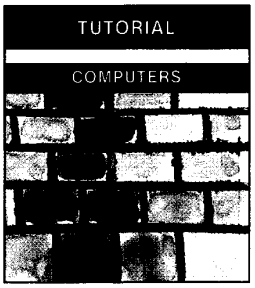
How to automate analog IC designs

Knowledge-based systems are relieving the labor-intensive bottlenecks usually associated with such building blocks as op amps and voltage references



In a matter of hours rather than weeks or months, basic analog integrated circuits can now be designed to suit the needs of the systems they are destined for. This sudden advance is due to new knowledge-based computer-aided design (CAD) tools. With the tools' help, system designers can employ common analog blocks, such as op amps and voltage references, without understanding their operation in much detail. Circuit designers also benefit from having the routine parts of a design done for them.

In a field that has traditionally been labor intensive, the tools embody the strategies of human design experts. In so doing, they speed up the design process. But perhaps more important, they enable a designer to improve overall system performance by exploring how one circuit relates to the others in a system and by using that knowledge to trade off performance specifications among circuits.

Knowledge-based design systems for analog components were formally announced only last year. First, the Centre Suisse d'Electronique et de Microtechnique (CSEM), Neuchâtel, Switzerland, described its Idac system. Later the same year, Carnegie Mellon University, Pittsburgh, revealed its Oasys project and the University of California, Berkeley, gave details of its Opasyn system. In all three cases, users simply feed in the performance specifications and the parameters of the fabrication process to be used—and the system designs analog circuit modules to match.

Results have been encouraging. When an Oasys-designed op amp was fabricated and its performance parameters measured, it fell within a few percent of the target design specifications or exceeded them. Yet the circuit design had not been tuned—that is, no small changes had been made to the specifications during the circuit simulation to try to improve performance—either manually or by automatic computer optimization.

The Swiss system, Idac, is already being marketed by CSEM and is being used by several customers in Europe and the United States. Opasyn and Oasys are still in development, but maturing rapidly as they are fed more design knowledge on a greater variety of circuits. Oasys's developers gave a preview of their system this June at the Design Automation Conference in Anaheim, Calif., and plan to release it this fall. Opasyn's developers report that their system is being tried out at an industrial user (AT&T) and should eventually be for sale. All three systems run on various manufacturers' workstations. Meanwhile, workers at such institutions as Canada's University of Toronto, the Georgia Institute of Technology in Atlanta, AT&T Bell Laboratories in Murray Hill, N.J., and the General Electric Research and Development Center in Schenectady, N.Y., are busy developing their own knowledge-based analog circuit design systems.

The impetus for these developments is the growing use of analog functions in digital ICs—part of the overall trend to replace circuit boards full of ICs with a single application-specific IC, or ASIC.

At AT&T, for example, over half the ASICs reportedly include analog circuitry. Even children's toys are part of the trend; one of the most successful ASICs of 1986 was the chip for Teddy Ruxpin, a talking stuffed bear. Just one chip contained nearly all of Teddy's circuitry.

But while designers can apply elaborate CAD techniques to digital circuitry, the absence of such aids for analog circuitry makes that part of the design a bottleneck, adding months to the design time even when the chip is 90 percent digital. In the fast-moving world of ASICs, such a delay can mean lost sales or even commercial failure of the end product.

