel: while one engineer worked on building the graphical user interface, another worked on encapsulating the individual application tools.

■ Icons and Templates

Each data file used or created by a tool has an icon and a corresponding object kind (for example, an HSPICE netlist) located in the Design Manager Server so that it can be managed within EDMS. The Design Manager Server stores high-level information about the design data such as location in the file system, the owner, and the last modified date, rather than storing the actual data. Also, each tool has a "tool template," a graphical interface that prompts the end user for all necessary inputs and options needed to run the tool.

The graphics for the icons and templates are created using the EDMS Graphic Interface Editor. Each icon was drawn to look like the object it represented. For example, an icon for a schematic design block actually looks like a schematic block normally seen on a schematic page. These object icons also are used on the tool template to depict the kind of data needed to run the tool.

The tool template for Verilog has "slots" into which design manager objects are added. Objects in this case are inputs and outputs needed to run the tool.

One graphical interface is common to all tools

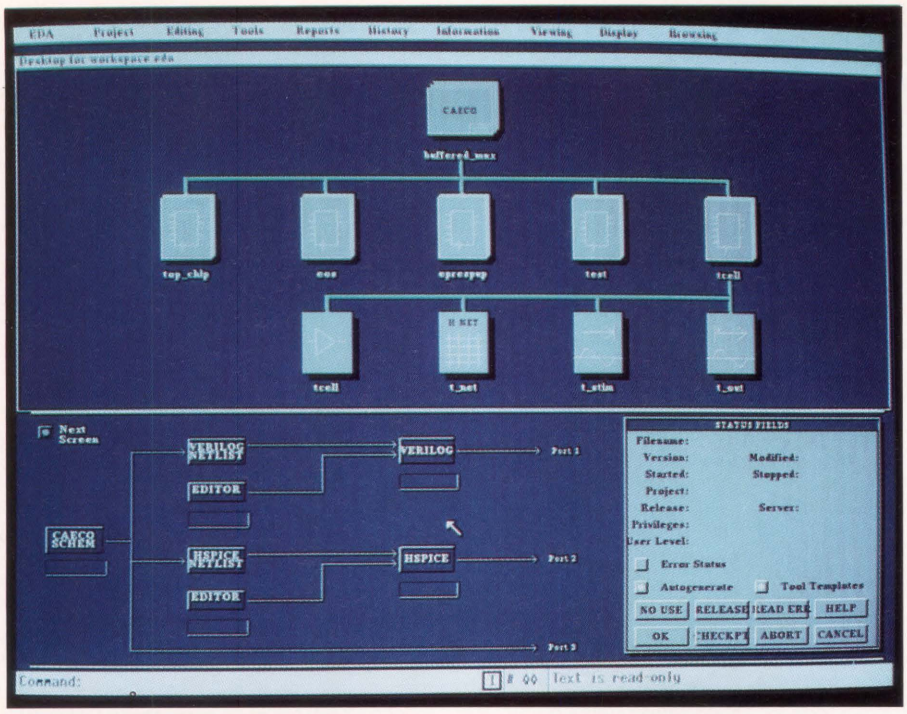


Figure 2. A higher-level template, called a process template, defines not only the different tools to be used, but also the logical flow of data between them.

```
Process_41

  stat = te_ts_check_tro(hspice_tro_id, dialog_id)
  if (stat != TS_ST_OK) return

  stat = te_ts_prepare_tro(hspice_tro_id, dialog_id)
  if (stat != TS_ST_OK) return

  /* Get the first module from the TRO. */
  mod_index = ts_first_argument(hspice_tro_id, "hspice_module")
  mod_info = ts_get_argument(hspice_tro_id, "hspice_module", mod_index)

  /* Get the derived netlist for this version of the module*/
  net_file = ee_hspice_get_netlist(hspice_tro_id, mod_info.vo_value)
  if (net_file == "") {
    "ERROR: Cannot find netlist for module."
    stat = te_ts_abort_tro(hspice_tro_id, dialog_id)
    stat = te_util_delete_args(hspice_tro_id, "hspice_net")
    return
  }
```

Figure 3. A portion of the E code that was written to encapsulate the HSPICE simulator is shown above. The entire program was only 134 lines.

Optional and mandatory data needed to run a tool are represented by dotted and solid lines, respectively, as shown in Figure 1. Checking is done automatically so that only the correct kind of data can be selected and added to a tool template. For example, the template will not let the user select a Caeco schematic icon and enter it into an HSPICE netlist slot. The Verilog tool template also has several "buttons." A button is a variation of an icon in which the state of the button controls the commands to be activated by the template. The buttons in the Verilog template control the commands to be issued to the tool, such as RUN, QUIT, and HELP.

Creating the graphics is only part of the story. As previously mentioned, the buttons have to be attached to appropriate actions. For example, the Verilog run button is a "sensitive" area that, when selected by the user, issues the command to invoke the Verilog simulator.

All the different tool templates are hooked up to create a higher-level template called a process template (Figure 2). The process template defines not only the different tools to be used by an engineer, but also the logical flow of data between them. In other words, it provides an overview of tool and design status.

For this project, the process template shows the Caeco schematic editor driving the HSPICE and Verilog simulators, with the appropriate netlisters in between. The benefits of a graphical interface are obvious. Users don't have to remember the complex invocation sequences, data and library requirements, options, and configuration files for each tool. This graphical interface alone can dramatically reduce the time to learn, or relearn, a tool.

However, the major benefits of using EDMS are obtained when you use its graphical capability in conjunction with the tracking and control mechanism of the Design Manager Server.

■ OBJECTS, POLICIES, AND RELATIONSHIPS

Setting up the Design Manager Server for the environment can take place in parallel with creating the graphical interface and includes defining objects, policies, and relationship to be used in a design project.

Objects in the Design Manager Server have specific data kinds associated with them (for example, HSPICE stimulus, Verilog netlist). Each kind of data is represented in the Design Manager by a different icon. Each object also stores information about the file it represents. For example, an object representing schematic data contains such information as the physical location of the file on disk, the kind of object it is (Caeco schematic), the owner, and the access rights of the file.

Policies are defined and associated with each object to ensure that a set of procedures is implemented within EDMS. For example, a policy can be set so that when a new version of an object is created, that version is marked as the current version.

It is important to note that EDMS does not copy the design data into its database, but instead creates objects representing the data much as index cards within a library card catalog contain information about the actual library books.

Relationships can be established between different data objects. A derivation relationship can associate a netlist with the appropriate schematic. In this project, the derivation relationship is used to check if an up-to-date netlist for a given schematic is available as input to a simulation run. If the up-to-date version is not available, the encapsulation code written for the simulator will automatically invoke the netlist program to create an updated netlist. All of this checking and netlist generation (if needed) is done invisibly.

Creating the objects, tool templates, and icons is the first step in encapsulating

a tool. To fully encapsulate a tool, a layer of software is written around the application using an interpretive C-based programming language called E.

■ Putting It All Together

An encapsulation program can be written using the following steps as a guideline:

- Define all data kinds for a given tool. (The HSPICE simulator, for instance, may need data kinds like the HSPICE netlist and the HSPICE stimulus file).
- Define each tool to the Design Manager Server by supplying the tool name and type (editor, transformer, or browser).
- Once the tool has been defined, describe all inputs and outputs to the tool.
- Write the E code to read the data from the tool template slots and prepare the data for tool activation. This code is referred to as the preprocessing section of the encapsulation program.
- Write the E code that deals with the management aspects of the data, such as tracking versions of the data and relating objects to each other. This code is the postprocessing section of the encapsulation program.

Functions that can also be included in the encapsulation are checking to see if the tool run was successful and reporting errors or any other pertinent information to the screen. In addition, EDMS provides the capability to save relevant information about a given tool run, such as what were the inputs, what were the outputs, who ran the tool, when the tool was started, and when it was completed.

The HSPICE encapsulation took three days to plan and one week to write. Figure 3 shows a portion of the E code. The entire program was only 134 lines (see the table).

For the HSPICE encapsulation, we had two input and two output files: netlist and error message files for input, and stimulus and waveform files for output. Each time the simulator was invoked, we chose to save all the relevant information about each tool run. This information is stored in a "tool run folder," which tracks the progression of the simulation runs and therefore can be used for locating or analyzing any particular run at a later date.

■ An Engineering Activity Analyzer

EDMS has a history mechanism that keeps track of all the activities that have taken place under its control. With the aid of this mechanism, users can produce bar charts analyzing engineering activities, such as which engineer ran a given tool, how many times the tool was run, and the duration of the tool run.

Time and Code Needed for Setting Up Various Tools in the EDMS Environment

	Lines of E Code	Time Needed
CAECO Encapsulation	479	12 Days
Verilog Encapsulation	159	7 Days
HSPICE Encapsulation	134	3 Days
Netlister Encapsulation	154	2 Days
Tool Templates	Automatic Builder	1 Day
Menus	175	1 Day

■ Benefits to Users

Tools encapsulated with EDMS can reduce design and analysis times of projects in several ways.

Since EDMS hides the complexities of computers and tools, an engineer can use tools without having to remember complex tool invocation sequences, tool options, and operating system commands; the interface to all tools becomes graphical, simple, and consistent.

EDMS makes data dependencies and tool sequencing invisible to the end user. For instance, an engineer wants to simulate a particular schematic block, and the simulation depends on the availability of an up-to-date netlist for that block. Using EDMS, the engineer only has to specify the block he she wants to simulate and EDMS will check for the most current version of the netlist for that block; if none exists, it will automatically generate a netlist.

Engineering project reports can be very useful for the engineer as well as the project manager. Tool usage information can be used to identify and solve design process bottlenecks, as well as to justify for hardware procurement.

In addition, EDMS allows the customization of the design environment, with the current process methodologies, specific to a project. Adding new design tools to the design environment becomes easier and quicker with EDMS whether or not you have access to source code for your tools. ■

About the Authors

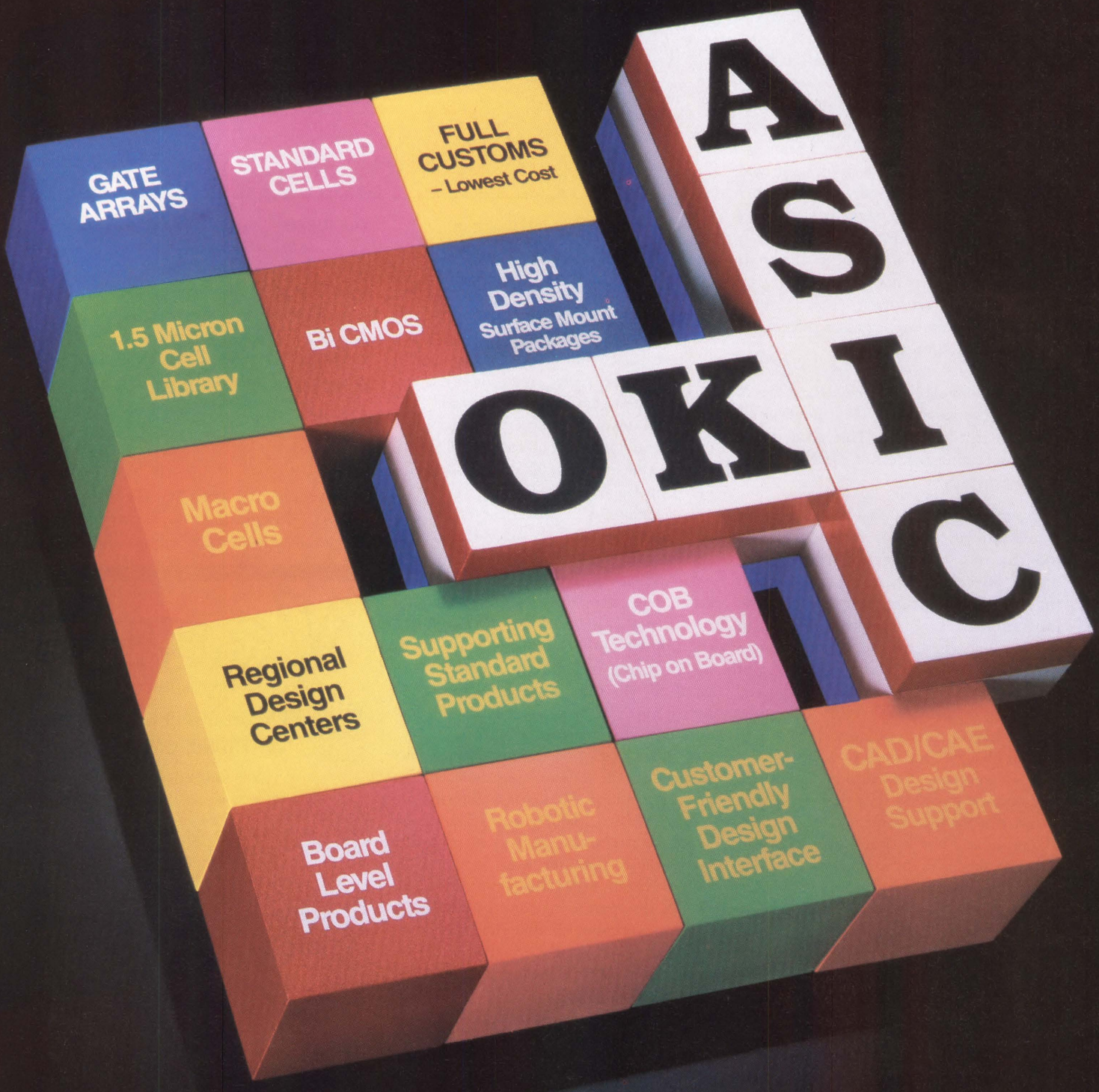
Ted Stout *is currently the product manager for EDMS at EDA. Before that, he spent four years at Daisy Systems as product manager of PCB layout tools and two years at Prime Computer as product manager for design management products. He holds an EE degree from California State University at Northridge.*

Kyle Goldman *is currently the manager of framework applications at EDA. Before joining EDA, she headed up the CAE/CAD workstation group at National Semiconductor for six years. For two years before that, she worked in software engineering for automated cartography for the U.S. government. Goldman holds a BS in mathematics from Arizona State University.*

Reference

Brouwers, J., and M. Gray. June 1987. "Integrating the Electronic Design Process," *VLSI Systems Design.*

Easy ASIC


GATE ARRAYS
STANDARD CELLS
FULL CUSTOMS – Lowest Cost
1.5 Micron Cell Library
Bi CMOS
High Density Surface Mount Packages
Macro Cells
Regional Design Centers
Supporting Standard Products
COB Technology (Chip on Board)
Board Level Products
Robotic Manu-facturing
Customer-Friendly Design Interface
CAD/CAE Design Support
A
S
I
C
OKI

Only from OKI:

- Most **complete** ASIC building blocks.
- Most **versatile** design/package options.
- Most **experienced** ASIC technology.

Nobody but nobody puts ASIC technology together like OKI Semiconductor can.

Ease into ASIC with OKI as your close working partner—and you instantly support your VLSI application with the most comprehensive ASIC capabilities on the world market today. Bar none.

From gate array, standard cell and full custom chips to standard components to integration to advanced board level products, OKI alone puts you on the leading edge of ASIC technology and its complete implementation.

OKI: the totally logical choice.

Opt for OKI ASIC, and you open up your options across the board. Only OKI now offers the system designer the unique security and entry ease that only a proven track record in CMOS ASIC problem-solving can provide. This history of performance built up since 1977 has produced the widest range of solid building blocks yet: advanced ASIC products and packaging including surface-mount, backed up with the most flexible cell libraries, CAD/CAE design tools and development aids.

As your working partner, OKI ASIC expertise is available to you at any stage of the development process. We'll help you define system requirements, determine the most cost-effective product solutions and supply complete design software—accessible at your own workstation or through our regional design centers. And then we take it from there: with high volume fabrication, assembly and testing completed in one of the world's most highly robotized manufacturing facilities.

Compare Total ASIC Capabilities	OKI ASIC	Source "B"	Source "C"
Gate Arrays to 10K Gates	●	●	
Standard Cells to 30K Gates	●	●	
Full Customs – Lowest Cost	●		
1.5 Micron Cell Library	●	●	●
Macro Cells	●	●	
Bi CMOS	●		
High Density Surface Mount Packages	●		
Board Level Products	●		
Supporting Standard Products	●		
COB Technology (Chip on Board)	●		
CAD/CAE Design Support	●	●	●
Customer-Friendly Design Interface	●	●	●
Regional Design Centers	●	●	●
Robotic Manufacturing	●		

ASIC Solutions from **OKI:**
You can't beat the logic!

Check out OKI ASIC data:

VLSI 688

Please rush complete technical data/specs on OKI capabilities in:

( ) Gate Arrays
( ) Standard Cells
( ) Full Customs

( ) Please call: we have immediate requirements:

Name/Title _____

Company _____

Attach coupon to business card or letterhead and return to: ASIC Customer Service, **OKI Semiconductor,** 650 North Mary Avenue, Sunnyvale, CA 94086. Tel: (408) 720-1900.

CIRCLE NUMBER 11

ENTRY Systems

FOR ANALOG DESIGNS

VLSI SYSTEMS DESIGN STAFF

Although analog circuits were popular long before digital came into its own, CAE entry systems for analog designs have continued to lag those of digital systems. In addition, the early design packages for analog circuitry usually consisted of very rudimentary tools that lacked the speed, accuracy, and convenience of their digital equivalents. Fortunately, the past few years has seen the introduction of more sophisticated analog design tools that are steadily approaching the performance of most digital design systems.

ANALOG DESIGN TOOLS ARE COMING OF AGE

An entry system is defined here as a system that not only captures the connectivity requirements of the circuit schematic, but also captures the basic design information and provides the necessary hooks to analysis tools as well. Many digital schematic capture packages can also "capture" analog circuits, but the systems are usually limited to performing only the most basic analyses. Of course, some of today's newer CAE systems can handle analog, digital, and mixed analog-digital designs. However, though an accurate analysis of a digital circuit can be performed with relatively crude models, that is not true for analog circuits.

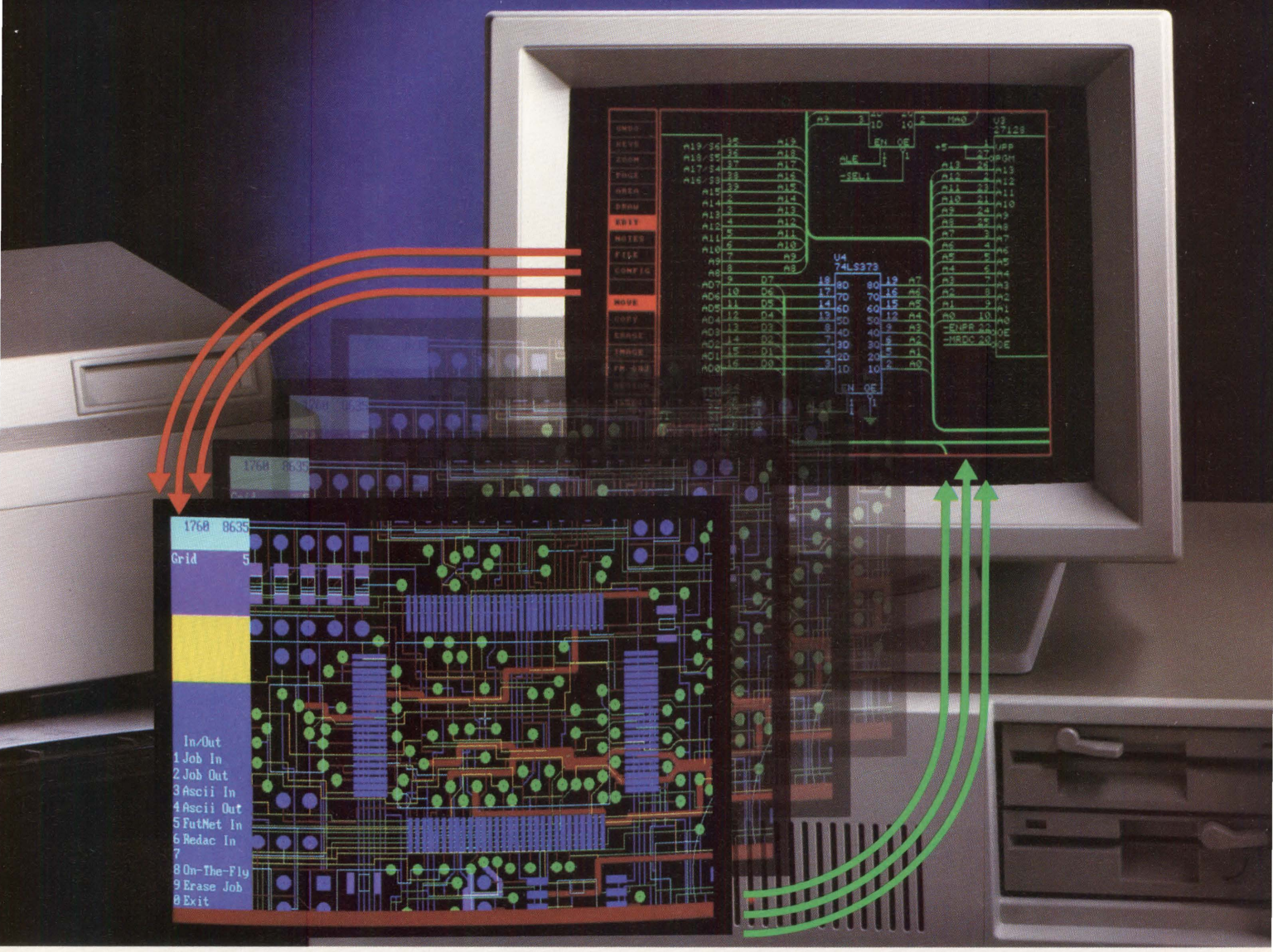
For example, a simple AND gate may consist of as many as 12 transistors and still be represented with reasonable accuracy by a relatively primitive model. On the other hand, an op amp that is to be used in a critical high-gain feedback application may also consist of only 12 transistors, but just to analyze the amplifier's performance may require calculating tens of thousands of operating points.

As can be seen in this month's directory, today's analog CAE entry systems provide a lot more than just a method for capturing schematics. They accept inputs from a mouse, keyboard, and netlists. They feature pop-up menus and can make almost every imaginable analysis on an analog circuit, such as frequency- and time-domain analyses, together with dc, ac, and noise characterization. In addition, they can perform Monte Carlo, sensitivity, worst-case, and thermal analyses.

Since a design system is no better than its libraries, most of the packages include standard and optional libraries that typically have 1,000 or more discrete components, such as low-level and power bipolar, JFET, and CMOS transistors and rectifiers, zener diodes, SCRs, UJTs, and an assortment of resistors, capacitors, and inductive components. Macros like op amps and comparators also are included.

The entry systems run on a variety of personal computers, workstations, and minicomputers. They have links to SPICE and other simulation programs. In addition, whereas some provide analysis only at the basic transistor level, others can simulate complete systems, including electromechanical components.

The Integrated CAE Solution From Just Under $500.



Nothing Captures Today's Solutions Like Schema.

OMATION, originators of the most popular design entry software SCHEMA II, presents an integrated solution to printed circuit board design.

Now you can enter a schematic using SCHEMA II, and then produce a finished printed circuit card with SCHEMA PCB and SCHEMA ROUTE. From start to finish *in one easy step.*

The integration, complete with automatic backward and forward annotation, makes the process simple and error free. All SCHEMA products operate on IBM personal computers or compatibles, yet they offer all the features of CAE workstations costing $50,000 *for a fraction of the cost.*

CALL TODAY and ask for our free demo disk.

CALL: 1-800-553-9119

As always, each member of the SCHEMA family of products comes with a *30-Day Money Back Guarantee.*

Capture your solutions by calling us Toll Free: **1-800-553-9119** (in Texas, 214-231-5167). Ask about our SCHEMA Family Demo Evaluation Kit.

OMATION

1210 E. Campbell Rd., Suite 100
Richardson, Texas 75081

CIRCLE NUMBER 12

DIRECTORY OF ANALOG DESIGN ENTRY SYSTEMS

VENDOR AND CONTACT	PRODUCT AND PRICE	HOST	SPECIAL-PURPOSE HARDWARE	INPUT METHODS
ANALOG DESIGN TOOLS INC. *1080 East Arques Ave.* *Sunnyvale, Calif. 94086* James Carro Senior Vice President, Marketing and Sales (408) 737-7300	**ANALOG WORKBENCH** Software only and turnkey $8k–$14.5k, software	Sun-3 and -4 (Unix 4.2 BSD); Apollo 3000, 4000, 10000 (Aegis or Domain IX); HP 9000/320 and 350 (HP/UX); IBM PC AT (Unix System V); VAXstation (VMS); Compaq (Unix System V)	Apollo, HP, and Sun: floating-point coprocessor optional. IBM PC AT: EGA graphics card required Compaq: coprocessor required	Menu- and mouse-driven
ANALOGY INC. *P.O. Box 1669* *Beaverton, Ore. 97075* Bob Ferguson (503) 626-9700	**SABER** analog simulator Software only $15k–$35k on most workstations	Sun-3 and -4; Apollo 3000, 4000; Digital Equipment VAX 700 and 8000, MicroVAX and VAXstation; Alliant FX	None	Graphical front end and graphics postprocessor use a mouse, pop-ups, and icons
APTOS SYSTEMS CORP. *5275 Scotts Valley Dr.* *Scotts Valley, Calif.* John Roth President (408) 438-2199	**CRITERION I** Software only $495 **RGRAPH** Software only $6.7k	Criterion I: PC AT, 386 RGRAPH: PC AT, 386	Criterion I: EGA, 640 × 350 RGRAPH: 19″ monitor, 1024 × 768; Artist I graphics card optional	Mouse, tablet, keyboard and user-definable menus, back annotation from PCB layout
CADNETIX CORP. *5775 Flatiron Pkwy.* *Boulder, Colo. 80301* Tim Gerchar Dan Damon Product Marketing Managers (303) 444-8075	**CADNETIX ANALOG DESIGN ENVIRONMENT** (based on Saber from Analogy) Software only and turnkey $5k–$30k, software only; $45k–$80k, turnkey	Software-only CDX-8130 and turnkey CDX-9600, CDX-9630, CDX-9640: Sun-3 or -4 with color monitor (SunOS) CDX-3000 and CDX-3200: IBM PC ATs or 100% compatibles (DOS)	For schematic entry only, no additional hardware is required. For the CDX-9630/9640/8130, a Sun-3 or -4 with color monitor is required; for the CDX-3200, a math coprocessor is required for the PC AT	Mouse with six soft keys; Cadnetix standard input keys (10); easy-to-use forms; keyboard; automatic forward and back annotation from PCB layout
CASE TECHNOLOGY *2141 Landings Dr.* *Mountain View, Calif. 94043* Melanie King Marketing Communications Manager (415) 962-1440	**VANGUARD CAE DESIGN SYSTEM** Software only $5.75k–35.6k	IBM PC, PC XT, PC AT, and compatibles; Compaq 386; PS/2 models 50, 60, and 80; MicroVAX 2000, VAXstation 2000, MicroVAX GPX/II; Sun-3/60, 260, 50, 110	IBM PC AT: VGA or EGA graphics card; 2-MB memory expansion for Hercules and Microfield graphics	Mouse with pop-up menus, user-definable keyboard mapping, on-line help
DAISY SYSTEMS CORP. *700 E. Middlefield Rd.* *P.O. Box 7006* *Mountain View, Calif. 94039* Cynthia Cauley Product Marketing Manager, Analog Design Automation (415) 960-7196	**VIRTUAL LAB** analog design environment Software only or turnkey $6k–$12k **DSPICE** analog simulation engine Software only or turnkey $5k–$41k	Virtual Lab: Sun 386i, Personal Logician 386, Logician 386 DSPICE: Sun 386i, Personal Logician 386, Logician 386. Accelerated versions may be run remotely across the network using the Sun-4/260S, MicroVAX II, or VAX	None	Direct input from either schematics or netlists (SPICE-compatible netlists are also accepted); schematics may be created using Daisy's advanced capture editor (ACE). Users have the option of using icons, pop-up menus, forms, or command-line entry to execute each step of the analog design process

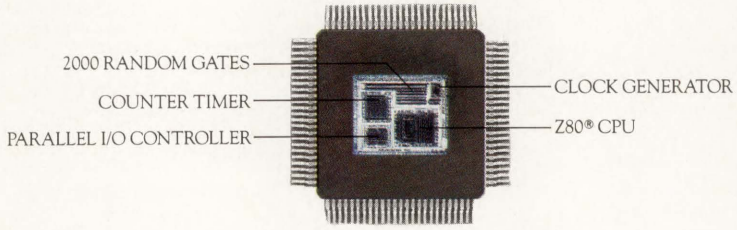
OUTPUT FORMATS	PRINTERS AND PLOTTERS SUPPORTED	LIBRARIES	OTHER FEATURES	SYSTEM CAPACITY AND MEMORY REQUIREMENTS
ASCII (standard); hi-res graphics on simulated test instruments (standard); Mentor interface (optional); HP Design Center (optional); FutureNet interface (optional)	HP pen plotters, A through E size; Houston Instruments pen plotter; various dot-matrix and laser printers	General device library, 1400+ components (optional); power device library (optional); standard device library, PC only (optional); basic device Library (optional); sample device library (included)	Parametric plotting; stress analysis; Monte Carlo and sensitivity/worst case analysis; special module for power design with nonlinear magnetics; mixed-mode handled at transistor level	Contact factory
Not specified	HP pen plotters and compatibles; also, laser printers through host screen dump	Diodes, 93; Power MOS, 109; optocouplers, 2; BJTs, 217; SCRs, 5; SPICE templates, 33; JFETs, 23; op amps, 63; Saber templates, 28; FETs, 9; comparators, 3	Saber/Cadat system- through circuit-level mixed-mode simulation; spectral analysis—noise, distortion, Fourier; time-domain modeling—ideal delay, schedule events, state-based models	Not specified
ASCII; Tegas; SPICE; Logis; Ilogs/Silos; GDS II; PSPICE	Not specified	Analog library consists of generic parts for use as template to build parts; total number 1200+; TTL, ECL, SMD, microprocessor libraries	Automatic power and ground; supports analog, digital, and mixed signals; 10 levels of hierarchy; fully integrated with Aptos PCB layout	Not specified
Saber netlist, ASCII (standard); EDIF output (standard); Postscript devices supported (standard); direct interfaces with Cadnetix CAD (standard)	Not specified	Generic component library: op amp, resistor, capacitor, inductor, transistor, MOSFET, 4-bit A/D converter, function generator, diode, transformer, sample-and-hold amplifier, zener diode (standard). Specific component library: 500 semiconductor devices (optional)	Standard analysis modes: dc, transient, ac, noise, distortion. Optional analysis: Monte Carlo. Standard: user-defined analog behavioral models; mixed technologies—electrical, mechanical, simple digital, chemical, etc.	Design size is limited only by disk size; several hundred transistors and macro models could be analyzed in a reasonable time period; PC AT-based systems may require a DOS extender to handle large analog designs
Netlists: Silos, Hilo, Lasar, SPICE, Cadat, EDIF ASCII format *PCB interfaces:* Cadnetix, Racal-Redac, Scicards, CBDS, CADDS 4X	Options: Gerber and HP printer-plotters	Standard: standard libraries for TTL, ECL, CMOS, PLD, ASIC and microprocessor families—5000+; 48 diodes, 45 bipolar transistors, 36 power MOSFETs, 26 saturable transformer models, 4 op amps, 28 comparators, 8 quality crystals	Electrical rule checker, design rule checker, analog oscilloscope, document editor, symbol editor, logic analyzer, timing verifier; handles digital circuits; link to Interleaf/Ventura and ASK's MANMAN	PC version of Vanguard CAE design system handles 30,000 gates, can simulate 20,000 gates; DEC and Sun versions handle 4 million gates
Virtual Lab: netlist to drive DSPICE analog simulation. Simulation results are stored as files for waveform display. Postprocessing results may be saved and displayed as graphical waveforms, tables of values, or in standard SPICE output file format	Not specified	1400+ parts—bipolar transistors, JFETs, MOSFETs, discrete power transistors, zener diodes, op amps, comparators, voltage regulators, and other ICs. Special-purpose libraries also available	Standard features of Virtual Lab and DSPICE: dc analysis, time- and frequency-domain analysis; component sweeper; parametric plotter; sensitivity analysis; unlimited waveform display (no limits to waveforms/window, number of displays); real-time marching waveforms during simulation; automatic spike detection; on-line calculator and arithmetic expressions package; interactive execution and acceleration on remote network nodes (Sun-4/260S, MicroVAX II, VAX); hierachical design methodology; block and critical-path analysis; FFT analyzer; gallium arsenide models	Design size is not limited by capacity

DIRECTORY OF ANALOG DESIGN ENTRY SYSTEMS (continued)

VENDOR AND CONTACT	PRODUCT AND PRICE	HOST	SPECIAL-PURPOSE HARDWARE	INPUT METHODS
HEWLETT-PACKARD CO. *Customer Information Center* *Inquiry Fulfillment Dept.* *19310 Pruneridge Ave.* *Cupertino, Calif. 95014* Local sales office (listed in white pages)	**HP DESIGN INTERFACE FOR ANALOG WORKBENCH** (bidirectional schematic link between the HP Design Capture System and the Analog Workbench from Analog Design Tools) Software only $2,040	HP 9000/319, 320, 330, 350 (HP-UX)	None	Mouse-driven with pop-up menus and the keyboard for assigning file names. Bidirectional layout and documentation capability are provided through the HP Design Capture System
INTERGRAPH CORP. *3160 Crow Canyon Rd.* *San Ramon, Calif. 94586* Wayne Sherman Director of Marketing (415) 866-0520	**CIRCUIT TOOLS** Turnkey CSPICE, $10k; Benchtop Instruments Interface, $5k; Analytical Interface, $5k; statistical module, $5k; comprehensive library $30k (cost of individual modules varies); system-level simulation package (requires CSPICE and Analytical Interface), $5k	InterPro workstation (Unix System V)	None	CSPICE performance simulator: Berkeley netlist. AAT simulation interface: function generator, power supply, DVM, oscilloscope, spectrum analyzer, and network analyzer; pop-up menus, and displays
MENTOR GRAPHICS CORP. *8500 S.W. Creekside Place* *Beaverton, Ore. 97005* Gary O'Dell Product Manager (503) 626-7000	**MSPICE** circuit simulator **MSPICE PLUS** analog simulator **MSIMON** MOS circuit simulator **ANALOG LIBRARY** Software options only MSPICE, $9.9k; MSPICE PLUS, $19.8k; MSIMON, $23.8k; Analog Library, first copy, $20k, subsequent copies, $2k	Mentor Graphics' Design Station and Idea station (Apollo)	None	Schematic input; interactive graphical output with menus, forms, and icons to control simulator functions. Works with PCB layout, thermal analysis, and schematic capture through a common database. Back annotation, parameter variation, and simulation control managed by a flexible macro language
MIETEC *Raketstraat 62* *B-1130* *Brussels, Belgium* Anne Windals Communications Manager	**MADE**	VAX 700 (VMS), VAX 8000, MicroVAX 2000	None	Mouse with pop-up menus and keyboard
OMATION INC. *1210 E. Campbell Rd.* *Suite 100* *Richardson, Tex. 75081* Kimberley Ammons Marketing Assistant (214) 231-5167	**SCHEMA II** **Schema-PCB** (CAD Software) Software only Schema II, $495; Schema-PCB, $975; Schema-Route, $850; Schema-Plot, $300; Schema-Drill, $300; Schema-Large, $250	IBM PC XT, PC AT, PS/2, and compatibles; Compaq; AT&T; Zenith	Schema II: parallel port Schema-PCB: parallel port, EGA with 16 colors, mouse	Schema II: keyboard and mouse Scema-PCB: mouse and limited keyboard support; forward and back annotation
PHASE THREE LOGIC *1600 N.W. 167th Place* *Suite 335* *Beaverton, Ore. 97006* Gary S. Kahl North American Sales Manager (503) 640-2422	**CAPFAST** Software only CF3000, $1,295; CF3550, $2,950; CF/IG-SPICE, $1,495; CF6000, $4,950; HSPICE interface option (Thompson Automation), $500; PSPICE interface, $500	PC AT, 386, PS/2, and compatibles (DOS); Sun-3	CF3000: EGA card and monitor CF3550 and CF/IG-SPICE: Metheus 1104 Ultra Graphics adapter and 48KH2 monitor	CF3000, CF3550, CF/IG-SPICE, CF6000: Mouse with static and pop-up menus, keyboard input, keyboard macros, long macros, startup file and customization capabilities HSPICE and PSPICE interfaces: CapFast schematic files
PRIME COMPUTER INC. *Computervision Division* *100 Crosby Dr.* *Bedford, Mass. 01730* Anthony Dolph Media Relations (617) 275-1800 x6228	**CIRCUIT DESIGN** (Saber analog circuit simulation and analysis from Analogy) Turnkey Professional Circuit Designer (Circuit Design software with Sun-3/60), from $46.9k	Sun-3 and -4 (Unix 4.2 BSD)	None	Icon-driven user interface with pop-up and pull-down menus, mouse, keyboard, and macro command language

OUTPUT FORMATS	PRINTERS AND PLOTTERS SUPPORTED	LIBRARIES	OTHER FEATURES	SYSTEM CAPACITY AND MEMORY REQUIREMENTS
Through the open Design Capture Database, users can customize netlists and system interfaces, as well as printer and plotter support as a standard part of the system. Standard: HP Printed Circuit Design System bidirectional interface. Optional: EDIF 2.0.0 bidirectional schematic view; Scicards bidirectonal interface; Calay bidirectional interface; Racal-Redac RINF bidirectional interface; Computervision bidirectional interface	Support includes Versatek and HP 7427/28, 7440/70, 7475A/B, 7570C/D, 7550A/B, 7580A-D, 7585/6A-D, and 7595/6A-E machines	R, L, C parts library and semiconductor library of approximately 1,400 parts (standard)	Standard features include full attribute-tagging capability and mixed digital-analog capture capability with on-line rule checking	Limited in capacity wholly by the capacity of the Analog Workbench itself
AAT simulation interface: dc, time-domain, and frequency-domain color plots on screen (i.e., dc and dc transfer; frequency response vs. gain and phase shift; group delay; voltage, current, and power vs. time); output direct to Intergraph's PCB and hybrid layout products	Not specified	Comprehensive library: diodes (amplifier macro models); BJTs (amplifier micromodels); JFETs (power devices); MOSFETs (specialty ICs, macro and micro)—1,700+ total	CSPICE performance simulator and AAT simulation interface: dc and time- and frequency-domain analysis; filter design package (optional), pole-zero analysis package (optional), arbitrary waveform calculator (optional). System-level simulation package: system-level simulation (e.g., electromechanical)	Simulator especially suited to very large problems (15× SPICE)
Common database suports EDIF formats, integration with IC and PCB layout, thermal analysis, and user-written postprocessing	Common database suports a variety of plotter and printer options	1,200 active devices (optional)	Rule checking during design capture; variety of postprocessing functions, such as Bode plots. Mixed circuits can reside on the same schematic; the simulator runs the analog sections without the need to separate analog and digital sections	MSIMON: 20,000+ MOS transistors
SDL (Silvar-Lisco) and GDS II netlists	Versatec, LNO3 (digital)	2.4-μm CMOS: digital, 730; analog, 70. SB1MOS: digital, 60; analog, 40	Mixed-mode simulator (digital-analog), switched-capacitor filter compiler, accelerated circuit simulator (40 times faster SPICE)	Standard cells: 5,000 gates (20,000 transisors) mixed digital-analog. No special memory requirements
FutureNet (netlist and pin list), EDIF, Racal-Redac, Datacon wire wrap, Visionics, Calay, Cadnetix, P-CAD PDIF, SPICE, Computervision, Telesis CBDS	Not specified	TTL, CMOS, discretes, memory, microprocessors, and IEEE. PCB has SMD decals	Design rule and routing checking for Schema. Space checking, electrical connectivity design rule checking, and via reduction for PCB	512 KB minimum memory; 640 KB highly recommended
CF3000, CF3550, CF6000: PADS-PCB, Scicards, Racal-Redac, Hilo-3, SPICE interfaces	More than 30 manufacturers	Symed symbol editor; all SPICE 2G6 (or IG-SPICE, HSPICE, or PSPICE) symbols and values easily modified in the schematic editor (or symbol editor) to represent any transistor, resistor, etc. Includes symbols for different analysis (ac, dc, transient), sources, meters	Electrical rule checking (standard), simulation grapher for interactive viewing of analog or digital waveforms (standard)	CF3000, CF3550, CF/IG-SPICE: SPICE and IG-SPICE netlister has limited capacity. With optional DOS extender, virtually unlimited capacity—2 MB minimum extended memory, up to 15 MB CF6000: virtual unlimited capacity
EDIF and ASCII netlists; full programmable netlist interface for all CAD system interfaces	Full range	Schematic design library (standard), 3000+ parts, adherence to IEEE standards; Saber analog simulation (optional), approx. 500 models and standard templates	Electrical rule checking and dynamic timing verifier (standard)	System limited only by available disk capacity (Sun-4 provides for over 1 GB of disk capacity and as much as 128 MB of main memory)

Leader Of



PACKAGE: 100 PFP, 32 MIL CENTERS*

®Z80 is a trademark of Zilog Inc.

