Work

The engineer's job: it moves toward abstraction

Automated modeling and analysis minimize the practitioner's contact with real hardware

Paul Wallich Associate Editor

As engineering tasks have become more complex and the tools available to grapple with them have become increasingly sophisticated, the job of electrical engineering has become more analytical and abstract. Far more emphasis is being placed on designing a product that will work properly the first time it is built, rather than on making a series of prototypes and debugging them. This trend toward analysis and modeling is changing the way engineers work, both individually and as members of organizations.

Today, an engineer designing almost any kind of circuit, from a simple analog amplifier or digital switching network to a complex single-chip signal processor or complete supercomputer, can use computer-based simulation tools to see how a design will work before it is built. Even large systems, such as power-distribution networks or telephone switching systems, are amenable to modeling. Although analytical tools do not aid synthesis directly, they can determine quickly whether the results of synthesis will work—where bottlenecks or critical paths occur in a design, and whether a design meets overall goals for power consumption, speed, linearity, or other qualities.

"The biggest change in engineering practice in the last few decades," commented James M. Early of Fairchild Camera and Instrument Corp. in Palo Alto, Calif., "is the increased reliance on modeling and the availability of far more exact models for circuit design." Digital design is the area of engineering that has been most visibly affected by increased computing power, Dr. Early noted, and thus gives a preview of effects in other areas. At the same time, digital logic has become the main area of interest for many EEs: "Ones and zeros have replaced the a + jb of complex impedances," Dr. Early noted—signal processing, for example, once almost entirely an analog field, has become largely a digital one in the last decade. Among the various advantages digital designs have over analog ones is that complex digital designs are more easily verified than analog ones. Verification of analog designs, with their continuously varying inputs and outputs, requires a great deal of computing power, and only in the last few years have extensive analog design tools become available. The lessons learned in applying these tools to digital problems thus hold, albeit perhaps not quite so strongly, for areas other than design of digital circuits.

With good computer models, a design can be simulated and exercised long before it is built. Furthermore, said Dr. Early, certain kinds of integrated circuits, such as read-only memories and programmed logic arrays, are so well understood that they can be generated from specifications entirely by computer, with no need for checking or simulation.

A different kind of abstraction spawned by increasing complexity is the partitioning of designs into smaller and smaller portions. In the semiconductor industry, for example, the traditional approach is to rely on a series of designers at different levels (horizontal partitioning): a top-level designer may specify the architecture and instruction set of a microprocessor, a middle-level designer the logic diagram, a third designer the implementation in silicon, and a fourth the actual chip layout. A complex design would be partitioned at each level as well (vertical partitioning), with some engineers finally taking responsibility for one segment of a particular level of a design.

A world inside the machine

Yet another kind of abstraction in engineering comes from the evolution of computer science as a major interest of EEs. More than a quarter of the IEEE's 250 000 members consider themselves computer scientists or engineers in 1984, as compared to less than one tenth 20 years ago. Computer science and engineering includes not only the design and manufacture of new computer systems but also the theory and practice of making computer systems perform useful tasks—software in all its ramifications. And software engineers work in a field that seems completely alien to the hardware engineer assembling components at the bench. (For some people, computer science denotes only software, but as new computer architectures are proposed that fit the hardware to a particular class of problem, or even reconfigure the hardware as required by the computational demands of the moment, the distinction becomes difficult.)

In the more arcane realms of computer science, a design may involve sophisticated mathematical analysis and consideration of object-oriented computational paradigms, with little or no consideration given to whether a machine may eventually be built to execute the programs that result. Some observers question whether the activities of mathematicians whose main interaction with electronics involves turning on their computer terminals should be considered as electrical engineering at all. Dr. Early, for example, said that some of his peers in the semiconductor industry feel that "it's not electrical engineering unless one is consciously affecting hardware."