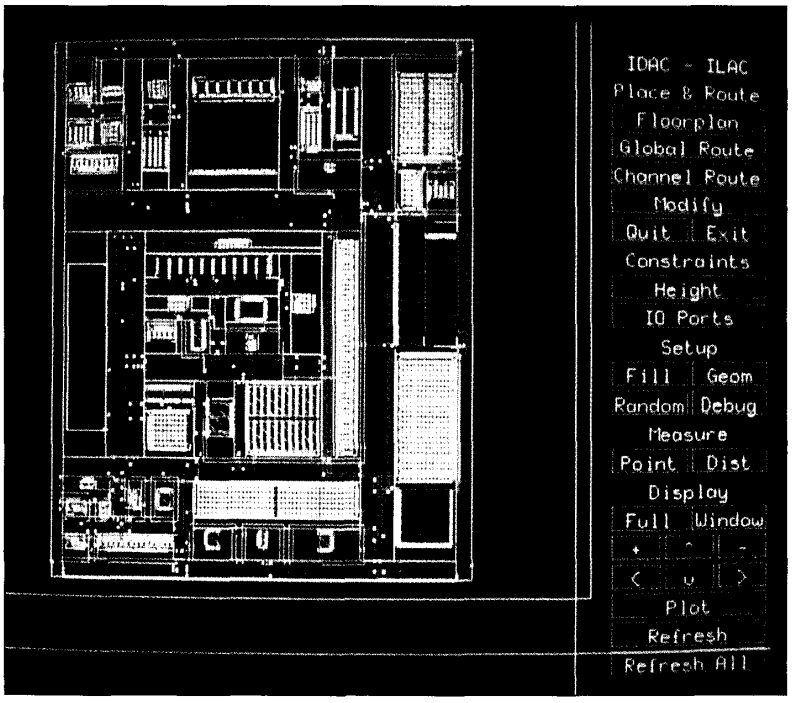
Although Idac, Oasys, and Opasyn differ widely in philosophy, all employ common building blocks and produce sized-schematic diagrams showing how transistors, capacitors, and so

L. Richard Carley and Rob A. Rutenbar
Carnegie Mellon University

Defining terms

Current mirror: two or more transistors connected so that current in one node is duplicated in another node. Two MOS transistors, for example, having sources tied together and gates connected by one of the drains, would duplicate the current in that drain in the second drain.
Design plan (for hierarchical systems): a set of operations, either heuristic or algebraic, that translates performance specifications for a block into performance specifications for its sub-blocks.
Differential pair: two transistors whose sources (or emitters) are tied together to a current source. The voltage difference between the two gates steers the current between the two drains.
Knowledge base (in circuit design): knowledge furnished to a computer system by an expert circuit designer.
Op amp: a high-gain differential-input amplifier.
Phase margin: a measure of op amp stability; specifically, phase of output voltage at unity-gain frequency relative to 180 degrees of phase lag.
Sized-schematic diagrams: a circuit diagram indicating interconnection of components and their sizes (for MOS transistors, for example, the width and length of the gate).
Slew rate: the maximum rate, in volts per microsecond, at which the output voltage of an op amp changes when a square-wave or step input is applied.
Synthesis (of a circuit design): generation of sized device-level schematic diagrams from performance specifications and process specifications.
Threshold voltage: the lowest gate-to-source voltage at which the channel of a MOS field-effect transistor becomes conductive.
Topology: an interconnection pattern for components such as transistors, capacitors, and resistors.
Unity-gain frequency: the frequency at which the gain of an op amp equals one.

0018-9235/88/0800-0026$1.00©1988 IEEE



Ilac, a companion tool to Idac, developed by the Centre Suisse d'Electronique et de Microtechnique, Neuchâtel, Switzerland, generates a geometrical layout for analog CMOS cells from netlist information created by Idac. This CMOS op amp, laid out by Ilac, contains 53 transistors, four capacitors, and a resistor. By using the main menu (right), the user can modify the design, zoom in on one part of the circuit, or exercise other options.

forth are connected, complete with the components' values—from which custom circuits can be synthesized. These building blocks, however, unlike the ones used in semicustom analog IC design, are not fixed designs from a library. Rather, they can be varied infinitely, according to rules given to the tools by human experts, so that they approach the ideal performance far more closely than is possible with a limited choice of fixed blocks.

All told, Idac, Oasys, and Opasyn can automatically synthesize analog circuits from 13 classes of analog building blocks and can produce over 100 distinct circuit topologies. Each topology represents an infinite number of designs because the tools will select devices of various sizes for it, depending on the application.

The analog building blocks synthesized so far range from simple op amps through band-gap voltage references to complex analog-to-digital converters. Many of these building blocks have been fabricated and performed much as predicted by the CAD tool. Others have only been simulated on a computer, but again with reassuring results.