Now 1.5 μ Hard Megacell ASICs.

Nobody packs as much function into a megacell custom circuit as Toshiba because we've got the broadest library of 1.5 μ CMOS megacells. We're Leader of the Packed.

We've been successfully producing complex megacell customs for over four years now. And we've shipped millions of them. So while others are just beginning their megacell efforts, we stand alone in experience and production.

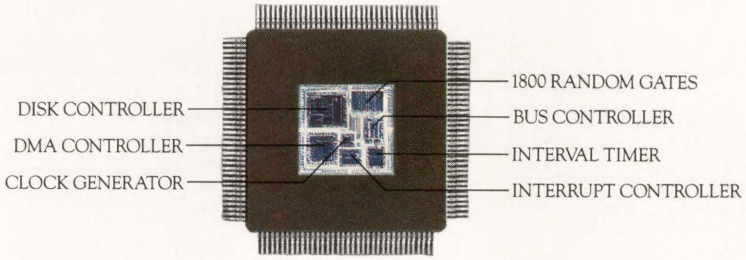
Our megacells are exact mask duplicates of our standard LSI discretes. Each megacell is tested to our standard data sheet specifications. New layout is only required for the random logic section, and total circuit testability is always assured. We offer your application the highest complexity at the lowest risk.

Z80 FAMILY MEGACELLS

Z80. PIO. SIO. CTC. DMA. We have them all. And you can mix and match components with random logic to create integrated solutions for your most complex applications. Like the solutions we've already provided for modems, printers, hand terminals and industrial controls. To name just a few.

AREA SALES OFFICES: CENTRAL AREA, Toshiba America, Inc., (312) 945-1500; **EASTERN AREA,** Toshiba America, Inc., (617) 272-4352; **NORTHWESTERN AREA,** Toshiba America, Inc., (408) 737-9844; **SOUTHWESTERN REGION,** Toshiba America, Inc., (714) 259-0368; **SOUTH CENTRAL REGION,** Toshiba America, Inc., (214) 480-0470; **SOUTHEASTERN REGION,** Toshiba America, Inc., (404) 368-0203; **MAJOR ACCOUNT OFFICE,** Fishkill, New York, Toshiba America, Inc., (914) 896-6500; **BOCA RATON, FLORIDA,** Toshiba America, Inc., (305) 394-3004. **REPRESENTATIVE OFFICES: ALABAMA,** Montgomery Marketing, Inc., (205) 830-0498; **ARIZONA,** Summit Sales, (602) 998-4850; **ARKANSAS,** MIL-REP Associates, (512) 346-6331; **CALIFORNIA** (Northern) Elrepco, Inc., (415) 962-0660; **CALIFORNIA** (L.A. & Orange County) Bager Electronics, Inc., (818) 712-0011, (714) 957-3367, (San Diego County) Eagle Technical Sales, (619) 743-6550; **COLORADO,** Straube Associates Mountain States, Inc., (303) 426-0890; **CONNECTICUT,** Datcom, Inc., (203) 288-7005; **FLORIDA,** Sales Engineering Concepts, (813) 823-6221, (305) 426-4601, (305) 682-4800; **GEORGIA,** Montgomery Marketing, Inc., (404) 447-6124; **IDAHO,** Components West, (509) 922-2412; **ILLINOIS,** Carlson Electronic Sales, (312) 956-8240, R.W. Kunz, (314) 966-4977; **INDIANA,** Leslie M. DeVoe Company, (317) 842-3245; **KANSAS,** D.L.E. Electronics, (316) 744-1229; **KENTUCKY,** Leslie M. DeVoe Company, (317) 842-3245; **LOUISIANA,** MIL-REP Associates, (713) 444-2557; **MAINE,** Datcom, Inc., (617) 891-4600; **MASSACHUSETTS,** Datcom, Inc., (617) 891-4600; **MICHIGAN,** Action Components Sales, (313) 349-3940; **MINNESOTA,** Electric Component Sales, (612) 933-2594; **MISSISSIPPI,**

The Packed.



PACKAGE: 144 PFP, 25 MIL CENTERS*

*Die shown larger than actual size.

82Cxx PERIPHERAL MEGACELLS

We can supply all the necessary peripherals you need for PC and compatible environments. To communicate. To control disks. To access memory. To drive buses. To manage interrupts. They're all in our library.

SPECIAL PURPOSE MEGACELLS

CRT controllers. LCD drivers. UARTs and analog circuits. And RAM and ROM. Our special purpose megacells offer these kinds of solutions for your special needs. Solutions not available in other ASIC offerings.

GET A PACKAGE OF INFORMATION

Stop fighting the battle of packing more onto a PC board. Integrate the entire board into one Toshiba Megacell ASIC. And take over as leader of your pack. For complete details contact Toshiba today. Call your Custom IC Product Manager at (714) 832-6300 or a Toshiba Regional Sales Office: NORTHWESTERN: San Jose, CA (408) 244-4070. SOUTHWESTERN: Newport Beach, CA (714) 259-0368. NORTH CENTRAL: Chicago, IL (312) 945-1500. SOUTH CENTRAL: Dallas, TX (214) 480-0470. NORTHEASTERN: Burlington, MA (617) 272-4352. SOUTHEASTERN: Atlanta, GA (404) 368-0203.

Toshiba. Leader Of The Packed.

TOSHIBA AMERICA, INC.

Montgomery Marketing, Inc., (205) 830-0498; **MISSOURI,** D.L.E. Electronics, (316) 744-1229, R.W. Kunz, (314) 966-4977; **MONTANA,** Components West, (206) 885-5880; **NEVADA,** Elrepco, Inc., (415) 962-0660; **NEBRASKA,** D.L.E. Electronics, (316) 744-1229; **NEW ENGLAND,** Datcom, Inc., (617) 891-4600; **NEW HAMPSHIRE,** Datcom, Inc., (617) 891-4600; **NEW JERSEY,** Nexus-Technology, (201) 947-0151; **NEW MEXICO,** Summit Sales, (602) 998-4850; **NEW YORK,** Nexus Technology, (201) 947-0151; Pi-tronics, (315) 455-7346; **NORTH CAROLINA/SOUTH CAROLINA,** Montgomery Marketing, Inc., (919) 467-6319; **NORTH DAKOTA/SOUTH DAKOTA,** Electric Component Sales, (612) 933-2594; **OHIO,** Steffen & Associates, (216) 461-8333; (419) 884-2313, (513) 293-3145; **OKLAHOMA,** MIL-REP Associates, (214) 644-6731; **OREGON,** Components West, (503) 684-1671; **PENNSYLVANIA,** Nexus Technology, (215) 675-9600, Steffen & Associates, (412) 276-7366; **RHODE ISLAND,** Datcom, Inc., (617) 891-4600; **TENNESSEE,** Montgomery Marketing, Inc., (205) 830-0498; **TEXAS,** MIL-REP Associates, (512) 346-6331, (713) 444-2557, (214) 644-6731; **UTAH,** Straube Associates Mountain States, Inc., (801) 263-2640; **VERMONT,** Datcom, Inc., (617) 891-4600; **WEST VIRGINIA,** Steffen & Associates, (419) 884-2313; **WASHINGTON,** Components West, (206) 885-5880, (509) 922-2412; **WISCONSIN,** Carlson Electronics, (414) 476-2790, Electric Component Sales, (612) 933-2594; **WYOMING,** Straube Associates Mountain States, Inc., (303) 426-0890; **CANADA, BRITISH COLUMBIA,** Components West, (206) 885-5880; **ONTARIO,** Electro Source, Inc., (416) 675-4490, (613) 726-1452; **QUEBEC,** Electro Source, Inc., (514) 630-7421.

CIRCLE NUMBER 13

DIRECTORY OF ANALOG DESIGN ENTRY SYSTEMS (continued)

VENDOR AND CONTACT	PRODUCT AND PRICE	HOST	SPECIAL-PURPOSE HARDWARE	INPUT METHODS
RACAL-REDAC INC. *238 Littleton Rd.* *Westford, Mass. 01886* David DeMaria Director, Technical Marketing (617) 692-4900	**VISULA** (mostly proprietary; ESQL from Informix, Cadat 6 from HHB Systems, Saber from Analogy) Software only and turnkey From $45k	Apollo 3000, 4000, DN580T; DEC VAXstation 2000, VAXstation II/GPX, 3200; Sun-3/60, 100, 200	MIPS application accelerators; CATS hardware modeler available	Keyboard, mouse, back annotation
SCIENTIFIC CALCULATIONS INC. *7796 Victor-Mendon Rd.* *Fishers, N.Y. 14453* Dennis George Product Manager (716) 924-9303	**SCIDESIGN** Software only $3k, fully supported copy; $1k, right to copy	IBM PC XT, PC AT, and compatibles (DOS); MicroVAX/GPX (VMS)	None	Mouse-driven hierarchical menu system that expands and contracts automatically, plus keyboard; ASCII input and output files; forward and backward annotaion
SDA SYSTEMS INC. *555 River Oaks Pkwy.* *San Jose, Calif. 95134* Geoffrey Indrajo (408) 943-1234	**SPICE INTERFACE WITH SPICE 2G.6** Software only SPICE Interface with SPICE 2G.6, $5k; electrical rule checker, $15k	Apollo 3000, 4000, DN570; MicroVAX II, VAX 3200, 3500; Sun-3 and -4	None	SDA schematics
	HSPICE INTERFACE Software only HSPICE interface, $5k;electrical rule checker, $15k	Apollo 3000, 4000, DN570; MicroVAX II, VAX 3200, 3500; Sun-3 and -4	None	SDA schematics
SILVAR-LISCO *1080 Marsh Rd.* *Menlo Park, Calif. 94025* Herman Beke Vice President, Marketing (415) 324-0700 (800) 624-9978	**ANDI** mixed analog-digital simulator Software only $25k+	Sun-3 (Unix); Apollo (Aegis); DEC VAXstations 700, 8000 (VMS); IBM 9370, 30xx, RT PC (VM/CMS)	None	Includes schematic capture for entry and waveform display for simulation results
	SWAP switched-capacitor filter simulator Software only $23k+	Sun-3 (Unix); Apollo (Aegis); DEC VAXstations 700, 8000 (VMS); IBM 9370, 30xx, RT PC (VM/CMS)	None	Includes schematic capture for entry and waveform display for simulation results
SPECTRUM SOFTWARE *1021 S. Wolfe Rd.* *Sunnyvale, Calif. 94086* Donald Chan CAE Development Engineer (408) 738-4387	**MICRO-CAP II** Software only $895	IBM PC XT and PC AT, PS/2 model or equivalent	EGA, CGA, or Hercules graphics cards; 8087 coprocessor support, also 80287/80387 hard disk recommended	Keyboard- and menu-driven

OUTPUT FORMATS	PRINTERS AND PLOTTERS SUPPORTED	LIBRARIES	OTHER FEATURES	SYSTEM CAPACITY AND MEMORY REQUIREMENTS
Generic postprocessing outputs; comprehensive outputs to manufacturing and ATE; links to Mentor, Daisy, Valid, Computervision; IPC 350, IGES	Gerber, Benson, Versatec, Emma, AEG, HP, Ferranti, EIE, Calcomp	SSI/MSI models, LSI behavioral models, 1000+ components	On-line reports; electrical rule checking, design rule checking, 21-state simulation, 90+ configurable primitives, mixed-level simulation, digital stimulus language	Virtually limitless capacity
ASCII output for the entire database (symbols, parts, netlist, tags, etc.); EDIF 1.0; interfaces to CADDS, Cadat, and SPICE via Access (standard product)	Directly supports Epson and HP plotters; via Scicards/UNV file, output can be generated for a host of other pen and electrostatic plotters	600+ parts (standard)	Scidesign provides checks for duplicate component designators, missing off-sheet connections, and sheet-to-sheet reference bullets. Access allows the user to create checks for his specific application or concern (company part numbers, simulator information, thermal parameters, etc.)	On GPX, limited to the amount of memory available on that system
Silos, Hilo, SPICE, HSPICE, Verilog	Not specified	None (standard basic elements: transistors, capacitors, etc.)	Electrical rule checking (option--can also be used to check layout)	Design-specific
Silos, Hilo, SPICE, HSPICE, Verilog	Not specified	None	Electrical rule checking (option—can also be used to check layout)	Design-specific
Netlist for Gards, Avant Gards, Cal-MP, SC II, and Silvar-Lisco PCB design packages; optional output to SPICE netlist and tool kit for user-defined netlists (EARS)	HP, Calcomp, and Versatec plotters	60 components, including MOS and bipolar transitors, R, L, C, sources, and digital logic devices (standard)	Schematic capture and output waveform display (standard); system handles mixed analog-digital circuits; SPICE integration package (optional)	20,000 components
Netlist for Gards, Avant Gards, Cal-MP, SCII, and Silvar-Lisco PCB design packages; optional output to SPICE netlist and tool kit for user-defined netlists (EARS)	HP, Calcomp, and Versatec plotters	15 component primitives, including MOS transitors, R, L, C, sources (standard)	Schematic capture and output waveform display (standard); distortion, noise, sensitivity, frequency and time analysis modes (standard); SPICE integration package (option)	Capacity is a function of number of components and time steps
Schematic netlists	Epson printers; HP 7470–7475	100 op amps, 100 diodes, 100 pnp/npn transistors, 100 MOS transistors, 50 programmable waveforms, 50 sinusoidal sources, 50 transformers, 50 polynomial sources (standard)	Transient, ac, dc, and fourier analysis; does not handle digital and mixed-signal circuits	150 nodes for transient and dc analysis, 75 nodes for ac analysis; maximum number of components handled in a circuit: 200 switches, 200 resistors, 350 capacitors, 50 inductors, 50 batteries, 1 user source, 10 sine sources, 10 programmable waveforms, 10 transmission lines, 50 polynomial sources, 50 diodes, 50 transformers, 50 op amps, 150 npn/pnp transistors, 150 FETs

DIRECTORY OF ANALOG SCHEMATIC CAPTURE SYSTEMS (continued)

VENDOR AND CONTACT	PRODUCT AND PRICE	HOST	SPECIAL-PURPOSE HARDWARE	INPUT METHODS
VALID LOGIC SYSTEMS INC. *2820 Orchard Pkwy.* *San Jose, Calif. 95134* Paul Giordano Product Marketing Manager (408) 432-9400	**ADVANTAGE** analog design environment Software only and turnkey $20k–$50k, typical software pricing	Sun and DEC platforms	None	Object-oriented, mouse control
VAMP INC. *6753 Selma Ave.* *Los Angeles, Calif. 90028* John Soluk Director of Marketing (213) 466-5533	**MCCAD** Software only McCAD Schematics V3.1, $495; McCAD Schematics/DSIM, $695; McCAD PCB-ST, $995; MicroSim PSPICE analog simulator, $1.4k–$2k	All Macintoshes, including Mac II	None	Typical Macintosh mouse and menu input; keyboard only when necessary
VIEWLOGIC SYSTEMS INC. *275 Boston Post Rd. West* *Marlboro, Mass. 01752* K.S Sriram Vice President of Marketing (617) 480-0882	**WORKVIEW** Software only $4k–$15k	IBM PC AT; Compaq 386; PS/2; NEC PC-XL; MicroVAX 2000/GPX	PC AT and compatibles, 80386 machines, and PS/2 need graphics card such as EGA, VGA, or 8514 to display graphics	Mouse with pop-up menu; keyboard; multiple windows
VISIONICS CORP. *343 Gibralter Dr.* *Sunnyvale, Calif. 94089* Alex Wellins Marketing Specialist (800) 553-1177	**EE DESIGNER III** Software only $3,995	IBM PC XT, PC AT, and compatibles; Compaq 386	CGA or EGA card	Mouse with hierarchical menu structure; full back annotation to multipage schematics
XEROX E.D.D.S *2441 Mission College Blvd.* *Santa Clara, Calif. 95054* Wynn Kageyame Sales Engineer (408) 562-2170	**EXPERT SCHEMATICS** Software only and turnkey $4.9k—software; $8k–$15k—turnkey	Xerox 6085and 8010	None	Mouse with pop-up menu; back annotation; multiple windows and multiple views of schematic

OUTPUT FORMATS	PRINTERS AND PLOTTERS SUPPORTED	LIBRARIES	OTHER FEATURES	SYSTEM CAPACITY AND MEMORY REQUIREMENTS
Simulator interfaces: Hilo-3, Lasar, Logcap, McLDL, Simon, SPICE, Tegas V. *PCB interfaces:* Allegro, Applicon, Cadam, CADDS 4, CADDS 4X, Calay, RINF, Scicards, Wire Wrap, CBDS, Intergraph, Redac. *Tester interfaces:* GenRad, Zehntel	Numerous plotters from Versatec (including color units) and Benson; pen plotters form Calcomp, HP, and Benson; printers from Epson; and A- and B-size Postscript	50 most commonly used discrete devices (standard); 1,500-part library (option); model parameter editor (standard)	Statistical sensitivity Monte Carlo, and expression analysis (standard)	4 MB main memory and 80 MB disk minimum
Compiles and extracts parts list and netlist in ASCII from single- or multiple-sheet projects; netlist output in standard text file form as well as Computervision, Gerber, Scicards, Calay; user-configurable SPICE netlist/command file generation; standard file preparation for McCAD PCB-ST and EDS-1; printed circuit design environments	Imagewriter; Laserwriter; Houston Instrument DMP 40, 41, 42, 51, 52, 60, 61; HP 7475, 7580, 7585, 7586; Draft-Pro and DraftMaster pen plotters	Includes transistors, FETs, acoustic devices, switches, vacuum tubes, op amps, transformers, light systems, transducers, etc.—approx. 350. McCAD Schematics comes with an on-line library graphics editor that permits rapid element creation and addition	McCAD Schematics can be used for analog, digital, or mixed-mode designs	In the project mode the system can track 32,000 devices or 32,000 nets, whichever comes first; 2 MB main memory recommended, 1 MB minimum
CAE/CAD graphics: Computervision CADDS 4, CADDS 4X, CADDstation, Mentor, Metheus, Prime EDMS, Valid *PCB layout—forward:* Redac (Redboards, Cadet, Maxi, Mini), Redac (DSM6). *PCB layout—forward and back annotation:* Academi, Applicon, Cadnetix, Calay, CBDS, Computervision, Intergraph, Merlyn, Omnicad, P-CAD, Prance, Redac RINF (Visula), Scicards, Telesis *PCB layout—placement forward:* Cadnetix, Calay, Scicards, DataCon *EDIF:* graphics, in/out, netlist *IC layout:* Ecad, Silvar-Lisco *Simulators:* Bolt, Cadat, Computervision, Helix, Hilo, HSPICE, IG-SPICE, Lasar, Precise, PSPICE, Silos, SPICE, SDL, Tegas, Touchstone, Zilos/Zycad	Epson printers and compatibles; IBM 4201 and 5152; HP Thinkjet, Laserjet+, and Paintjet; Versatec electrostatic (from VAXes only); Houston Instruments DMP41, DMP42, DMP51, DMP52; Hewlett-Packard HP7440, HP7470, HP475, HP7550, HP7570, HP7580, HP7585	Transistors, 221; diodes, 95; op amps, 77; JFETs, 23; power bipolar transistors, 21; power MOSFETs, 86; MOSFETs, 9; cores, 25; crystals, 8 (all options)	Optional: ac, dc, and transient simulation and analysis; Monte Carlo analysis; stress analysis; mixed analog and digital simulation, non-linear magnetics simulation; support for simulation on a variety of mainframe and supercomputers	Up to 4 GB of memory available for design capture and analysis
Netlist in ASCII format, can be edited; database contents in ASCII	HP Laserjet series II; others not specified	Extensive multipurpose library, including surface-mount parts (standard); seperate multipurpose module, 200 parts; TTL, approx. 1,000 parts (option); microprocessors, 200 parts (option)	Standard: full busing in schematics; list-of-materials processor for bill of materials; 2 autorouters; 25- and 50-mil grid with memory routing on both; user-definable design rule checking; power and ground signal routing separate	Will handle large boards up to 12 circuit layers, plus silkscreen and masks; need LIM expanded memory card to handle large boards
Netlist in ASCII format bill of material; data extract list	Electrostatic and laser printers, pen plotters	Library contains 1,200 symbols (standard); library creation and editing ability (standard)	Busing; special power and ground handling	Handles schematics from 1 to 50 pages; A-, B-, C-, D-, and E-size drawings

RISC BALANCE Changes the

The quest for speed has driven microprocessor architectures

BOB CUSHMAN, SENIOR EDITOR

The message behind the current avalanche of RISC processors is that what counts is not what the suppliers are presenting today, it is what they might present next year or the year after. None of the current RISC introductions represent finalities; they are but first iterations of product lines that are certain to be enhanced. Their suppliers are engaged in a battle of promises, each trying to outdo the other with predictions of more future performance and less cost.

ALL RISC MODELS ARE NOT ALIKE

One way to get a handle on RISC progress is to study the architecture. This article kicks off a series that will start with a general look at some of the elements of RISC architecture and then, in following articles, delve into the details of the architectures of specific RISC chips.

Figure 1 is the starting point. It provides a simple model that will help define RISC features and serve as a basis for comparing different RISC families. It will also help in comparing RISCs with competing CISCs and DSPs.

Three basic categories of software instructions are listed above the diagram in Figure 1: program manipulation instructions, data movement instructions, and data manipulation instructions. The arrows relate these to the parts of the model to which they apply. It is appropriate to introduce software into the picture because the goal of RISC is to create a superior, more synergistic software-hardware combination.

Below the model are listed the three main steps in a computer's operating cycle: instruction fetching, instruction decoding, and instruction execution. These steps are also related by arrows to the parts of the model to which they apply, and by inference to the three categories of instructions above. Indicating the relationships is appropriate, for it will help localize the various RISC features.

■ JUST WHAT IS RISC ANYWAY?

Before stepping through the operating cycle in Figure 1, let's review the RISC principles. The table, based on a presentation by Randy H. Katz of the University of California at Berkeley for a 1986 ISSCC evening panel session, lists the objectives of RISC design. It can be seen that the thrust of "pure" RISC is to reduce and streamline the instructions so that they can be compiled efficiently and executed quickly.

Whereas it is easy to find these "pure" RISC principles in early R&D RISCs (such as those that have been reported on at ISSCC—from Stanford University and the University of California at Berkeley and from IBM and Bell Laboratories), it is not so easy to isolate these principles in present commercial RISC chips. The fact is that designers have found that they couldn't show enough of a performance advantage over established CISC microprocessors with "pure" RISC designs. They have had to add so many CISC embellishments to the RISC concept that the so-called RISC machines are now in reality RISC-CISC machines.

The main RISC concept that designers have held onto is that of completing an instruction every cycle. In fact, now the thrust is to use parallelisms to complete more than one instruction per machine cycle. This move is causing some recent RISCs to look a bit like DSPs.

The computer cycle of Figure 1 begins with the instruction fetch. The address is

ILLUSTRATION BY MICHELLE BARNES

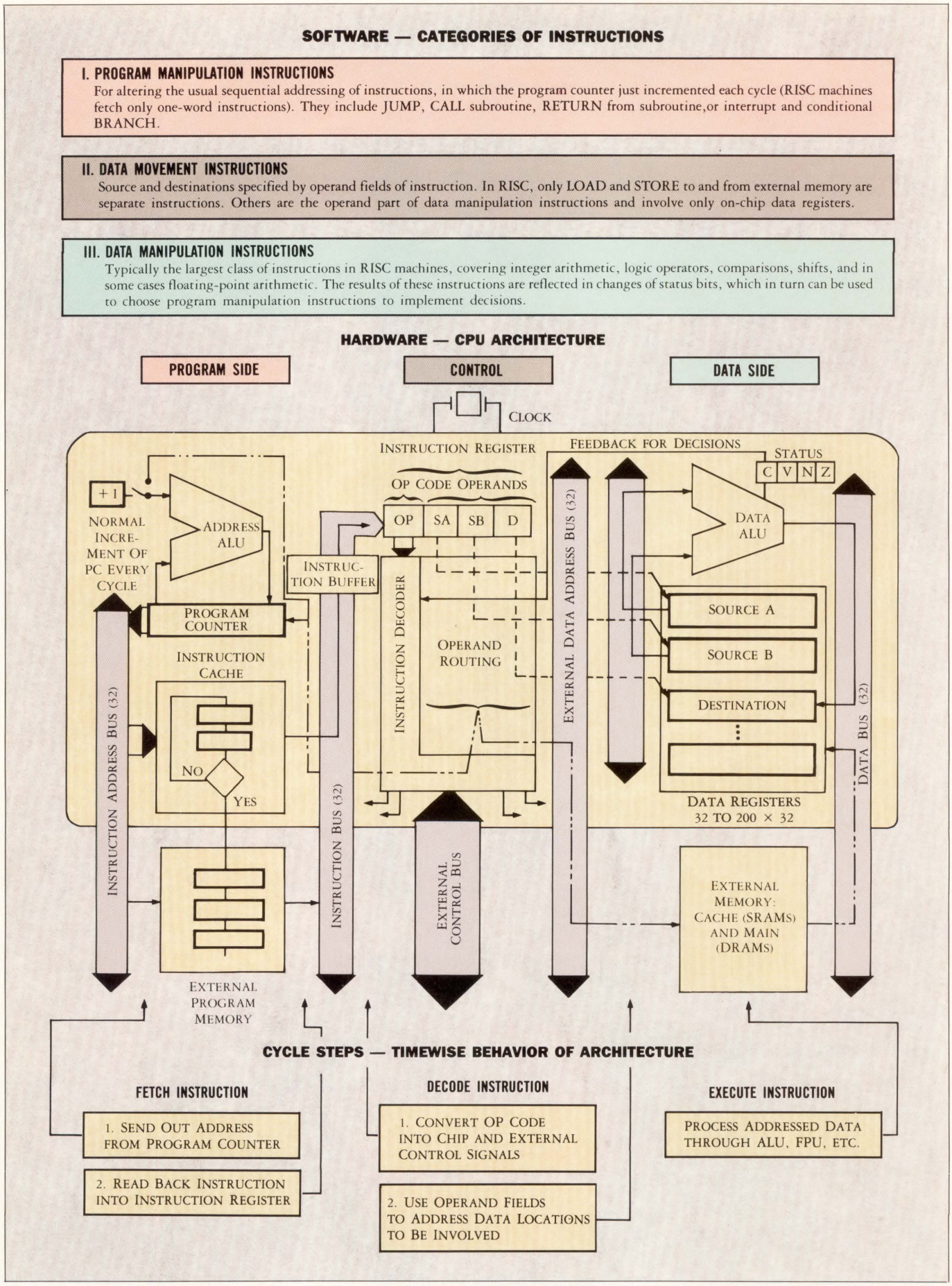


Figure 1. This simple model of a stored-program digital computer is the starting point for analyzing and comparing RISCs. The architectural diagram in the middle is surrounded by software categories (top) and timewise behavior (bottom), so that these other ways of looking at RISCs can be tied to the hardware involved.

ATTENTION CALMA OWNERS:

The buyout of Calma Corporation's IC layout operation poses some tough questions to current Calma owners. After all, Calma pioneered the development of IC CAD, including interactive layout, and played an important role in the development of most of today's integrated circuits.

But now, an era is ending. Calma's IC layout operation will be run by a different company. So it's an ideal time to evaluate which company and layout system can take you through the 1990's — and beyond.

The answer is simple: Mentor Graphics. And to make your decision even easier, Mentor Graphics' "Secure Future"* program now offers significant discounts, **up to 50%,** to Calma customers who trade in their IC CAD systems for Mentor Graphics' Chip Stations.®

Sure, the financial incentive is nice, but closer inspection gives you even more reasons to move to Mentor Graphics.

Fact: Mentor Graphics stands ready to help you make the change. From compatibility with your design data base, to extensive training and application support, we are prepared to smoothly transition you to our systems. And to make it even easier Chip Station can be programmed to emulate familiar commands of your current system.

Fact: Over the past two years, Mentor Graphics has become the fastest growing IC CAD vendor, with approximately 1,000 IC layout workstations installed worldwide.

Fact: Mentor Graphics is the worldwide leader in electronic design automation (EDA), with more than 12,000 installed workstations.

Fact: Mentor Graphics has proven that it's here for the long haul, with 22 consecutive quarters of revenue growth. And Mentor Graphics has been profitable every quarter since 1983, not to mention current reserves of over $100 million and no long-term debt.

Fact: Mentor Graphics has a customer list that reads like a "Who's Who" of the chip industry. And, with Mentor Graphics' tools, service and support, those customers return with repeat business.

Fact: Chip Station also brings you both the capacity and methodology to deal with the coming generation of ULSI designs:

- Structured Chip Design lets you represent cells symbolically, so the organization and management of large designs is greatly simplified.
- Device-level editing defines and manipulates entire groups of polygons in single operations.
- REMEDI™ integrates DRACULA II LVS checks to include graphic correlation of errors in both schematics and layouts.
- Cell Station® gives you access to automatic place and route.
- For VLSI color plotting, Chip Station's new rasterization capabilities offer the industry's fastest solution.

Make your next step a little easier. Discover why Mentor Graphics offers a "Secure Future" program that lives up to its name. As a Calma owner, more information on *your* secure future may be one phone call away.

Don't let a corporate acquisition make up your mind for you: call 800-547-7390 today.

* "Secure Future" program details available on request. Offer ends 12/31/88. Calma IC layout systems must have proprietary software erased before they are turned over to Mentor Graphics.

Mentor Graphics®

Your ideas. Our experience.

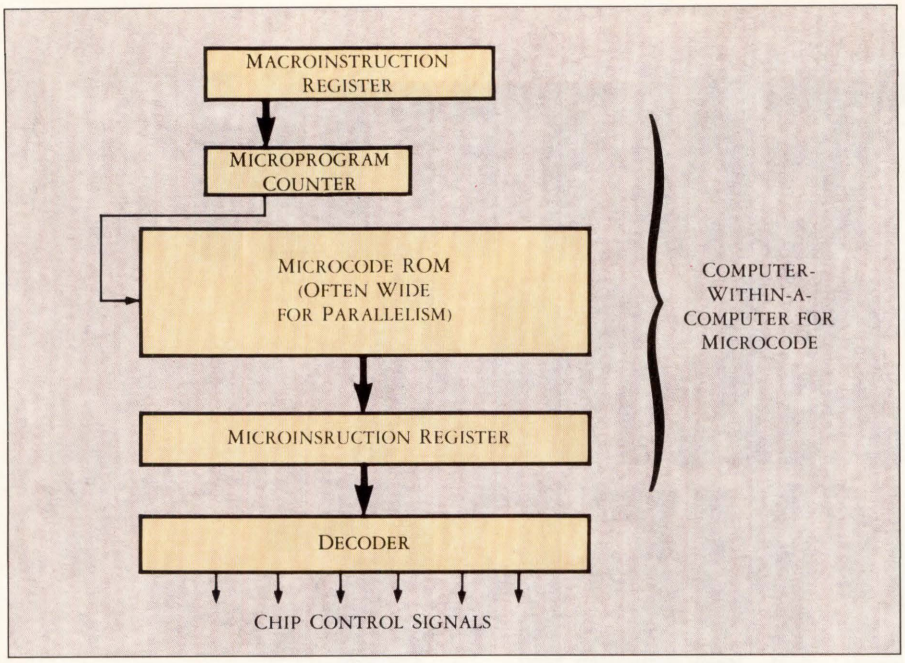


Figure 2. Microcoding inserts a "computer within a computer" in the instruction-decoding path. The approach shown is but one of several.

sent out from the program counter (PC) and the instruction is read into the instruction register (IR). RISCs don't appear to vary appreciably from CISC or other architectures with respect to instruction fetching. Also, because they are 32-bit general-purpose machines, RISCs must, like other 32-bit machines, be prepared to fetch instructions from a large memory space (for one thing, the Unix operating system, which user-reprogrammable RISCs are expected to support, is itself in the megabyte range).

■ CACHE AID

The differences that do exist between RISCs and CISCs are due to RISC's emphasis on putting a new instruction into the instruction register each cycle—even despite interruptions in the program flow. As RISCs reach for ever higher performance levels, expect to see more complexity in the fetch operation. Expect to see increased use of separate, Harvard-architecture, fetch buses, instruction prefetch buffers, and most important, levels of cache memories.

Figure 1 indicates in a somewhat sketchy manner the help that a cache memory can give in the very typical situation in which a software loop is being executed over and over. As long as the cache can contain the loop instructions, the fetching can proceed at full speed. But when a comparison instruction (the diamond shape) causes the execution to fall out of the loop, there could be a performance-damaging hesitation when the fetch has to go out to main memory.

The problem is not just that of the cache logic realizing that the next instruction isn't within the cache, plus the slowness of the main memory, the problem is also that the decision instruction itself poses a problem to the pipelining invariably found in modern RISCs (as well as CISCs and DSPs).

