PART TWO
DECADE
90
The future of
system design

Developing CAE tools target top-down design of complex systems

he next decade's billion-transistor ICs vill rapidly make manual system-design nethods obsolete. If engineers are to tackle arger, more complex electronic designs, they will need automated design tools. To-lay's CAE tools help engineers develop sys-tem components. Tomorrow's tools will help lesigners develop entire systems.

Steven H Leibson, *Regional Editor*

The growing complexity of standard and custom ICs is making the task of designing electronic systems ever more arduous. Many vendors now offer CAE tools that elp engineers manage such complexity, but today each ol handles, at best, only part of the job. These CAE roducts help you design ASICs or pc boards, or aid in e coding of software, for example, but no one product n yet help you do the total system design from top to ttom. Furthermore, today's automated IC-design ols can't transform inexperienced engineers into chip signers, and merely using a pc-board design system esn't ensure that you'll come up with a good board sign.

Over the next decade, however, CAE-tool vendors ll start distilling design expertise into their tools so that inexperienced engineers can design noncritical parts of systems, and seasoned engineers can vastly improve their productivity. At the same time, system-level CAE tools will give designers better control over the definition and attainment of project goals.

Today's CAE tools work

Although most engineers still don't use CAE tools for any kind of design, a few designers are already using today's tools to create very complex systems. For example, Sequent Computer Systems (Portland, OR) used CAE tools from Mentor Graphics (Beaverton, OR) to develop two backplanes, three ASICs, and three pc boards for its Symmetry Series multiprocessor computers. The Symmetry computers are compatible with Sequent's earlier Balance Series.

Each Symmetry CPU board incorporates two processor subsections based on Intel (Santa Clara, CA) 80386 µPs. Sequent developed three ASICs for the Symmetry processor board: a 5000-gate cache-memory controller based on a gate array, a 10,000-gate bus-interface controller also based on a gate array, and a 14,000-gate data-path bus controller built with standard-cell technology. The company also used a 6000-gate serial-link-controller IC that it had designed for its earlier Balance Series computers. The company's engineers used Mentor Graphics tools to design all four of these ASICs.

Paul Gifford, manager of central systems engineering at Sequent, says the Mentor Graphics ASIC-design

Today's CAE tools are helping engineers design and simulate complex systems. Engineers at Apollo Computer Inc used Mentor Graphics tools to develop the CPU board for the Apollo's DN4000 Series workstations.

tools are "pretty robust" for developing chips at the 9000-gate level and are quite capable of developing ASICs having as many as 20,000 gates. Beyond that number of gates, he says, "the tools start to let go."

According to Gifford, Sequent also used Mentor Graphics tools to design the dual-processor boards for the Symmetry Series. Engineers performed simulations on about half of the processor board's circuitry, which is the equivalent of about 70,000 gates. A typical design iteration, including simulation, evaluation, and design editing, required about 24 hours.

Expanding the backplane

Sequent's engineers used MSpice analog simulation on the Mentor systems to help develop the Symmetry's backplane. They used bus-driver and -receiver models from the IC vendors and simulated several termination schemes to optimize the backplane's design. As a result, Symmetry's backplane has 26 slots. If the engineers hadn't been able to use analog simulation to verify the feasibility of the bigger backplane, says Walt Mayberry, Sequent's director of engineering, time-to-market pressures would have forced the company to use a more conservative 16-slot backplane, which would have limited the capacity of the machine.

Another computer manufacturer, Apollo Computer (Chelmsford, MA), also employed Mentor Graphics tools to design the processor board of one of its latest products, the DN4000 workstation. The DN4000's CPU board incorporates Motorola's (Phoenix, AZ) 68020 μP, 68881 floating-point coprocessor, and 68851 memory-management unit (MMU), plus other LSI components and 40 programmable-logic devices (PLDs).

Assorted device models

Apollo engineers used an assortment of device models to simulate the CPU board. They used hardware models for the 68020 and 68881, behavioral models for some of the other VLSI parts on the board, gate models for glue logic, and downloaded programs for PLDs and PROMs. Using this wide assortment of models, the designers were able to make design changes and perform simulations on the revised design in about 30 minutes.

Ted Elkind, a section manager in charge of Apollo's CAE logic-design tools group, estimates that the company saved one to two months of the 6-month prototyping cycle by using these CAE tools. Engineers brought the system debugger up on the CPU board in one day—for an earlier project, that task required about a week—and they had the operating system running on the new system in three weeks, instead of the usual three months.