On the other hand, Robert W. Lucky, executive director of the communications sciences division of AT&T Bell Laboratories, Holmdel, N.J., said, "It is important that electrical engineering be associated with higher-level conceptual work, not just circuit design and hardware." He said that more than half the EEs in his department write software, but contended that their work is as much a part of electrical engineering as any hardware designer's.

The work of software engineers is at a different level of abstraction from that of hardware engineers, even that of hardware engineers who design circuits using computer-aided design tools, said Stephen Trimberger of VLSI Technology Inc. in San Jose, Calif. Not only do software engineers work with fundamental components that differ from those of hardware engineers—bytes, procedures, stacks, and heaps rather than wires, chips, and boards—but the relationship of their designs to their final product is also much different. The final specification for a computer program is represented as words in computer memory just as is the compiled object code that the computer actually executes. For hardware engineering, the execution medium—silicon and circuit boards—is completely different from the design medium—paper or a computerized data base. A hardware engineer working with a computer-aided design system is working with a level of abstraction between the design and the final product: the circuit elements placed on a video schematic and tested with a circuit simulator are metaphors for the actual design in silicon or on a printed-circuit board. The software engineer works with a level of abstraction between the product and the actual physical processes that go on in a computer: the program statements placed in a software design are identical to the ones that will be executed, but the very idea of program statements is a metaphor for the appropriate movements of electrons.

IEEE Center for the History of Electrical Engineering

International Electrical Exhibition, *1884*

In addition to creating entirely new engineering disciplines, computer systems are also changing the ways EEs in more traditional fields do their jobs. Power engineers, for example, using computer systems for analysis and modeling, have tackled many of the stability problems involved in interconnecting ever larger portions of the United States into regional power grids, Dr. Early pointed out. Although ensuring that such large systems will be stable and will not fail catastrophically if small segments break down is a difficult task, it is one that could not even be contemplated without powerful modeling capabilities. Computer systems must be capable of dealing with the immense volumes of calculation required for detailed simulation, and engineers must also be able to place confidence in those simulations, since thorough prototype testing of complex systems is usually not possible.

Peter Van Olinda, general applications manager in the computer applications section of Consolidated Edison's research and development group in New York, explained that engineering work stations have replaced mainframes for analyzing load flows because they offer more consistently accessible computing power, and because their high-resolution graphics displays can present information far more comprehensibly than a large printout. For example, the utility's secondary networks, 52 of which serve the boroughs of New York, are each served by a dozen or more feeder lines, each connected to about 20 transformers; a single net may have over a thousand connections, and it is difficult to understand what is going on without a graphic display. Furthermore, if a particular portion of a network is carved out to make a simulation more manageable, an engineer can look at the graphic display to see whether out-of-bounds conditions are real or are caused by computational misbehavior near the edges of the simulation area.

Mr. Van Olinda said that, in addition to simulation of load flows, data on actual flows are transmitted from each transformer in the secondary networks every 15 minutes, using recent developments in signal processing to recover the information from the relatively noisy power lines over which it is piggybacked. The actual load-flow data can be used as input to load-flow simulations both improving their verisimilitude and testing their accuracy. The graphic analysis and data gathering, he said "are essentially free" because the work stations pay for themselves within a year through the timesharing savings alone.

Even the engineer's nonengineering tasks are being changed by the computer. Meetings in some companies, for example, are held by linking participants with computer-based communications networks rather than convening them in a single room. Computers add a level of abstraction to discussions by reducing participants' perceptions of each other to words typed on a CRT screen, much as telephones in a previous generation added a level of abstraction by replacing face-to-face contact with voices.

At the same time that they make information sharing more abstract, computers also make it much more efficient. Erich Bloch, vice president for technical personnel development at IBM Corp. in Armonk, N.Y., claimed that the vastly increased ability of engineers and others in companies to share information is "more revolutionary than evolutionary" and predicted that sharing of information among engineers working on projects and between engineers and managers at different levels will significantly improve engineering productivity as data bases, computer systems, and intelligent work stations become more pervasive.