How can the architecture sail past such breaks in instruction execution? This problem is a tough one for RISC designers, as they are expected to keep their execution pipelines full. The designers appear to be heading toward brute-force solutions, such as going to the extreme of prefetching into the instruction buffer both instruction streams that emanate from the branch decision (both the No and Yes paths coming out of the diamond in Figure 1)—or to at least be sure that both instructions streams are at hand, waiting in the cache. This follows a general trend away from the early RISC dictum of "let the compiler do it" and toward increased hardware sophistication. Thus, while the designers of earlier RISCs might have been content to let the compiler figure out the solution, the trend in more recent RISC chips is to see that the problem is solved in hardware and thus inherently faster as well as transparent to the compiler.

■ DECODING SEPARATES RISC

The instruction-decoding process is where RISC in principle departs most noticeably from the CISC architecture that has been so widely used for 16- and 32-bit microprocessors. As stated in the table, one of the cornerstones of the RISC philosophy is that decoding should be simple and therefore fast. There should be no computer-within-a-computer microcoding, as in CISCs.

Microcoding involves the regular instruction register directing a microprogram counter to address instructions in a microcode ROM, which then feed a microinstruction register (Figure 2). Not only does this procedure insert extra steps in the decoding process, but for very complex CISC instructions it could involve a multiple microcycle sequence of microcode ROM locations.

One of the reasons that RISCs have become popular for 32-bit microprocessors is that the 32-bit word is wide enough to hold all the fields needed for complete one-cycle control. If we assume the 32-bit word is divided up into four byte fields, as is shown in Figure 1, the four fields are sufficiently wide for the instruction operation code (op code) and three operands. Having three operands means that the two data sources needed to feed an ALU and the one destination to receive the ALU result all can be addressed. In Figure 1, the dashed lines from the instruction register represent the control and addressing, and the solid lines represent the flow of data from two selected source registers to the ALU and then from the ALU back to a selected destination register.

The 8-bit fields, with their 256 "degrees of freedom," are enough to specify all the single-cycle operations of today's RISCs. The op code can handle the 100-plus instructions of current RISCs (the goal of a "reduced instruction set" has rather gotten lost). The operand fields can ad-

RISCs at CICC

RISC cores were being proposed for advanced ASICs by several exhibitors at last month's Custom Integrated Circuits Conference in Rochester, N.Y. LSI Logic Inc. (Milpitas, Calif.), said it now had both the SPARC and MIPS RISC processors in its ASIC library and was demonstrating how to implement custom variations of these processors. Similarly, VLSI Technology Inc. (San Jose, Calif.) was showing how its tools could be applied to the 86C0X0 (ARM) RISC core.

Also at CICC, Silicon Compiler Systems Corp. (San Jose) was telling attendees how its GDT system helped produce the Motorola 88000 RISC chip (see last month's *VLSI Systems Design*).

WORKING DEFINITION OF A VLSI RISC PROCESSOR		
SIMPLIFIED:	The number of instructions ■ Small number of "orthogonal" instructions ■ Unbundling of independent operations into separate instructions ■ Avoidance of "kitchen sink" instructions ■ Implicit assumption: instructions can be divided into two classes—(1) simple and frequent; (2) complex and infrequent trapping to software for the latter is not too awful The instruction format ■ Small number of easy-to-decode formats ■ Separate register-register from load-store instructions ■ Required pipeline resources are obvious from the instruction ■ Predictable instruction timing simplifies instruction scheduling ■ Attempt to issue an instruction every clock cycle The addressing modes ■ Small number Control ■ Only two levels: software sequences directly interpreted by control hardware; no "in-between" control	RESULTS: the control is simplified and the basic execution cycle time of the machine is reduced
DONE IN SOFTWARE:	More help from the compiler: delayed branches, software interlocks	RESULTS: simplified and regularly running pipeline
SAVED SPACE USED TO:	Reduce demands on critically limited chip bandwidth by incorporating more memory on chip in the form of registers and caches Smaller processor area equals faster circuitry with higher yield	RESULTS: increased system throughout RESULTS: faster circuitry (and therefore higher system throughout) with higher yield (and therefore lower cost)

dress all the registers that can be squeezed onto current RISC CPU chips. Recall from the table that one of the RISC principles is to have enough on-chip registers to contain most of the data being processed so that only occasional wasteful LOAD/STORE instructions are necessary to move data to and from external memory (depicted in Figure 1 by dashed-dotted lines). With 32 bits, it's not that necessary to go to a wider microcode ROM to have the control parallelism needed for high throughput. Note that one by-product of the 32-bit instruction word is that the operations of the three categories of instructions listed at the top of Figure 1 can be combined into single, more powerful instructions.

■ PIPELINED EXECUTION

However, though it's possible to start new instructions each cycle, many instructions actually take a number of cycles to complete. The RISCs (as well as many CISCs and DSPs) solve this problem by pipelining. The pipelining in current machines is on the order of four stages deep. Think of a pipe that is long enough that it takes four shoves of the cycle clock to push data from one end to the other. Better, think of a number of four-stage pipelines in parallel. What counts is that the operations are so staggered among these pipelines that results come out every cycle.

Coordinating the flow of data through concurrent pipelines can be tricky, especially when branching decisions must be made. How can you keep a pipeline full when you won't know which of two possible next instructions to fetch until a few clocks in the future? One popular technique used by RISCs is delayed branching. In the Am29000, for example, the branch instructions indicate that the branch won't be taken until an intervening delay instruction has been executed. That gives the execution process time to fetch the proper instruction indicated by the decision. Advanced Micro Devices claims that 90% of the time the compiler can insert a useful instruction into the delay slot. This juggling of code to fit tight pipelining is what makes assembly-language programming so difficult that it is impractical for anything but I/O operations. The I/O operations must still be done at the assembly level because they usually are not included in an operating system, like Unix, that is invariably paired with RISCs.

■ TURNING TO CISC FOR ANSWERS

Bare-bones RISC has so many Achilles heels that either slow down its overall benchmarks (for example, overly crude multiplication instructions) or drive up its system cost (for example, the need for very fast but expensive SRAMs), so that it hasn't really been able to compete in the 32-bit arena. Indeed, few of the original bare-bones RISCs that came out of university or corporate research projects have made a dent in the 32-bit marketplace. They have either been so upgraded that they in fact have become RISC-CISC machines or they have quietly faded away.

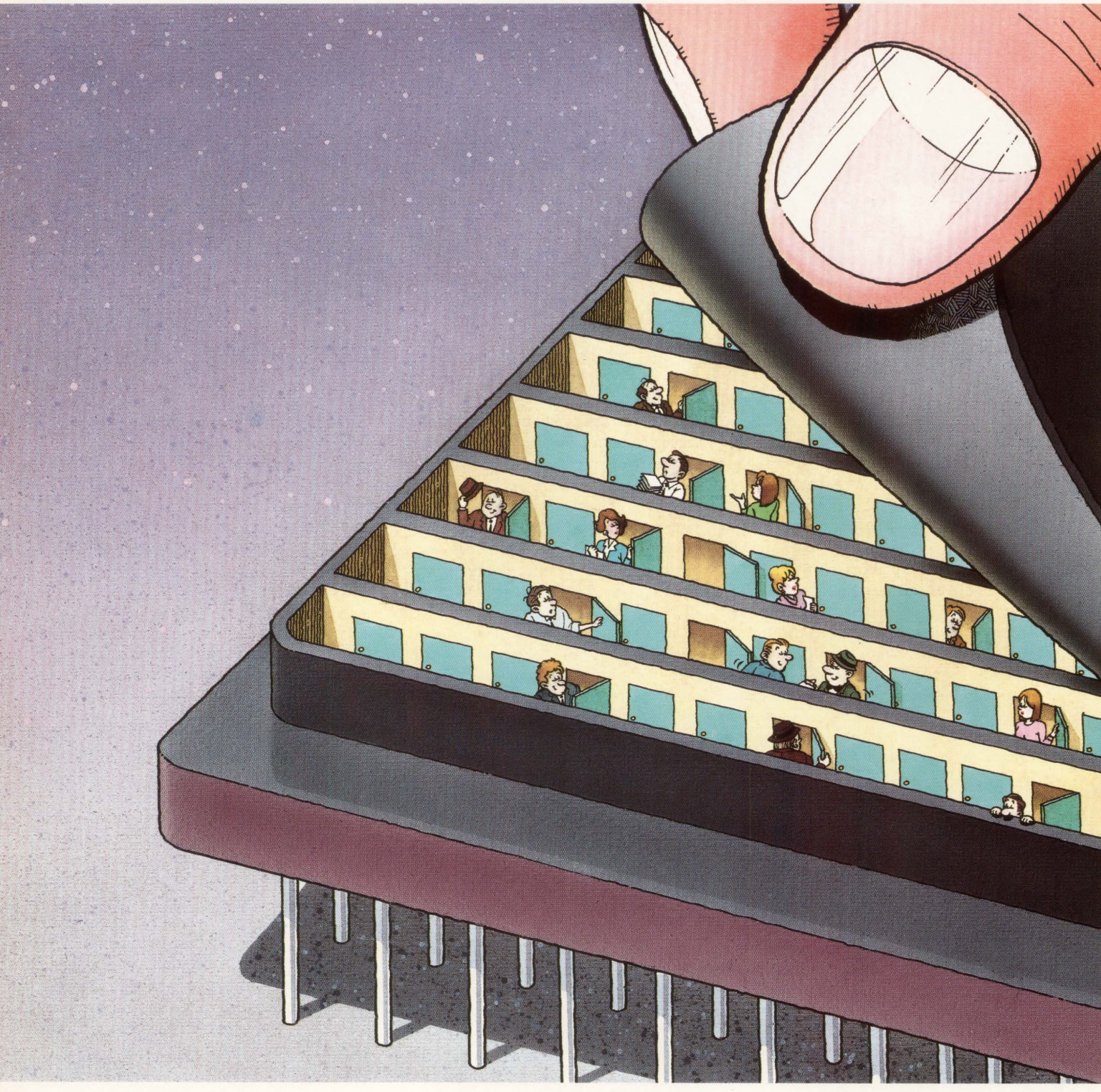
Many of the designers of so-called RISC machines have privately told us that their machines were never intended to be pure RISC, only semi-RISC. The designers of the Inmos Transputer and the Intergraph Clipper are cases in point.

Now we are hearing the same disclaimers from the designers of the new Motorola 88000 and Intel 80960. These designers haven't hesitated in adding non-RISC features—for example, microcoded control—wherever their analysis showed the system would benefit.

Meanwhile the designers of CISC machines are seeing how they can streamline their critical instructions through the use of RISC principles. No one yet has proven that a National 32532 (one of today's higher-performance CISC processors) or the extremely popular Intel 80386 couldn't have more single-cycle instructions. By the same token, the new 32-bit DSP chips from AT&T, Motorola, NEC, and Texas Instruments can—sometimes with CISC architectural features—easily outperform the best of RISCs in single-cycle number crunching, such as the multiply-accumulate of DSP algorithms. Thus there is much homework us in studying, analyzing, and comparing RISCs, CISCs, and DSPs before we have the insight to see which direction microprocessor-based architectures should go for various end uses. ■

Nobody makes it denser

45,000 gates in a CMOS channel-type gate



C&C Computers and Communications

array

High integration and high speed—you get it all from NEC's new CMOS-5 and -5A channel-type gate array families. Gate counts start at 2000 and range all the way up to 45000, the world record for density in a channel-type CMOS device. Gate utilization is over 90% (μPD65450).

Speed rivals ECL performance. Internal gate delays are 1.0ns for a standard gate; 0.65ns for a power gate (F/O = 3, L = 3mm).

CMOS-5 and -5A families protect your system with high resistance to latchup—over 1A (typ). That's more than enough for security in most applications. Other features include:

- ☐ High drive capability: I_{OL} = 3—18mA.
- ☐ 5K-ohm pull-up resistor-incorporated buffers; ready for TTL bus lines.

NEC now offers you the broadest CMOS gate array lineup—six families, with 37 masters, including the world's densest channel-type gate array. For more information, call NEC today.

Family	Master	Gate Count	I/O Pads
CMOS-5	μPD65025	2016	71
	032	3366	81
	044	4440	112
	051	5292	120
	061	6348	132
	071	7500	144
	082	8748	152
	103	10800	164
	140	14256	188
	180	18144	224
	240	24000	256
CMOS-5A	μPD65300	30600	256
	450	45012	256

For further information, please contact:
USA Tel:1-800-632-3531. In California: Tel:1-800-632-3532. TWX:910-379-6985. W. Germany Tel: 0211-650302. Telex:8589960. The Netherlands Tel:040-445-845. Telex:51923. Sweden Tel:08-732-8200. Telex:13839. France Tel:1-3946-9617. Telex:699499. Italy Tel:02-6709108. Telex:315355. UK Tel:0908-691133. Telex:826791. Hong Kong Tel:3-755-9008. Telex:54561. Taiwan Tel: 02-522-4192. Telex:22372. Singapore Tel:4819881. Telex:39726. Australia Tel:03-267-6355 Telex:38343.

NEC

Writing Your Own CAE Tools

MARTY DENHAM, TEXTRONIX INC., LAB INSTRUMENTS DIVISION, BEAVERTON, ORE.

Even hardware designers can write software that really pays off

Using computers to automate design tasks can allow engineers to spend their time more fruitfully; can cut the time needed to design, manufacture, and service a product; and can reduce the time needed to get the product to the marketplace. Quicker development, in turn, can extend the product's life cycle in the market and perhaps allow it to sell at a higher price. All these factors contribute to greater profitability.

Although it would be ideal if a product development team could acquire all the design automation software it needs off the shelf, there are many product design tasks for which no suitable CAE software is readily available, and there are frequently other reasons for designing your own. As an example, hardware design methods are constantly changing, often as a result of innovative engineering thinking. Therefore, new methodologies are often necessary to adopt new technologies or to push existing technologies to their performance and integration limits.

Further, there are times when one recognizes that a manual task with a known algorithm should be automated. Still further, roadblocks that may come up during product design are often best solved by automation; automating your own design frequently saves lots of time, as

ILLUSTRATION BY TERRY ALLEN

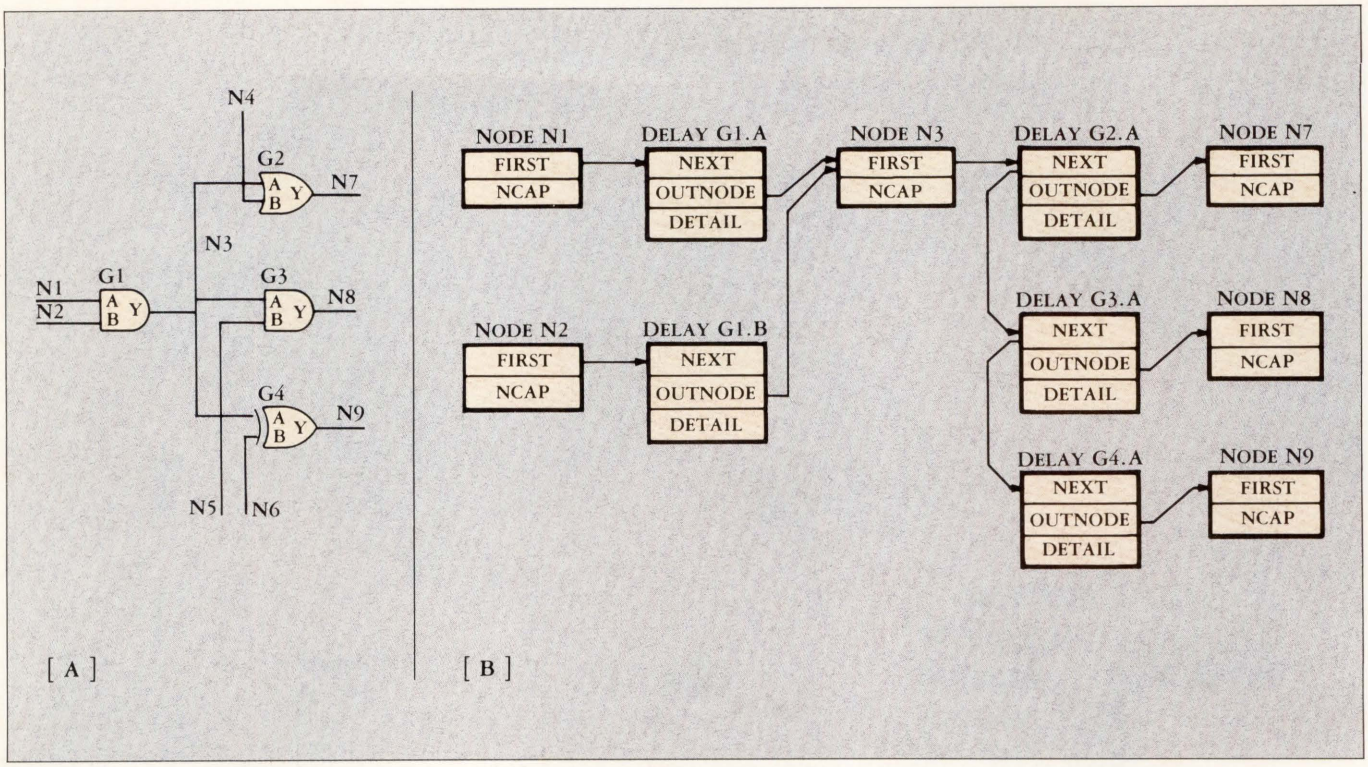


Figure 1. A data structure diagram like this one, for a linked-listing of delay structures for fan-out visualization, can help structure coding.

it may require a significant amount of time for engineers to evaluate available commercial software. Even when turnkey software packages are available, Murphy's law suggests that none of them will completely meet all the needs of a specific project.

The good news is that hardware designers can write special-purpose—or tactical—CAE tools for the numerous product design tasks for which there are no readily available CAE programs. Though the algorithms embodied within a tactical CAE tool may apply to a variety of problems, each tool is targeted at a specific design problem. The tool can be modular, maintainable, and expandable.

Deciding when to write a CAE tool is a matter of straightforward economics; so is enhancing a CAE tool to be more comprehensive or to tackle a different problem. Designers should evaluate design alternatives and select the approach that takes the least time and cost.

Estimating the time to write a tactical CAE tool is difficult but necessary. One suggestion is to write a small program, debug it, document it, and keep track of the total development time, then use this experience as a rough gauge for future estimations. The gauge does not represent time per lines of code. It represents time per piece of code of approximately the same complexity. Large programs are composed of small pieces, and overhead is associated with glueing the pieces together and designing the way the pieces interact. As programs get larger, the time to implement them grows exponentially.

The prerequisite for hardware engineers to write CAE applications is a working knowledge of both a suitable programming language and an available operating system. I used C and the Unix operating system. In developing the software, it's wise to start by setting up a specific target that can be stated in a simple sentence—for example, "This program will be a batch-mode timing verifier that will operate on the XYZ integrated circuit and will print critical timing paths to a file."

One should identify possible extensions of the program to similar designs, again with a simple sentence. "This program will also evaluate the XYZ IC and the ALU IC." Avoid overgeneralization. It's easy to fall into the trap of generalizing a tactical program to meet unforeseen needs.

Think through how the program will be invoked and controlled, then design the architecture of the program and segment its large tasks into smaller pieces. It is easy to dive right in and solve details of the program before creating the architecture, but this approach inevitably leads to rewriting code.

■ Setting Real Schedules and Milestones

It is best to set an aggressive schedule with demonstrable milestones that are no more than two weeks apart, preferably one week apart. One cause of failure to complete a program on time is losing track of the target and beginning to design to unnecessary specifications. When a stringent, weekly milestone is set, it is easier for the writer to throw everything overboard that is not pertinent to the specific task at hand. A good schedule might include something like the following:

Week 27: Figure out a straightforward, simple algorithm to do the task along with a program data structure and be able to explain it to a colleague.
Week 28: Work out with the computer resources support staff the three technical details for file interfaces that are new to me. Begin writing code and be able to run the analysis on a simple test circuit and verify the algorithm.
Week 29: Add the remaining circuit primitives needed for the XYZ IC and begin debugging and screening of code on the chip. Be able to show successful and verifiable results of running the program.

Set a feasibility time line within which to complete the tactical software program. In some cases, when the code is badly needed and there are simply no alternatives, failure to meet the feasibility time line is an indicator that either the schedule should be adjusted or the algorithms should be re-examined.

Code review is not mandatory for tactical CAE software, but it is strongly recommended. Code review helps clean up a program in several ways. As the code writer explains the code, he or she discov-

In the race to market, GenRad's HILO gives the smooth hand-off you need from your ASIC design to your board design.

Getting your new product to market faster than your competition means added profits for a longer time. And that's what HILO® is all about.

The HILO Universal Logic Simulator helps you design quality into your ASICs fast. And you'll find many leading ASIC foundries using HILO in their in-house simulation.

This means your ASIC design via HILO is the ideal connection with the foundry's system. The result is faster production of high-quality custom and semi-custom devices.

And since HILO can feed forward to your board design, your HILO-designed devices move quickly onto your HILO-designed pc boards where they are complemented by the GenRad HICHIP™ physical device modeler.

Think about it. You'll get quality designed devices and boards in less time. And your finished product will get to market faster. Call 1-800-4-GenRad to find out more about HILO.

GenRad

CIRCLE NUMBER 14

See us at DAC Booth #236

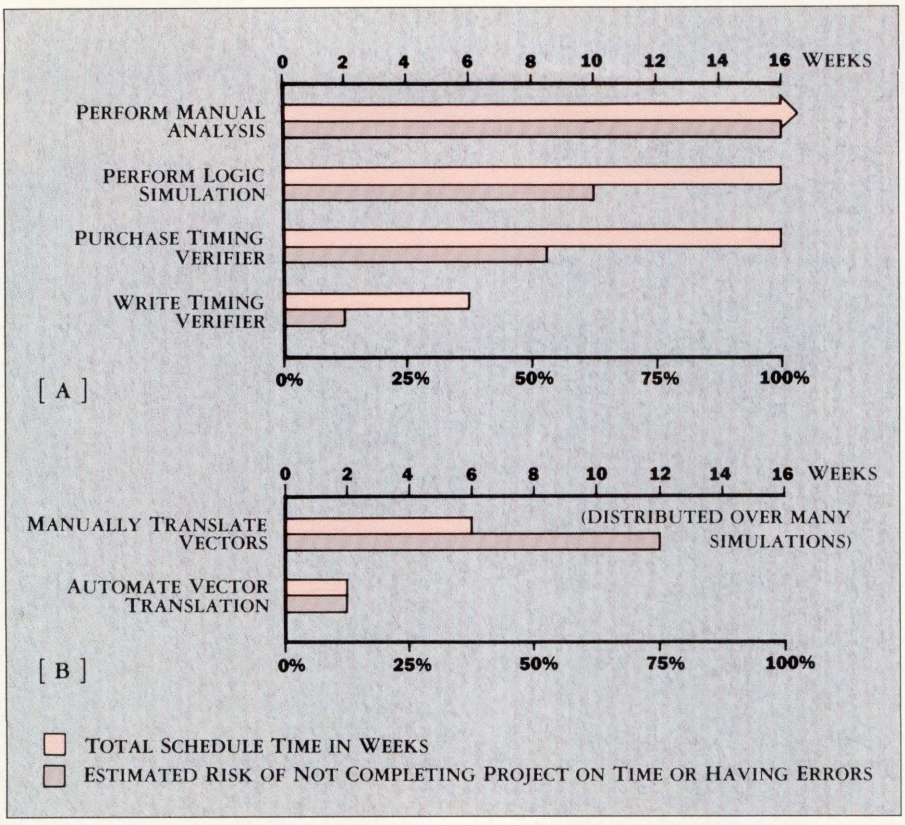


Figure 2. These bar graphs helped determine whether or not to write programs for timing analysis (A) and simulation vector translation and analysis (B). The upper bars show estimated schedule time in weeks, and the lower bars show the estimated risk, in percent, of not completing the task on time or of having errors.

ers errors and inefficiencies. What the coder does not catch is usually caught by the person reviewing the code. This process greatly minimizes the chances for failure of the code to work properly and the chances for catastrophic operation.

Reliability should be given top priority, along with maintainability and in-code documentation. Proper emphasis on maintainability allows limited extensibility (small extensions are inevitable) and provides for thorough debugging.

■ CPU Utilization: A Low Priority

Optimum CPU utilization should be given low priority. It's easy to lose perspective and try to trim five minutes off a ten-minute execution time, forgetting that shaving those five minutes might take another week of coding.

If possible, for data used by the program during execution, it's wise to use CPU main memory and avoid complicated disk-based storage structures. Data files, such as netlists, are probably easier to work with when stored in ASCII text formats. The need to use disk-based database structures arises when the program task has to manipulate tens of megabytes of data. The task of writing a tactical software program is to move the design of a product from point A to point B in the design flow, not to figure out clever ways of sorting data records on disk. The simplest system accessories of Unix, for example, will suffice for tactical development.

There are other useful rules:

- Use data structures to represent logically grouped data fields. Data structures allow much easier handling of data than groups of arrays. It might take some effort to fully understand how to use them, but the data structures have tremendous payoffs in terms of code simplicity.
- Use dynamic memory allocation when you don't know how much memory will be needed. (Even if you know how much memory will be needed, some machines limit static array sizes, so it's wise to find out what those limits are.)
- Develop and use simple memory allocation and deletion subroutines for each structure type. After writing these routines for one data structure, you will write similar ones for other data structure types much faster. Using centralized memory allocation routines will greatly simplify code writing and debugging.
- Use data structure diagrams. When a writer has cleanly structured the data that the tactical program will manipulate, the code will be implicitly structured as well. Conversely, poor data structuring increases the difficulty of writing code and decreases the reliability of the code.

An example is shown, simplified, in Figure 1. This diagram was used to visualize fan-out for a verification program. In Figure 1A we have a combinatorial circuit with fan-out at node N3, the input to three gates—G2.A, G3.A, and G4.A. In Figure 1B we have boxes representing data structures in main memory and subdivisions of boxes representing data fields. For simplification, nodes N4, N5, and N6 are not shown in the data structure diagram.

The NCAP fields in the node structures represent node capacitance. The DETAIL field is a pointer to structures not shown that contain further information about that delay path (like rise time and fall time coefficients and delay constants). The arrows represent links, which are addresses in main memory.

■ The Need to Document While Writing

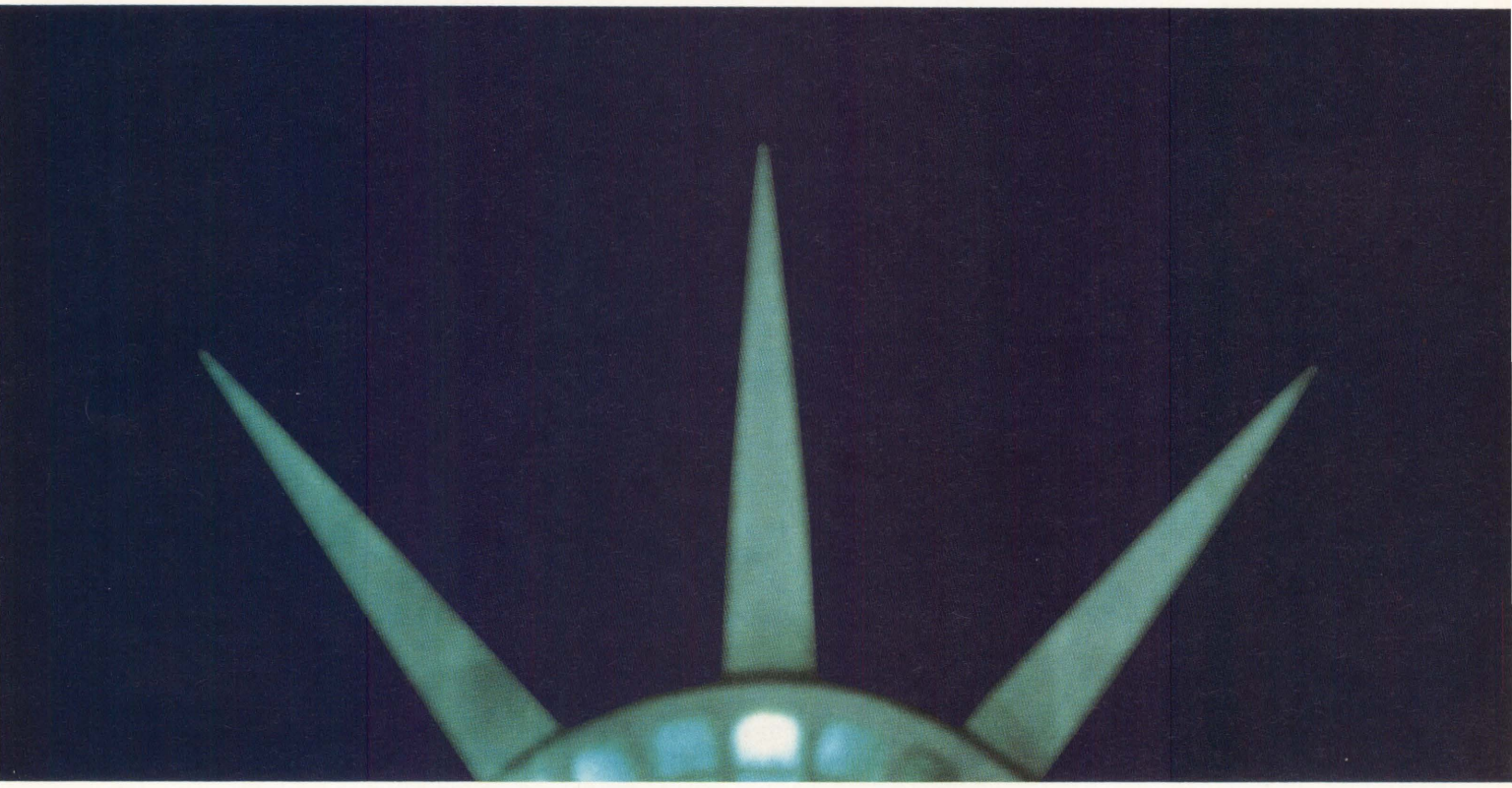
Document the code operation from a user's standpoint as you write the code. It's much easier to document code operation at coding time than it is to do so later. Further, the process of documenting helps a writer think through the program from a high level and uncover inefficiencies and complicated usage patterns; if the writer cannot explain how the program works, it is probably too complicated. It is also easy to forget how to use the program after a period of unuse.

Parsing is the process of analyzing a user's input, checking the input for correctness of syntax, and creating or modifying data structures that represent the input. If the parser for a tactical program is written from scratch, the user input syntax should be as simple as allowable. Examples of simple syntax constructs are using simple one-word commands; placing every command in a file (batch mode); separating commands, one per line; and limiting the size of input lines.

As your programming skills grow, you'll find it worthwhile to use standard utilities to create parsers. On Unix systems, the appropriate utilities are YACC and LEX. They are powerful but it takes time to learn them, and it helps to become familiar with compilers before using these utilities. Using system utilities will increase a code writer's productivity and increase the maintainability of tactical code.

When debugging the tactical program and verifying its function, start with very small test cases with purposely injected errors. If any program output is contrary to expectations, the code is not yet reliable. Every allowable user option must be tested, whether it be on a small test case or on the intended design case, as an untest-

Most simulation systems tell you as much about your ASIC design as this tells you about the Statue of Liberty.



Get the full picture of your ASIC design's performance. Get the IKOS Simulation System. IKOS gives you more simulation information in less time than any other system. Simulate 15,000 gates with 34,000 vectors in 34 seconds. Or 8,500 gates with 5,100 vectors in 4 seconds. Complete with detailed timing information!

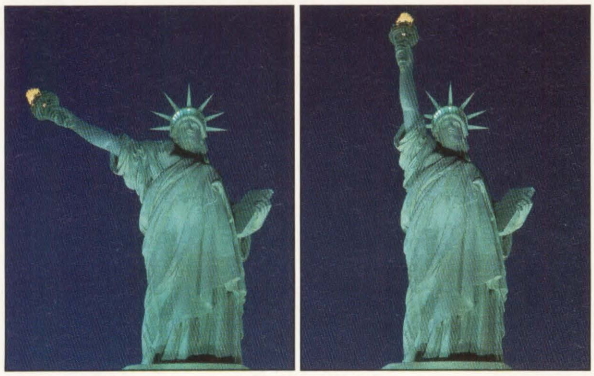
IKOS is a comprehensive ASIC simulation system incorporating a combination of sophisticated software and hardware components that run on the Sun, Apollo, or IBM PC/AT platforms. It provides you with a streamlined, high-speed approach to stimulus entry, logic and fault simulation, analysis, and ASIC library support. And, it integrates easily into your existing design environment.

IKOS gives you a tremendous performance advantage.

You enter stimulus the way you visualize it—by drawing waveforms. Powerful signal modeling features allow you to easily and realistically simulate asynchronous signals and interfaces.

With IKOS' real time logic analyzer you can run simulations interactively, pinpointing signals and conditions of interest. Gone are the tedious hours spent pouring over long output listings.

There's no limit to the number of test cases you can generate and analyze with IKOS. You'll be able to simulate seconds, *even minutes* of real time system operation. And, gain the confidence that your ASIC design works as you intended it to.



Without the IKOS Simulation System, most ASIC design mistakes show up after production.

IKOS comes complete with all the simulation tools you need—at a low price. Use IKOS to get your ASIC design right *before* committing to manufacturing. It's almost as easy as snapping a picture.

Give us a call.

DAC Booth #565

IKOS

Now available on Sun!

408 • 245 • 1900

145 N. Wolfe Road · Sunnyvale, CA 94086

Sun is a trademark of Sun Microsystems. Apollo is a trademark of Apollo Computer, Inc. IBM PC/AT is a trademark of International Business Machines Corporation.

CIRCLE NUMBER 15

COMPARING KEY ASPECTS OF TWO CAD TOOLS		
	VERIFY TIMING ANALYZER	MUPPET VECTOR TRANSLATOR
INTENDED USE	DETECTION OF PATHS THAT ARE TOO LONG IN IC LAYOUTS. ALSO, DETECTION OF 4 OTHER TYPES OF ERRORS	TRANSLATION OF OUTPUT VECTORS FROM LOGIC SIMULATOR INTO READILY UNDERSTOOD VECTORS
COMMAND SYNTAX	KEY WORD AND PARAMETER SEPERATED BY A SPACE. COMMANDS SEPERATED ONE PER LINE	COMMANDS SEPERATED BY SEMICOLONS; MULTIPLE COMMANDS ALLOWED PER LINE
PARSER	BUILT FROM SCRATCH	YACC- AND LEX-BASED
DEVELOPMENT TIME, INCLUDING DOCUMENTATION	4 WEEKS	6 DAYS
NUMBER OF LINES OF DOCUMENTED CODE	2.7K	2.7K (APPROXIMATELY 20% INHERITED)
NUMBER OF BYTES OF DOCUMENTED CODE	70K	63K

ed option can waste time and destroy files or databases.

■ ERRORS SHOULD SHOW

An error should produce a distinguishable error message and then immediate termination of the program. As the program is used, it can be changed to print multiple errors in a single run so that the user can run the program fewer times.

Use available debugging tools and program checkers that are standard on most systems. The "lint" program for C is invaluable in checking a program before compilation. There are also several C interpreters and run-time analyzers available for most popular computers.

We can see examples of the factors in deciding whether or not to write your own programs in two tactical CAE tools I wrote, a timing verifier, called Verify, and a logic simulation vector translator, called Muppet. Figure 2 presents a "back-of-the-envelope" analysis for the decisions. The two graphs show the alternatives the design team examined and compare the merits of each. The top bars show an approximate total number of weeks for getting the task done, and it includes programming, debugging, timing-model encoding, purchasing, and so on. The bottom bars represent an estimated risk factor in percent, the risk being a probability of errors occurring in the analysis, of not completing the task in the estimated time, or both. We can see, for instance, that six weeks was expected for writing the timing verifier and that the risk of errors or failure to complete the task was estimated at about 12%. The time and risk estimates are based on previous experience.

The need for the timing verifier arose in one custom IC project I worked on. When we were almost finished laying out the chip, our group discovered that it would have significantly more parasitic capacitance than had been anticipated. We were faced with four choices to verify that the design operated at the required clock frequency: (1) Change the gate delay rules and manually analyze the design. (2) Change the simulation models and resimulate. (3) Purchase a timing verifier program. (4) Write a timing verifier program.

Manual analysis, using new gate delay rules, was obviously undesirable; the effort would have taken a great deal of time and was certain to contribute errors. Allowing timing errors would have set the schedule back a great deal. At the time of the decision, detailed logic simulation was a standard practice for ASIC designs, but the performance of logic simulation was not as good as that of timing verification, and with logic simulation there remained a risk that insufficient vectors would be run to uncover all timing errors, as well as the risk that an error would be overlooked.

■ THE REAL COST OF BUYING

Purchasing a timing verifier would have taken a long time because none had yet been evaluated locally and there was a great risk that any available package would not be compatible with our simulation environment. Writing a timing verifier was therefore the most promising choice, especially as I had extensive experience in writing CAE tools and understood the design problem.

One necessary part of the timing verifier code was a set of procedures that read an ASCII netlist and built the timing path investigation network. Figure 1 is one of several data structure diagrams that helped me think through how data would be represented internally. In the diagram, each node structure in an array of node structures points to a delay structure representing one path through a gate. When a node has a fan-out, the possible gate delay paths are represented by a linked list of delay structures.