Analog dynamics

CAD for analog circuits has been slow in coming because the process of analog design is far more complicated than digital design. Digital logic design needs only a small set of fixed building blocks, simple elements like AND gates, OR gates, and storage registers, or more complex blocks like arithmetic and logic units and memories. Analog design needs a profusion of customized building blocks, ranging from simple transistor pairs to op amps, to achieve the best circuit performance.

Now that analog tools are becoming available, however, both system and circuit designers stand to benefit. System designers can use analog building blocks without having to understand their internal workings. Circuit designers can now concentrate on designing the more unusual parts of a system, knowing that the CAD system will customize the basic parts to their needs.

Moreover, when block design is fast and automatic, many variations of the same block can be reviewed until the best set of tradeoffs among characteristics is found. For example, gain can be traded off with bandwidth.

This ability to design, inspect, and abandon many suboptimal designs is completely new to analog design. It is made possible by automated synthesis tools and may prove to be as great a benefit as shortening the design cycle.

As the tools mature, large libraries of design knowledge will accumulate, and more and more analog circuit designers will exploit this computerized knowledge. Perhaps two-thirds of all analog designers will eventually use such tools as Idac, Oasys, and Opasyn; their primary users are likely to be custom ASIC designers and high-performance system designers who want to explore tradeoffs between individual specifications—a significant and growing section of the analog design field. Many of the same designers are likely to enlarge their company's knowledge base by adding their own expertise on proprietary topologies.

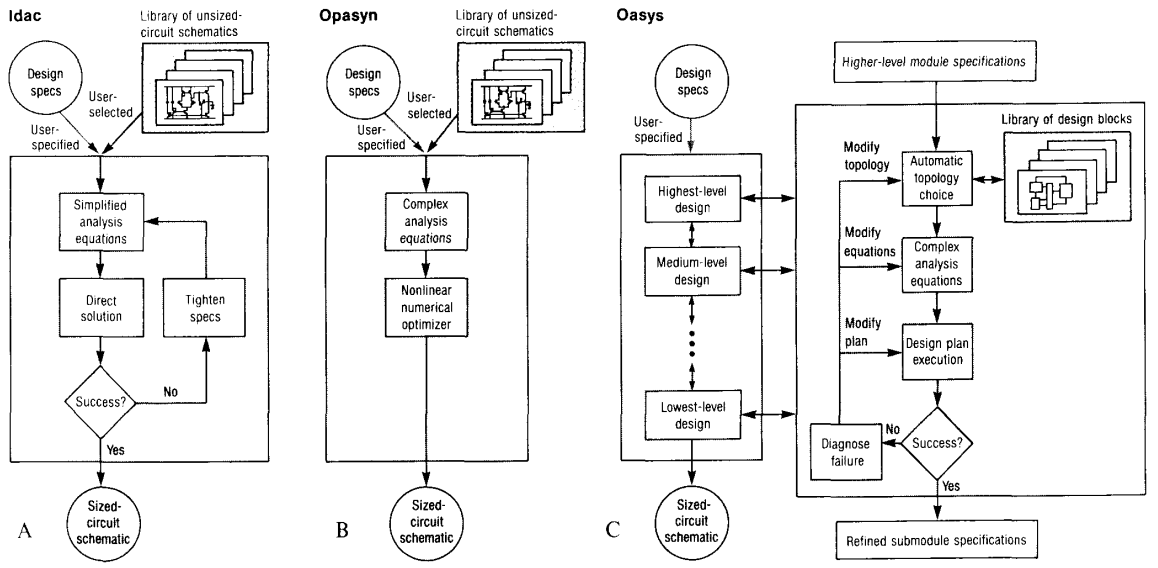
To be sure, these users will find that knowledge-based design has its limitations. Creating the design knowledge for a circuit topology takes a lot of time—much more than expanding a library with a fixed analog cell that meets only a single set of specifications. In effect, creating design knowledge is economical only for frequently used analog blocks. The seriousness of this limitation depends on the tool; it is less severe for a hierarchical system like Oasys, which uses blocks, sub-blocks, sub-sub-blocks, and so forth. For example, an op amp consists of differential pairs and current mirrors, which are in turn composed of individual transistors.