Some engineering managers are not as enthusiastic about the effects of computers on engineering communication. Gene Amdahl, who designed IBM's 360 computer in the early 1960s, founded Amdahl Corp. to build cheaper, more powerful IBM-compatible mainframes in the 1970s and is currently president of Trilogy Systems Corp. in Cupertino, Calif., a company that intends to make mainframe computers based on integrated circuits covering an entire silicon wafer, said that the degree of communication between engineers using computer-based methods for their work is not significantly greater than it used to be. "People may be communicating by electronic mail," he said, "but that's not such a big change."

Other observers point out that electronic mail is a relatively unimportant side effect of doing one's work on a computer—more significant is shared access to data bases and design files, so that a team of engineers working on a project all have access to the same information, even though they do not communicate explicitly.

In many areas of the field, EEs are using engineering work stations, computers specifically designed to emulate traditional paper-and-pencil methods of working. These machines use high-resolution graphics displays to simulate a desktop, right down to individual windows—work areas much like a sheet of paper—that can be scattered about the screen, buried under each other, or arranged neatly in overlapping piles. These displays are coupled with some kind of pointing device, either a mouse or a digitizing tablet so the engineer can point to objects on the screen and select commands to perform operations on them. A typical example is the selection and placement of logic elements on a schematic diagram. The work-station screens can give the illusion of being a window onto a much larger sheet by panning across the "sheet" in response to commands. Unlike sheets of paper, however, the electronic drawings may be shrunk or expanded to show fine detail or to see a design in overview. This shrinking or expansion can be done in either a physical or a logical fashion—the engineer can command simply that components be drawn very small on the screen, or that they be consolidated into functional blocks previously defined.

In a properly designed work-station system, the information in the drawings can also be used for input to other portions of the design process: for example, a logic simulator can be invoked to see whether the design as drawn matches the functional specifications desired. A simple continuity checker could make sure that no wires remain unconnected and that power and ground go to the proper places. If the components placed in the schematic have had electrical data attached to them, then a more detailed simulation can examine timing problems, fanout adequacy, and other issues. By abstracting information and reducing the overhead involved in manipulating large, complex designs, computer-aided engineering systems significantly increase the complexity of designs a single engineer can handle.

In one sense the work station distances the designer from the product, because all interaction is with a computer model of the design rather than with actual components or chips, but by allowing an individual engineer to handle larger designs, the work station may bring the designer closer to the product in another sense by reducing the number of pieces a system must be split up into. One example of such changes is the concept of the "tall, thin designer" advocated by the proponents of the Mead-Conway style of structured VLSI design. The tall, thin designer is responsible for all aspects of a given design, starting with the functional specification and going down to the actual layout, as contrasted with the more traditional methods of parceling out levels of a design among a large group of engineers.

Whereas the traditional approach maximizes the utilization of each individual designer, proponents of the tall, thin methodology claim that valuable information is lost each time the design goes from one level to the next: the logic designer can only work to the formal specification delivered by the architect, and the silicon implementers can only work to the schematic delivered by the logic designer, rather than working to the functional specification of what a circuit should do. A tall, thin designer, on the other hand, given enough computing power to take a circuit from initial design to layout, can make tradeoffs between levels, always keeping in mind the functional requirements. Thus, the power of engineering work stations could change the structure of integrated-circuit design from a series of engineers, each working on the same level of every chip designed at a company, to a group of engineers each taking a particular portion of an entire chip from concept to silicon. (The design of the RISC [reduced-instruction-set computer] chip at the University of California at Berkeley, where a group of graduate students designed a microprocessor, each responsible for a given functional block, is an example of the tall, thin methodology. The total time from initial design to first silicon and subsequent testing for the RISC was approximately one year.) Of course, it is also possible that networks of engineering work stations might solidify the current organization of engineers based on short, fat designers: since all the information about a design is available through the same work station, information need no longer be lost by moving from one level of design to another, even if different designers execute each level.