The timing verifier employs a depth-first analysis, completely penetrating a signal path from state element to state element by means of recursion, then backing off one gate and moving down the linked list of gates sharing a common input. When done, the algorithm exhaustively examines all paths from a specified clock node back to that same clock node.

■ VECTOR TRANSLATOR

To achieve a required clock frequency for one ASIC, I applied a number of different pipelining techniques. However, the resulting encoding of control signals and data made it very difficult to understand what was happening in logic simulations. The alternatives considered for evaluating the simulation vectors were manual translation of the vectors and automated translation of the vectors controlled by a set of user-input translation formulas.

There was no question that writing a vector translator was the better choice. Though the manual task was distributed across several simulations, the total time for all manual translations was estimated to be larger than the time for writing the CAE tool. Even if the total times were equal, the CAE tool would still have been worth pursuing because of its possible use in future applications. In fact, it was used extensively later.

The table summarizes key aspects of the timing verifier and the vector translator

Programming Styles

Programming style can be a matter of personal preference for an ad hoc programmer or a matter of enforced standards for a group of programmers working on a common code library. There is a wealth of information on programming style for the C programming language.

My choice for learning C is *The C Primer*, written by Hancock and Krieger and published by McGraw-Hill, New York. It is comprehensive yet very readable. A good choice for a reference manual is *The C Programming Language*, Kernighan and Ritchie, Prentice-Hall, Englewood Cliffs, N.J. After having learned the language, an engineer will get further value from *Reliable C*, written by Thomas Plum and published by Plum Hall Inc., Cardiff, N.J. The Plum book offers practical tips for attacking complex problems.

Learning C is the obvious first step; there is, however, another important consideration—style. The syntax of C is extremely flexible. (In fact, the annual Obfuscated C-Code Contest is held to see who can write the most undecipherable piece of code that still performs a specified function.) The point here is that C allows so many permutations of style that a communications barrier can develop among code writers.

To improve the ability of other programmers to read and understand the writer's code, it is a good practice to conform to established style. One good publication for learning effective style is *C Programming Guidelines*, Thomas Plum, Plum Hall. The beginning programmer will find that adopting a well-honed style will improve his ability to read his own code at a later date.

A limitation of most implementations of C is that they do not have run-time error checking, such as array bounds checking and self-modifying code prevention. Here are four suggestions for dealing with the problem of building reliable code in C. First, practice reliable C programming techniques as outlined in the *Reliable C* book. Second, use the Unix "lint" utility or a similar precompilation code checker. "Lint" analyzes C programs for errors that may not be reported by a C compiler. Third, use run-time C program evaluators that are available on most popular computers. Fourth, when a bug in a program cannot be found by any other technique ("illegal instruction" can be one such bug), then planting trace statements in the program to show where the program is and what the program "thinks" it is doing may be the final necessary step in weeding out that bug.

Several tools in Unix help programmers manipulate programs from a high level. The "make" utility allows the programmer to establish complex time-related dependencies so that incremental code changes cause only incremental compilation. "Make" also allows very large and tedious code building procedures to be controlled and managed easily. The "rcs" (revision control system) utility maintains a history of code changes. These are just a couple of examples of available tools for helping the programmer mangage programs; there are many more. Though the original user's manuals for many of these tools are sketchy, better documentation is being made available. Consult local systems administrators or computer vendors for more information. ■

programs. One interesting point of comparison is the development time for each program and the number of lines of original documented code. Notice that the codes are approximately the same size, but the Muppet development took about 25% of the time needed for developing Verify. This difference is attributable to three factors: Muppet was written nearly two years after Verify, so that my experience improved my productivity; Muppet used system utilities (YACC and LEX) to speed development; and it inherited code from earlier CAE tools (nearly 20%).

■ When to Consider Buying

In general, if the software for a design task is already available, the immediate reaction should be to use it; the task of writing and documenting a CAE tool is not trivial. However, if the software is not readily available and if the task is limited in scope, then writing a CAE tool is worth considering.

Searching the marketplace for an analytical design tool, especially if there are peculiar requirements, can take several months. Then there are licensing agreements to be made and installations to be performed, as well as learning curves to overcome, and almost always bugs to be reported to the vendor. A project team can wait several weeks for bug fixes from the vendor, and so on. The net result is that tool searches can vary anywhere from a month to a year.

On the other hand, an engineer with computer programming skills who understands the design problem can solve it in anywhere from a week to less than three months. Incremental bug fixes are as simple as walking over to an engineer's desk and watching over his or her shoulder while it is done. Peculiar enhancements of almost any kind can be accommodated within days.

One more advantage of hardware engineers writing CAE tools is that the resulting "knowledge base," that is, the combination of algorithms and code, can be a market barrier to competitors attempting to match the products built with them. These programs can be part of a proprietary design methodology. If a CAE tool is generally available on the open market, a competitor can also use it; if a CAE tool and corresponding design technique are known only locally, it could be months or years before a competitor can match those capabilities. ■

Acknowledgments

I am indebted to David Eby for opening my eyes to the usefulness of programming in C. His patience with my never-ending questions was remarkable. Brad Needham, Bill Campbell, Mike Moser and Marc Frajola, four of our resident software engineers, also invariably found time to answer my questions about programming style, data structuring techniques, memory management, and many other programming details. Dan Romike's knowledge about our local systems was invaluable.

About the Author

MARTY DENHAM *is a VLSI IC designer for Tektronix's Lab Instruments Division. Previously, he spent 17 months as a designer for Texas Instruments in Dallas, where he had his first industry exposure to IC design. Between designing ICs and writing ad hoc CAE tools, he finds time to take graduate courses in computer science and go freshwater fishing for steelhead and salmon with his wife and three children. Denham received his BS in electrical and computer engineering from Oregon State University.*

Hardware-Software

Trade-Offs in Real-Time Systems

CHRIS W. MALINOWSKI AND PETER S. DANILE,
HARRIS CORP., SEMICONDUCTOR SECTOR, MELBOURNE, FLA.

The design and development of real-time systems involves maintaining a fine balance between hardware and software: hardware functions are less economical to implement than equivalent software functions, but they execute more quickly. The trade-offs for partitioning systems into hardware and software portions are being affected by powerful processors that are available in many cell libraries. With RISC processors that approach the 10-MIPS performance range available as macrofunctions, where does the designer draw the line between hardware and software?

HIGH-PERFORMANCE PROCESSORS INCREASE THE RANGE OF OPTIONS

Partitioning real-time system functions between hardware and software requires the analysis of performance bottlenecks for each of the tasks to be handled by the processor. It also calls for detailed investigation of the timing of those sets of instructions most critical to the application. Harris has developed the RTX Toolbox (formerly called the FORCE Toolbox), which simplifies these partitioning problem by providing a highly visible architecture as well as straightforward methods to modify hardware and software.

This article will show how the use of core processors can be used by designers to implement real-time (RT) systems efficiently without sacrificing either software portability or the ability

TRATION BY RENEE KLEIN

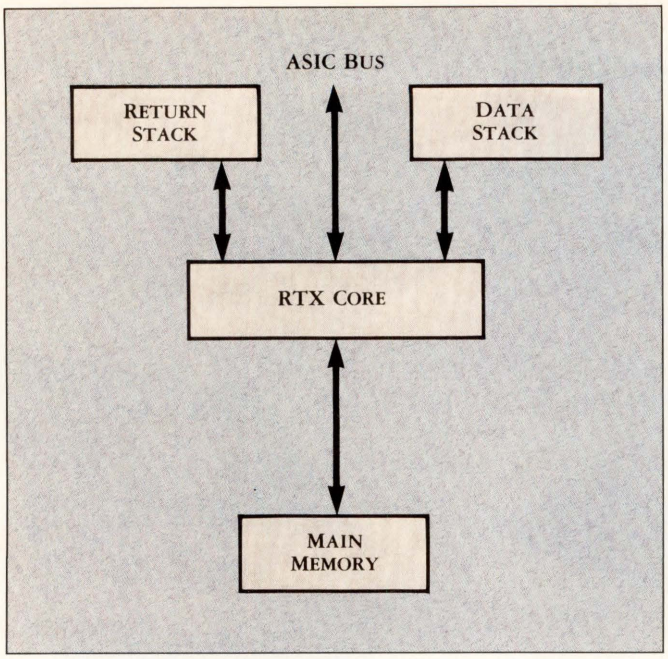


Figure 1. The RTX architecture adds two stack-memory buses and an ASIC bus for hardware assists of real-time computations.

to migrate their designs to other applications. Some software examples are also included to give the designer an idea of the complexity of the type of software required.

■ THE REAL-TIME ENVIRONMENT

Real-time applications place unique requirements on the processors and the software environment in which they run. Software and hardware for accelerating specific tasks, of minor significance in general-purpose computing, are crucial to the acceptable performance of a dedicated real-time processor. Moreover, the relationship between the code and the processor in the real-time environment is much more intimate than that in the general-purpose computing environment, where the processor can be made almost totally invisible to the user.

This intimate relationship has several immediate consequences. First, many control-related functions are interchangeable between hardware and software. Such basic functions as multitasking, task suspension, task switching, and intertask communication can be implemented in either software or hardware or both, depending on task priorities and timing constraints of the environment. For his specific application, a real-time system designer must make a decision whether the execution time for a given task is satisfactory when performed by code or requires further hardware acceleration.

Features that are inherently software-related, such as the ability to resume execution of an interrupted routine from an intermediate state ("reentrancy"), may be supported by hardware for greater memory and housekeeping efficiency. The same applies to other real-time functions normally under software control: concurrent execution support, interrupt handling, and task prioritization. High-speed real-time systems put additional constraints on the designers, requiring difficult compromises among minimum chip count, throughput, and code size.

The second consequence (which results from the first) is that the designer must have the ability to exercise full control over the processor, its memory, and its peripheral devices. To design a real-time system that executes all tasks within its specified time constraints, the designer must be able to account for every clock cycle of the task execution and to predict the processor's behavior under every conceivable circumstance, including during interrupts and task suspensions. These demands are inherent in real-time systems, which, by definition, have to be responsive to asynchronous task service requests of varying priority. A processor in which the execution of tasks can be fully predictable and observable is said to have good "observability."

This constraint limits the potential application of many modern RISC architectures in time-critical RT applications. The heavy use of pipelines and caches within these architectures can make them unsuitable for applications where the context-switching time must be fully predictable—and short. The unpredictable suspension times involved in flushing pipelines or reloading caches may be intolerable in many high-speed real-time applications (Small, 1988).

In addition, most general-purpose operating systems and high-level languages don't offer explicit access to the processor and often purposely hide its low-level operations (Thompson, 1987; Kernighan and Ritchie, 1978). Thus such systems and languages are somewhat useless for high-performance real-time applications. Standard Unix, the operating system of choice for most high-end computing, is inherently nonreentrant; for example, if x users request the same task under Unix, x copies of the task will be loaded into the memory, rather than one copy that, in the reentrant approach, serves all users. Moreover, the most touted feature of Unix and its language of choice, C, is its portability, which implies lack of processor-specific manipulation instructions. Yet the user's capability to directly manipulate the processor's data paths and registers is vital for the design of high-performance real-time systems.

In general, 90% of the execution time in a real-time application is spent executing 10% of the code. The most popular approach to developing real-time software has been to write that code in the assembly language of the chosen processor and the remaining 90% of the code in a high-level language like C. This approach has offered the best compromise between the need to execute the time-critical functions as fast as possible—by manipulating the hardware directly via assembly language—and the desire for code portability.

The mixture of assembly and high-level code executed by most real-time processors is the consequence of several deficiencies of C as a language for RT applications: C lacks support for interrupts, and it is fully disassociated from the hardware. In addition, although C does support recursion, most C compilers do not produce reentrant code.

A final difficulty in the development of a real-time system stems from the nonuniformity of the software development environment. Continuous switching among disjointed software tools, such as editors, compilers, assemblers, debuggers, and disassemblers, consumes the majority of the development time. Such a diversified environment impedes the development of predictable and repeatable code, not to mention the accurate timing analysis of completed, embedded software.

■ MARRYING HARDWARE WITH SOFTWARE

Current high-performance RT system designers are therefore inevitably faced with the following trade-offs during the design process:

- Form-factor limitations can dictate the trade-off of fast hardware for space-saving embedded firmware or software that executes more slowly.
- The schedule and cost of programming in high-performance assembly language

A close look at digital testers will give you a new Outlook for design verification.

The DAS 9200 from Tektronix is a high performance digital test system. But for even higher performance, take a close look at the T-100 from Outlook Technology.

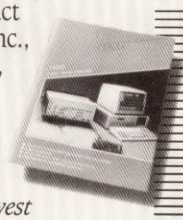
Both products have a maximum recording clock of 2 GHz. But only the T-100 uses intelligent sampling for precise data recording with 100 ps resolution. That's up to five times the resolution of the 9200... the difference between just seeing what happened and finding out why.

Both instruments find timing problems, but only the T-100 can trigger on and track down setup and hold time violations to save countless hours of searching for logic problems.

Both products can be used in automated setups to test boards and chips at high speed and high resolution. But the T-100 can perform up to ten times more tests per hour. And it can act as a 250 MHz pattern generator (stimulus), 250 MHz logic recorder (response), or both.

The T-100 also comes with a friendly human interface, including LogicProbe, a new utility program that makes setup and use faster and easier than ever before.

For a new outlook on digital testing, look into the T-100 family, with prices starting at just $15,000. Contact Outlook Technology, Inc., 200 E. Hacienda Ave., Campbell, CA (408) 374-2990.



Discover the newest measurement techniques in our 12-page brochure.

OUTLOOK

CIRCLE NUMBER 17

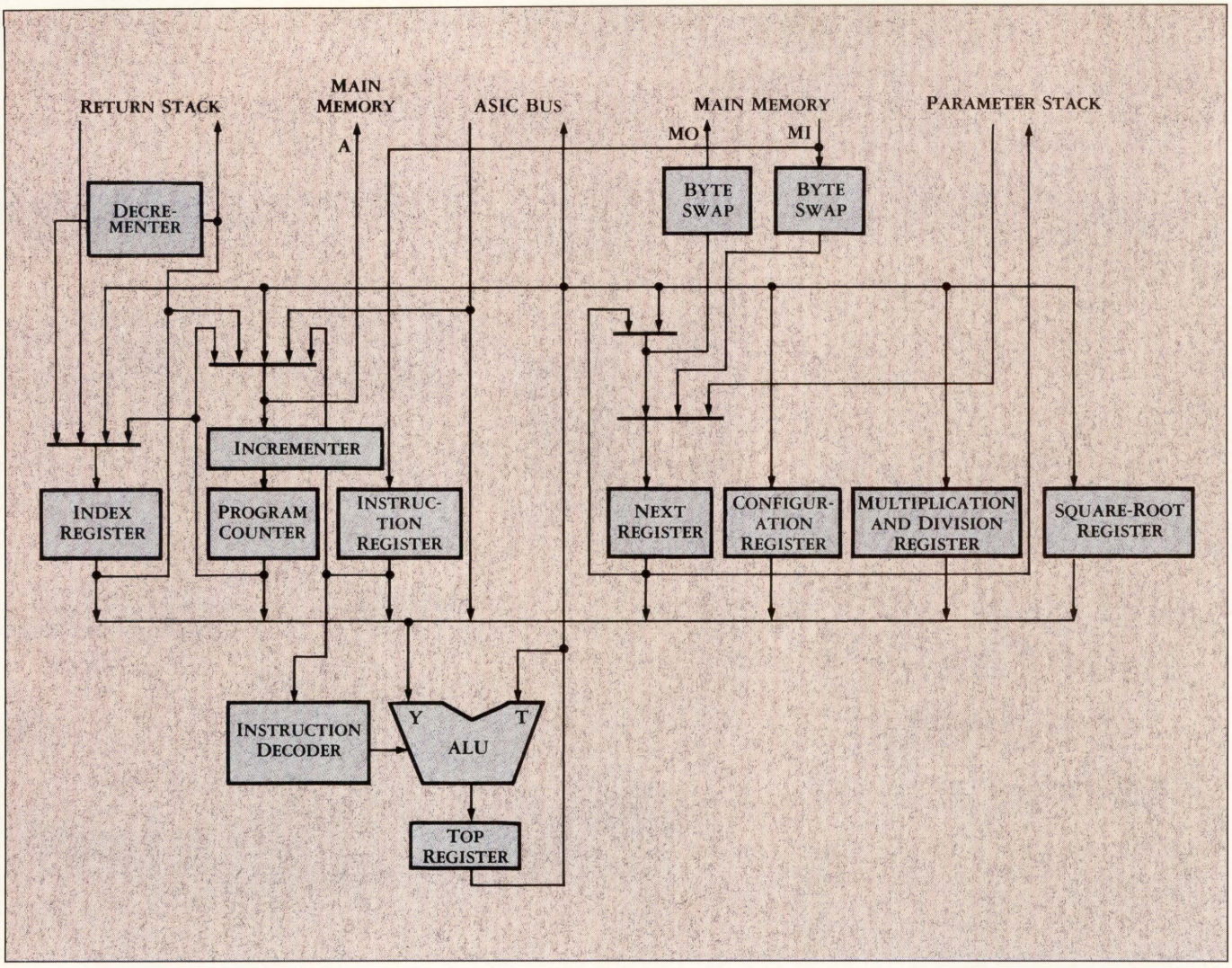


Figure 2. Within the RTX core, the registers, stacks, and ASIC bus can supply data for single-cycle instruction execution.

may be too great, necessitating the lower performance of less cumbersome high-level languages.

- The reduction of the versatility and flexibility of the software to make it more compact can cut requirements for memory size and address space.
- Compromising code reentrancy (often at the expense of memory space) can reduce the need for housekeeping functions.
- Increasing a system's response time can allow the designer to accommodate worst-case interrupt latencies and context-switching needs.
- The designer can limit program memory space and performance to improve the level of execution observability and to simplify the system response to unpredictable contingencies like pipeline flushing or cache misses.

In applications with critical time requirements, the attempts to make up for inefficiencies (such as long interrupt latencies or slow arithmetic operations) often fail, because of increased I/O bus congestion of the processor. As a result, the system becomes obsolete long before its time unless the processor manufacturer offers a hardware and software migration path, usually tied to newer technology and faster building blocks.

■ THE RTX SOLUTION

The RTX core processor addresses the issues discussed above in a variety of manners. First, the architecture is highly parallel and, most distinctively, is not pipelined. Because the RTX processor does not employ pipelines or internal cache memory, its cycle-to-cycle behavior is fully predictable, even during such events as interrupts or nested subroutine calls and branches. Deterministic execution allows the effects of hardware and software trade-offs to be measured and makes it easier to redesign hardware.

The architecture's strengths couldn't be exploited without an instruction set optimized for the engine's internal data flow. Each RTX machine instruction is equivalent to one or more instructions of the high-level language Forth. Forth's execution speed, efficiency, and compactness offer the designer a means of making trade-offs between hardware and software.

The RTX's hardware support for such crucial performance factors as multitasking and interrupt handling, as well as single-cycle subroutine calls and returns, eases the programmer's burden. Code development is also simplified by Forth's inherent reentrancy, a feature vital for producing compact and fast real-time multitasking code. Unlike conventional real-time operating systems, Forth provides its own complete operating environment. Moreover, the threaded code generated by Forth can be executed interactively for real-time applications development.

An RTX development system permits the system design team to partition the application between hardware and software. The system includes the processing engine and some prepackaged peripheral cells from the toolbox, such as

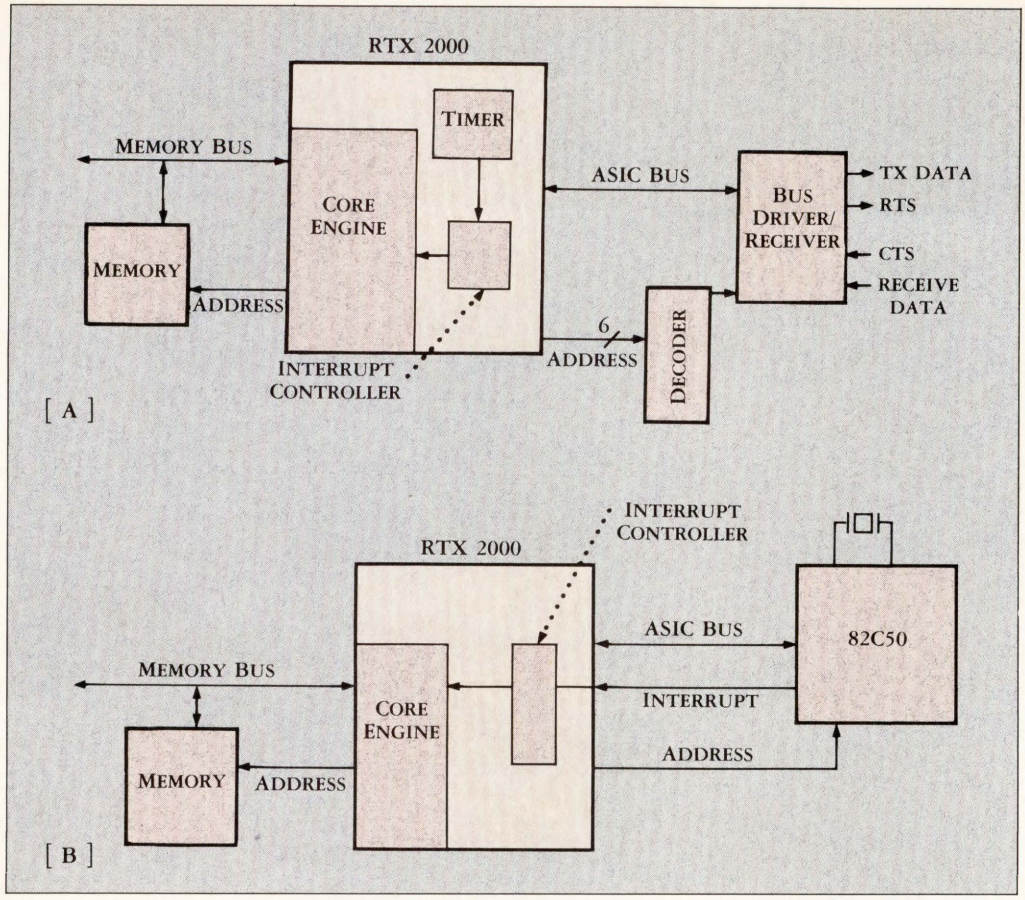


Figure 3. A software version of a UART (A) is more flexible and compact than a hardware version (B), although slower.

stack controllers, interrupt controller, 16 × 16-bit multipliers, plus stack and memory RAM. Also, the first RTX stand-alone real-time microcontroller, the RTX 2000, can be embedded into the development environment.

The system comes with a completely integrated software environment that is designed to produce highly optimized real-time executive kernels (or operating systems for nonembedded applications). The software package includes a Forth cross-compiler, debugger and monitor, disassembler, and host-interface drivers. Because of the interactive nature of Forth and the availability of an on-line disassembler, the system designer can trace the execution of the application code on a cycle-by-cycle basis.

Those designers wishing to port their existing C code can use an upcoming C cross-compiler, thereby providing an easy migration path for software originally written in C for other machines. In instances where part of the code was written in assembly language, it will need to be replaced by the RTX processor's own "machine language," Forth.

The core of the hardware part of the toolbox is a 16-bit processor engine with a number of unique features optimized for use in real-time control systems. Besides having a main 16-bit memory bus, the processor also supports two independent 16-bit stack-memory buses (Figure 1). The two-stack configuration is ideal for fast data manipulation, interrupt response, subroutine calls, and return handling. The data stack can operate directly on the data with implicit addressing, a capability that yields a 50% time savings compared with corresponding main-memory data operations.

The engine also has a dedicated I/O bus, called the ASIC bus. This bus is specifically designed to interface with the custom acceleration hardware that may be necessary for some applications. It accounts for much of the potential processing power of the engine, because it offers a direct path for the peripheral data to the 16-bit ALU, and from there to the "top" of the engine's stack (Figure 2; see also "Parallel Developments," p. 88). Also, the following peripheral devices that interface directly with the ASIC bus are under development: a high-speed memory element, a peripheral controller, and a support chip for multiprocessing applications that use the RTX engine.

The computational power of this 10-MIPS processor offers new solutions to the hardware/software partitioning problem. The ability of the processor to fetch data from the ASIC bus within a single processor cycle allows the designer to add specific ASIC chips to the ASIC bus of a stand-alone RTX processor, improving system performance without the large NREs of a custom processor IC. If the designer has chip count constraints, the processor and peripherals can be embedded in a single chip.

In the examples to follow, some hardware/software tradeoffs for implementing real-time systems, we compare. In general, the software approach will be based on the RTX 2000 as a stand-alone device. The performance of this device will be contrasted with the performance that could be obtained with a processor that works in conjunction with application-specific dedicated hardware.

■ Trade-Offs in Real-Time Systems

Context-switching time is a critical aspect of real-time processing. Designers can implement context switching using hardware or software structures, depending on the timing constraints of the system. For most conventional processors, the amount of time required by the processor to switch from one task to another is difficult to determine precisely. The overhead associated with context switching is directly related to the number of on-chip registers whose contents need to be stored, the status of pipelines within the processor, and the status of the cache memory. The overhead also depends on the type of instruction, and the stage of its execution, which is interrupted when the context switch is initiated.

Determining all these factors for complex microprocessors is very difficult; they have a lack of "low-level observability." Not only is it difficult to determine worst-case context-switching latency, but also trading a software-controlled context switch for a faster, hardware-assisted one is a formidable task.

For a microprocessor like the RTX 2000 with good low-level observability, the tradeoff is clear cut. The number of on-chip registers is relatively small, and there are no pipelines or cache memories to worry about. Moreover, most instructions execute in either one or two clock cycles. The three exceptions—MULTIPLY,

```
: SETBRG                    \This routine sets the baud
                            \rate and the interval for the
                            \timer_1_int routine
  VALUE TIMER1 G!           \Value is equal to 1/16
                            \of a bit period in clock
                            \cycles
  16 VALUE * 1BIT !         \Store the
                            \count requred for a bit
;                           \period in 1bit
: TIMER_1_INT               \This routine is used
                            \to sample the incoming
                            \data 16 times per bit
                            \and checks for a
                            \start bit
  UART G@                   \Read uart register
  0001 AND
  IF NODATA ELSE
  CHECK_START_BIT THEN
;
: NODATA                    \Tells us we do not have a start
                            \bit
  0 2NDTIME !
;
: WAIT1                     \Sets the timer to wait 1 bit
  1BIT TIMER2 G!            \period before interrupting again
;
: CHECK_START_BIT
  2NDTIME @ IF GETCHAR
  ELSE FFFF 2NDTIME G!
                            \Sets flag to show
                            \we had a start bit
  1BIT 2/ TIMER1 G!         \Sets the timer to
                            \interrupt 1/2 cycle later
                            \to recheck the start bit
  THEN
;
: GETCHAR                   \This routine does the
                            \character fetch from the
                            \uart register
  UART G@                   \Read the uart register
  COUNT @                   \Find what bit we should
                            \be receiving
  OF( 2*                    \shift left from 1 to 8 bits
                            \depending on count
  RXCHAR @ OR RXCHAR !
                            \Merge the data into the
                            \receive register
  COUNT @ 1 +               \Increment the bit count
  DUP COUNT !               \and store it
  8 - IF ENDCHAR            \Check to see if this
                            \is the last bit
  ELSE WAIT2
;
: WAIT2                     \Set the timer2 interrupt
                            \interval
  1BIT TIMER2 G@
;
: ENDCHAR                   \Finish up the character
  0 COUNT !                 \Reset the count
  FFDF INTMASK G!           \Disable the interrupt
                            \until a new start bit is detected
;
: TRANSWORD                 \Transmit the character
                            \from the top of the stack
  2* 1 + 2* 1* +            \Add 2 stop bits to the word
  TWORD !                   \Store the word to be
                            \transmitted
  UART G@                   \Check the cts bit for a 0
  0040 AND IF
  NOT CLEAR ELSE TRANSMIT
  THEN
;
: TRANSMIT                  \Transmit the word on the
  0 UART G!                 \top of the data stack
  FFFF INTMASK G!           \Enable the
                            \interrupt bit for timer 3
  1BIT WAIT3                \Set the timer and begin the
                            \start bit transmission
  0 TCOUNT !
;
: TRANSMIT_INT
  TWORD @ DUP               \Fetch the word to be
                            \transmitted
  UART G!                   \Transmit a bit
  2* TWORD!                 \Store the next
                            \bit into the
                            \tword register
  TCOUNT @ 1 + 9 =          \Disable interrupt
  IF DISABLE                \Rearm the timer
  ELSE WAIT3                \for another interrupt
;
```

Figure 4. The code for a software version of a UART requires more machine cycles than does the code that drives a discrete UART (Figure 5), but its execution occupies only 1% of the processor's available time.

DIVIDE, and SQUARE-ROOT—can't be interrupted, so they need not be considered.

If the software-controlled context switch is not fast enough in the RTX 2000, the designer can accelerate it by, for example, providing a last-in, first-out (LIFO) memory on the ASIC bus for each task. In such a configuration, the contents of the internal registers would be dumped directly over the ASIC bus to the appropriate LIFO for the task being interrupted. No explicit memory addresses need be generated to store the data associated with that task. This scheme uses 33% fewer clock cycles than a software-based context switch, which must generate those addresses.

ASIC bus LIFOs can also accelerate intertask message passing (called "mailboxing"), another common characteristic of real-time systems. Ordinarily, the software assigns a segment of data memory to serve as a mailbox for all tasks ("public mailbox"), or it splits a series of segments among a group of tasks, creating "private mailboxes." Tasks can put messages for other active tasks in message stacks or place them in the private mailboxes of inactive tasks.

In either case, software-based mailboxes residing in main memory are difficult to design and can reduce the overall task execution rate of the processor. If LIFOs on the ASIC bus are used as hardware mailboxes tasks can read and write messages without addressing data memory. This approach eliminates the overhead of generating addresses and accessing a memory bus.

■ IMPLEMENTING A UART

For I/O functions, a 10-MHz RTX 2000 can be used to emulate a UART, or it can interact with a hardware UART. For the first option (Figure 3A), the processor needs a chip-select decoder and a 4-bit latching bus driver. The chip-select decoders are usually present in most systems, and the bus drivers may be incorporated in unused bits of an existing parallel port.

The state-of-the-art in gate array design is a Silvar-Lisco product. Again!

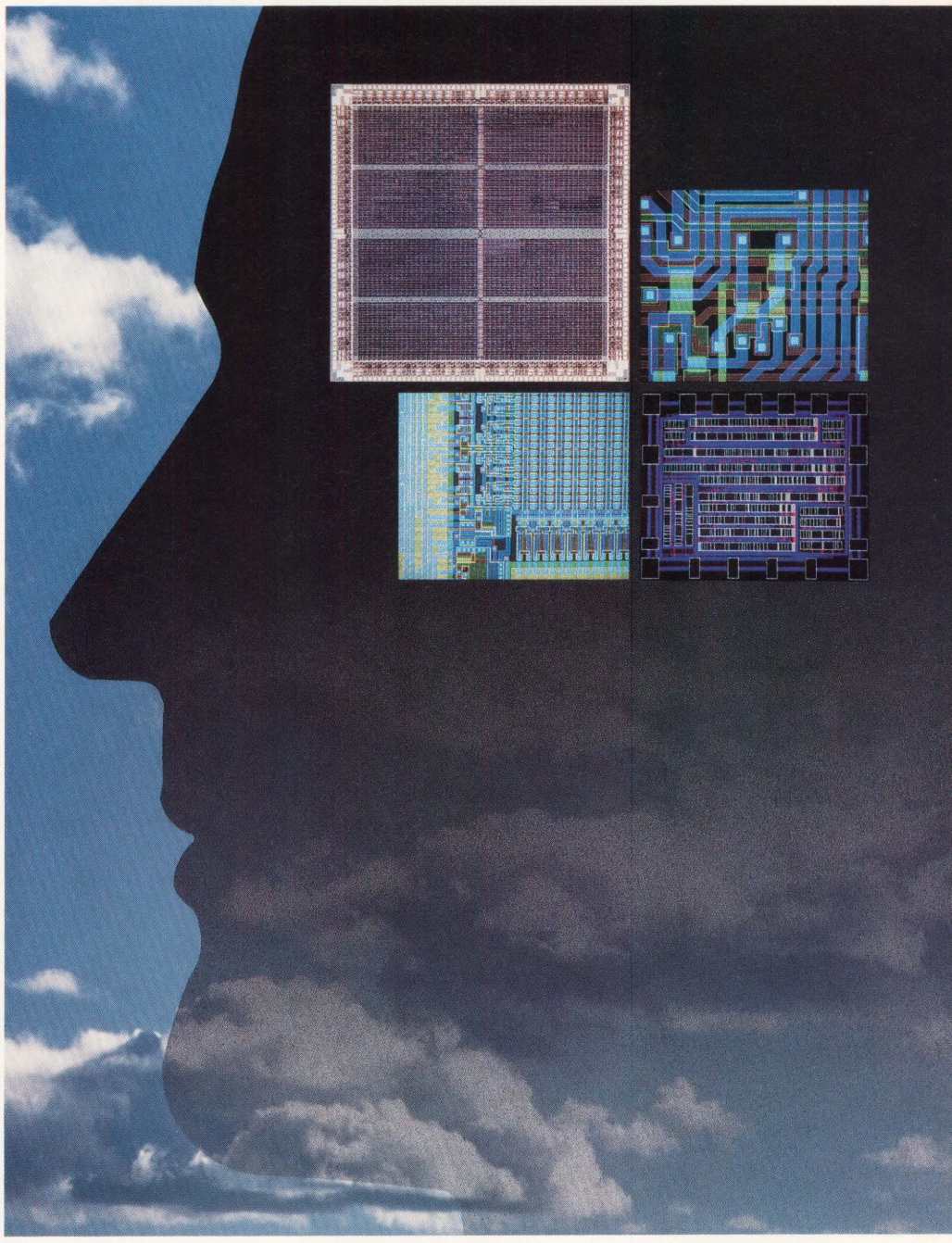
Introducing AVANT GARDS.™ For VLSI sea-of-gates arrays. With 150,000 gates on a chip. And more to come.

Until now, the state-of-the-art in gate array design systems was Silvar-Lisco's GARDS.™ But not anymore.

AVANT GARDS,™ Advanced Gate Array Design System can place and route over 150,000 gates on a single chip using sea-of-gates and three-layer designs. With gate utilization as high as 90%. Adhering to your timing specifications. In less than a week. And supported by the most popular foundries.

That's the *new* state-of-the-art in gate array design. And it's also a Silvar-Lisco product.

But AVANT GARDS™ isn't alone. Silvar-Lisco's cell-based and full custom chip layout software are state-of-the-art as well. And they get better all the time because we improve them all the time. All of them fully integrated with Silvar-Lisco's full line of CAE software that takes you from design capture through mask data preparation. All production proven in thousands of designs.



Silvar-Lisco Offices

Corporate Headquarters
1080 Marsh Road
Menlo Park, CA 94025
Telephone: (415) 324-0700

European Headquarters
Brussels, Belgium
Telephone: 32-16-200016

Far East Headquarters
Nihon Silvar-Lisco
Tokyo, Japan
Telephone: 81-3-449-5831

SILVAR-LISCO

CAE Software products for electronic design

Call or write to get detailed literature about the IC layout products that have made Silvar-Lisco the world's leading CAE software company.

CIRCLE NUMBER 18

```
: INIT
      PARAM1 UCMD G!        \Initialize parameters
      PARAM2 UCMD G!
      UDATA G@ DROP         \Clear data port
;
: @STATUS ( -- n)           \Read status/command port
      UCMD G@ ;
;
: RECEIVE ( -- n)           \Poll for character
      BEGIN @STATUS RRDY    \Wait for RRDY = 1
        AND UNTIL
      UDATA G@ 007F AND     \Read character
;
: TRANSMIT ( n -- )
      BEGIN @STATUS         \Wait for transmit ready
        TXRDY AND UNTIL
      UDATA G!
;
```

Figure 5. A discrete UART requires far less code the a software version (Figure 4).

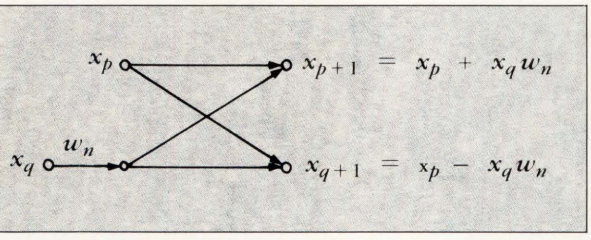


Figure 6. The butterfly operation of a fast Fourier transform requires four multiplications and two additions.

In this configuration, the remaining portions of the UART function are implemented completely in software, with a hardware assist from the RTX timers and interrupt controller. The software required to implement the UART is shown in Figure 4.

The interrupt controller and the timers keep track of when to check for start bits and when to receive the next data bit, enabling the processor to perform other tasks between incoming data bits. This particular implementation would occupy about 100 words of program storage to implement and use three of the timers aboard the RTX 2000.