All the new tools are based on design expertise obtained from analog-circuit designers. Human experts are adept at developing simplified but useful equations to describe the design goals, eliminating unimportant effects or deferring them for later consideration. They pin down critical interactions, set up goals and subgoals and assign priorities to them, and steer a path through the design task, drawing on their experience all the while. When they arrive at a promising circuit design, they simulate it on a computer, using complex device models, so that they can assess its performance accurately.

Capturing human expertise

To compete, a computer must somehow capture this design knowledge and apply it. Computer representation, of the kind used in these three systems, generally follows a scheme like this: asked to synthesize a building block (an op amp, for instance), a computer will call up from its knowledge base a specific circuit topology, simplified models of the devices that make up the topology, and a set of equations describing the behavior of the circuit. From these, the tool will assign the best possible values to the components, creating a sized-schematic diagram.

As yet, there are no mechanisms for automating the entry of human expertise into the system. Such mechanisms are crucial to a knowledge-based system that must grow to accommodate new technologies. A major goal is to insulate experts from com-



The process used by each of three analog-circuit module design systems to arrive at a sized-circuit schematic differs, sometimes substantially. Each system selects appropriate simultaneous nonlinear equations from its knowledge base that incorporate the user's design parameters. Using an unsized-circuit schematic selected by the user, Idac (A), from the Centre Suisse d'Electronique et de Microtechnique, breaks the analytical equations down into sets of simpler equations from its knowledge base to arrive at a solution; if no solution is found immediately, it imposes more severe specifications until it finds a solution or determines that none exists. Also using a user-selected unsized-circuit schematic, Opasyn (B), developed at the University of California, Berkeley, solves the equations using a numerical optimizer that chooses the design that comes closest to the specifications. Oasys (C), designed at Carnegie Mellon University, Pittsburgh, uses a hierarchical approach, breaking each design down into sub-blocks, which have progressively simpler sets of equations. It creates connections between design blocks rather than using complete, preconstructed schematics. If Oasys has trouble finding workable values for a particular sub-block, it alters its plan for solving the equations; if this, too, fails, Oasys retreats a step to the next-highest level in the design and varies its design plan to find and correct the problem.

puter code: they should be able to interact with a tool on a higher level, using familiar concepts and abstractions while the machine generates the code. Workers at universities and in industry are developing mechanisms that will make such interaction possible. Meanwhile, however, human expertise is added to a tool directly, as lines of code.

The circuit topology itself is part of the design knowledge. The computer need not invent a new topology; it need only automatically select a topology from the library of topologies and choose the component values.

The computer uses analytical equations from the knowledge base to express the block's performance in terms of designable parameters—component values such as transistor sizes and simplified device model parameters such as threshold voltages.

Solving these analytical equations is anything but simple. They are often nonlinear, with each variable dependent on many others. Idac, Opasyn, and Oasys handle equation-solving difficulties in different ways [see diagram above], and so have differing capabilities.

Idac does not solve simultaneous nonlinear equations directly. Instead, the creator of the design knowledge must solve for all of the design equations explicitly. That is, a human expert must provide detailed program code that explicitly, step by step, solves each equation. This means that someone has to decouple and decompose each set of analytical equations in such a way that a step-by-step procedure can solve simpler design equations for everything of interest.

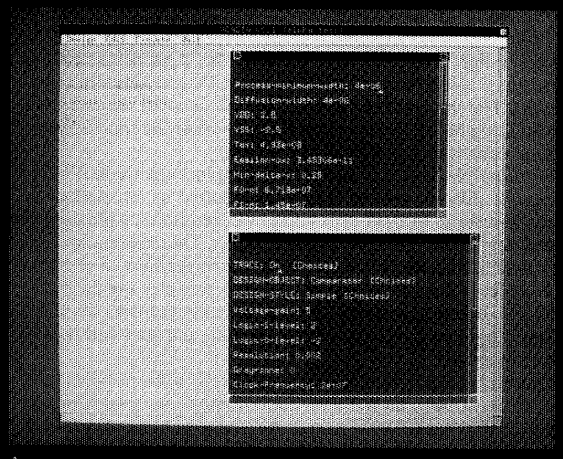
If Idac cannot achieve the required circuit performance, it attempts another solution for new performance specifications. If it fails to meet a phase margin specification, for example, it will tighten the internal phase margin specification and repeat the calculations. Since the key to improving performance may actually be changing another specification that has already been met, this type of iteration will not always improve performance. On the other hand, Idac is straightforward and designs rapidly. A typical op amp design will take only a fraction of a second on the central processing unit of a Digital Equipment Corp. VAX-8800.