Some shortcuts cost time

Apollo could have saved even more time by making better use of the simulation tools, Elkind notes. For example, because of time and resource constraints, Apollo didn't create hardware models of the 68851 MMU and the cache-tag comparator. Instead, the engineers wrote a simple, "pass-through" model of the MMU and a behavioral model of the cache-tag comparator.

When debugging the newly fabricated CPU board, the engineers spent a disproportionate amount of time finding and fixing design flaws centered around the MMU, and they discovered that the cache-tag comparator didn't operate according to its specifications. The comparator vendor had to redesign the part to meet the specifications and, fortunately, the revised part was ready in time for Apollo's first production run. Gifford says that better modeling of these two parts would have revealed these problems earlier in the design cycle and would have further reduced the time required to debug the new system.

Because of their experiences, both Sequent and Apollo strongly advocate the use of CAE development tools. Both companies intend to make even more use of simulation in future projects to further reduce debugging efforts and overall development time. However, the 24-hour design cycle required for the Sequent design shows that workstations can have a tough time coping with complex system design. In fact, most engineers would even consider the 30-minute time that Apollo achieved to be far too long.

Expert design assistance

These two examples illustrate the way that today's CAE tools can help engineers who have design expertise to create new designs more quickly. The CAE tools of the 1990s will go a step further: Just as general-purpose software, such as a spreadsheet, allows casual computer users to perform complex computational tasks without programming, the next decade's CAE tools will allow system-design engineers to create ICs without becoming IC designers.

Dr Prabhu Goel, president of Gateway Design Automation Corp (a Westford, MA, CAE-tool vendor) believes a dichotomy will emerge in IC-design CAE tools. He believes that some tools will mask the details of IC design so that inexpert chip designers can easily create noncritical portions of a chip. The critical portions of a chip—the ones that directly affect the IC's performance—will still be created by expert IC designers, who will use tools that allow for fine sculpturing of silicon, Goel says.

Artifical intelligence is already providing one way for engineers with some experience to become better IC designers. For example, NCR's (Fort Collins, CO) Design Advisor, an expert-system tool, can review IC designs and suggest improvements. The design rules built into the software were gathered from NCR's own IC engineering staff. The Design Advisor checks for several different types of design problems and can offer a wide range of advice concerning speed, testability, manufacturability, and silicon-area usage **(Fig 1)**. To build the Design Advisor, NCR used the Proteus expert system developed by the MCC (Microelectronics and Computer Technology Corp) Artifical Intelligence Laboratory in Austin, TX.

But engineers use CAE software for more than just IC design. Some CAE tools aid in the development of pc boards or PLD programs. Unfortunately, the databases for these design tools are generally incompatible, creating an electronic Tower of Babel. One solution companies employ to overcome this mass of incompatible data is translation. For example, Aida Corp (Santa Clara, CA) offers software that performs file-format conversion, which allows Aida's system-design tools to accept design information from Mentor, Viewlogic, Tegas, and Hilo systems.

A standard language will allow for interchange

Translation is not the solution to the basic problem of incompatibility: If each CAE-tool vendor were to write translators for selected data-file formats, the industry would end up with an incomplete solution at a horrendous cost in development time. Instead of using translation, CAE-product vendors will employ standard interchange languages such as EDIF—the Electronic Design Interchange Format—to help eliminate database-format incompatibilities in the 1990s. EDIF allows CAE systems to exchange engineering information, and it encompasses several types of engineering documents or "views," including mask layouts, documents, behavioral descriptions of circuits, schematics, and net lists.