Use of a hardware UART like the 82C50 (Figure 3B) would require less code (Figure 5). The hardware UART would also simplify the implementation of such features as parity checking and overrun detection. However, it will also cost about 1,500 gates of logic in an ASIC application or an extra package as a discrete device. The extra 1500 gates could force the the user to sacrifice approximately 4K bits worth of usable RAM space or 32K bits of ROM space on the chip.

Because the software implementation would take four to five times more processor time than the hardware implementation, a circuit designer would chose the hardware option as the best possible approach. However, even with the extra overhead required, the processor is only actively executing the UART code less than 1% of the time available for processing. Therefore, even with the added overhead of the software, the processor's performance is essentially unimpaired and a hardware implementation (from the point of view of a software programmer) is virtually useless. The designer should use the area that would have been dedicated to the UART in a manner that is more useful to the overall system. The user could add 4K bits of RAM, eliminate the hardware UART, and not affect his processor price a penny.

He could also use the UART code more effectively than the hardware version. It would be a much simpler task to integrate the hardware and software for a chip using the RTX core (which can implement eight UART functions on a single chip) than to integrate eight separate 82C50 macrocells. In addition, the overall die size would be smaller when the software approach is taken. To prevent system obsolescence, the designer would be able to use the software-configured device to implement a variable number of asynchronous channels merely by reprogramming its I/O interface. If the designer had chosen the hardware version of the UART, he would be stuck with his original choice.

■ AN FFT EXAMPLE

Many real-time systems in signal-processing applications must compute fast Fourier transforms (FFTs). The following example will show how the butterfly arithmetic operations used to implement the FFT can be implemented in FORTH with the RTX 2000.

Because the processor has a 16 × 16-bit hardware multiplier, it can perform the multiplication of two complex numbers in 30 processor clock cycle cycles, a figure 20 to 50 times faster than most conventional microcontrollers. For example,

Parallel Developments

Designers can use the ASIC bus on the RTX 2000 to implement hardware acceleration for specific operations. Through direct access to the ALU and stacks in the RTX 2000 core, circuits on the ASIC bus can access data as easily as RTX 2000 execution units. Being independently implemented, they can execute concurrently with the other execution units. This method of equal access and independent operation implements one form of fine-grain parallelism within the architecture.

Such parallelism (the same phenomenon that the designer of the Am29000, Mike Johnson, is developing at Stanford—see People, this issue, p. 14) will determine the performance of the next generation of microprocessors. For example, the new RISC architectures from Motorola Inc. (see *VLSI Systems Design*, May 1988, p. 24), and Intel Corp. (see this issue, p. 18) are built around a register stack with a "register" or "operand" bus that transports data to execution units in their architectures. These execution units have equal access to the data in the registers, and the flow of data is coordinated by scoreboarding techniques that keep track of the completion of operations (and hence accessibility of results).

The RTX 2000 does not have a scoreboarding mechanism, so designers of extensions through the ASIC bus must implement some method of control for the flow of data. The determinism of the architecture (no assembly language or pipelines) makes such implementation straightforward. Because designers of real-time systems need to oversee cycle-by-cycle operation anyway, and in many cases have peculiar system requirements, the freedom to implement any type of control may be welcome. ■

—David Smith

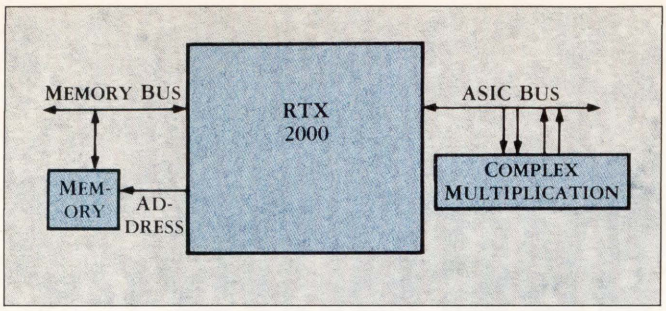


Figure 7. Adding hardware that performs complex multiplication to the ASIC bus speeds fast Fourier transform execution by roughly 10× over a software version.

the 80C196 microcontroller requires 2.3 ms to perform a single 16 × 16 multiplication, whereas the RTX 2000 requires only 100 ns.

In this particular example, the FFT is computed with complex numbers. If the FFT were performed with only real numbers (as is done in some benchmarks), the performance of the device would improve. However, because the most popular requirements for FFTs include recovering phase information as well as amplitude information, this example uses complex variables. As with the UART example, the software-only version of the performance is rated on the RTX 2000 without additional hardware tied to the ASIC bus.

The calculations required for a single butterfly operation, the fundamental calculation in the FFT, are shown in Figure 6. In the figure, the variables x_p, x_q, and w_n each contain both a real and an imaginary part. To calculate the value for $x_q w_n$ therefore requires one complex multiplication (instead of a one-cycle real-number multiplication). The complex multiplication in turn requires four multiplications and two additions. At each successive point in the FFT, the previous value for x_n is used to calculate the next value in the FFT.

The value of a fast multiplication routine is apparent when the complex-number multiplication routine is used in an application to solve a butterfly operation. For the example shown, the code calculates the values of x_{p+1} and x_{q+1} in 80 cycles, and an 8-by-8 FFT can be accomplished in 1,670 cycles, for a rate of 6,000 FFTs/s at a clock frequency of 10 MHz. A 1,024-point complex FFT would take 20 to 30 ms with the processor as a stand-alone device—substantially faster than most general-purpose microcontrollers or microprocessors, but not as fast as dedicated digital signal processors.

To accelerate the FFT to greater throughput, the designer can assign some of the functions performed in software to custom logic that is tied to the processor's ASIC bus (Figure 7). In this configuration, the ASIC bus connects the RTX core to hardware that performs the complex multiplication. The RTX core would load the real and imaginary portions of the numbers to be multiplied and then read the hardware to obtain the results.

This implementation would allow the software to replace a complex multiplication routine with one that merely writes and reads the ASIC bus. The complex multiplication can be performed in this manner in only six cycles, providing 10 times the multiplication performance and improving the speed of the FFT substantially. With this hardware/software partition, a 1,024-point FFT could be calculated in 10 ms. This time includes the total overhead required for the bit reversal, memory fetching, and storing. In this way, by adding application-specific logic to the RTX's ASIC bus, a designer can achieve performance meeting or exceeding that of many signal processors.

As these examples have indicated, it is quite possible to improve the performance of systems by incorporating the proper hardware to assist the application. Ultimately, the system performance requirements will become the mechanism for driving the decision toward software or hardware approaches. The ASIC designer must balance the cost of the additional hardware, in terms of die area, gate count, and final packaged cost, with the hardware's superior performance. ■

REFERENCES

KERNIGHAN, B., and RITCHIE, D. 1978. *C Programming Language*, Prentice Hall, Englewood Cliffs, N.J.

RATHER, E. 1985. "Forth Takes Aim at Real-Time Applications," *Computer Design*, October 15, 1985.

SMALL, C. 1988. "Real-Time Operating Systems," *EDN*, January 7, 1988.

THOMPSON, E. 1987. "Microcontroller Programming in C," *Computer Design*, November 15, 1987.

WILSON, R. 1987. "Embedded Systems Manipulate Distributed Tasks," *Computer Design*, September 1, 1987.

ABOUT THE AUTHORS

CHRIS W. MALINOWSKI *is a senior scientist in Harris's semiconductor research and development department., where he is responsible for advanced architecture development. He holds an MS degree in nuclear electronics and a PhD in solid-state physics from Warsaw Technical university in Poland.*

PETER S. DANILE *is a section head of semicustom design at Harris Semicondutor. Before joining Harris in 1981, he worked for Northern Telecom and Motorola Communications. A graduate of the University of Southern Florida, he holds an MSE degree from Florida Atlantic University.*

Optoelectronic ICs

A New Era in Data Communications and Computing?

RODERIC BERESFORD, EDITOR EMERITUS

THE GLOBAL TELEPHONE network goes lightwave. Computer designers are harnessing optical beams for low-noise interconnections. Across the industry, electronics plus photonics looks like a good marriage. What will come of it and why should you care?

In some not too distant but not yet specified future era, optoelectronic ICs will lower costs and improve reliability in fiber-optic local-area networks and long-haul communications. Optoelectronic ICs and optical communications may also bring the benefits of extremely low noise and high throughput to intermodule, interboard, and interchip signaling within computers of all types. No one yet knows all the new kinds of chips and applications that might ultimately be made possible by the large-scale integration of optics and electronics. What the experts do know is that basic optoelectronic IC technology still faces considerable challenges.

Optoelectronic integration is the next logical step in a chain of development reaching back to the early 1970s. The demonstration of low-loss propagation in optical fibers was the spur that led technologists to improve laser diode material, eventually to achieve reliable continuous-wave operation at room temperature. The next step is to increase modulation frequencies to better exploit the bandwidth of optical fiber.

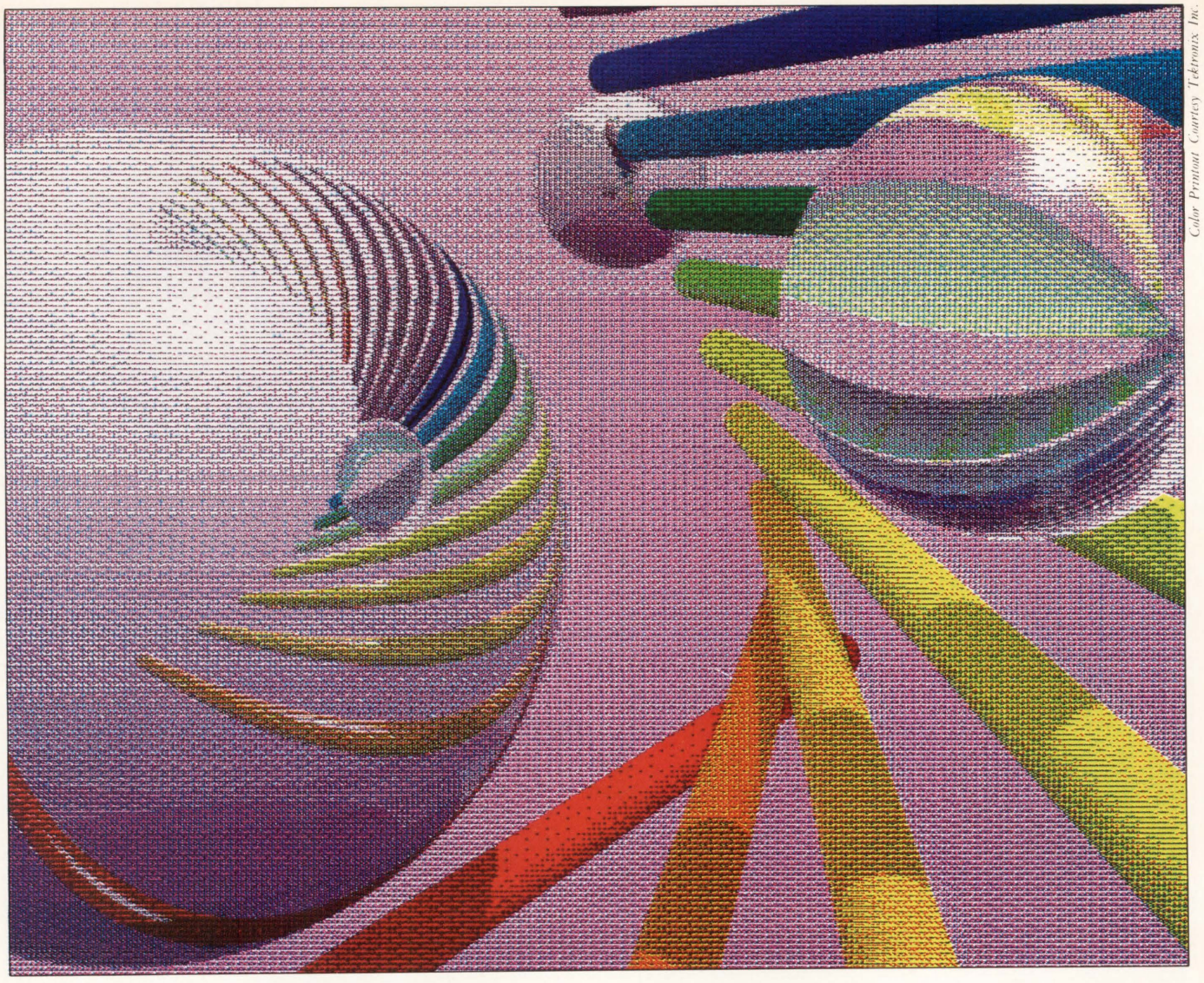
Electromagnetic energy propagating in silica fibers at a 1.55-μm wavelength has a frequency of about 115 THz; the low-loss band at this frequency has a width of about 15 THz. Only a small fraction of this enormous bandwidth is utilized in commercial telecommunications—a fraction dictated by the much lower operating frequencies of the modulating circuits that transmit information on the optical carrier. With discrete lasers and photodetectors, parasitic inductance and capacitance may prevent designers from achieving the intrinsic performance limits of the optical devices. Integration offers a solution. In any case, since the network is going lightwave, optics eventually will be invading almost every electronic system, even if only at a connection point with the outside world. Naturally, the question arises whether optical devices can be integrated with electronic devices.

The term *optoelectronic IC* (OEIC) is coming to mean a semiconductor integrated circuit that has both electronic circuits and optical devices—lasers or photodetectors, or both, as well as passive optical and perhaps electro-optic components—fabricated on a common substrate. In the related field of *integrated optics*, researchers are exploiting nonlinear interactions in semiconductors and other optical solids in attempts to use inherently fast optical interactions in all-optical computers and switching networks. All-optical switching and computing may involve not only electro-optic and passive optical components on ICs, but also other important areas, namely, optical-fiber components and holography.

Both OEICs and integrated optics are burgeoning research fields. At a symposium entitled Optoelectronic Devices for Future Telecommunications Systems sponsored by the National Science Foundation and held in early April at Columbia University in New York City, Fred J. Leonberger of United Technologies Research Center said that an integrated photonic device—a gallium arsenide substrate hosting electronic devices, electro-optic modulators, and interfaces for optical fibers—has constituted "the holy grail of this business for the last 10 years." Along the way to that ultimate goal are a number of potentially interesting applications. Here, the focus is only on OEICs, and we consider the problems these chips might solve, the devices required, the technologies under development, and some of the problems that must be overcome for OEICs to be able to make a contribution to system design. Other topics, including integrated optics and the all-optical computer, are left to future articles.

■ WHAT ROLE FOR OEICS?

Optical technology is rapidly penetrating electronic systems. By the year 2000,



Color Printout Courtesy Tektronix Inc.

the long-distance network will be essentially 100% lightwave. Sometime in the first part of the next century, there will be no copper wire left anywhere in the entire telephone transmission network. Other applications of optical signaling—in isolators, motion sensors, and optical disks—continue to grow.

All these applications are making use of discrete light sources and photodetectors. For approximately the past five years, several Japanese companies, preeminently Fujitsu Ltd., have been working on integrating lasers with their transistor drivers, and likewise photodetectors with their transistor amplifiers. In both cases, the primary motivation has been to eliminate the parasitic reactance of gold bond wires, which, with inductances of about 1 nH and capacitances of about 1 pF, limit frequency responses in the gigahertz range. Secondary motivations are to equip multiple channels economically and reliably. The resulting transmitter ICs and receiver ICs are rapidly maturing as optoelectronic components. For example, at the Conference on Lasers and Electro Optics in April, researchers N. Suzuki et al. from Toshiba Corp. displayed a landmark 5-Gb/s transmitter IC based on InGaAsP.

Such demonstrations, however, do not satisfy the "urgent" needs for practical devices for immediate applications. Also at CLEO, Osamu Wada's group from Fujitsu showed how they flip-chip–bond a photodiode detector directly to a GaAs receiver IC—a hybrid approach that eliminates parasitic capacitance effects. Their photodiode's bandwidth of 19 GHz is limited by the carrier transit time only (Makiuchi et al., 1988).

In long-distance telephone transmission, sensitivity and bandwidth are critical, so demonstration projects emphasize those specifications. The first big challenge for optical devices integrated with electronic devices is to achieve parity or superiority compared to hybrid implementations using discrete optical devices.

The second big challenge is that both laser sources and high-gain photoreceivers are power-hungry components that will cause heating problems on chip. Integrated laser sources face the additional problem that both terminals must be contacted from the top surface of the chip. Because of these current-handling constraints, low-threshold lasers are critical for OEICs.

Although the existing applications for experimental OEICs are in long-haul networks, some engineers envision optoelectronic ICs as an important low-cost component for the day when lightwave technology makes it to the congested local office switches (Tomlinson and Brackett, 1987). These chips will likely comprise one or a few lasers, with drive electronics (bias and feedback stabilization) plus multiplexers to merge lower-rate electronic signal channels.

The desire to make best use of the available fiber bandwidth argues for a network employing wavelength multiplexing. The bandwidth in the 1.55-μm window corresponds to about 120 nm, so that laser sources stable to a few nanometers would permit several tens of channels at different wavelengths to use a single fiber.

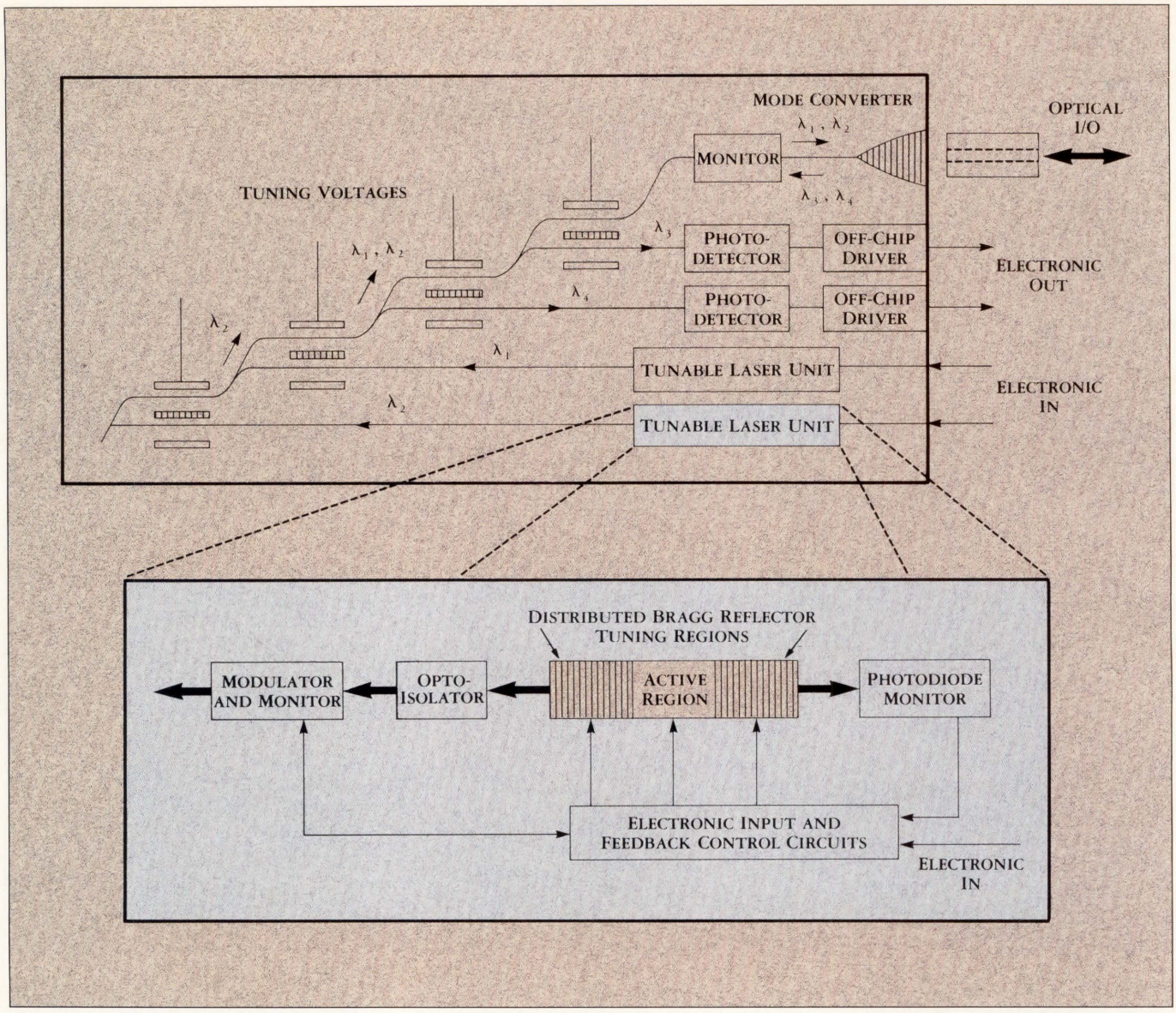


Figure 1. Future multichannel optoelectronic transceiver for a wavelength-multiplexed optical fiber transmission system, after Suematsu and Arai (1987).

A conceptual diagram of such an application is shown in Figure 1. This photonic integrated circuit of the future implements a multichannel optical transceiver, containing tunable laser diode sources, optical combiners, photodetectors, optical I/O, and electronic control and I/O (Suematsu and Arai, 1987). Some of the components are discussed further below.

A related concept is the use of optical coherent detection to select the frequency at the receiving end. In this case, each node broadcasts on a unique frequency to all other nodes; a local oscillator (laser) signal is mixed with the received signal to decode the desired channel.

Another obvious slot for optoelectronic ICs is in a crossbar switch. Here, optical-fiber inputs and outputs and a high-speed electronic switching matrix are required (Figure 2). The table shows that current performance of GaAs crossbar switches is at up to 2 Gb/s. Recently, OEIC transmitters and receivers were used in such a demonstration at 2 Gb/s (Wada, 1987). Complete integration of this application would confront power dissipation problems, based on current performance levels of the optical components.

There are potential applications in computing as well, where optical signals could provide high-bandwidth low-interference communications among chips and circuit boards. Optical interconnect, as this application is usually called, is being pursued in diverse forms (Fossum, 1987). Some schemes advocate free-space propagation of the optical beams, using holograms or mirrors to effect connections. Alternatives are to use optical fibers, or perhaps waveguides integrated in the circuit board itself, with direct mechanical connections to the circuit elements. Some envision the light sources on chip; others see a hybrid approach, with OEICs mounted close to VLSI silicon circuits. Still others want to use "remote" lasers, with a small number of primary beams split and piped to many chips, where modulators impress their signals on the beams, which are then routed as required. Depending on the variant selected, this application requires laser sources, photodetectors, and modulators integrated with drive electronics and perhaps, with digital logic as well. Figure 3 gives an example.

In even further out applications, researchers are planning all-optical computers that rely on transformations of many parallel and spatially separated beams (Sawchuck and Strand, 1984). Two-dimensional arrays of lasers, detectors, and light modulators would be required for signal processing and conversion. These architectures, involving hundreds or thousands of parallel channels under electronic control, are certainly impractical without the density of optoelectronic ICs (or analogous all-optical devices). Whether they

or...

The choice is really quite simple. Dozens of different PLDs expensively stocked to meet every need. Or the generic GAL® family of programmable high-performance logic devices.

Reprogrammable CMOS.

Behind this family is our Generic Array Logic (GAL) architecture. First developed and produced by Lattice, this architecture lets one GAL device replace all common 20 or 24 pin PAL® devices.

Fabricated with Lattice's proprietary E²CMOS™ technology, GAL devices give you bipolar speed and lower power. Half-power parts are available with maximum gate delays of 15ns. Quarter-power delays are 25ns. Both reduce heat without loss of performance.

Gone are discarded fuse-link parts caused by misprogramming. Gone is the need for expensive window packaging. Gone are long UVerase cycles. Instead, you simply plug your GAL device back into the programmer and erase and reprogram in less than a second. Design changes are simple and easy, too.

Inventory reductions.

From a manufacturing and management perspective, the GAL family offers even more. Now PAL and other PLD inventories can be dramatically reduced. At the same time, design alternatives multiply.

Because GAL devices are reprogrammable, they are also reuseable. And last-minute design changes mean simple reprogramming rather than costly replacement.

100% tested.

Unlike other PLDs, GAL devices are 100% tested for optimum system quality. There is simply no need to overstock in anticipation of a high failure rate.

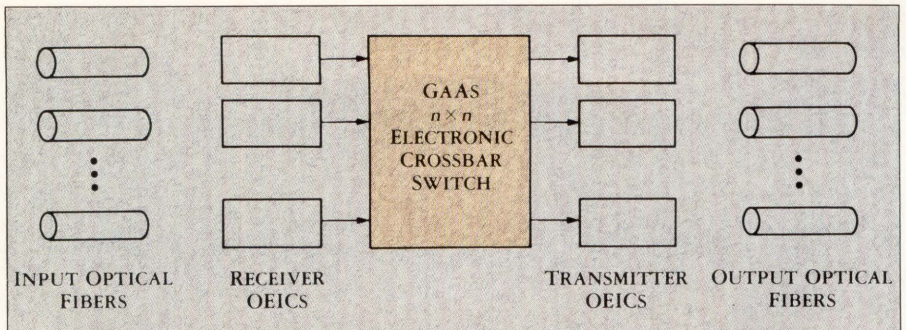
So forget about that pile of "Eithers." And learn more about the Lattice "Or." To find out how GAL devices fit all of your PLD needs, ask for a free copy of the Lattice GAL Data Book today.

Lattice Semiconductor Corporation™

5555 N.E. Moore Ct. • Hillsboro, OR. 97124

The GAL Revolution Starts Here℠

CIRCLE NUMBER 35



COMPANY	PER-CHANNEL THROUGHPUT (Gb/s)	CROSSPOINTS	TOTAL THROUGHPUT (Gb/s)	POWER (W)
NEC	1.2	8×4	9.6	2.4
FUJITSU	2.0	4×4	8	1.1

Figure 2. Hybrid optoelectronic crossbar switch configuration. The table gives parameters for GaAs IC implementations of the switching matrix. See Wada (1987).

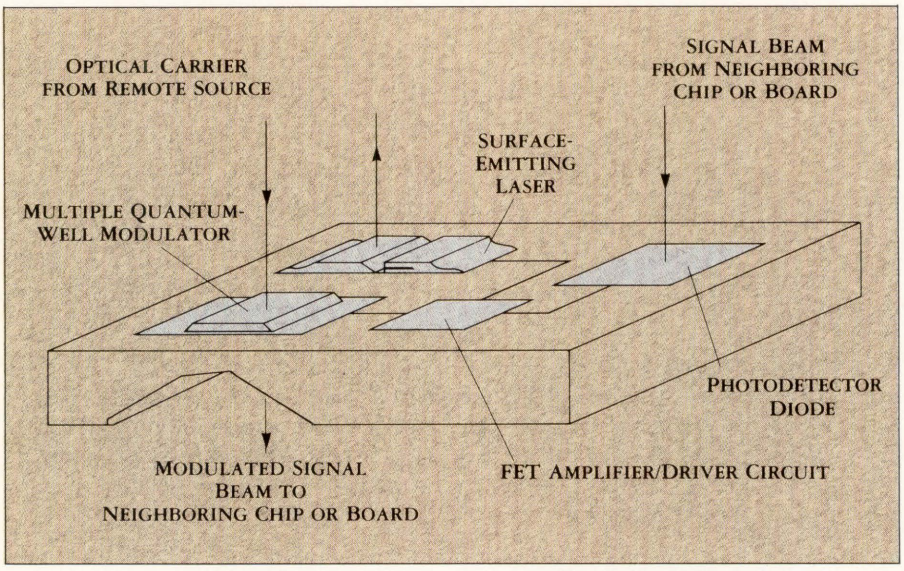


Figure 3. OEIC components for managing optical interconnect in a computer. Optical beams propagate chip-to-chip in fibers or in free space.

are practical even with such ICs is also being debated.

As described above, then, the key devices of future optoelectronic ICs include laser sources, photodetectors, light modulators, and couplers, as well as conventional electronic components: transistors, passive components, and interconnections. How rapidly such applications can be demonstrated depends largely on the development of the underlying technology. Material quality and the difficulties in integrating optical with electronic devices appear to be the gating factors now. Before considering fabrication issues in more detail, however, it is useful to examine some of the device structures required.

■ Devices for Optoelectronic Applications

Of the necessary devices, the electronic ones of course are the most highly developed today. In addition, integrated photodetectors are generally considered to be well advanced, although their specifications do not always meet those of discrete devices. Here, instead, we will focus some attention on more critical or more experimental components: the laser, the passive optical components, and quantum-well modulators.

First and foremost is the laser diode, which has evolved to a high and acronymous art. Even discrete diodes today are being fabricated with an "integrated optics" approach, meaning that the complete device is not just a lasing cavity, but also includes additional passive or active optical devices, such as distributed Bragg reflectors and current tuning. An example of a late model laser diode exhibiting this approach is shown in Figure 4 (Tohmori et al., 1986). This "BIG-DBR-DSM," for bundle-integrated-guide distributed-Bragg-reflector dynamic single-mode laser, can be wavelength-tuned by injecting current into the Bragg reflector regions on either end of the laser cavity.

The corrugated surface of the waveguides in these regions introduces periodic variation of the effective index of refraction of the guide. The variation acts to couple modes of the waveguide; for a particular wavelength, dependent on the period of the grating, the forward propagating optical wave is coupled to the backward propagating wave of the same mode, giving reflection. Injecting free carriers into the DBR region changes the index of refraction, in turn changing the wavelength of the mode that satisfies the Bragg reflection condition, and the diode lases at a different wavelength.

Passive components like DBRs are particularly important because they can substitute for the cleaved-facet mirrors of discrete laser diodes. More generally, if future optoelectronic ICs are to integrate more than a single optical transceiver, a variety of passive optical components will be required. Optical waveguides of several configurations have been fabricated in semiconductor substrates. Losses are usually high compared with "optical" materials like lithium niobate. Other unsolved problems include how to make mirrors, so that optical signals can take sharp turns on the chip; how to couple waveguides; and how to make optical isolators.

Passive optical components are entirely obviated in some schemes, in which all of the optical propagation is orthogonal to the IC surface. As mentioned above, some in the research community favor the concept of an off-chip light source, with optoelectronic ICs merely modulating and detecting optical signals propagated to the IC either through fiber waveguides or free space. In these scenarios, a multiple-quantum-well modulator is almost certainly going to be a key on-chip component.

Furthermore, most researchers believe that only quantum well lasers will be able to achieve the low threshold currents required for optoelectronic integration.

As currently fabricated, a quantum well is simply a layer of GaAs sandwiched between layers of AlGaAs, which, because of its wider bandgap, provides confining barriers to carriers in the well. Figure 5A pictures electron and hole distributions superimposed on an electron energy band diagram of a quantum well. When light hits this structure, carriers are excited across the absorption gap. If the well is biased with an electric field, the energy bands are tilted as shown in Figure 5B; the absorption gap is reduced (shifted) and weakened because of the spatial separation

META-SOFTWARE

Has Taken A Bold, New Step...

From design to silicon, Meta-Software provides the best circuit simulation tools in the industry. Meta has recently repositioned its product offering to better meet the demanding needs of today's design engineers.

For convenient, one-stop shopping, the innovators at Meta-Software now offer:

HSPICE: The industry's leading analog circuit simulator for integrated and discrete circuit design. HSPICE includes a multi-target optimizer supporting all SPICE and HSPICE models.

RADSPICE: HSPICE plus radiation effects modeling provided by SAIC. Effects include total dose, ionizing photocurrents and neutron radiation.

HSPLOT: Meta's high-resolution interactive graphics post- processor for HSPICE and RADSPICE. HSPLOT provides graphic terminal and hardcopy support for a wide range of display services.

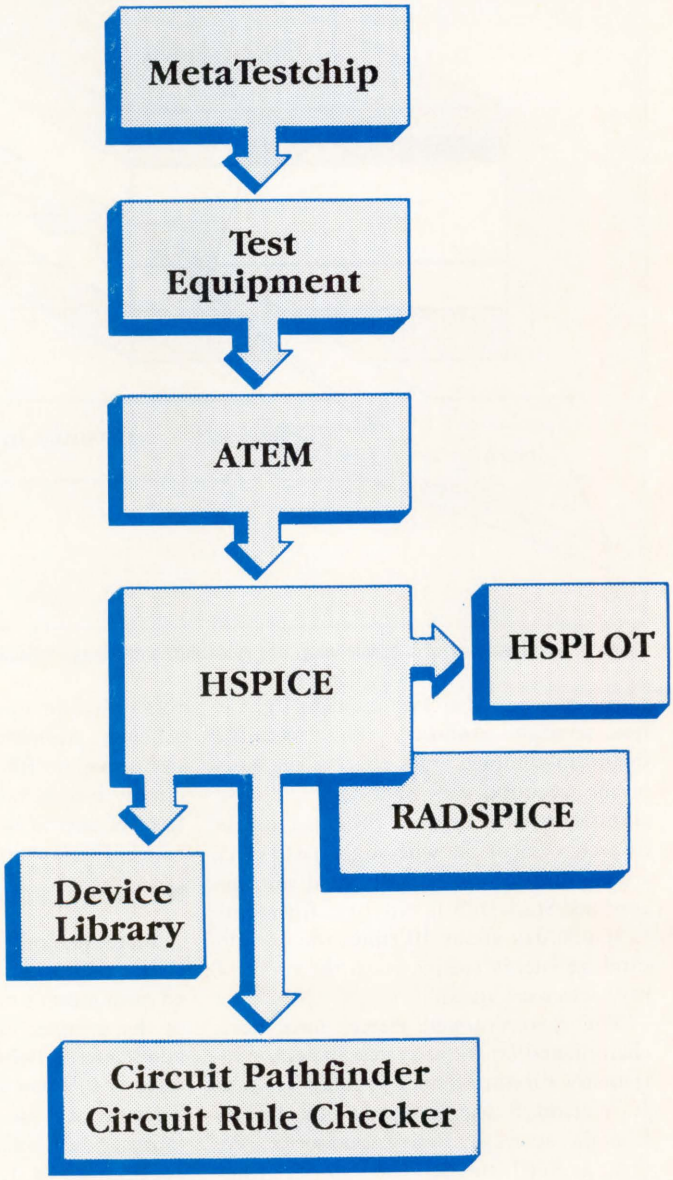
Discrete Device Library (DDL): Includes more than 750 models of discrete components for use with HSPICE. Included are BJT, MOSFET, HEXFET, Diode, JFET, Op Amp. Comparator, A/D converter, D/A converter, Timer and SCR models.

ATEM: Meta's lab test equipment interface program which creates measured data files and initial guesses for optimization features of HSPICE. ATEM provides an easy method for scanning transistor characteristics and selecting devices for full optimization.

MetaTestchip™: A test chip tailored to customer's design rules, providing all structures necessary for complete, automated process and device characterization.

Lab Services: Products and services for discrete device and wafer level characterization.

Circuit PathFinder (CPF): A path timing analysis program, providing full chip analysis at interactive speed. Circuit Rule Checker module locates slow nodes and gates, and a variety of circuit-configuration rule violations.



Meta-Software provides all essential support services. For answers to technical questions from experienced engineers, please call Meta's toll-free HOTLINE SUPPORT number: (800) 346-5953

HOTLINE SUPPORT is also available at our main number: (800) 371-5100

Get your circuit simulation and characterization problems resolved quickly and professionally. Call Meta today!