Opasyn solves a set of nonlinear design equations using an optimization function that measures how "good" a design solution is in terms of how close it is to the target parameter values. Opasyn does not solve equations analytically. Instead, it uses a numerical nonlinear-problem-solver program to calculate results for a complete mathematical description of the circuit to be designed. It adjusts the variables until it finds a solution with a minimum difference between the user's specifications and the design's performance.

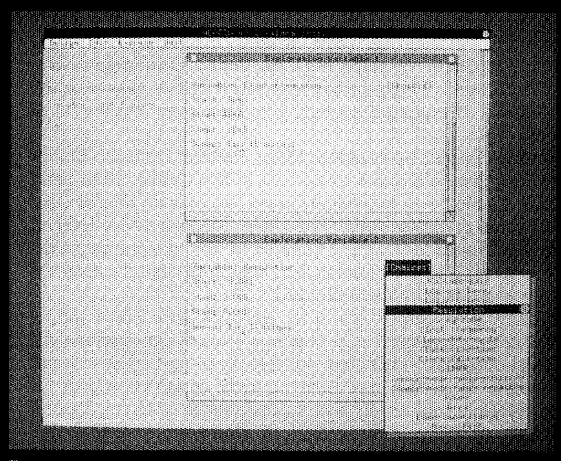
The optimization algorithm makes Opasyn much slower than Idac; an Opasyn op amp design takes from 70 to 280 seconds on a VAX-8800. Opasyn is therefore less attractive than Idac (or Oasys) for exploring alternative designs. Nevertheless, it produces op amp designs whose performance is comparable to those developed by human experts.

Oasys is different from both Idac and Opasyn in that it operates in a strongly hierarchical way, breaking down even simple analog modules such as op amps into several sub-blocks. Each block has a relatively small set of design equations to be solved simultaneously, and Oasys solves each set by starting with assumptions based on design knowledge and using a planning mechanism that goes through the equations to adjust the assumptions. Like Opasyn and Idac, Oasys produces op amps that rival those designed by humans. Because of its hierarchical treatment, however, it is much faster than Opasyn, taking only about 1 second of VAX-8800 time to design an op amp.

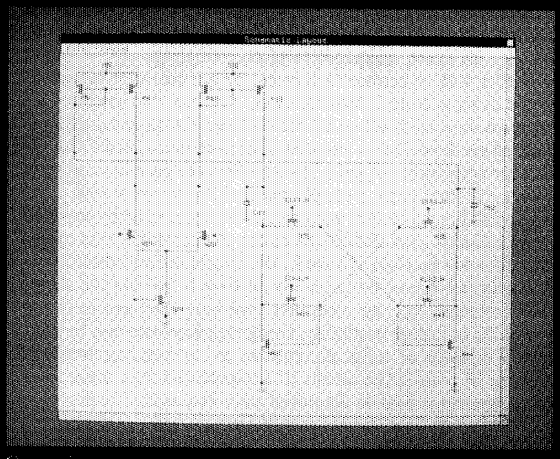
Oasys's hierarchical structure stands it in good stead when it comes to design variety. Hierarchy, even at the low level of an