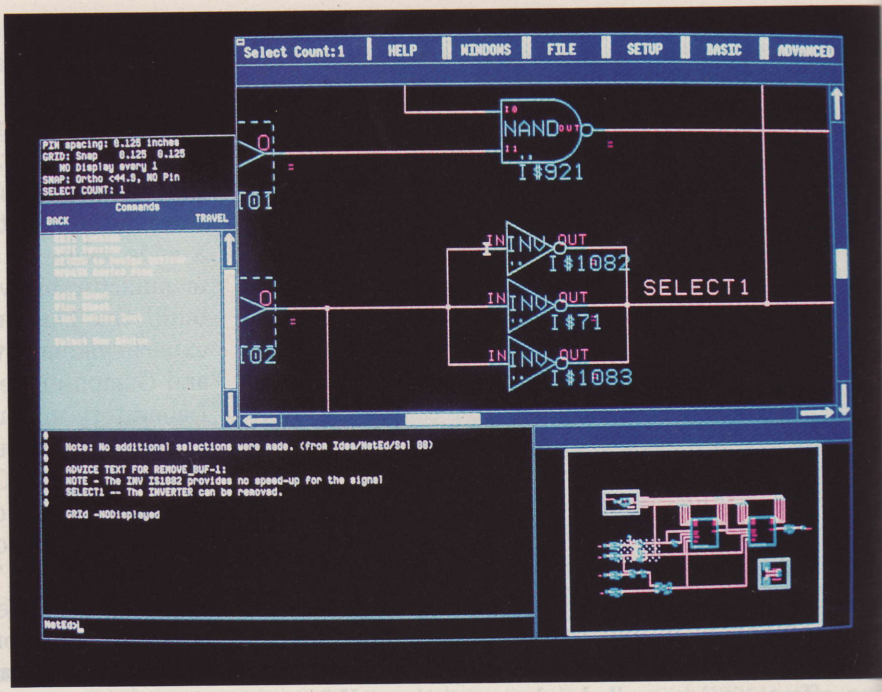
Some companies plan to take EDIF beyond hardware design. Cadre Technologies Inc (Providence, RI) has proposed a set of extensions to EDIF version 2 0 0 to accommodate software developed with CASE (computer-aided software engineering) tools. The EDIF Technical Committee, sponsored by the Electronic Industries Association (EIA) is currently considering that proposal.

EDIF allows CAE systems to exchange only low-level design information. Hardware definition languages

PART TWO

DECADE 90

The future of system design



(a)

(b)

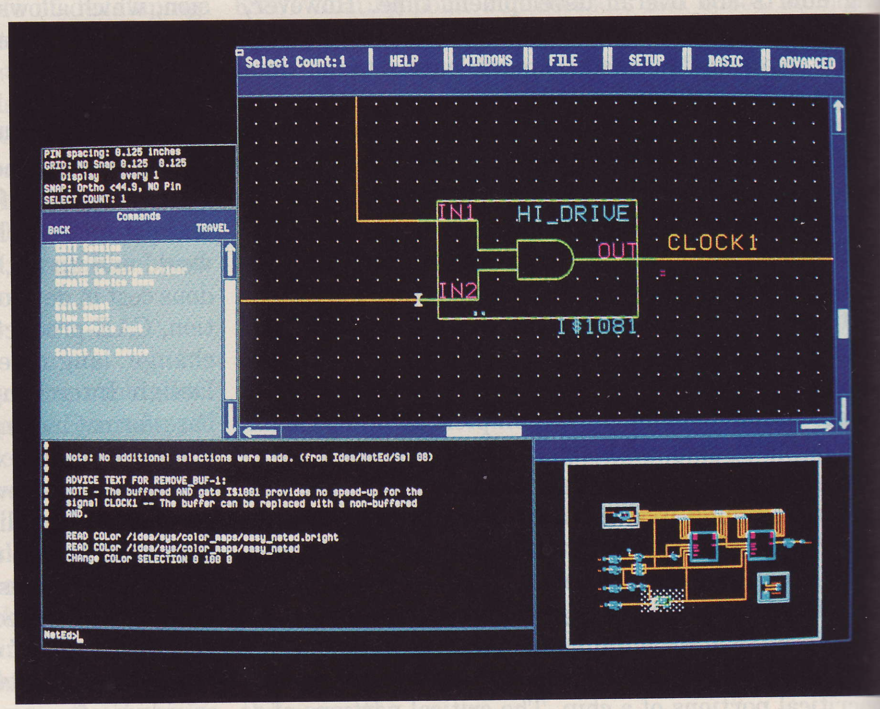


Fig 1—Based on artificial intelligence, *NCR's Design Advisor CAE tool provides expert design assistance for IC development. The program uses several criteria to evaluate IC designs, and it also gives advice. Here, for example, it recommended that the designer eliminate one of three parallel inverters **(a)** and suggested the use of an unbuffered instead of a buffered gate **(b)**. Design Advisor made both suggestions because the designer selected components that don't speed circuit operation yet use more silicon than necessary.*

(HDLs), however, promise to allow designers to manage and exchange information at several levels of abstraction, from the system level down. Engineers will need system-level CAE tools such as HDLs to help them design systems around billion-transistor ICs. System development of that complexity demands a structured approach such as that supplied by HDLs.