Meta-Software, Inc. • 50 Curtner Avenue, Suite 16 • Campbell, CA 95008
Phone (408) 371-5100 • FAX (408) 371-5638 • TLX 910-350-4928

CIRCLE NUMBER 20

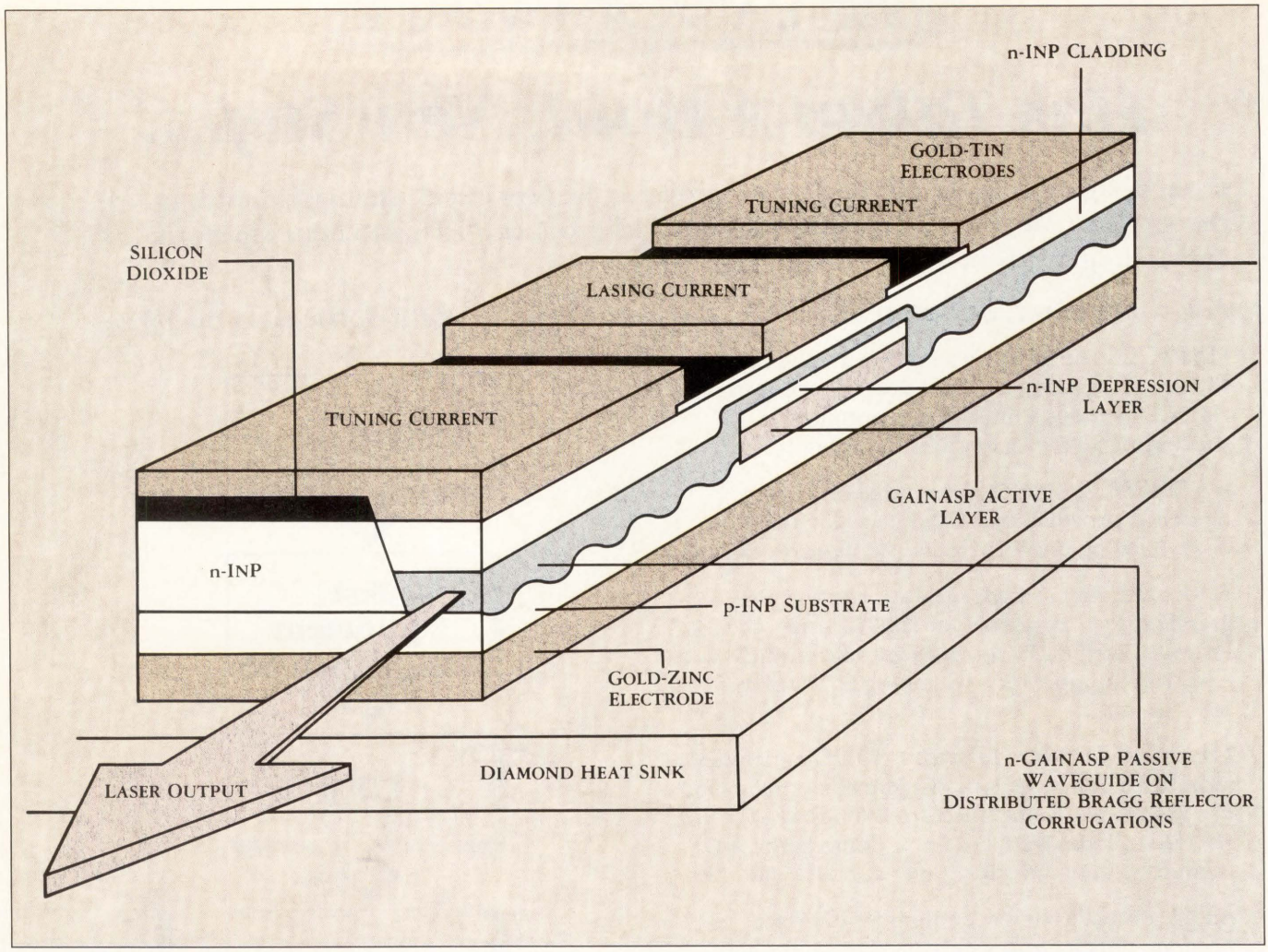


Figure 4. Sectional view of a bundle-integrated-guide distributed-Bragg-reflector dynamic-single-mode laser diode, after Tohmori et al. (1986).

of the carriers. Therefore incident light is less strongly absorbed. This so-called quantum-confined Stark effect is the basis of the quantum-well modulator. It is so significant first because the structure is directly compatible with high-speed electronics and second because the quantum-confined Stark shift is very large (by atomic standards): about 40 times the exciton binding energy (which is on the order of milli-electron volts).

The quantum-well effects have been championed by industry researchers, particularly Emilio Mendez at IBM's Thomas J. Watson Research Center and Daniel Chemla at AT&T Bell Laboratories. At CLEO in April, the Bell Laboratories group discussed a quantum-well modulator integrated with an FET and a photodiode. This experiment demonstrates optical I/O perpendicular to the chip—the photodiode detects an optical signal and drives the FET, which drives the quantum well, modulating a second beam. This optoelectronic three-terminal amplifier showed a gain of 25. Arrays of such multiple-quantum-well modulators could provide the technology for the spatial light modulators called for in some optical computing schemes. Alternatively, if combined with free-space or fiber propagation between circuit boards, such OEICs could be the basis of optical interconnects in otherwise conventional computers.

■ UNDERLYING TECHNOLOGY

Many demonstration experiments have shown the various optical devices integrated with MESFET or MODFET circuits. Most of the progress so far is in the AlGaAs alloy system, which is sensitive to radiation at 0.9 μm and shorter wavelengths. These devices are used in short-haul communications. InGaAsP alloys are used for applications at 1.3- and 1.55-μm wavelengths, which are the locations of the low-loss windows of optical fiber used in long-distance networks. The most highly developed technology is the AlGaAs/GaAs system, which covers most of current optical device applications. For both optical and electronic characteristics, InP-based devices might ultimately be preferred over the GaAs-based ones. Although this material system is employed to make long-wavelength lasers and detectors today, it is generally less well advanced. Finally, some effort is being applied to GaAs-on-silicon epitaxy as a means of integrating GaAs optical functions with silicon electronic functions.

In any technology, one overriding problem in integrating optical sources and electronic devices is the grossly different vertical profiles of the two types of device. Although laser diode active layers are thin, the associated buffer and cladding layers may add up to 5-10 μm of thickness. In contrast, MESFETs and MODFETs are usually fabricated within a few hundred angstroms of the surface. If the laser layers are etched away, leaving mesas of optical devices, fine-line patterning of electronics on the steeply stepped surface is extremely difficult. Some OEIC schemes involve "digging a trench" in which to regrow epitaxial layers for the laser diode. Toshiba's approach for building the transmitter IC mentioned previously is to do the fine-line lithography before the mesa layers are grown.

Another key obstacle is how to produce the mirrors on the ends of the laser cavity. In some cases, one may be able to obtain

one end mirror by conventional cleaving, but certainly not both. The Fujitsu group has tried "microcleaving," in which the active layer is undercut by etching and then vibration or pressure is used to snap off the end. The alternative is to use gratings to obtain reflection, as described above in connection with laser sources.

In assessing materials systems for optoelectronic ICs, both the material quality, which is critical for laser sources, and the isolation of electrical and optical elements must be considered (Forrest, 1987). InP substrates, which host the InGaAsP alloys, are generally smaller and of poorer quality than GaAs substrates. However, InP offers an advantage in terms of crosstalk: the electronics can be fabricated in InP, which has a wider bandgap than the InGaAsP alloys used for laser diodes. Therefore light emitted on chip won't be absorbed by the electronics. Just the opposite situation obtains in GaAs, where the light sources are in the wider-bandgap AlGaAs, whose radiation is absorbed easily by GaAs-based electronics.

Although the InP system is attractive for its transparent substrate and for the high electron velocity that can be obtained in InGaAs, it has some shortcomings. For one, Schottky barrier heights are too low to make MESFETs easily, and passivation is too difficult to make MISFETs. So we're left with the less common JFET, which is undergoing some development for OEICs. Ultimately, having a two-wavelength network (AlGaAs in short-haul or LAN applications and InGaAsP in long-distance ones) may be a necessary inconvenience, particularly if the GaAs-based technologies can achieve low-cost OEICs.

GaAs-on-Si technology may be the way to achieve that goal. It has evolved very rapidly over the past four years. Workers in both Japan and the United States are achieving results that suggest GaAs-on-Si may make it possible to exploit the superior electronic and optical properties of GaAs, yet retain many of the very significant manufacturing advantages of silicon technology. GaAs wafers are relatively small, brittle, and expensive compared with silicon wafers. Because of the special care and special equipment required, as well as the smaller wafer area for devices, GaAs manufacturing is generally much more expensive than silicon manufacturing. GaAs-on-Si, ideally, would provide GaAs layers on standard 6-inch substrates that could be handled by standard fabrication line equipment.

To succeed in this area, the precise control and ultrahigh-vacuum conditions of molecular beam epitaxy appear to be essential. By MBE, you can deposit Ga and

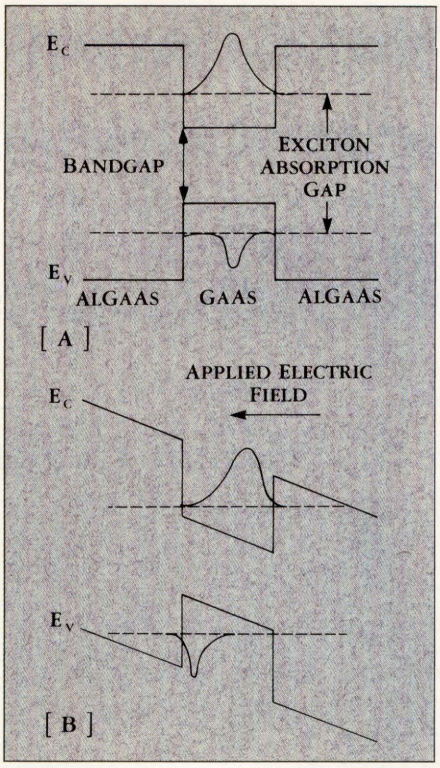


Figure 5. Energy bands and carrier distributions in a quantum well under zero bias (A) and with an applied field (B).

As on clean Si substrates. Because the GaAs and Si crystal lattice constants differ by 4%, misfit dislocations must form as the GaAs layer grows. The substrate surface preparation and the detailed growth conditions will affect the number of defects and their propagation through the growing crystal. In addition, researchers have found that strained-layer superlattices of GaAs-based compounds can be effective in absorbing the defects or turning them away from device active regions.

The other major outstanding problem in GaAs grown on Si is the 2.5 times larger coefficient of thermal expansion of GaAs compared with Si. Because growth takes place at an elevated temperature, on cooling the two materials contract differentially and tremendous tensile stress is applied to the GaAs layer. Unless the GaAs layer is very thin, the result is cracking of the epitaxial film. Clearly, these are serious problems. However, recent results continue to be promising. At CLEO in April, J. Paslaski, H.Z. Chen, H. Morkoc, and A. Yariv of the California Institute of Technology reported the first ac performance measurements on GaAs-on-Si p-i-n photodiodes fabricated by MBE. They found an impulse response with a 45-ps pulse width and a −3-dB corner frequency above 4 GHz.

The real acid test, however, is laser sources. Experience with GaAs in the 1970s clearly correlated laser diode reliability with material quality. At present, GaAs-on-Si material is coming out with dislocation densities of about 10^7 cm^{-2}, and the defects are limiting the performance that can be achieved.

■ PROSPECTS FOR OEICS

Drive electronics integrated with laser diodes and amplifiers integrated with photodetectors allow the optical devices to be properly impedance-matched to the prevalent electronic environment. For the near-term applications in long-haul communications and switching, a hybrid approach to optoelectronic integration is most likely, with GaAs- or InP-based OEICs performing transceiver functions only. A hybrid approach would also be the most direct path to optical interconnections in computers. A longer term interest is to develop optical I/O perpendicular to the IC surface, for high capacity and low-interference interconnections. Not until the GaAs and InP material systems have developed much further will it be possible to assess the best way to reach this goal. ■

REFERENCES

FORREST, STEPHEN R. November 1987. "Optoelectronic Integrated Circuits," *Proceedings of the IEEE.*

FOSSUM, ERIC R. May 1987. "Photonic Interconnections: More Bandwidth, Less Real Estate," *Photonics Spectra.*

MAKIUCHI, M., T. KUMAI, H. HAMAGUCHI, O. AOKI, Y. OIKAWA, and OSAMU WADA. 1988. "Flip-Chip Integrated GaInAs PIN Photodiode/GaAs Preamplifier Photoreceiver for Gigabit-Rate Communication Systems," *Conference on Lasers and Electro Optics.* Anaheim, CA.

SAWCHUCK, A.A., and T.C. STRAND. January 1984. "Digital Optical Computing," *Proceedings of the IEEE.*

SUEMATSU, Y., and S. ARAI. November 1987. "Integrated Optics Approach for Advanced Semiconductor Lasers," *Proceedings of the IEEE.*

TOHMORI, Y., H. OOHASHI, T. KATO, S. ARAI, K. KOMORI, and Y. SUEMATSU. January 1986. "Wavelength Stabilization of 1.5 μm GaInAsP/InP Bundle Integrated Guide Distributed Bragg Reflector (BIG-DBR) Lasers Integrated with Wavelength Tuning Region, " *Electronics Letters.*

TOMLINSON, W.J., and C.A. BRACKETT. November 1987. "Telecommunications Applications of Integrated Optics and Optoelectronics," *Proceedings of the IEEE.*

WADA, Osamu. March 1987. "Recent Progress in Optoelectronic Integrated Circuits," *Proceedings of SPIE.*

Sequent's Symmetry Series

'Software Breadboarding' Caught 95% of the Design Errors

PAUL GIFFORD, SEQUENT COMPUTER SYSTEMS INC., BEAVERTON, ORE.

IN DEVELOPING ITS new Symmetry multiprocessor computer system, Sequent Computer Systems integrated board-level simulation into its design methodology. Just 11 months after the start of Symmmetry's design implementation phase, we installed beta site units. Much of this compressed product development cycle can be attributed to extensive simulation at both the board and the ASIC level.

The three complex ASICs developed for the project were prototyped and powered up in just seven months, and the final production version was readied in parallel with the alpha and beta site evaluations. The three PCBs involved were sent straight to prototyping without the construction of an intermediate breadboard, and like the ASICs, the production versions were readied in parallel with the system evaluation phase.

Usually, a project of this scope would require several breadboard iterations that would add months to the schedule. Without "software breadboarding"—that is, board-level simulation—the Symmetry design group could not have achieved its time-to-market goals and would have spent more money in the development process.

■ SYMMETRY ARCHITECTURE

The Symmetry series is a follow-on to Sequent's original Balance product line. The Balance series machines are shared-memory multiprocessor systems employing up to 30 processors and are based on the 32-bit National Semiconductor 32032 microprocessor family. The Symmetry architecture is similar but is based on the Intel 80386 microprocessor family (Figure 1) and an extended system bus structure and protocol. All processing units (packaged two to a PCB) have symmetric access to up to 240 MB of main memory via the common system bus. The I/O portion of the architecture is divided among three units: an SCSI controller card, which also handles serial and Ethernet interfaces and diagnostics; a dual-channel SMD disk controller card; and a Multibus interface card that attaches to a separate Multibus card cage.

The I/O sections were essentially carried forward from the original Balance design, but the rest of the system required an extensive design effort. The major elements in the design are the processor PCB, which holds either one or two identical 32-bit processing units; the 16-MB memory controller PCB; and a companion 24-MB memory expansion board. Finally, the design required three new ASICs: a 14,000-gate standard-cell chip, a 10,000-gate gate array, and an 8,000-gate gate array. All three were implemented in 1.2-μm CMOS technology and are packaged in 180- or 208-pin PGAs.

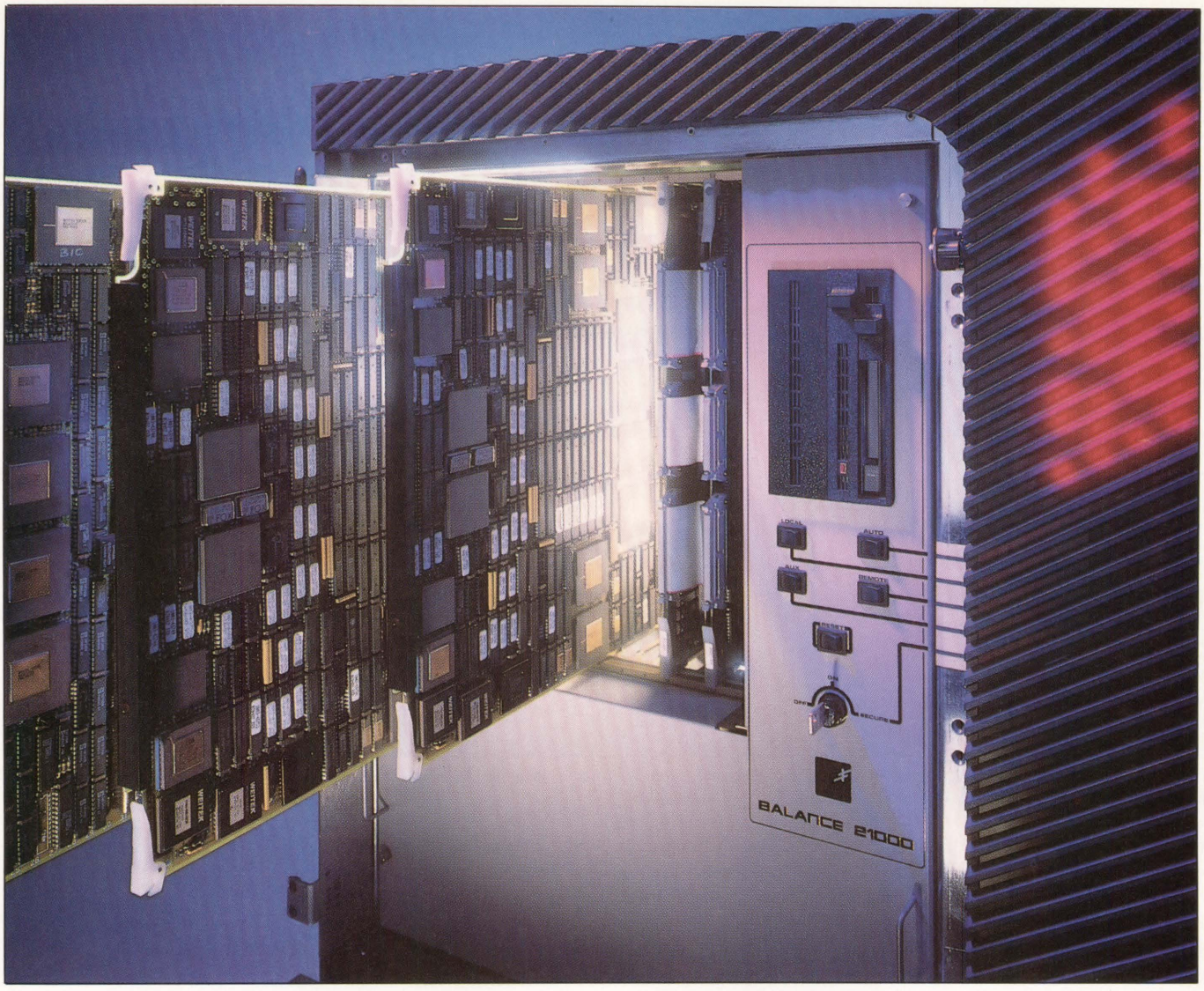
ASIC development was required to deal with one of the more unusual features of the design, a write-back, or copy-back, cache memory system. Such systems have frequently been described in the technical literature, but a bus-based copy-back system for a parallel computer was never commercially implemented until now.

■ WRITE-BACK MEMORY

A shared-memory multiprocessor system must have a method of ensuring that the individual processing units remain consistent, or coherent. The usual approach is a write-through caching protocol, which requires that all memory writes by a processor be sent both to its local cache memory and to the global memory. All other processors observe this write activity on the system bus and update or invalidate their local caches as required.

As the number of processors grows, however, the write-though scheme generates a large amount of write traffic on the system bus. This traffic, which also is an inefficient consumer of the available bus and memory bandwidth, is usually the primary limitation on processor scaling, or the ability of a particular system bus to support additional processors.

The copy-back approach removes this write-through overhead by requiring each processor to perform writes locally (into the cache). Consequently, writes that miss at the cache are translated into memory reads followed by a cache installation (allocation) and a local write. All subsequent writes to this "modified" cache line may be performed locally, without system bus traffic. Since writes are performed only during cach replacement (reallocation of the line to another area of memory) of



these modified lines (to write back the modified data into the global memory), processors observe memory reads on the bus instead of memory writes to effect cache consistency. When a processor detects a bus-level read that is accessing a locally modified copy of the data, it aborts the memory controller's response and sends its cache data to the processor initiating the read. The control logic to implement this write-back system requires interaction among all three of the ASICs.

■ SYMMETRY PROCESSOR

Figure 2 shows a high-level block diagram of one of the processing units on the processor board (each PCB also contains a second processing unit identical to the first). The highlighted blocks represent the sections implemented as ASICs. The bus data path (BDP) chip is the standard-cell IC, and the cache memory controller (CMC) and bus interface controller (BIC) chips are gate arrays. This chip set, along with the cache memory itself, act as an intermediary between the system bus and the 80386 processor. These same system bus interface components—the BDP and BIC—not only are used on the new processor and memory controller boards, but also will be used on future I/O boards. Other major functional blocks on the processor board include a serial link interface controller (SLIC). The SLIC provides a system-wide serial data path that is independent of the main bus; it is used for certain control and diagnostic operations, including interrupt prioritization.

The design goals for the processor board included a 5× increase in processing speed over the 32-bit processor boards in the original Balance system and up to an 8× boost in floating-point performance. The floating-point goal was met through the use of a standard 80387 coprocessor, augmented by an optional Weitek floating-point accelerator based on the WTL-1167 chip set. The physical constraint was 11 × 14 inches for a PCB that would hold two complete processor subsystems. Additionally, the design had to accommodate future performance upgrades, such as higher clock rates, larger caches, and next-generation microprocessors.

■ BOARD-LEVEL MODELING

The design automation environment consisted of eight Mentor Graphics Idea series workstations configured in a network with two 500-MB DSP90 server disks for centralized data storage. This environment includes schematic capture and simulation tools for both the boards and the ASICs. The simulation strategy was to simulate the ASICs in a board-like "surrounding" environment and also to simulate the board design as a whole, including its interaction with the system bus.

The key to the board simulation was the mix of models that produced accurate results yet allowed simulation runs to be accomplished in a reasonable amount of time. The three boards contain a wide variety of devices, including TTL drivers and buffers, PROMs, PALs, the 80386, the 80387, the 1167 VLSI chip set, multiple instances of the three ASICs, the SLIC chip, some high-speed static RAMs, error detec-

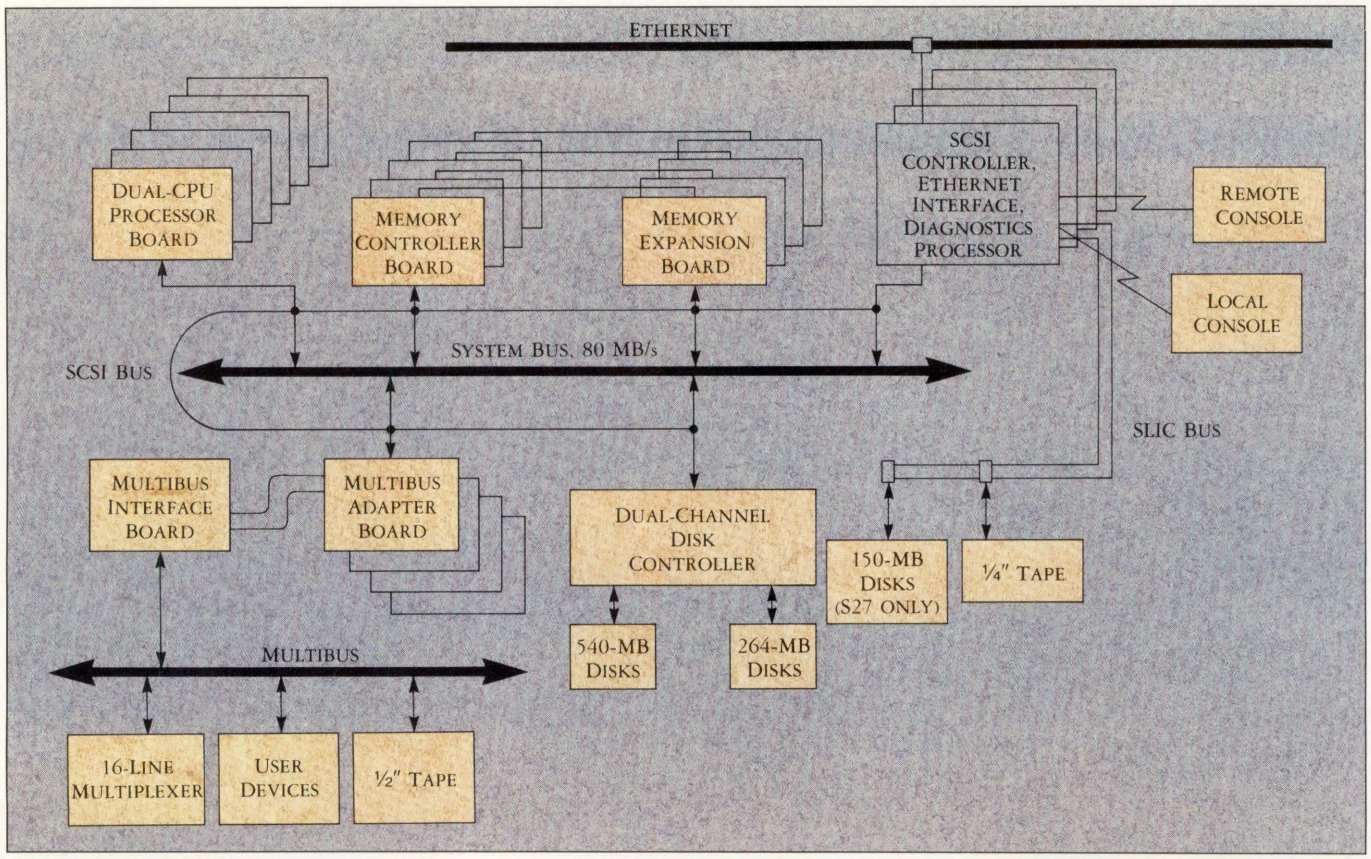


Figure 1. The architecture of the shared-memory multiprocessor Symmetry series is similar to that of the earlier Balance series. All processing units have symmetric access to as much 240 MB of main memory through the system bus.

tion and correction chips, and 1-megabit dynamic RAMs.

The 80386 was modeled through a bus-level model suppied by Logic Automation Inc. to run on the Mentor Graphics QuickSim digital simulator. The 80387 was not modeled, because it was assumed that its transactions with the 80386 were well established and tested. However, the transactions of Weitek's floating-point chip set with the CPU were simulated. This simulation was done through a bus-level model, also developed for Sequent by Logic Automation, that simulated the WTL1163 interface chip, which handles CPU–floating-point transactions. In addition, Logic Automation produced a transactional model of the 1-Mb DRAMs for the simuation of the memory controller board.

Many of the other parts were modeled through standard-parts libraries, either purchased from outside vendors or developed internally. Other, more complex parts used Mentor Graphics' QuickParts models, which are compact functional and timing descriptons for devices of medium-scale complexity. Finally, certain aspects of system's behavior were modeled through behavioral-language models (BLMs), as were many of the memory devices used within the board design.

One of the more important aspects of the simulation environment was the use of a system bus model, which allowed many board-level functions involving bus transactions to be simulated. For example, it was possible to simulate an operation where the processor would initiate read and write accesses as if it were executing a program. This execution sequence allowed the cache memory controller and its surrounding environment to be stimulated, so that various cache hits and misses could be investigated. In the case of a cache miss, the entire cycle, from processor board to memory controller board and back, could be simulated and investigated in detail.

The ASICs were developed using a simulation strategy that complemented the board simulations. Since a board-level simulation environment existed, it was possible to simulate and analyze the functionality of the ASICs in the context of their target system. The gate arrays were designed on Mentor Graphics' workstations from vendor libraries that included simulation models targeted at the QuickSim digital simulator.

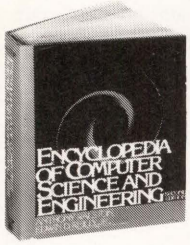
■ ASIC Simulation

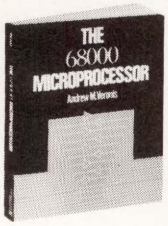
The ASICs were modeled using gate-level representations based on the schematics constructed from vendor libraries. The team had intentions of developing behavioral models of the ASICs using Mentor Graphics' BLM approach, but the task proved to be too time-consuming. The availability of a robust and well-integrated hardware description language (HDL) would have simplified the job.

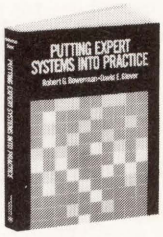
During development, the individual ASICs were either "surrounded" by a behavioral shell or model or embedded in a mixed gate and behavioral model, which allowed the designer to exercise them in a manner representative of the intended application. The gate-level models and BLMs were "wired" together through schematic sheets. The state of the individual signal inputs to a BLM was then a function of the overall design and the applied stimulus during a test sequence.

For example, the cache memory controller gate array was embedded in a schematic sheet, with various three-state buffers and RAM BLMs. It was stimulated within this environment by simulated processor requests and system bus requests. As the requests occurred, the CMC's output was monitored. The RAM BLMs were accessed, updated, and reaccessed during the test sequences. Essentially, the surrounding environment provided a means of automating the analysis process to a degree. Device outputs could be used to realistically control external address and data paths. Also, the state of the simulated cache RAMs could be al-

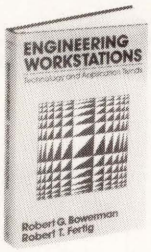
HIGH-PERFORMANCE SYSTEMS DESIGN:

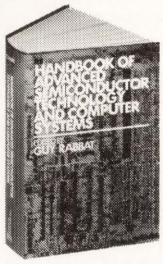






Expert guidance—
FREE for 15 days





From microchips to full-scale computer systems
HANDBOOK OF ADVANCED SEMICONDUCTOR TECHNOLOGY AND COMPUTER SYSTEMS
Edited by Guy Rabbat

$64.95, 960 pages, illus., January
For the first time, one source details computer system development by starting at the microchip level. It covers everything from semiconductor processing methods to very large scale integration (VLSI), computer-aided design of computers, and computer systems architecture. This source provides information on VLSI routing techniques, cell-based VLSI design, and VLSI design verification and logic simulation. Software design techniques complete this chip-to-final-system reference.
To order, circle #126 on reader service card.

Modern microcomputing—what it is and how it works
MICROPROGRAMMING AND FIRMWARE ENGINEERING METHODS
Edited by Stanley Habib

$39.95, 560 pages, illus., July
Here is clear documentation of key concepts used in microprogramming. Habib covers control function design and examines the refinement, development, and current role of microcomputers in modern computing systems. Drawing on the expertise of leading authorities in the field, the book reflects over twenty years of experience in microprogramming methodology.
To order, circle #127 on reader service card.

An indispensable aid for designing and installing—
THE 68000 MICROPROCESSOR
By Andrew M. Veronis

$29.95 paper, 352 pages, illus., April
All the information you need to use the world's most advanced 16-bit microprocessor in a new system or product is at your fingertips! *The 68000 Microprocessor* is the first guide to cover both the basic operation and programming of the 68000. It also shows you how to use several popular members of the 68000 family and peripheral devices in electronic systems.
To order, circle #128 on reader service card.

Today's programs, techniques, and models
ANALYSIS AND DESIGN OF ELECTRONIC CIRCUITS USING PCs
By John Greenbaum, Bruce Pollard, Less Besser, Brian Biehl, and Robert Osann

$34.95, 336 pages, illus., June
Discover how the PC can be used for design with this application-oriented guide. It describes available programs, their strengths and limitations... techniques for ensuring optimum results from simulations...recently developed techniques for analyzing and designing electronic circuits... analog, digital, and microwave programs...and computer models for bipolar, FET, and other semiconductors or digital and analog ICs.
To order, circle #129 on reader service card.

Proven methods to ensure desired results
LOGIC DESIGN WITH PASCAL
Computer-Aided Design Techniques
By Thomas Downs and Mark F. Schulz

$42.95, 512 pages, illus., May
This powerful "tool" shows how to use tested Pascal programs to design combinational and sequential logic circuits. It offers clearly explained CAD techniques; many ready-to-use Pascal programs to facilitate synchronous sequential design; and coverage of state minimization, design using MSI devices, digital system design, and the incorporation of multiplexing, ROM, and programming logic arrays into system design.
To order, circle #130 on reader service card.

Detailed coverage of what you need to know about—
ENGINEERING WORKSTATIONS
Technology and Application Trends
By Robert G. Bowerman and Robert T. Fertig

$39.95, 304 pages, illus., March
A modern, in-depth work on engineering workstation design, technology, and applications. It examines a user view of productivity, how workstations can improve engineers' work, CAD, CAM, CAE, local area networks, server-based architecture, Smalltalk influences, solids model databases, drafting, finite element analysis, pre- and post-processing, machine tools, robotics simulation, and control.
To order, circle #131 on reader service card.

Take full advantage of expert systems
PUTTING EXPERT SYSTEMS INTO PRACTICE
By Robert G. Bowerman and David E. Glover

$39.95, 384 pages, illus., April
Get the strategies and tactics you need for putting expert systems into practice in the most practical, cost-effective, and profit-making way! This real-world guide clearly covers the seven phases of a recommended expert system development cycle, demonstrating how a project develops from conception to successful completion!
To order, circle #132 on reader service card.

A convenient source for fast, accurate information
ENCYCLOPEDIA OF COMPUTER SCIENCE AND ENGINEERING
Edited by Anthony Ralston and Edwin D. Reilly, Jr.

$93.95, 1,678 pages, illus., 1982
Drawing on the expertise of more than 300 respected authorities, this one-volume reference answers a wide range of questions about computer science, engineering, and related fields. It offers over 550 informative entries, plus over 700 explanatory illustrations, tables, and charts. Useful appendices focus on abbreviations, acronyms, high-level languages, research journals, and university departments.
To order, circle #133 on reader service card.

TO ORDER FOR 15-DAY FREE EXAMINATION: Simply circle the appropriate number(s) on the Reader Service Card at the back of this magazine. Your book(s) will be sent to you for your 15-day examination. If you are satisfied, keep the book(s) and pay the purchase price plus postage and handling. Otherwise return the book(s) by the end of the 15-day period and owe nothing. You may also order by writing directly to *VLSI*.

To order, use the card provided or call toll-free **1-800-645-6278** and ask for Kathie Conlon.

6/88 C 4460

Van Nostrand Reinhold

tered, and a new stimulus that is dependent on this altered state could be generated. The bus interface controller and bus data path devices were developed in a similar manner.

After the individual ASICs were developed in the surrounding environment, and before tape-out of the designs to the vendor, they were linked into the board-level environment and the files expanded for simulation. This allowed the devices to be exercised as a complete chip set. In addition, it ensured that any errors introduced by the stand-alone surrounding environments were uncovered.

■ Simulation Performance

The ASIC foundry accepted the Symmetry design team's netlists without performing its own prelayout simulation. Once the ASIC layout and routing process

MSPICE Backplane Simulation

In addition to the board-level logic simulation of the entire Symmetry system, the design team performed a number of analog simulations of the system backplane using Mentor Graphics' MSPICE, an interactive analog simulator based on Berkeley SPICE. The goal of the MSPICE simulations was to determine the maximum number of backplane board connections that could be supported and to analyze extender card performance. More backplane connections would allow customers to expand the systems as their needs expanded, ensuring a long product life. Additionally, by verifying that the backplane delivers the same signal quality when extender cards are used, we could ensure ease of testing in the engineering lab and on the manufacturing floor.

Simulation results indicated that we could reliably implement 21 slots on the backplane. Also, we learned that we could loosen tolerances on the backplane impedance without affecting performance and thus reduce manufacturing costs.

Model development time for the backplane simulation was minimal. The impedance of the unloaded backplane was modeled by taking into account only significant capacitance and inductance effects. A SPICE model of the output driver was supplied by the IC manufacturer and receiver gate models were developed with the MSPICE modeling language. Because only simple high-level models were needed, the backplane designers were able to progress from knowing nothing about MSPICE to a working simulation in three weeks.

MSPICE provided a convenient interface for defining component values, circuit power supplies, and the type of analysis to be performed. Setup and run commands were preprogrammed using Mentor Graphics' macrocommand capabilities, allowing conditions under varying loads to be entered and simulated with a few keystrokes. Sequent's most recent backplane simulations were performed on an Apollo DN460 platform and took under an hour to run. Achieving the same analysis conditions with a physical breadboard would require weeks of work and considerable expense.

Various backplane configurations were simulated to address questions such as, Should connections be strung out or packed in tightly? How long and wide should traces be? and What is the optimal termination method?

In the final design, connections for 21 slots are staggered on the front and back sides of the backplane. A 4/10-inch trace is allowed between each connection and a 2-inch trace is specified from the backplane to ICs on each board. To keep the low–to–three-state transition within 21 seconds, signals are terminated to 5 V using a 220-Ω resistor, with a diode added to clamp the signals at the 3-V transition point. To maintain signal fidelity, the signal path along the bus is kept to approximately 1 foot.

Since customers can configure a Symmetry system with anywhere from very few boards to a fully loaded backplane, it was important to simulate how the backplane would perform under varying loads.

Worst-case measurements of backplane performance were made for the three-state–to–low transition with varying loads while driving from slot 1 and analyzing results at slot 20. As loads are added to the backplane, this transition becomes more difficult to make, because backplane capacitance increases. Multiple end-to-end propagation delays were specified for the bus to allow for the drivers' inability to drive the very low effective impedance. This problem is magnified during a low–to–three-state transition.

The MSPICE simulation revealed that intrinsic backplane impedance could vary within $\pm 20\ \Omega$ without significantly affecting the electrical characteristics of a loaded system. This conclusion was surprising given that backplane tolerances for a high-performance computer system are typically kept within $\pm 5\ \Omega$. However, the results were supported by the fact that the effective impedance of a configured system varies in proportion to the load. For example, if the intrinsic impedance of the unloaded backplane were 100 Ω, once load capacitance was considered the effective impedance might only be 20 Ω—an 80% reduction. If the intrinsic impedance were only 80 Ω, load capacitance would again reduce it by four fifths, so that the effective impedance would be 16 Ω.

Though the simulation results seemed accurate, the backplane designers verified them before loosening the backplane tolerances by comparing the MSPICE-generated waveforms with actual measurements made with a prototype backplane. The two sets of waveform measurements could almost be overlayed, indicating the high reliability of the simulation method and models.