Engineering work stations have had little discernible effect on engineering organizations thus far because just a few are in use. Only a few thousand engineering work stations have been shipped in the United States, for example, though an estimated 15 000 to several hundred thousand engineers could benefit significantly from using them. Although shipments increase each year, it will be some time before work stations significantly affect the way the average engineer works. Furthermore, the definition of engineering work station is beginning to shift as the power of personal computers increases with each generation of new hardware. Mr. Bloch at IBM said that he uses a PC for much of his work, for example [see "Bolsheviks and Mensheviks," p. 35]. As standard personal computers become more powerful, the major difference between them and engineering work stations will be the work stations' specially designed software, in much the same way that machines are still sold solely to do word processing, even though word-processing programs are available on personal computers for a fraction of the cost of a dedicated machine.

Chickens and eggs

One activity that stretches the capabilities of computer-aided engineering tools is the design and manufacture of computers—large computer systems are among the largest digital designs. Trilogy's Dr. Amdahl pointed out that the rapidly decreasing cost of memory has changed computer architectures and made it possible to build systems far more complex than those he designed a decade or two ago. Such systems require computer aids to comprehend them. And furthermore, since they are built from integrated circuits, they must be thoroughly understood long before they are built. "You can't reconnect the wires anymore, because they're on silicon," he noted.

The need for thorough simulation will continue to grow with the increasing size of computer systems, said Dr. Amdahl, adding, "As circuits become more complex, it becomes harder to understand the possible deviations from expected function. You need simulation both for performance and for function." And the requirements for simulation and analytical tools are only going to get worse, he predicted, because "as machine organization gets more complex, you start having to think very carefully about second-and third-order effects on performance."

Increased simulation and analysis as integral parts of the design process are hallmarks of the effects on engineers' working styles of increased levels of integration in the circuits they design. Even if the engineer is designing a board-level part, using only parts already existing, the increased functionality of each of those parts will make it more difficult to design by making breadboards and rewiring them until they work properly. Instead, the iteration must be done with a simulator to make sure the design works before a breadboard is built.

If the circuit being designed is ultimately to become a chip, then breadboarding may be more trouble than it is worth, depending on the cost and turnaround time involved in making silicon prototypes, Dr. Lucky said. "A wire-wrapped board takes a month or more to build and debug," he claimed, "and it can't be simulated before it's built, so you spend a lot of time chasing down errors." Furthermore, there is no assurance that a successful board-level prototype will work when it is reduced to silicon.

As the design of circuits becomes more and more based on analytical methods, simulations, and computer-aided engineering work stations, the skills involved in building circuit boards and debugging them may go out of style among most EEs. In the case of integrated-circuit design, the final step of drawing geometries for transistors has been in disfavor for some time.

Design-automation systems on the horizon are intended to take over tasks starting all the way up at the architectural level where chip function is initially specified and to carry designs down to layout with a minimum of human intervention.

Bolsheviks and Mensheviks in the computer revolution

The computer revolution has been occurring for the last 30 years: only in the last five has it reached the trenches. Erich Bloch of IBM Corp. in Armonk, N.Y., maintained that there is a very important difference between the existence of computing power in an organization, even if it is made accessible to engineers, and the placing of that computing power directly on engineers' desks. Personal computers and engineering work stations, he claimed, "are completely different from a dumb terminal. Their capabilities are much better than those of a terminal, and they allow many different things to be done without going through complex procedures."

Because the personal computer or work station is not separated from the engineer by communications lines, it can display results far more quickly than could a mainframe. even if the portion of a mainframe allocated to an individual engineer might be able to compute those results faster than the personal machine—the personal machine can rewrite a 2000-character screen in a small fraction of a second, while standard terminal-communications links are limited to 480 or 960 characters per second, and ordinary telephone links have a maximum rate of 120 characters per second. Furthermore, computing on a desktop machine is essentially free, in contrast to the access charges levied for mainframes, and so an engineer need have no qualms about leaving a desktop computer on all day, turning to it as needed. The ease of using desktop machines, Mr. Bloch said, makes engineers and others willing to use them even for the simplest tasks, tasks that would never justify the trouble of logging on to a mainframe with a dumb terminal.