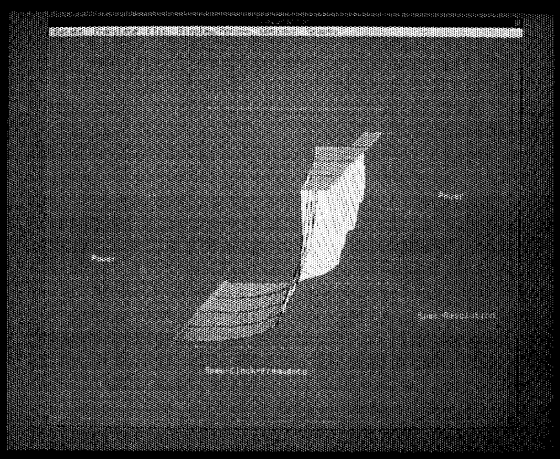
A



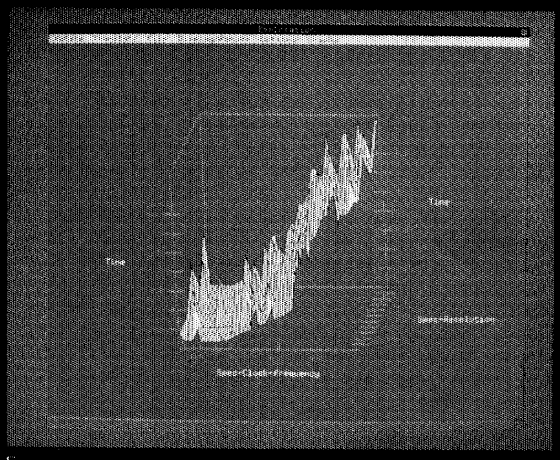
B



C



D



E

In designing a comparator—like other components—with Oasys, the user starts with a main menu showing process specifications for fabrication and performance specifications for the circuit itself (A). Oasys allows the user to explore certain of the comparator's design parameters, such as clock frequency and resolution; in this case, over 200 different comparator designs result from this exploration (B). The user is then shown the schematic for the comparator resulting from the initial specifications (C). Power is plotted against clock frequency and resolution to show the projected performance of this comparator exploration; each comparator's estimated power consumption is shown (D). The user can also display information about Oasys's performance; here, the amount of CPU time consumed for each design by Oasys on a Digital Equipment Corp. VAXstation II/GPX is shown versus clock frequency and resolution (E).

op amp, allows the sub-blocks to be designed in several possible topologies. For example, Oasys currently offers only two block-level topologies for op amps, but it can design 72 different device-level topologies. Thus, the tool can more closely approach the proficiency of human designers, who have more topologies to choose from.

Both of Oasys's op amp topologies are built from sub-blocks such as differential pairs or current mirrors rather than from specific topologies of individual devices. In contrast, Idac or Opasyn requires complete design knowledge for each device-level op amp topology. (Idac employs a coarse version of hierarchical design; only high-level circuits such as analog-to-digital converters are broken down into sub-blocks.)

Hierarchical design has other benefits, too. Since simple elements—current mirrors, for example—are designed separately, it is a simple task to use these sub-blocks as part of new circuit topologies. Moreover, improvements in the design knowledge for a sub-block automatically become included in all circuit topologies that use that sub-block. And, of course, many small design tasks are easier to carry out than one big design.

When a design doesn't work

What if the design system cannot create circuitry that performs as desired? Each tool has its own response to this situation.

Idac states which specifications could not be met. Because its design plans are fixed, it can always associate the failure with a particular design step.

Oasys, though, cannot always assign a failure to a particular step because it modifies its design plan repeatedly to find a workable solution; thus, it's very difficult to trace a failure back through the iterations. It halts a design when its planning mechanism detects that computations are not converging to a workable solution.

For example, suppose a user, in asking Oasys for a simple two-stage CMOS amplifier, specifies both a large bandwidth and an op amp voltage swing very close to the two power-supply voltages. Oasys may not be able to meet both conditions and still guarantee that the op amp will be stable—that is, that it will not oscillate. In fact, Oasys will report that the phase margin, which determines stability, could not be met.

The true cause is less obvious, however. Even an expert human designer may not recognize that the problem could be solved just by decreasing the output swing by a few tenths of a volt.

Opasyn avoids all these difficulties because it treats the design task as a large numerical optimization problem. It always completes a design—not always satisfactorily—since it simply tries to minimize the difference between the required specifications and the performance predicted by its analysis equations. However, Opasyn lets the user assign weights to the differences being minimized so that it can favor the more important performance goals in a compromise.