One instance where intrinsic impedance does have a significant impact on signal quality is when extender cards are used. Extender cards lengthen the distance from a PCB to the backplane by approximately 16 inches. When the backplane team simulated some existing extender card designs, the results showed that extender cards translated slight reflections caused by a mismatch between backplane impedance and the impedance of the etch from the driver to the backplane into very large signal swings.

The design team concluded that new extender cards were needed with the impedance matched to that of a fully loaded backplane. They then ran additional simulations, each time varying parameters to approach a matching impedance. Based on the results of these simulations, we now build 18-Ω extender boards to a tolerance of $\pm 3\ \Omega$. Although these boards are more expensive to manufacture than conventional designs, they allow accurate and easy debugging of both the Balance and Symmetry product lines. ■

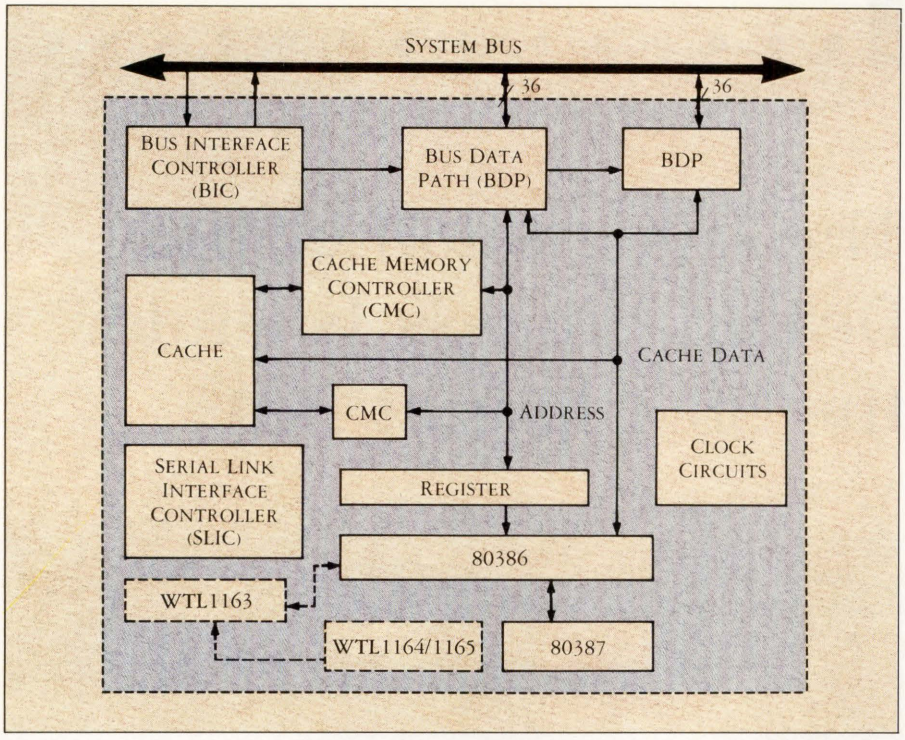


Figure 2. Each processor (there are two to a PCB) contains multiple VLSI chips. Besides the 80386 and 80387 and the optional WTL1167 floating-point chip set, there are three semicustom devices: the bus data path, a standard-cell design, and the bus interface controller and cache memory controller, implemented with gate arrays.

was complete, the design was back-annotated to include timing information based on actual metal run lengths and fan-out. This updated file was then used to perform postlayout simulations using the vendor's simulator so that the best- and worst-case timing could be examined and verified.

The ASICs were developed on Apollo 3000 platforms with either 4 or 8 MB of memory and a 170- or 380-MB hard disk. The board-level simulations were run on DN460 platforms with either 12 or 16 MB of memory and a 380-MB disk.

The ASIC simulations were typically decomposed into groups of test sequences (scripts, macros, or force files), with each sequence intended to exercise some particular function or operating mode. An individual test took from tens of minutes to several hours, depending on the number of vectors applied, the complexity or size of the design, and the speed and memory capabilities of the simulation platform.

The board-level simulations were likewise decomposed into groups of test sequences. Owing to the size of the design being simulated (70,000 to 100,000 instances), the amount of physical memory on the platform had a large impact on simulation throughput. With more memory, less paging across the network is necessary. The DN460s were used for board simulation primarily because they could hold twice as much physical memory as the new 3000 series platforms.

Initializing the board-level simuation of the Symmetry processor was a complex task. For example, various BLMs, such as certain RAMs and PROMs, required initialization. Also, individual scripts were required for each of the Logic Automation bus-level models. Additionally, the system bus model had to be initialized. As a result, the task of initializing the simulation environment took several hours. Once it was accomplished, the actual board-level simulation was started.

The simulation process began by applying a RESET command and then initializing the modes of the VLSI models using a simulated connection to the system's boot and diagnostic processor subsystem. Only then was the simulation environment ready to run the desired test sequences. Usually, these sequences consisted of processor and/or system bus requests and associated reference information. An overall simulation run, including a few hundred system bus or processor cycles, typically required 8 to 16 hours of wall-clock time to complete. Another 2 to 4 hours would then be needed to interactively examine the results and identify any problems.

One advantage of ASIC simulation in the board environment is the ability to re-create and examine functionality problems discovered in the physical prototype. When a problem with an ASIC occurs in the laboratory, it becomes very difficult to infer the cause of the problem from the stimulus and response at the pin level on the board. It is often easier and more complete to re-create the situation in the simulator in terms of the pin-level I/O and then examine the resultant internal activity at the gate-level inside the ASIC circuitry. The QuickSim simulator allows the schematics to be "probed" so that activity at any internal circuit node can be selectively observed. By rapidly pinpointing problems, this capability saved many days of trial and error during the project.

Board simuation also served as a tool for testing basic diagnostic code and power-on sequences. Diagnostic software sequences were used to emulate the output of the serial link interface controller's board-level interface (Figure 2). The SLIC is used to interface with the ASICs and to provide a remote link for diagnosing, initializing, and configuring. These sequences simulated the instruction flow to the BIC, CMC, and BDP chips necessary to configure them, test them, and run diagnostic sequences. As a consequence, it was possible to test and debug several of the critical power-up and initialization sequences before the hardware ever reached the laboratory.

■ Evaluating the Results

Both quantitative and qualitative benefits derived from the board simulation strategy. On the qualitative side, it allowed the verification of several key design decisions. One was the use of the copy-back cache scheme. Also, software breadboarding was mandatory to verify the functionality of the three ASICs. These chips were essential to implementing the product and were far too complex to design and analyze without design automation tools. Since they functioned as a chip set, it was important to simulate their operations in a board environment.

On the quantitative side, at least 95% of the design errors were identified before we developed the initial prototype hardware. Further, we estimate that the board simulation strategy cut the time for product development—which was done using a core hardware development team of only four design engineers—by up to a third. ■

About the Author

PAUL GIFFORD *is the manager of central systems engineering at Sequent and the program manager for Symmetry platform developments. He designed the copy-back cache and the processor subsystem for the Symmetry series. Before joining Sequent in late 1985, Paul was chief technical officer at Saturn Computers Inc. He received his BSEE in 1973 and an MSEE in 1979, both from the Rochester Institute of Technology.*

Common Design Environments Are Embracing Many Tasks

The Integration of the Design Process Is under the Spotlight at DAC

THE DESIGN Automation Conference consistently disgorges a multitude of new products that are too numerous to categorize. However, the trend this year clearly seems to be integration: the integration of design and simulation, of hardware and software design, of simulation capabilities, of design and test, and of design environments.

Not all of this year's products attempt to link individual steps in the design process. New approaches for specific tasks—such as VHDL simulation support—will also appear. We've attempted to compile some of the significant announcements to be made at the show.

■ Integrated Design and Simulation

Through new database techniques, vendors of simulators are beginning to eliminate some of the compilation steps normally required between schematic capture and simulation. Older design systems use separate databases for graphical schematics and list-oriented simulation. The two products eliminating most compilation, therefore, are relatively young products.

In its behavioral simulator, Vantage Analysis Systems Inc. (Fremont, Calif.) combines a unified database structure with concurrent execution of schematic capture, simulation, and waveform-viewing programs. All programs are initiated at the beginning of a session and reside in separate windows on the screen. Users move from one program to another by changing windows. Updated schematics can be simulated immediately, with results appearing in the waveform window. The unified database structure eliminates lengthy compilation between steps.

Vantage's simulator is the first to execute directly on VHDL models; models in the VHDL format will be available from Logic Automation Inc. (Beaverton, Ore.). The combination of behavioral modeling and no compilation steps speeds up the design process through a concept the company calls "incremental design": iterative changes in the schematic are immediately analyzed through simulation, and their effects can be viewed immediately in the waveform viewer.

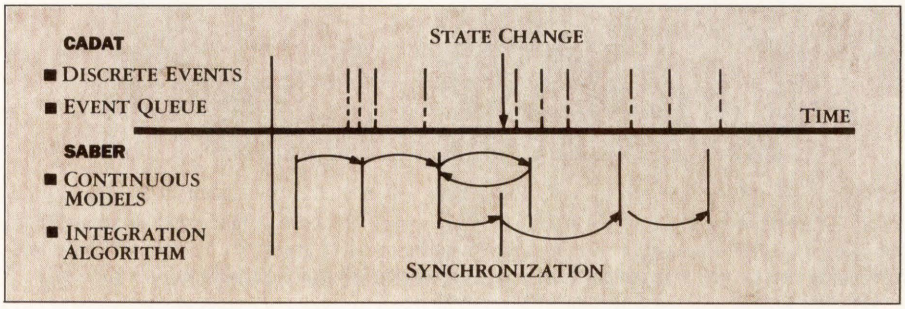


Figure 1. The Calveras algorithm allows the Saber analog simulator and Cadat digital simulator to use the time steps most natural to each. Only when an event in one simulator affects the other simulator does the algorithm lock Saber to a time step in Cadat.

Aldec (Newbury Park, Calif.) has updated its Susie logic simulator to minimize compilation and to execute in the 80386 native mode. Susie-5 can simulate designs as large as 100,000 gates that have been created with PC-based schematic capture tools. Users can toggle switches, replace chips, load JEDEC fuse maps and hex files for PLDs, and move jumpers without recompiling the design for resimulation. Susie-5 is rated at 265,000 AND-OR evaluations/s on an 80286-based PC and more than 500,000 evaluations/s on an 80386-based PC.

■ Integrated Embedded System Design

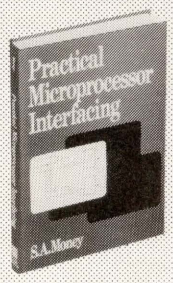
The System Design Environment from Interact Corp. (New York) could represent the prototype of future design systems that combine hardware and software design. According to company information available at the time of this writing, designers will be able to use the system to design, simulate, and debug embedded systems, including hardware and software components. It allows software engineers to run their programs on simulated hardware designs, and hardware designers to test their designs, integrated with software components, before building prototypes. It also provides a feedback mechanism between hardware and software tools to simplify verification.

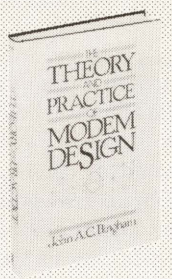
The design environment contains models of standard microprocessors that are capable of running as many as 5000 instructions/s, whereas most software models of microprocessors can execute only a few instructions each second. The System Design Environment models can work with behavioral models of other system functions for a complete system simulation. The environment also includes software development tools, design capture tools, a software debugger, and a library of standard components.

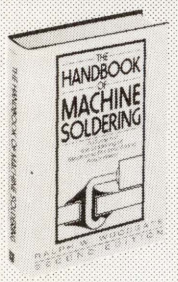
The design environment is based on an object-oriented database that, with configuration management and version control, tracks the library elements and user designs. An integrated "rule base" can automate the control of the execution of the design tools, making it easier for the user to concentrate on the design process.

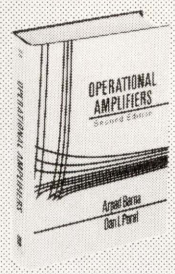
ESSENTIAL REFERENCES from WILEY-INTERSCIENCE

Electrical Engineering Texts . . . Examine FREE For 15 Days!









BUILT-IN TEST FOR VLSI
Pseudorandom Techniques
P.H. Bardell, W.H. McAnney & J. Savir

Provides comprehensive, state-of-the-art information on the analysis and design of built-in self-test theory and practice. Includes scores of solved examples.

1-62463-2 1987 360pp $49.95
(circle #: 110)

DIGITAL TEST ENGINEERING
J.M. Cortner

Presents the faults most likely to occur and the methods used to detect them. Describes test equipment, including automatic test equipment, interface hardware and test programming languages.

1-85135-3 1987 337pp $45.95
(circle #: 111)

MICROWAVE SOLID STATE CIRCUIT DESIGN
I. Bahl & P. Bhartia

A comprehensive discussion of the design of passive circuits, solid state devices and microwave solid state circuits. Examples and problem sets.

1-83189-9 1988 944pp $59.95
(circle #: 112)

ANALOG MOS INTEGRATED CIRCUITS FOR SIGNAL PROCESSING
R. Gregorian & G.C. Temes

Explains the operating principles of analog MOS integrated circuits and presents tested design techniques.

1-09797-7 1986 598pp $49.50
(circle #: 113)

THE THEORY AND PRACTICE OF MODEM DESIGN
J.A.C. Bingham

Examines the new theories, techniques and applications of modem design. Includes practical examples of how to choose and design all the major components of a modem.

1-85108-6 1988 384pp $49.95
(circle #: 114)

DIGITAL FILTER DESIGN
T.W. Parks & C.S. Burrus

Provides a step-by-step approach to the most powerful methods used to design FIR and IIR digital filters. Detailed design examples included.

1-82896-3 1987 342pp $32.95
(circle #: 115)

THE 80960 MICROPROCESSOR ARCHITECTURE
G.J. Myers & D.L. Budde

The authoritative book on Intel's new high-performance microprocessor series, the 80960 architecture family, written by its leading designers. Emphasizes the "whys" and "hows" of these devices.

1-61857-8 1988 240pp $31.95(tent)
(circle #: 116)

BIPOLAR AND MOS ANALOG INTEGRATED CIRCUIT DESIGN
A.B. Grebene

Treats fundamental design principles, capabilities and applications of monolithic integrated circuits.

1-08529-4 1984 894pp $62.50
(circle #: 117)

THE HANDBOOK OF MACHINE SOLDERING
Second Edition
R.W. Woodgate

Presents in one volume, all the information required to select, install and operate machine soldering equipment. Features a more rigorous review of the use of computer control, and an expanded section on Zero-Defect soldering.

1-85779-3 1988 275pp $31.95
(circle #: 118)

HANDBOOK OF SURFACE MOUNT TECHNOLOGY
S.W. Hinch

A comprehensive and practical handbook which shows how to design for surface mounted components, how to set up an assembly system, and how to keep it running problem-free.

0-21094-x 1988 512pp $74.95(tent)
(circle #: 119)

PRACTICAL MICROPROCESSOR INTERFACING
S.A. Money

Offers a solid treatment of the principles and techniques for transferring data to or from the CPU. Reviews important interface circuit arrangements.

1-63788-2 1987 247pp $31.95
(circle #: 120)

OPERATIONAL AMPLIFIERS *Second Edition*
A. Barna & D. Porat

Presents principles and applications, as well as a detailed treatment of the high speed properties of the latest operational amplifiers.

1-84715-1 1988 350pp $32.95(tent)
(circle #: 121)

PHYSICS OF SEMI-CONDUCTOR DEVICES
Second Edition
S.M Sze

Provides thorough descriptions of the underlying physics and operational principles of all major semiconductor devices. Contains critically evaluated references and over 600 technical illustrations.

1-05661-8 1981 912pp $59.95
(circle #: 122)

VLSI FABRICATION PRINCIPLES
Silicon and Gallium Arsenide
S.K. Ghandhi

Covers the principles underlying semiconductor devices made from both silicon and gallium arsenide. Emphasizes those processes which are useful for VLSI schemes, using extremely small, densely packed devices and components.

1-86833-7 1983 665pp $58.50
(circle #: 123)

TO ORDER:
Simply circle the appropriate numbers on the readers' service card at the back of this magazine. Your books will be sent to you to examine FREE for 15 days.

Prices subject to change without notice & higher in Canada

WILEY

For further information, write: WILEY-INTERSCIENCE
605 Third Avenue, New York, New York 10158
Attn: Sharon Nelson, 5th Floor

In Canada: 22 Worcester Rd., Rexdale, Ontario M9W 1L1

To order by phone, call Toll-Free

1-800-526-5368

In New Jersey, call collect 201-342-6707

092/9-0062

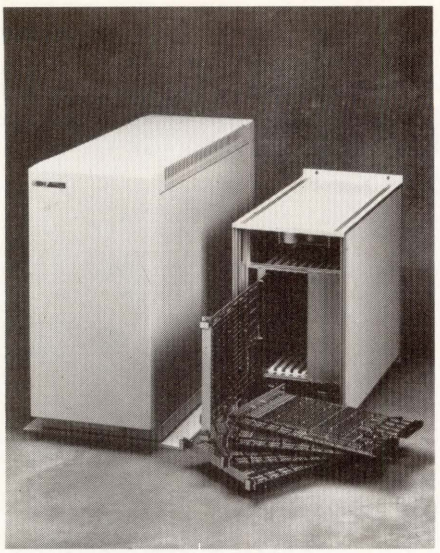


Figure 2. Aida's Memory and Logic CoSimulator processor can model circuits with up to 8 million gates.

Interact says that its new design environment is extensible, so that users can integrate other tools into the environment.

■ Integrating Simulation Tools

HHB Systems (Mahwah, N.J.) will show some of the fruits of its push to span the full range of analysis of electronics designs. Most significant is its analog-digital simulation capability, developed in conjunction with Analogy Inc. (Beaverton, Ore.). The Saber/Cadat simulator provides modeling from the behavioral level to the implementation level for both digital and analog portions of systems. It accepts a single EDIF netlist, automatically parses the netlist into analog and digital portions that execute within the Cadat and Saber simulators, respectively.

Users can interact with either simulator's interface to initiate and control simulation. Both simulators are active under the control of a proprietary "Calaveras" algorithm (Figure 1). The algorithm allows the simulators to execute asynchronously—using the most efficient time steps for each—but still remain in synchronization. Similar products usually require the time steps of the analog simulator to match up with events in the digital simulator.

Saber/Cadat provides models that are automatically inserted at the boundary between analog and digital circuitry. These models permit the user to control such details as how X states are presented to analog circuitry.

The simulator runs on platforms from Apollo Computer, Sun Microsystems, and Digital Equipment. Prices range from $35,000 to $67,000.

HHB has also acquired Simucad (Menlo Park, Calif.), a company that provides logic and circuit simulation for MOS IC designs. Simucad's Pacsim circuit simulator and Silos digital simulator (for logic and fault simulation) will be integrated into HHB's design environment, providing some balance to HHB's higher exposure in PCB design.

For design capture and model development, HHB acquired the rights to the Dash-Cadat Plus product line and the Acculib software from Data I/O's FutureNet Division (Redmond, Wash.). HHB has hired Data I/O engineers who will continue to develop the Dash-Cadat design entry and simulation system for personal computers. Acculib will be used to create ASIC libraries for HHB's simulators.

For high-end system simulation, HHB is bolstering its behavioral modeling by developing a link to the Helix behavioral language and simulator from Silvar-Lisco (Menlo Park, Calif.). More details of the link are expected to be revealed at DAC. In addition, Logic Automation has announced that it is creating versions of its SmartModel behavioral models for Cadat.

In terms of hardware, HHB not only has moved its Cadat simulator to the Sun-4 workstations, but it also has implemented its CATS hardware modeler on the Sun-4. The new modeler is called the Model 12000. Both products are available now. In addition, behavioral models from Logic Automation are becoming available for Cadat.

HHB has assimilated this variety of disparate tools to become more competitive with broader-based CAE suppliers. Hopefully, in Anaheim the company will give us a clearer picture of how it will tie all of them together.

■ Integrating Simulation and Test

Compass Development (San Jose, Calif.) produces a set of tools to simplify the development of stimulus vectors for simulation and test. Its SAV program allows designers to identify mnemonics that correspond to particular sets of waveforms. For example, a mnemonic may identify a setup-and-hold test, within which the designer specifies only the relative timing of data and clock edges.

The designer builds a stimulus vector set from these mnemonics. The tools then verify that the stimulus vectors can be recreated on the targeted production tester and produce an input file for simulation. After testing, the tools can also compare the output of test and simulation to highlight discrepancies. The software runs on a personal computer, a workstation, or VAX computer and starts at $5,000.

Compass's other program, ATOP, serves as an interface between simulators and testers. It interprets simulation output files to build tester files, automatically creating timing sets and formatting vectors. It can also translate files from one tester to another. It has other features that assist the creation and analysis of tester input and output, such as highlighting glitches in output waveforms. The tool costs $15,000 for a basic version that provides an interface between one simulator and one tester; additional tester outputs cost $7,000 to $8,000.

■ Integration from Accumulation

The three design automation arms of Teradyne Inc. (Boston)—recently acquired Case Technology Inc. (Mountain View, Calif.) and Aida Corp. (Santa Clara, Calif.), plus Teradyne's Data Systems group, which manages the Lasar logic simulator—will be demonstrating interactivity among their design environments. Teradyne has applied its Middleman translation program to bring the large Lasar libraries to the Aida simulation accelerator. Using Case's Common Simulation Data Format (CSDF) interface, Lasar can now operate in Case's Vanguard design environment. The company plans to show design files passing between the three design environments. The software will be running on networked platforms ranging from a personal computer to a VAX.

Case will also demonstrate an updated PCB layout program with both higher design completion rates than its predecessor and support for surface-mount packages placed on both sides of a PCB. The software also incorporates fine-line routing and rip-up-and-retry routing. On an 80386-based personal computer, the software package, which can use up to 16 megabytes of main memory, is priced at $5,500. Versions are also available for workstations from Apollo Computer and from Sun Microsystems ($10,000).

Aida will show its new CoSim/ML simulation accelerator (Figure 2), which executes as many as 10 million evaluations/s, five times faster than earlier CoSim products. It has a capacity of 8 million gates and supports hardware modeling of memories, a first for Aida. The first model uses the PC AT bus and connects to Apollo Domain 3000 and 4000 workstations; this fall, a new version using the VMEbus will connect to workstations from Sun Microsystems. A 1-million-gate configuration is tagged at $135,000, and an 8-million-gate version runs $275,000.

■ Disparate Developments

Silvar-Lisco has unveiled the successor to the Cal-MP standard-cell layout system.

The new tool, called SC II, adds more routing layers; better placement algorithms, including horizontal and vertical standard-cell rows; and better silicon efficiency. It can combine big blocks, such as memories, with custom blocks and fixed-height standard cells, a process that Silvar-Lisco has dubbed "silicon assembly."

The user supplies SC II with a hierarchical netlist, and the program uses the hierarchy in developing a placement strategy, creating a floorplan from the netlist automatically. The floorplan influences the execution of subsequent placement algorithms to deliver the higher silicon efficiency. It also provides global and local routing with tapered power and ground buses. As many as three layers of routing are currently supported, and Silvar-Lisco expects to expand the number of routing layers in the future.

SC II can lay out designs with as many as 20,000 cells in each layer of hierarchy. According to Silvar-Lisco, there are no limits on the number of hierarchies. This $60,000 product is compatible with existing Cal-MP design libraries.

A new fault simulator from Gateway Design Automation Corp. (Westford, Mass.), Verifault-XL, uses the same mixed-level hardware description language and algorithms as the Verilog-XL logic simulator, so it should demonstrate the same rapid execution rate as the logic simulator. It supersedes the previous fault simulator from Gateway, TestGrade, by providing full use of the hardware description language, concurrent and distributed fault simulation, and the passage of faults through behavioral models.

Since behavioral models within Verifault can propagate the effects of faults, it allows fault simulation to begin earlier in the design process. Concurrent fault simulation techniques evaluate several faults during each simulation run; distributed fault simulation divides the fault list into sections, each of which are simulated on separate workstations (for more information on these techniques, see *VLSI Systems Design,* October 1987, p. 28).

Verifault-XL models and simulates faults at both the gate and switch levels. Fault collapsing minimizes the number of faults in each pass, while incremental simulation allows interactive control and modification of each pass. Workstation versions of Verifault-XL start at $30,000.

Daisy Systems Corp. (Mountain View, Calif.) will show its design tools running on the recently announced 386i workstation from Sun Microsystems. This move signals the end of Daisy's development of workstation hardware, although it will continue to sell and support its lower-cost, 80286- and 80386-based, proprietary platforms, to interested users. These proprietary boxes, incidentally, will be running under the SunOS operating system from Sun by the end of the year. Design systems based on the Sun machine won't be shipped till September, with bundled system prices starting at $23,000 for a 20-MHz 386i including 8 MB of main memory and 155 MB of disk storage.

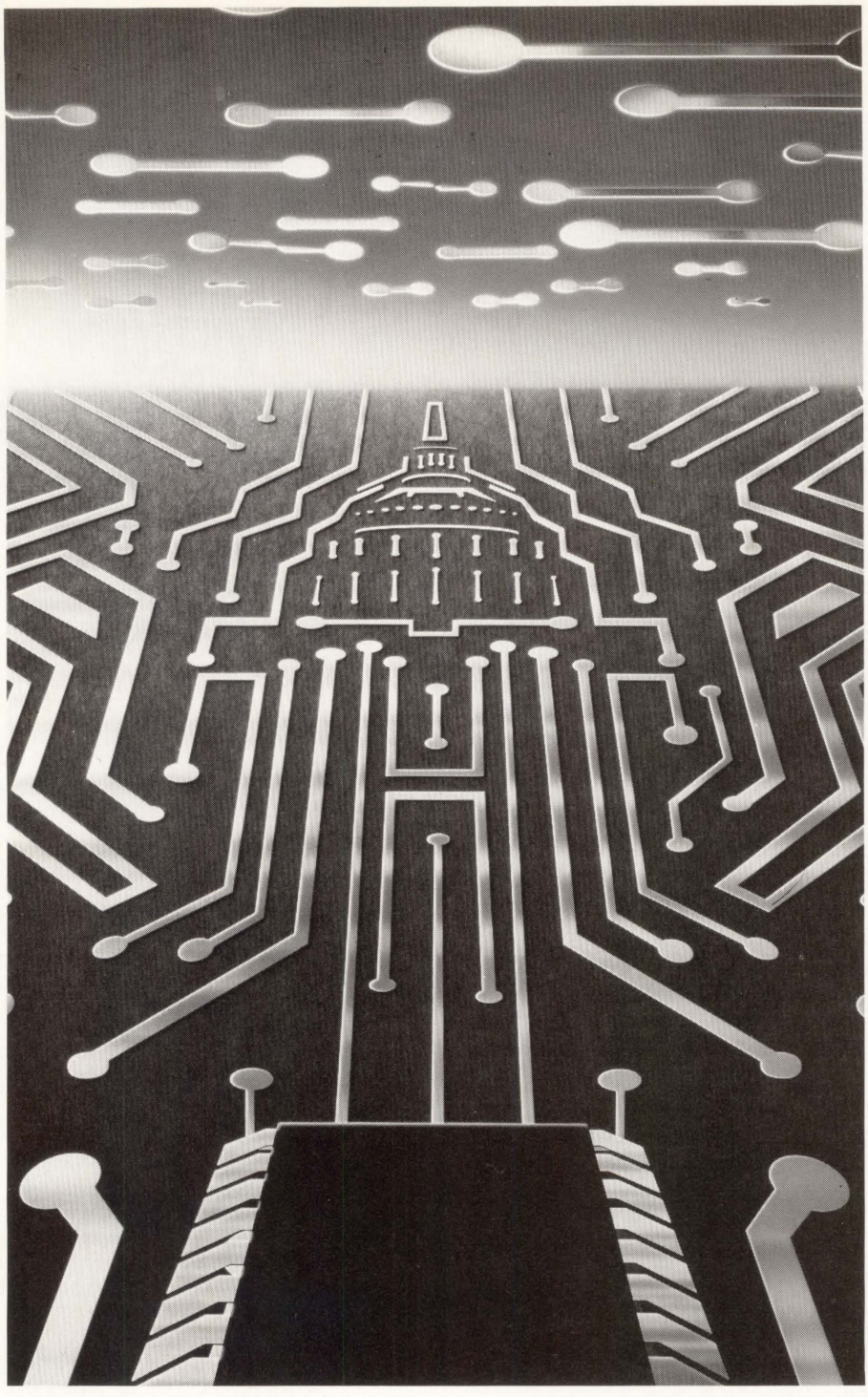
Daisy will also continue to develop acceleration hardware, such as the Gigalogi-

EIA Boosts EDIF

The Electronic Industries Association (EIA) is sponsoring a booth at DAC that will host a multivendor demonstration of EDIF 2.0.0. The EDIF schematic technical subcommittee selected six of the companies supporting EDIF to participate in the demonstration: Cadnetix, Hewlett-Packard, Mentor Graphics, Texas Instruments, Valid Logic Systems, and Viewlogic Systems. They will transfer library parts and schematic information in a multivendor software and hardware environment.

The 2.0.0 version of the EDIF standard, which was approved by the EIA this past winter, was approved by the American National Standards Institute on March 14. It is now officially ANSI/EIA RS-548-1987 (SP2086), or ANSI-RS-548 for short. Final approval thus comes less than five years after the EDIF technical committee held its first meeting at the International Conference on Computer Aided Design.

In addition, this year for the first time, the Design Automation Conference is offering paid, full-day tutorials at the end of the conference. The tutorial, "Using EDIF to Describe Electronic Design Data," will be held in Grand Ballroom A of the Marriott Hotel in Anaheim, Calif., on Thursday, June 16, starting at 9:00 A.M. The registration for the tutorial, as well as for the other three tutorials that the DAC is offering, will be open only to those professionals who register for at least one day of the conference. ■



Explore the world's most advanced test site.

ITC '88 Washington, DC

The International Test Conference is where testing's technical and business leaders meet—and reach new frontiers. For almost 20 years, ITC has remained the best place to learn, discover and improve electronics testing techniques of integrated circuits, boards and systems.

In 1987, over 3,500 participants demonstrated the importance of testing in today's marketplace. This year, over 150 technical papers are scheduled for presentation to attendees from around the world.

Those who've made ITC'88 a priority recognize the value of leadership... technical excellence... and practical suggestions. Explore the world's most advanced test site at ITC'88: Important discoveries are waiting in Washington.

While September 12-14 is not far off, there is still time to register. Simply call or write today for full conference details.

International Test Conference, Millbrook Plaza, Suite 104D, P.O. Box 264, Mount Freedom, NJ 07970. September 12, 13, 14, 1988

(201) 895-5260

INTERNATIONAL TEST CONFERENCE 1988

Out in front

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.
IEEE

THE COMPUTER SOCIETY OF THE IEEE

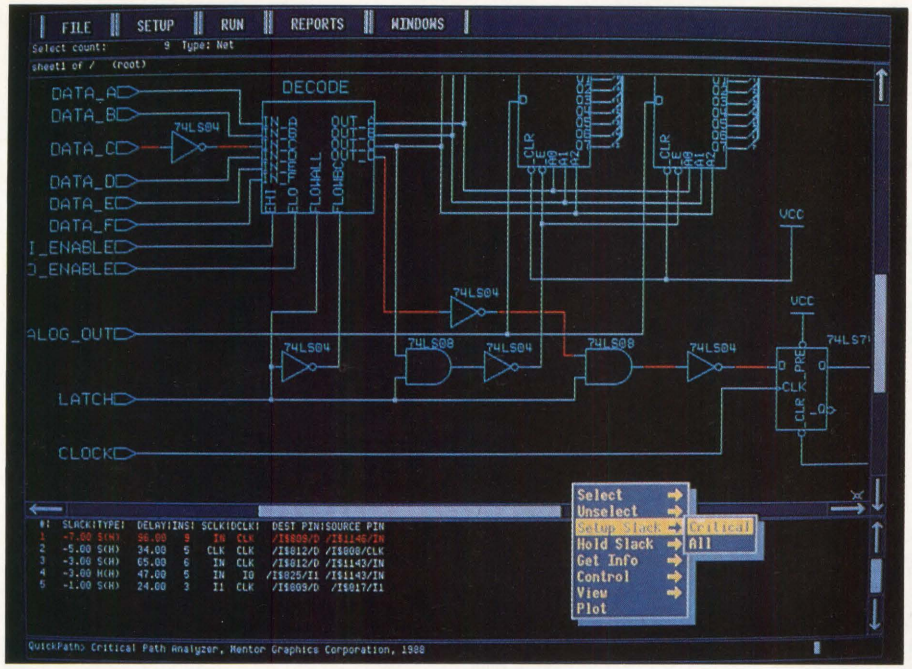


Figure 3. Mentor Graphics' QuickPath static timing-analysis tool examines signal path timing and provides worst-case min/max signal path analysis.

cian. The Gigalogician (see *VLSI Systems Design,* May 1988, p. 64) should be present on the floor of the Design Automation Conference.

Mentor Graphics Corp. (Beaverton, Ore.) has developed a static timing-analysis tool, QuickPath (Figure 3), that examines signal path timing without input stimulus patterns. It performs worst-case min/max analysis along all signal paths in a design, looking at paths between clocked elements and comparing path delays with clock specifications and setup- and hold-time requirements. It also analyzes circuits with multiple clocks and clock phases and removes common ambiguities.

QuickPath works with all types of Mentor systems. It runs in the batch or interactive mode and displays timing violations graphically on schematic diagrams. It is priced at $9,900. However, users trading in Mentor's Tver timing analysis program will receive a 50% discount on the new package.

Mentor also has enhanced its recently announced packaging analysis product, Package Station. The AutoTherm thermal analysis tool now automatically calculates convection coefficients and ambient temperatures for each component on a board, given specifications for airflow rate and temperature.

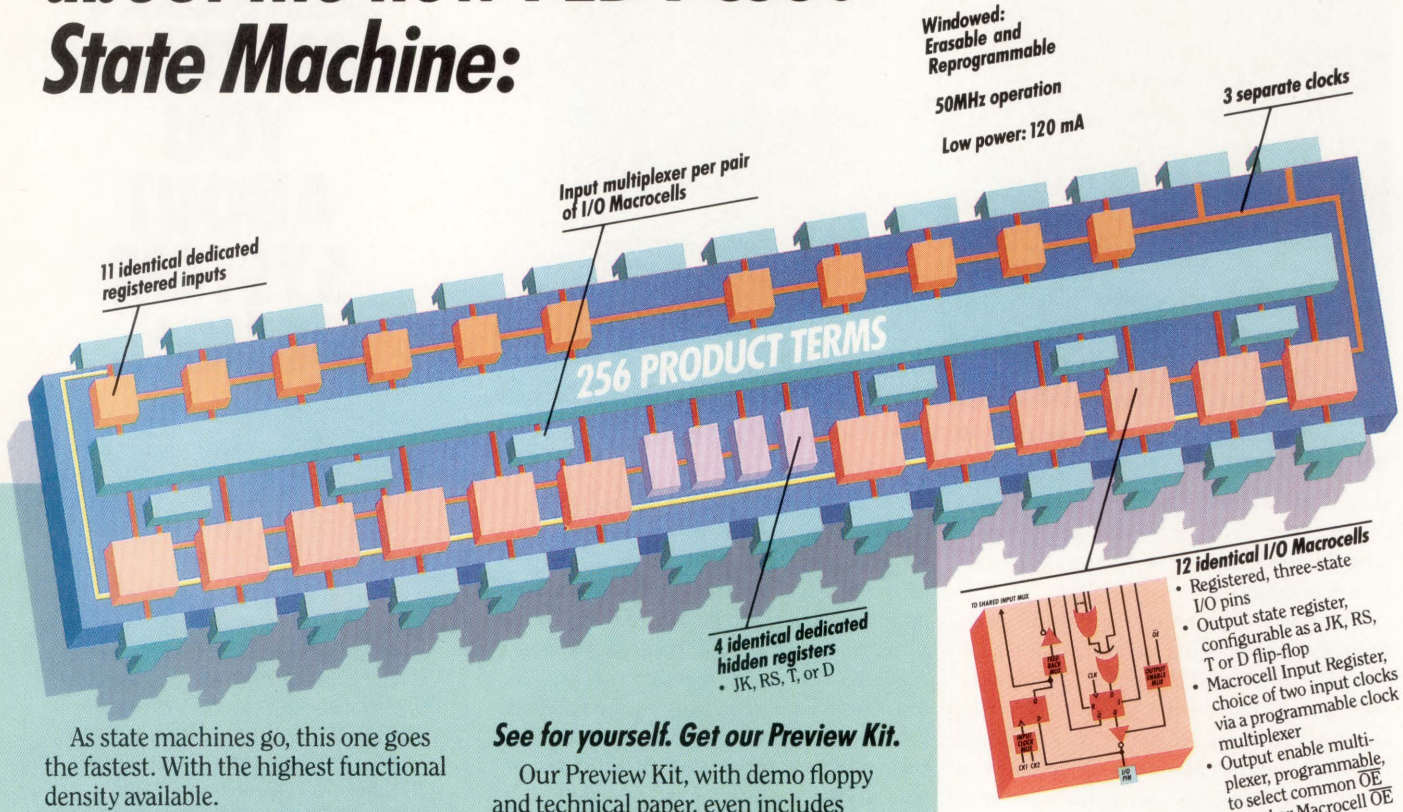
Valid Logic Systems Inc. (San Jose, Calif.) will be unveiling a new line of hardware modeling products in its DAC exhibit this month. Valid Logic has released its "second-generation" hardware modeling system, called Realchip II. Second-generation hardware modelers are characterized by utilization as a network resource (instead of as an adjunct to a workstation) and by "virtual" pattern memory, a technique whereby all available pattern memory can be utilized within any simulation. Less sophisticated hardware modelers partition memory among the resident devices, resulting in the potential for wasted memory if some devices need relatively few simulation patterns. Realchip II has a price tag of $65,000.