Robert W. Lucky of AT&T Bell Laboratories, Holmdel, N.J., on the contrary, rejects the idea that log-in procedures inhibit computer use. "You just type your name at the beginning of the day," he said. He asserted that the minicomputer is a useful vehicle for distributing computer power within a single building or department. Since high-speed data links operating over a limited area connect each terminal to the minicomputer, such a system is (at least for the present) faster and has more memory than a personal work station, and, best of all, it is taken care of by someone else, relieving the individual of the tasks of system administration. Dr. Lucky suggested that, in the future, most people would use powerful personal work stations communicating over a local area network.

Though the instantaneous response of a personal machine makes it very attractive for performing tasks requiring a limited amount of computing and storage, the power of a mainframe is still sometimes required for tasks such as large-scale computation or searches of large data bases, especially corporate data bases that may be maintained by a completely separate organization. The capability of the personal machine to communicate with the mainframe, or with other computers via local area networks, means that none of the qualities of the dumb terminal are lost; in fact, the desktop machine can enhance those qualities by transferring information to a mainframe for batch jobs, such as a net list for a large-scale simulation, or downloading data for subsequent interactive manipulation, such as corporationwide design files. In many cases, personal machines are connected in a network, allowing both computing power and data to be shared: multiple machines can work on a single task, thus relieving the load on any single engineer's computer, and in a properly designed network the engineer does not have to worry about where a particular piece of data or a particular program actually resides: the computer system can simply make it appear when requested. *—P.W.*

C. Lester Hogan, a director of Fairchild, rejects the idea that such systems may eliminate engineering jobs. "I don't see design automation as replacing engineers at all," he said. "I see it as permitting engineers to get rid of the terrible drudgery and to do a much better job."

Broadening the engineer's scope

Integrated-circuit design, once the domain only of a few arcane specialists, can now be practiced (at least in theory) by virtually any systems engineer, thanks to the emergence of computer-aided design tools and structured methodologies for designing ICs. This changes the ways that systems engineers work, not by redefining what they do, but by redefining the medium they work in.

What effect does that redefinition have on the way engineers work? Perhaps not much. If a systems engineer is building circuits using a typical standard cell system, for example, the functional blocks available are very similar to those that could be picked out of a standard catalog of medium-scale integrated parts. If the designer is aiming for a custom integrated circuit, there will be a certain set of relatively simple rules, picked for easy understanding and implementation restricting what kinds of circuit elements can be used and how they can be placed. The prevalence of "cookbook" design rules raises questions about exactly how tall the tall, thin designer actually might be. If a systems engineer learns how to design integrated circuits simply by applying a rote series of transformations to a logic diagram, then little would be gained, especially if the chips thus built ran significantly slower and were significantly larger than those laid out by experts.

Another viewpoint might come from considering the "thin" attribute of the tall, thin designer espoused by structured-VLSI-design advocates; while short, fat designers know all about the most arcane layout design rules and efficient geometries, they trade off knowledge about logic design or machine architecture. Exponents of the tall, thin approach have claimed that an understanding of what kinds of structures work well in IC design may have payoffs in improving top-level architectures that outweigh the small losses in low-level implementation that may come from simplified design rules. Thus, the systems engineer turned chip designer mixes systems-level thinking with understanding of low-level consequences.

As design tools become even more powerful and handle more of the low-level detail work, the systems designer will be able to spend more time on systems-level thinking, specifying an architecture and a set of functions, and letting a piece of software generate the actual circuit. In contrast to earlier systems-level work, where many decisions were arrived at by accumulated judgment and knowledge of particular constraints on system structure, the designer can then see the low-level consequences of a systems-level specification and make adjustments accordingly.