A few HDLs already exist; they include Gateway's Verilog-XL, Aida's ADL, VHDL (the VHSIC hardware definition language developed for the Department of

Defense's VHSIC (very-high-speed IC) program), and ISP' from Endot (Cleveland, OH). Verilog-XL and ISP' incorporate a syntax resembling the C software programming language. VHDL uses an Ada-like syntax.

According to Dr Thomas A Zimmerman, director of VHSIC programs at TRW's Electronics and Technology Div, VHDL's current lack of a good simulator is a major shortcoming of the language. Dr Zimmerman feels that HDLs will be extremely important for communicating the specifications and requirements of systems and system components. In fact, he says, in much the same way that Ada has become the software language of choice for military systems, VHDL may become mandatory for systems commissioned by the Department of Defense.

Intermetrics Inc (Cambridge, MA) developed the existing VHDL tool set under a VHSIC contract. In December 1987, a revised version of VHDL became IEEE standard 1076. Intermetrics plans to introduce software during 1988 that supports this IEEE standard; the software will include an analyzer that's somewhat like an HDL compiler, as well as a reverse analyzer, and both interactive and batch simulators. Intermetrics also plans to support other CAE vendors who wish to develop VHDL tools—the company will share its front-end software technology with them.

HDLs support many levels

HDLs allow you to describe circuits at the gate, functional, and behavioral levels. For top-down designs, you can use an abstract, behavioral description that avoids the clutter of implementation details and enables you to focus on how the circuit performs. Later, you can design the detailed circuit by using the HDL's lower-level syntax.

Because HDLs give you the ability to describe a system component behaviorally, you can simulate a complex system before designing the circuitry. For example, Gateway's Verilog-XL includes a mixed-mode simulator that accepts behavioral, functional, and gate-level models simultaneously. According to Dr Goel, hardware designers working for some of Gateway's clients have become software gurus while learning to define systems by using Verilog.

One of the benefits of HDL-based hardware design, says Goel, is that it permits the easier partitioning of large designs into blocks that one person can handle. Goel also believes that HDLs help move much of the design process to the conceptual level, freeing engineers from design details that frequently bog them down.

Other CAE experts, however, do not believe that today's hardware engineers will quickly abandon the design techniques they use today. Schematic circuit representations are firmly embedded in most engineers' work habits, and system-design tools that support those familiar ways are also appearing. Further, several CAE-tool vendors allow you to mix schematic and HDL representations. You should expect to use each design technique where it fits best. As usual, most engineers will adopt the available tools in ways that suit them, not the tool designers.

Graphic system design

An example of a system-level, graphic design tool is the ADAS (architecture design and assessment system) CAE tool set developed by the Research Triangle Institute (RTI, Research Triangle Park, NC). ADAS allows you to develop schematic representations of system data-flow and hardware configurations. Petri net simulators and analyzers then verify your design and simulate your system's performance. By using various definitions to map the execution of abstract operations onto system hardware resources, you can experiment with different levels of parallelism to find the optimum combination of hardware and software.

Honeywell (Minneapolis, MN) used ADAS to optimize the design of a video-image processor that will be used on NASA's space station. Honeywell simulated systems employing different network topologies (1-, 2-, and 3-bus systems, hypercube, and both unidirectional and bidirectional braided rings), different numbers of processors (four, eight, and 16), different processor speeds (2, 5, and 10 MIPS), and different bandwidths for interprocessor communications (2M, 5M, and 10M bytes/sec).

Honeywell used ADAS to simulate systems built with these various attributes. The company then decided that the optimum system configuration would have sixteen 10-MIPS processors that communicated over a dual bus at 5M bytes/sec. Clearly, system-level design tools such as ADAS allow you to make informed architectural decisions for very complex systems.