An interesting application of Realchip II is the design of computer products around the SPARC architecture originated by Sun Microsystems. Valid now offers a hardware model for two of the SPARC chips offered by Fujitsu Microelectronics Inc. (Santa Clara, Calif.): the MB86900 SPARC integer unit and the MB86910 floating-point controller. Each model is priced at under $5,000.

Quadrupling the capacity in its acceleration products, Ikos Systems Inc. (Sunnyvale, Calif.) now offers a 64K-primitive evaluator board that increases the maximum capacity in its Ikos 1900 simulation system to about 1 million primitives. Ikos has also added four more signals and four more states for signal modeling. The Design Automation Conference will be the first opportunity for many users to see Ikos's recently announced versions of simulation systems based on workstations from Sun (joining products based on PC ATs and Apollo workstations). Ikos also is expected to make an announcement at DAC about behavioral modeling support.

Chip designers can get a look at the Desire IC design translation tool from Do-

Why high performance designers are so excited about the new PLD 7C330 State Machine:



As state machines go, this one goes the fastest. With the highest functional density available.

A system that lets you design state machines that can execute control sequences at a full 50MHz without even breathing hard.

High performance designers are excited by the extensive features, led by the richness of 256 product terms.

By this part's ability to deliver twice the performance and density of previous generation PLDs, with about half the power.

By the ease of programming, using popular programming languages and machines.

And by the ability to design the highest performance state machines, with 1000 to 1500 gate functionality, using reprogrammable PLD technology.

See for yourself. Get our Preview Kit.

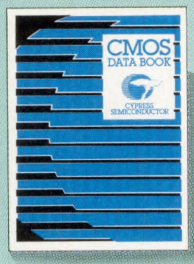
Our Preview Kit, with demo floppy and technical paper, even includes application ideas, like our step-by-step design example showing you how to create a 16 x 4 programmable multiplexer using a single CY7C330.

You'll also get the databook that has the information you need on this great part, and on ALL our high performance parts.

CMOS high speed SRAM.
CMOS high speed PROM.
CMOS high speed PLD.
CMOS high speed Logic.

PLD 7C330 Preview Kit and DataBook. Yours free, for a toll-free call.

DataBook Hotline:
1-800-952-6300
Ask for Dept. C103
1-800-423-4440 In CA
Ask for Dept. C103
(32) 2-672-2220 In Europe
(416) 475-3922 In Canada



CYPRESS SEMICONDUCTOR

Cypress Semiconductor, 3901 North First Street, San Jose, CA 95134, Phone: (408) 943-2666, Telex: 821032 CYPRESS SNJ UD, TWX: 910-997-0753.

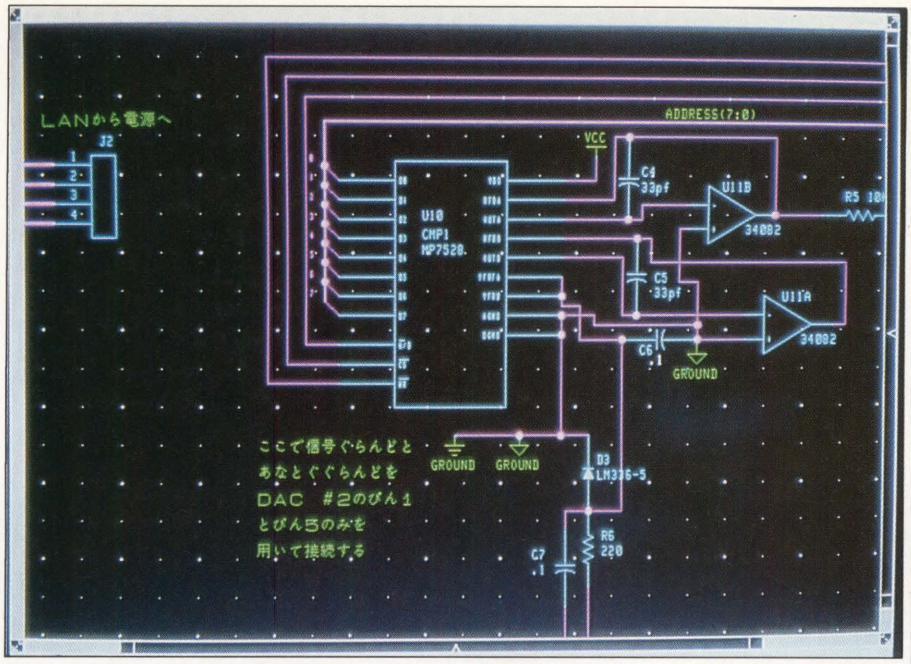


Figure 4. Hewlett-Packard's Japanese version of its Design Capture System displays schematics and generates reports with kanji characters.

sis GmbH (Dortmund, West Germany). Desire accepts an IC design file in the GDS II format, a set of target design rules, and specifications for special handling of user-defined objects. The program translates the design, even those including nonorthogonal and circular structures, to the new design rules. Dosis will also show a mixed-level simulator called Dacapo-III.

Designers of microprogrammed systems should take a look at the Step-50 microprogram development station from Step Engineering (Sunnyvale, Calif.) Its 160-bit-wide writable control store (4K deep) is built with 25-ns RAMs. A real-time trace memory stores 4K bits for 80 channels running at 20 MHz, and a built-in state analyzer provides clocking and breakpoint controls with multiple breakpoint trigger levels. Controlling the PC AT–based station through a menu running under Microsoft Windows, users create microprograms with Step's Metastep language. The Step-50 costs $15,000.

Logic Automation has developed a behavioral model of the 88000 RISC family from Motorola. The SmartModels for the 88100 processor and the 88200 cache and memory management unit are available for operation with simulators from Mentor Graphics, Valid Logic Systems, and Gateway Design Automation.

Logic Automation also has announced upcoming support for Daisy Systems' and Hewlett-Packard's simulators. It is developing translators that will create models for those simulators from its SmartModel library. Daisy is adding an interface to the behavioral models for its simulation environment; the interface and SmartModel libraries are scheduled for release in January. A 12-month subscription for the SmartModels for HP's Electronic Design System will run $6,000; deliveries will begin in the fourth quarter of this year.

■ HP Becomes Multilingual

Hewlett-Packard Co. (Fort Collins, Colo.) will be attempting to woo Japanese visitors at the Design Automation Conference with a new version of its design capture system that provides schematics with kanji characters (Figure 4). The kanji version will also generate reports such as bills of materials in the Japanese characters. To demonstrate its worldwide commitment, HP will also demonstrate a Spanish version of its HP ME Series 10 design and drafting system and a German version of the HP Engineering Graphics System.

HP's support of GenRad's latest Hilo simulator and Logic Automation's SmartModel behavioral models will be featured at the company's booth. In addition, the company will roll out its revised HP printed circuit design system, which adds a new rip-up–and–retry automatic router that rips up only traces that were placed in the current routing run. This approach protects critical routes that were manually placed in previous routes. The revised tool kit will include adaptive gridding to allow traces to be bent around obstacles and between different grids; real-time panning and zooming; enhanced SMT capabilities, such as board flipping, larger pin counts, and metric support; and a link to HP's ME30 solids modeling system for 3D visualization of PCB layouts. ■

—David Smith

Continued from page 12

in writing an equivalent algorithm is justified by an increase in throughput. Algorithms give the compiler a potential performance advantage that is important in an iterative design environment.

Like Socrates, the Design Compiler accepts netlists, truth tables, and equations similar to Boolean expressions. Optimization is aided by three types of RC timing models and several modes of timing analysis. The output is expressed in netlists or schematics for gate arrays and standard-cell ICs from LSI Logic and Harris Semiconductor. It runs on platforms from Apollo, Digital Equipment, and Sun Microsystems. A single-user license costs $35,000.

The SilcSyn program from Silc Technologies breaks a design expressed in behavioral statements (as well as the other input formats described for the Design Consultant) into control and data path resources of a "data control machine." Control sections are optimized through minimization, and data path functions that can be shared among portions of the design are identified. The design is then mapped into cells for a specific library. The use of a behavioral description language and a behavioral simulator for functional verification are unique to SilcSyn.

NCR Microelectronics (Fort Collins, Colo.) is expanding its expert-system-based tool suite, called the Knowledge-Based Engineering Environment (KE^2), with the SilcSyn-based NCR Design Synthesis. Like NCR's expert-system tool, the Design Advisor, NCR Design Synthesis runs in a window in the Mentor Graphics design environment. The two tools even share some common design rules, so they complement each other. The output is described in standard and compiled cells from NCR's 2.0- and 1.5-μm libraries, which also include analog cells. As a service, the program runs $5,000 per functional block; it's $51,500 for a tool license.

■ Optimization for PLDs

Logic synthesis tools for PLDs, including DS23 and the FutureDesigner from Data I/O FutureNet (Redmond, Wash.) don't accept RTL or behavioral descriptions, although we can expect upgrades to these types of entry formats in the future. At this point, such formats aren't necessary because the majority of users of PLDs are implementing relatively simple designs that are adequately expressed in Boolean equations or state machines.

DS23 assists designers who are converting designs built in TTL logic and PLDs into generic logic proper for Xilinx's programmable gate arrays. The user can build schematics using Xilinx's generic models or TTL function models. DS23 accepts completed schematics as netlists from schematic entry programs from several CAE vendors, including FutureNet, Viewlogic Systems, Daisy Systems, Valid Logic Systems, and Case Technology. Other portions of the PLD designs can be entered in the PALASM format or as logical equations.

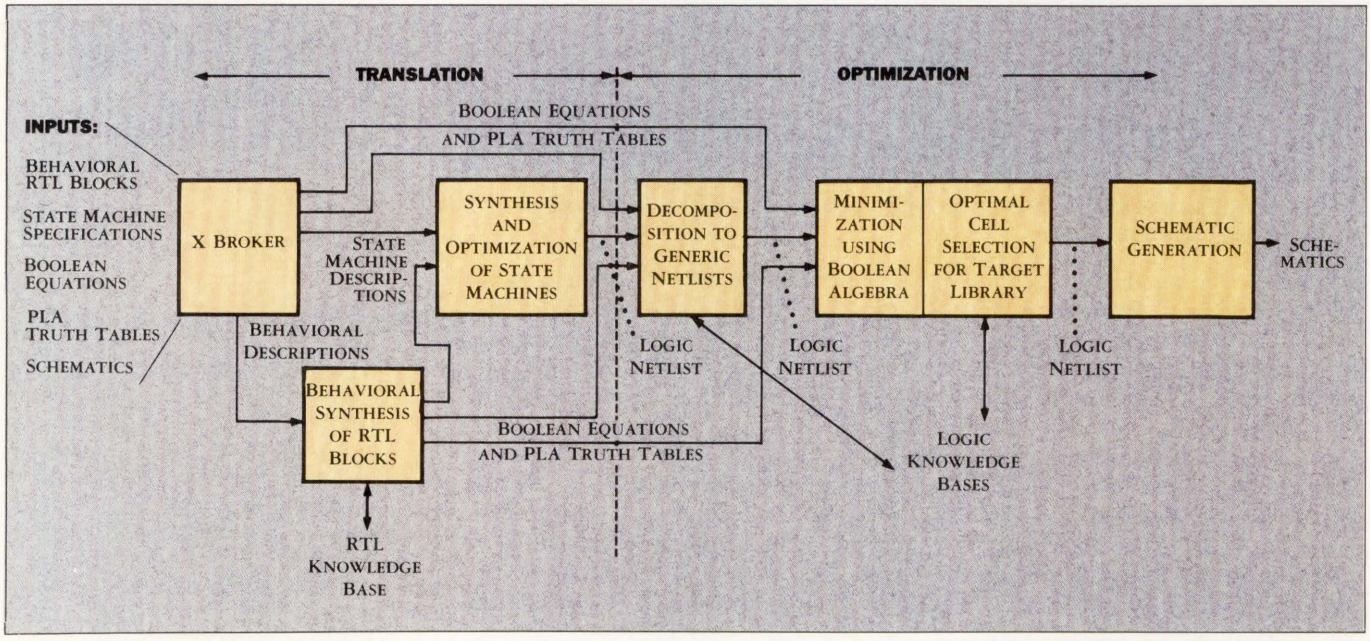
Designs are optimized using algorithms based on the Espresso and MIS software from the University of California at Berkeley. Subsequent tools within Xilinx's tool suite simulate, place, and route the design on the LCA family of devices.

■ Front End to Compilation

Logic synthesis is the front end that silicon compiler tools should have had. Complaints about the first silicon compilers—that they enforced a particular design style, that they required some IC-level expertise, and that they produced results that were inferior to custom designs—highlight the potential benefits of logic synthesis. Designers can create designs in the format they are most comfortable with or in the format that is most efficient for the type of circuit they are creating, and because optimization and mapping to a semicustom IC is automatic, they need know little about the specifics of IC design. Vendors of logic synthesizers offer numerous case studies in which, in terms of size, power consumption, and propagation delay, synthesized circuits were as good as, or better than, circuits produced by chip designers.

The efficacy of logic synthesis, in comparison to nonsynthesized methods, depends a good deal on the type of design. Logic synthesis can, however, always produce designs much faster, enabling designers to make more iterations on a design. Synthesis proponents point out that such iterations provide the insight for designers to find better solutions to their system requirements. ■

—David Smith



Logic synthesis consists of two steps: translation (on the left), which replaces high-level descriptions with lower-level implementations, and optimization (on the right), which tunes the design for size, power, and timing constraints of the system. Trimeter's Design Consultant adds the translation step to its optimizing Logic Consultant.

Continued from page 8

processors, printed circuit boards, wafer-scale assemblies, hardware and software, process and test data management, fault modeling and simulation, design for testability, and computer-aided test generation. For more details, contact Doris Thomas, Executive Secretary, International Test Conference, Millbrook Plaza, Suite 104D, P.O. Box 264, Mount Freedom, N.J. 07970. (201) 895-5260. ■

2ND INTERNATIONAL WORKSHOP ON VLSI DESIGN

December 15–18
Bangalore, India

This year's workshop will be held in Bangalore, which is also known an the "Silicon Valley" of India. It will feature tutorials and panel discussions, as well as technical presentations. Topics will include design rule checking, layout and routing, testing and test program generation, CAE/CAD systems, implementation algorithms, logic and circuit simulation, timing analysis and verification, silicon compilation, fault modeling, and fault simulation. By August 1, authors should submit a manuscript, not exceeding 4,000 words, that clearly explains the problem, solution, and final results. To submit a manuscript or to obtain additional information, contact Dr. Ravi M. Apte, Valid Logic Systems, M/S 01, 2820 Orchard Pkwy., San Jose, Calif. 95134. (408) 432-9400. ■

DEXPO EAST '89

February 8–10, 1989
New York, N.Y.

The theme of the third annual DEXPO East conference is "The Corporate VAX." Papers are invited on topics that include symmetrical multiprocessing, networking in multivendor environments, on-line transaction processing, distributed databases, distributed system services, storage and output technology for corporate systems, VMS vs. Unix, open and closed architectures, Apple-DEC computing, graphics, MIS programming environments, managing PC-VAX environments, and end-user programming tools. By July 31, interested authors should submit a 150-word abstract, session length description, audio-visual requirements, and biographical data to Sandy Krueger, Conference Director, Expoconsul International Inc., 3 Independence Way, Princeton, N.J. 08540. Additional information may be obtained by calling Mr. Krueger at (609) 987-9400. ■

IEEE PHOENIX CONFERENCE ON COMPUTERS AND COMMUNICATIONS

March 22–24, 1989
Scottsdale, Ariz.

This international conference provides a forum for the presentation and exchange of current work on a wide variety of topics in computers and communications. Topics of interest include theory, design, implementation, applications, and evaluation aspects in such areas as array processors, data-flow architecture, storage devices, fault tolerence, operating systems, performance evaluation and measurement, integrated software development environments, software tools, design specification and code generation, real-time systems, database systems, artificial intelligence, computer applications, VLSI/VHSIC, and networking technology. Interested authors should submit five copies of both an abstract and a complete paper by July 15. For more details about submitting a paper, contact Dr. Antonio Pizzarello, Program Chairman, Honeywell Bull Inc., P.O. Box 8000, M/S Z-10, Phoenix, Ariz. 85066, (602) 862-5281. Those interested in presenting a tutorial at this conference should contact Dr. Paul Jorgensen, Computer Science Dept., Arizona State University, Tempe, Ariz. 85287. (602) 965-6385. ■

EUROPEAN TEST CONFERENCE '89

April 12–14, 1989
Palais des Congrès
Paris, France

Preceded by a day of test-related tutorials, ETC '89 will provide a forum for the discussion of technical developments in electronic product testing. Papers are invited on topics that include application-specific ICs, microprocessors, memories, surface-mount technology, fault models, simulation, fixturing, failure mechanisms, automatic test generation, hardware and software, contactless probing, diagnostic techniques, scan design, built-in self-test, data collection and analysis, testability analysis, silicon compilers, wafer-scale technology, and expert systems. By August 1, authors should submit a 50-word abstract and either a 500-word summary or a full paper, including figures, that does not exceed 10 pages to ETC '89 c/o SEE, 48 Rue de la Procession, 75724 Paris Cedex 15, France. Further information may be obtained by contacting Colin Maunder, British Telecom Research Labs, Martlesham Heath, Ipswich, Suffolk IP5 7RE, U.K. Phone: (44) 473-642706. ■

Some managers are never satisfied. We need four of them.

We're running this ad because we have four immediate openings in the Design and Analysis Division (DAD) of Mentor Graphics.

The people who do well here are looking to break out of the pack. You know the type. Rule Changers. Pathfinders. Pioneers. People with a vision and passion so intense it makes others enthusiastic.

That's why to succeed in these opportunities you'll need to share our vision of design automation. A vision of a world-class systems solution to design tools. A vision where application and integration are equally important.

If you live for challenge, explore these opportunities at Mentor Graphics' headquarters in the Portland, Oregon area:

Logic Simulation Manager

You have professionally honed management skills. A strong understanding of the logic simulation market. Now come set the technical direction for the development of our QuickSim logic simulator. Some enhancements to our existing product will have to be made, of course. But mainly you'll oversee the development of new simulation tools. An MSCS or MSEE and a minimum of 6 years' industry experience including hands-on development of software products are required. At least 2 years in a managerial role are essential.

Engineering Manager

Graphical Design Capture

You've successfully brought an electronic CAE/CAD schematic capture system to market. So tell us about it. You may qualify to lead the design and implementation of a product to capture system, architectural and logic schematics. To do this, you must have the ability to assemble a design team and help team members develop a product. Your innovative approaches to design capture will find an attentive audience here.

Engineering Manager

Design Interfaces

Only those with proven project management experience and software talents need apply. That's you if you can successfully lead our Design Interfaces team in the development of software applications. You'll also be working with consultants which means you'll negotiate the functional design, test and acceptance specifications and schedules. You will be responsible for sticking to that schedule and making sure everyone else does. Your background must include experience with database and interface development in multi-organization or multi-company teams.

Engineering Manager

Design Creation

Taking a graphical design capture project, you'll supervise its evolution, including architecture and planning for system design aids, graphical behavioral modeling, logic synthesis, system design process managers and more. To make this a success, you'll need a thorough understanding and experience with the development of several delivered CAE/CAD products. Software expertise and excellent interpersonal skills are crucial. Your ability to visualize future developments will be key.

Satisfied?

We hope not. Because we want you to learn more about the many benefits to a career with Mentor Graphics. Located in Beaverton, Oregon, we offer you everything from a private office to an affordable house — with a very short commute between the two.

To apply, please send your resume to Professional Staffing, Mentor Graphics Corporation, 8500 S.W. Creekside Place, Beaverton, Oregon 97005-7191. Principals only.

We are an equal opportunity employer committed to affirmative action.

Mentor Graphics®

The whole world designs on us.

MOS/VLSI OPPORTUNITIES

To Meet The Sharp Upswing In Demand For Circuits, We Have Excellent Career Opportunities At All Levels At Our Suburban, Long Island Headquarters!

Standard Microsystems Corporation is a leading developer and manufacturer of advanced MOS/VLSI circuits for computer peripheral, communications, semicustom and full custom applications. High levels of funding in new product and semiconductor development in the past three years as well as an on-going emphasis on world-wide business development have positioned Standard Microsystems at the forefront of our rapidly growing markets. We are seeking highly qualified individuals for the following positions:

MANAGEMENT

- **Director of Computer-Aided Engineering**
- **Manager of Product Engineering**
- **Area Sales Manager** (San Jose based—11 state territory)

ENGINEERING

- **VLSI (ASIC) Design Engineers**
- **Product Engineeers**
- **Field Applications Engineers** (San Jose, Europe, or Far East)
- **CAE Engineers**
- **Test Engineers**
- **Process Engineers**
- **Failure Analysis Engineers**

You must have the appropriate educational background with a stong academic record (a master's degree is preferred) plus significant achievements in the area in which you are interested.

Standard Microsystems, one of the most successful and stable companies in the industry, offers excellent salaries, a full range of insurance coverages, financial benefits programs such as tax deferred stock purchase and profit sharing, bonus, and stock options plans; comprehensive relocation assistance; advancement opportunities; a progressive, employee-oriented work environment; and job security. You and your family will also appreciate our desirable Suffolk County, Long Island location, approximately 50 miles from New York City and all its advantages while close to the finest beaches and boating plus educational institutions in the country. To apply, please send your confidential resume, including your salary history and the area of your interest to: **A. P. Solowey, Manager, Recruitment, Dept. V, Standard Microsystems Corporation, 350 Kennedy Drive, Hauppauge, New York 11788.** SMC is an equal opportunity employer m/f.

STANDARD MICROSYSTEMS CORPORATION

The One To Watch—
In The Place To Be

ADVERTISERS' INDEX

This index is provided as an additional service. The publisher does not assume any liability for errors or omissions.

VLSI Systems Design

PUBLISHER
NORM ROSEN
DIRECTOR OF MARKETING
TIM L. TOBECK

DISPLAY ADVERTISING
JONNO WELLS, District Manager
1054 Saratoga-Sunnyvale Road
San Jose, CA 95129
(408) 252-6191
KAREN TALLARIDA
7 Wells Avenue, Newton, MA 02159
(617) 244-5333
WALTER OLSON
DAVID JANOFF
600 Community Drive, Manhasset, NY 11030
(516) 562-5000
CLAIRE FLORA
5151 Beltline Road, Dallas, TX 75240
(214) 661-5673
WILLIAM HIGGENS, III
MARY SHUTACK
225 N. Michigan Ave, Chicago, IL 60601
(312) 565-5943
RICHARD CARLISLE
4500 SW Kruse Way, Lake Oswego, OR 97035
(503) 636-7694
KENNETH D. BEACH
209 Arlene Court, Newark, DE 19702
(302) 368-3116

INTERNATIONAL ADVERTISING
DANIEL H. LEEDS, Vice President
(516) 562-5000
STEVE DRACE, International Sales Manager
FAX 852-5-861-0668
PacificMedia, Inc.
Fax: Tokyo 252-2780 Telephone: 256-8456
Telex: J25472
Young Media, Inc., Seoul, Korea
Telephone: 756-4819 Fax: (02) 757-5789
Mr. Owen Wang
Ace Marketing, Inc.
Telephone: 751-3636 Telex: 78514142

CLASSIFIED ADVERTISING
NEW YORK (516) 562-5000
SANDRA HERMAN/National Manager
MICHAEL ZERNER, East Coast District Mgr.
NEW ENGLAND N.Y. N. Central District Mgr.
KATHY HEALY, Southeast, South Central Space Sales
SONJA WONG, Adverstising Coordinator
FELICIA SABATINI, Graphics
SAN JOSE (408) 252-6191
LAURIE MUNCE, West Coast District Mgr.
DIRECT MARKETING SERVICES
JOANNA BRANDI, Manager
PETER CANDITO, Sales Manager
ARLENE BROWN, List Sales Representative
(516) 562-5000
MARKETING SERVICES
DORIANNE WALTHER-LEE/Director
CUSTOMER SERVICE/DISPLAY ADVERTISING
LYNDIANE HERVEY, Group Manager
TRACY RENK, Ad Coordinator
(516) 562-5000
PROMOTION
GENEVIEVE HIGGINS
CIRCULATION
GRACE MONAHAN, Director of Operations
NANCY MERRITT, Circulation Director
ROBERT ANDERSEN, Circulation Manager
SUBSCRIPITION SERVICES
(516) 562-5882
RESEARCH
DANIEL R. CAMPBELL, Electronics Group Manager
REPRINTS
BESSIE RHODEN
(516) 562-5000

CMP ELECTRONICS GROUP
KENNETH D. CRON, Vice President/Group Publisher
Electronic Buyers' News
Electronic Engineering Times
VLSI Systems Design

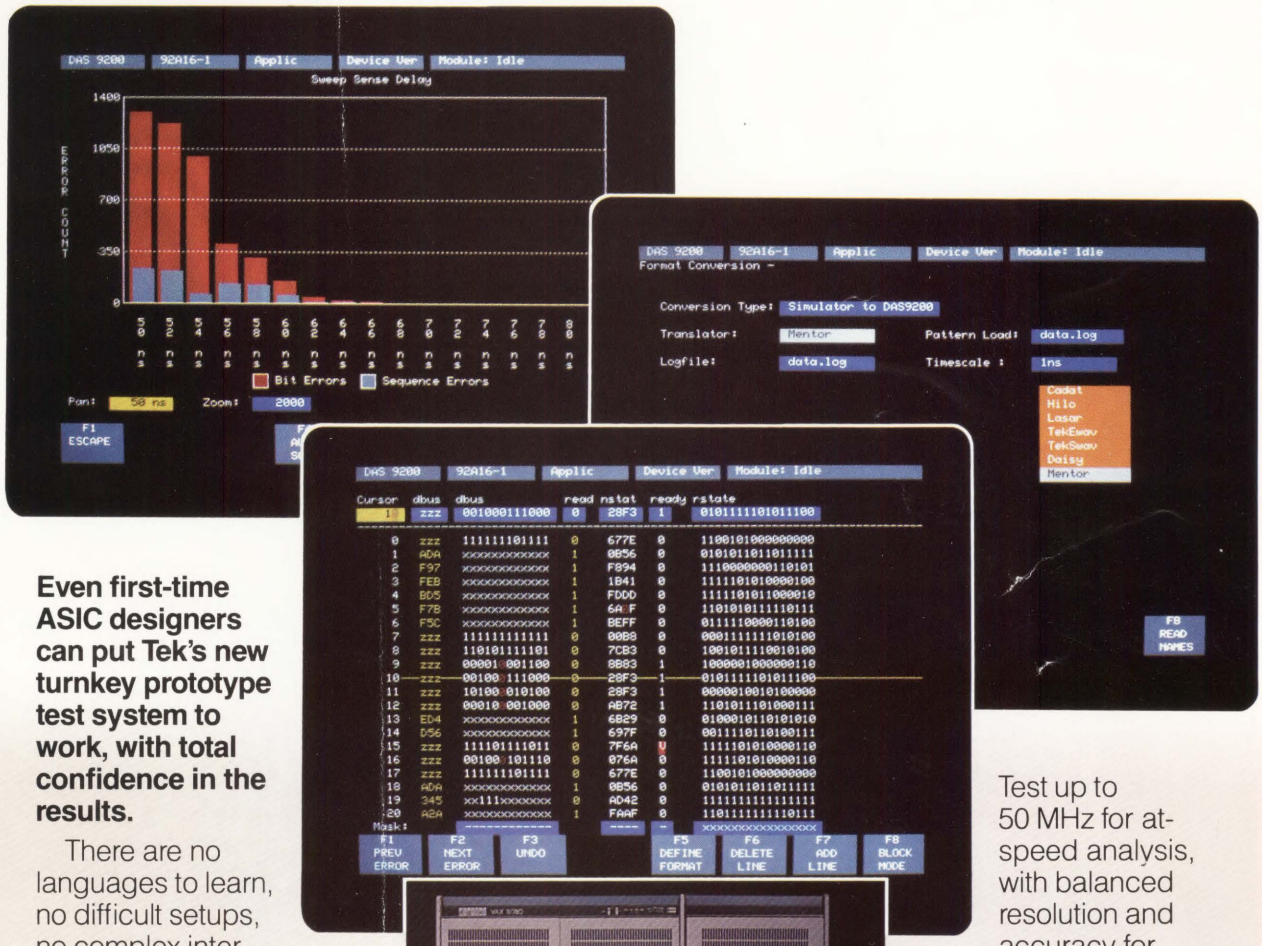
CMP Publications, Inc.
MICHAEL S. LEEDS, President
PEARL TURNER, Vice President/Treasurer
DANIEL H. LEEDS, Vice President

LILO J. LEEDS, GERARD G. LEEDS
Co-Chairpersons of the Board

NANCY GRABOW
Assistant to the President

BPA

HOW TO SUCCEED IN ASIC PROTOTYPE VERIFICATION. FIRST TIME. EVERY TIME.



Even first-time ASIC designers can put Tek's new turnkey prototype test system to work, with total confidence in the results.

There are no languages to learn, no difficult setups, no complex interpretations, no incomplete answers.

You simply advance from one menu to the next. Use the same test vectors you developed during design. Debug with highly flexible, interactive pattern editing and timing characterization routines.

Test vector data from your logic simulator is translated and checked for tester compatibility automatically.

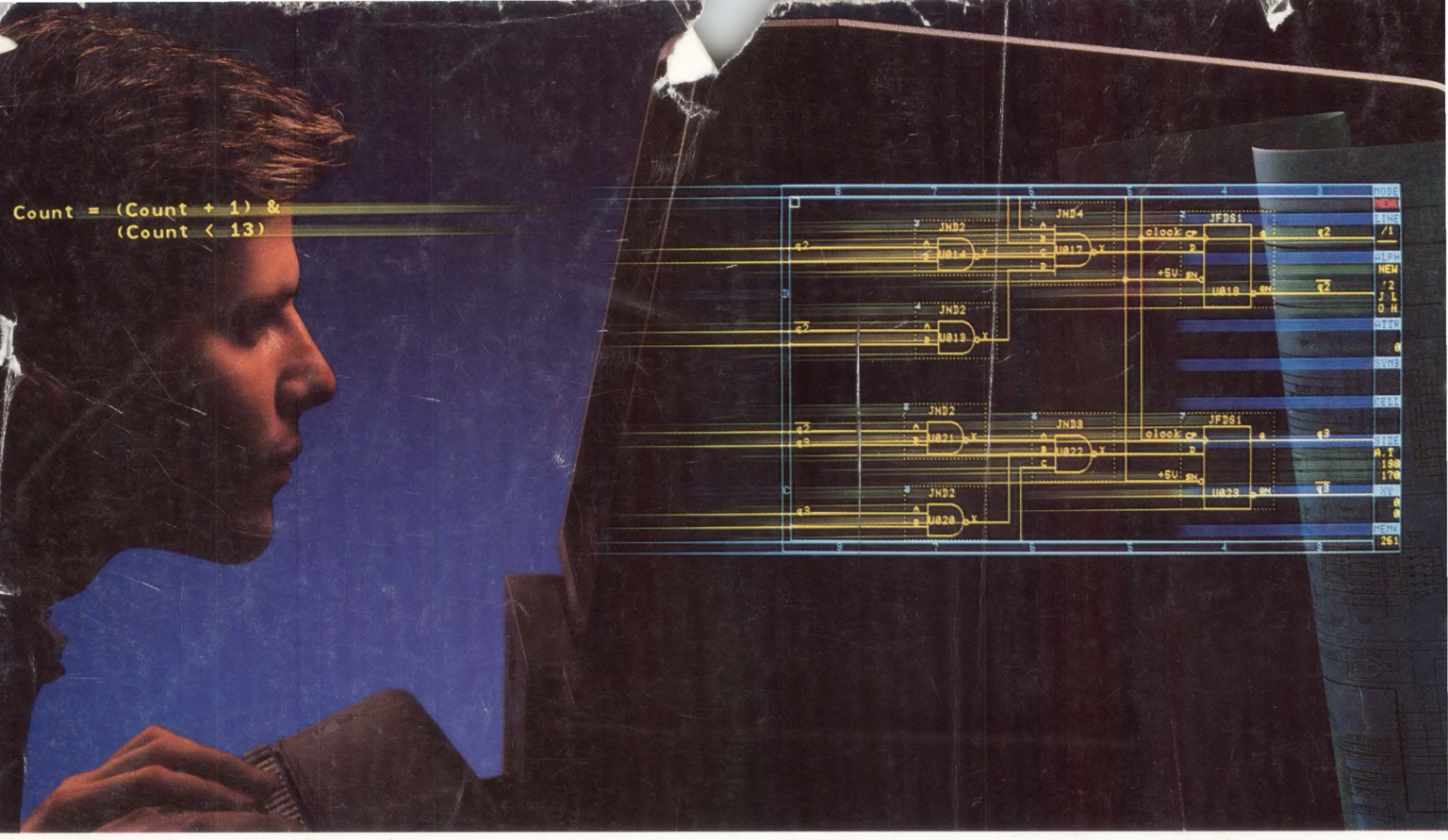
Test up to 50 MHz for at-speed analysis, with balanced resolution and accuracy for unprecedented signal measurement quality. The test fixture adapts to your DUT quickly.

The system can be easily reconfigured to help with many other complex problems—including integrating your ASIC onto the circuit board.

To learn more about Tek's DAS9200 Personal ASIC Verification System, call: **1-800-245-2036**
In Oregon, 231-1220

Tektronix®
COMMITTED TO EXCELLENCE

CIRCLE NUMBER 25



AUTOMATIC SCHEMATIC.

FUTUREDESIGNER: DRAW LESS. DESIGN MORE. Introducing FutureDesigner™ — the only advanced design entry workstation that lets you describe your circuit in compact, high-level terms and create more complex designs faster. FutureDesigner's flexible, new techniques encourage creativity and experimentation, helping you produce innovative products quickly and more accurately.

MULTIPLE DESIGN ENTRY MODES FOR SPEED AND FLEXIBILITY. Describe your circuit with any combination of structural and behavioral representations. Use schematics to enter the structural portions of the design, such as data paths in a memory array. For portions easier to describe behaviorally, like sequencers or decoders, simply enter equations, truth tables or state diagrams using on-screen input forms.

ADVANCED DESIGN VERIFICATION HELPS YOU GET IT RIGHT THE FIRST TIME. For the behavioral portions of your design, use FutureDesigner as a "what if" tool to try different design approaches. Immediately verify that your circuit works as you intended. For the structural portions, design check tools detect and help you correct connectivity and other common design errors. Together these features significantly shorten the design iteration cycle.

LOGIC SYNTHESIS CONVERTS YOUR EQUATIONS INTO SCHEMATICS. Once you've entered equations, state diagrams or truth tables, FutureDesigner's logic synthesizer eliminates redundant circuitry and optimizes your design for size/speed trade-offs. FutureDesigner is the only design entry workstation that will then automatically produce the correct schematics and integrate them with the total structural design.

MORE CHOICES IN TECHNOLOGIES, VENDORS AND SYSTEMS. FutureDesigner is technology independent. Choose the most convenient mix of TTL, PLDs, gate arrays or other ASICs from a wide range of semiconductor manufacturers. You can easily migrate from one technology to another without redesign.

FutureDesigner output is an industry standard, widely accepted by engineering service bureaus and semiconductor vendors. You'll also have access to both FutureNet® and other CAD systems for simulation and PCB layout.

Call us today and learn how a FutureDesigner workstation gives you the flexibility and accuracy to design innovative products faster.

1-800-247-5700
Dept. 125

Data I/O Corporation 10525 Willows Road N.E., P.O. Box 97046, Redmond, WA 98073-9746, U.S.A. (206) 881-6444/Telex 15-2167
FutureNet 9310 Topanga Canyon Boulevard, Chatsworth, CA 91311-7528 (818) 700-0691/Telex 910-494-2681
Data I/O Canada 6725 Airport Road, Suite 302, Mississauga, Ontario L4V 1V2 (416) 678-0761/06968133
Data I/O Europe World Trade Center, Strawinskylaan 633, 1077 XX Amsterdam, The Netherlands (20) 622866/Telex 16616 DATIO NL
Data I/O Japan Sumitomoseimei Higashishinbashi Bldg., 8F, 2-1-7, Higashi-Shinbashi, Minato-ku, Tokyo 105, Japan (03) 432-6991/Telex 2522685 DATAIO J

FutureNet
A Data I/O Company

CIRCLE NUMBER 26