Engineering abstractions

The emergence of computer science as a major discipline within electrical engineering and the increasingly abstract nature of such pursuits as VLSI design is part of a trend away from hands-on practice and toward analytical and theoretical grounding that has characterized the evolution of EEs' work since the 1920s, according to a number of observers. Jack Ryder, former dean of engineering at the University of Michigan, noted that prior to 1940 virtually the entire electrical engineering profession was in power engineering, with a few communications engineers involved in the telephone and telegraph networks. During World War II, observed Jerrier Haddad, formerly a vice president of IBM and now a consultant, "many if not most of the great engineering advances were made not by engineers but by physicists and chemists. People began saying that engineering was getting much too sophisticated and to do it you had to have that science base." By 1950, said Dr. Ryder, almost 50 percent of engineers were involved in some form of electronics; by 1960, the figure was near 90 percent. When he graduated from college in the 1930s, Dr. Ryder recalled, all his classmates took positions at power companies. The "mysteries of the electron," he said, began the evolution of EE as we know it today. "You couldn't see it, and so you had to think about it abstractly. Thus the field attracted the abstract thinkers, not the shop types but the mathematical types, and that made possibilities for EE and EEs to expand." Dr. Ryder suggested that the abstract thinkers who came to electrical engineering to work in electronics also thought differently in other areas—they were entrepreneurs, whereas the "shop types" opted for stable long-term employment.

Within electronics, the same kind of progression has occurred. In the late 1950s and early 1960s, students learned in classes about semiconductors and bandgaps and considered the properties of switches with no moving parts and amplifiers without tubes. They then went out into the world and built transistor-based circuits. In a parallel experience, students in the 1970s were told to consider the properties of a circuit containing an infinite-gain amplifier and a negative-feedback loop. They went out into the world and designed circuits with operational amplifiers. Today, engineers work with VLSI logic devices whose operation is difficult even to intuit, much less observe directly, but the abstractions they use, coupled with powerful computer aids, make the actual nature of the devices beneath the abstractions less important than before.

Dr. Amdahl said, for example, that "the current generation of computers is the first generation in which designing a computer is actually designing a computer. It used to be designing circuits, packaging, wiring, and so on—designing reliable circuits was a difficult job in itself. Today, you only get into packaging if you're a semiconductor manufacturer." Although the complexity of computer systems has increased enormously, the new tools and the power they have given engineers to reason abstractly allow engineers to concern themselves with issues of logical complexity rather than becoming bogged down in details of physical implementation.

As the trend toward more abstract treatment of circuits and systems continues, with the electrical engineer moving farther and farther away from the actual circuit, and the technician or a computer system handling many details of implementation, a different kind of person may become attracted to electrical engineering. The entrepreneurial spirit that Dr. Ryder characterized as part and parcel of the abstract thinking required by the newly emerging field of electronics in the 1930s is even more rampant today in computer software and semiconductors than in traditional areas of electrical engineering. It is not at all clear what the EE of the future will be like, nor is it certain that the work that the EE may be doing will be recognized as electrical engineering by today's standards. Once the overwhelming problems of implementing large systems become more manageable, the engineer may have more time to spend on the initial specifications—deciding, along with other members of the engineering organization, exactly what it is that a particular product should *do*. When the process of implementing a specification is largely automated, then it becomes much more important to get the specification right, rather than becoming committed to a large project with the assumption that a design will somehow emerge by the time a system is built.