Getting chips from system tools

ADAS is strictly a system-level design tool, however. It produces descriptions of system components, but it doesn't help you design those components. You must

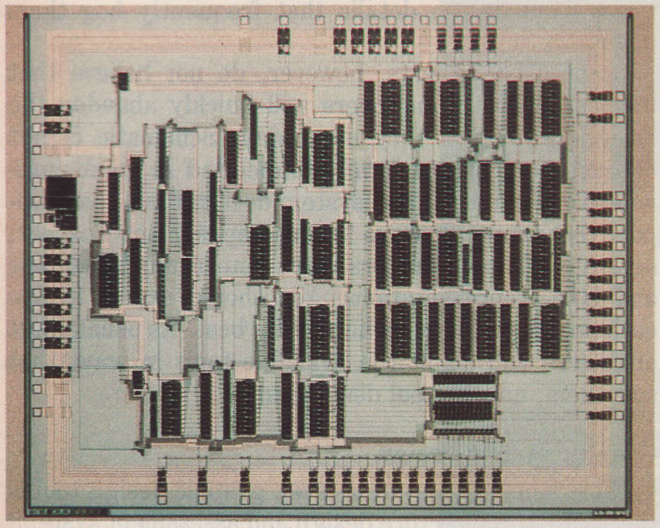


Fig 2—To develop the architecture for this edge-detection image-processor IC, engineers at the Research Triangle Institute (RTI) used ADAS, RTI's system-level CAE tool. To create the IC, they then transferred the architectural design to Silicon Compiler Systems' Genesil silicon compiler. In the future, standard hardware definition languages such as VHDL will make the conversion process automatic.

use other CAE tools for the individual component designs. For example, RTI recently created a system definition of an optical processor that employs the Sobel algorithm for edge detection. After using ADAS to create, simulate, and verify a system architecture, the RTI engineers transferred the architecture to Silicon Compiler Systems Corp's (San Jose, CA) Genesil compiler and implemented the various functional blocks in the optical processor (registers, multiplexers, adders, etc) with elements from the Genesil library. They then used Genesil to simulate the compiled version of the image processor as part of the verification process.

In an additional verification step, the engineers created a VHDL description of the edge-detection chip and verified the design with a VHDL simulator. Of course, the working chip provided the ultimate verification of the design (**Fig 2**). In the future, tools such as ADAS and Genesil will exchange design information directly, using automatically generated VHDL files. Both RTI and Silicon Compiler Systems are currently working on VHDL interfaces for their CAE tools.

Bearing the cost

Although CAE tools such as those mentioned here clearly boost engineering productivity, it's not clear how many companies will bear the cost of these tools. Most CAE tools run on workstations that cost at least $10,000. Add to this several thousand dollars—or tens of thousands of dollars—for the software, and you arrive at a total outlay that represents a substantial investment per design engineer.

Companies that believe that a shortened development cycle and optimized designs justify such a large outlay are already adopting these tools. If you're waiting for the tools to get a bit better before you adopt them, consider that at any point in time, CAE tools will always seem inadequate for the task of designing leading-edge systems. However, engineers routinely push their tools beyond commonly accepted limits to design state-of-the-art systems, and semiconductor technology shows no signs of slowing down to wait for the CAE tools to catch up.

From the Sequent and Apollo success stories, you can see that engineers at some firms have taken the sometimes difficult measures necessary to master today's design tools, and are already making plans to use tomorrow's offerings. Companies that cannot or will not invest in CAE tools may find themselves less able to tackle the large projects that will become more common in the next decade. Such companies will find themselves at a competitive disadvantage in the 1990s. **EDN**

References

1. Conner, Margery, "ASIC simulation," *EDN*, February 4, 1988, pg 118.
2. Hines, Dr John, "Where VHDL fits within the CAD environment," *Printed Circuit Design*, December 1987, pg 8.
3. Marx, Esther, Hart Switzer, and Mike Waters, "EDIF format brings uniformity to CAE/CAD data," *EDN*, January 22, 1987, pg 153.
4. Marx, Esther, Hart Switzer, and Mike Waters, "Use EDIF to create test vectors and verify your logic," *EDN*, February 19, 1987, pg 185.
5. Marx, Esther, Hart Switzer, and Mike Waters, "EDIF readers/writers handle data transfer between CAE systems," *EDN*, March 4, 1987, pg 171.
6. Mosley, J D, "Artificial-intelligence techniques boost utility of CAE/CAD tools," *EDN*, February 19, 1987, pg 8[illegible].
7. "Video Image Processor for Space Station Applications," *ADAS News*, October, 1987, pg 1.
8. Waters, Mike, Esther Marx, and Hart Switzer, "Use an interchange format to port component libraries," *EDN*, February 5, 1987, pg 175.

Article Interest Quotient (Circle One)
High 494 Medium 495 Low 496