Even if one of these tools does not indicate a specific cause for a design's failure, it may be able to help a user—especially one with little knowledge about the circuit being synthesized—by doing a sensitivity analysis of the failed design. Interacting with the user, the tool can show how sensitive a critical performance specification is to variations in all other specifications. Because tools like Idac and Oasys are quite fast, many different sensitivities can be computed.

For an op amp, if the system designer indicated that the bandwidth was a critical specification, the sensitivity analysis would proceed according to this two-step agenda. First, the tool would decrease the bandwidth, holding all other specifications fixed, until it succeeded in creating a design. Secondly, the tool would vary each of the design specifications by a small amount and increase (or decrease) the bandwidth to determine the threshold at which it cannot complete the design.

Exploring design space

An analog circuit in the process of being designed can be looked on as a multidimensional space in which each performance specification is a dimension with its own axis. Each point within the space represents a complete set of performance specifications for the design. The feasible design space for a particular circuit consists of those points for which the tools can actually build circuits. The shapes of the feasible design space for various analog building blocks tell a system designer what the tradeoffs are among constituent blocks in the system.

For example, Idac has been used to explore the tradeoffs between the silicon area and the power required by an analog-to-digital converter and the area and power required by the digital decimating filter that supports the converter. The decimating filter prevents aliasing, the folding of high frequencies onto lower ones. The less complex the converter, the more complex the decimating filter. In one experiment, Idac aided a system-level designer in selecting the point at which the total system area and power were minimized.

One way to explore the design space is to fix some performance specification, vary all the rest, and examine how performance varies. This procedure yields multidimensional plots of the fixed specifications versus the variable ones and, in the case of Oasys, which can alter topologies as the design changes, can show where the tool has made a change in the topology. Although experienced designers always have a general qualitative feel for tradeoffs, the tools give an exact, quantitative picture of them.

Another way to explore is to seek the surface that separates feasible designs from infeasible ones. The tool concentrates on two parameters, such as bandwidth and gain for an op amp, holding all others fixed. It first calculates the maximum feasible bandwidth for a low value of gain. It then raises the gain in steps, calculating the maximum possible bandwidth for each step.

The tool then comes up with an envelope curve of bandwidth as a function of gain. Below the curve, the combinations of design values are feasible. Above it, the combinations are infeasible. Again, an expert designer would have a feel for the shape of the envelope, but the computer tool pinpoints precisely the outer limits of design feasibility. Design surfaces can also be constructed in three or more dimensions as more parameters are added in; of course, the results of explorations with more than three dimensions can be displayed only in tabular form.

To probe further

For more about Idac, see *IEEE Journal of Solid State Circuits*, Vol. SC-22, December 1987, pp. 1105–1116. Idac's supplier will furnish a user's guide on request. Contact: Centre Suisse d'Electronique et de Microtechnique, Maladière 71, CH-2000 Neuchâtel 7, Switzerland.

For more about Opasyn, see the *Proceedings of the 1987 IEEE International Conference on Computer-Aided Design*, pp. 502–505. For more about Oasys, see the *Proceedings of the 1987 IEEE Design Automation Conference*, pp. 42–49. For information about Oasys and related analog-design tools, contact: SRC/CMU CAD Center, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, Pa. 15213.

A good source of background information on knowledge-based expert systems is *Building Expert Systems* by Frederick Hayes-Roth, Donald A. Waterman, and D. B. Lenat (Addison-Wesley, Reading, Mass., 1983). For details about designing CMOS op amps, see *Analog MOS Integrated Circuits for Signal Processing* by Roubik Gregorian and G. C. Temes (John Wiley & Sons, New York, 1986).

About the authors

L. Richard Carley [M] and Rob A. Rutenbar [M] are assistant professors of electrical and computer engineering at Carnegie Mellon University. Carley received S.B., M.S., and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge. Rutenbar received a B.S. in electrical and computer engineering from Wayne State University, Detroit, Mich., and M.S. and Ph.D. degrees in computer engineering from the University of Michigan, Ann Arbor. ◆