Changing times

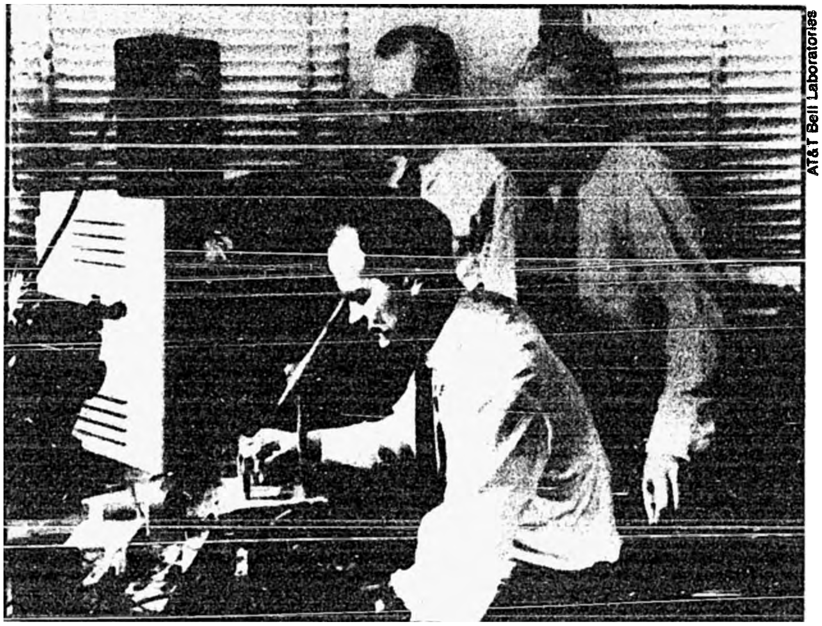
"An organization is simply a means for achieving a goal, not an end in itself," noted Mr. Bloch of IBM. Thus, as electrical engineering changes, the structure of organizations that do electrical engineering will have to change also, he said. But how will that structure change? One important requirement of a new structure would be to allow information to be shared as rapidly as possible among people with different expertise, so that all the different factors bearing on a decision could be taken into account; this would be the management equivalent of the multidisciplinary team required for doing current EE. Such a structure is similar to matrix-management structures that have been adopted by a number of companies already, with mixed results. On the other hand, the capability of computer-based information systems to collect data rapidly from many sources and filter it could allow individual project managers to manage more complex projects, just as it allows individual engineers to deal with more complex engineering problems. Higher-level managers might be able to use the same technology to oversee more lower-level managers, with the result that the canonical pyramid shape of organizations might remain, but much broader and shallower than before. Even so, noted Bell Laboratories' Dr. Lucky, it is not clear that the information managers need most will be amenable to storage in computerized form. "Management information is mostly intangible," he said; "unstructured things that don't fit into computer records."

The shape of future engineering organizations will be determined not only by the capabilities conferred by the computer technology such organizations have built up to now, but also by the requirements of future markets for engineering products. Most observers agree that future product lifetimes will be even shorter than current ones; Dr. Trimberger of VLSI Technology suggests, for example, that in the semiconductor industry, standard products as they are now known may become far less important, displaced by standard design modules—large blocks of silicon layout that may be replicated time and again, but always with a slightly different mix of peripheral circuits, so that rather than buying a standard microprocessor for a high-volume product, for example, and building a circuit board around it, engineering organizations will contract for a custom circuit, one portion of which happens to be an industry-standard microprocessor. Marketing demands including such "virtual standard products" would require a far faster flow of information among different parts of a manufacturing organization to bring lead times down to acceptable levels.

Dr. Bloch of IBM suggested that decision making will be pushed down to lower levels of the engineering hierarchy by advances in computer technology, because a manager would be more likely to allow a subordinate to make a decision if it were easier to keep track of what the subordinate was doing. If things went wrong, it would be easier to retrieve the situation. He also suggested that, because most engineering resources and information required for a project would be available on line, the entire relationship of a project manager to the people in the project would change, because the manager would spend less time mediating between team members and information coming down from above (or across from other teams) or between higher-level managers and information coming up from below. Similarly, Mr. Bloch suggested that the relationship between the individual engineer and technicians working on a project would change, because of the rapid flow of information to everyone working on a problem. AT&T's Dr. Lucky questioned whether information would be shared so freely between different levels of a corporate hierarchy, noting that even so simple a task as scheduling meetings by computer has foundered in many companies because managers insist on access to all their subordinates' calendars, but refuse to allow anyone else to see their own. Mr. Bloch asserted that the use of a new technology is not bounded by its availability, but by the ready acceptance it receives from members of an organization.

Avoiding obsolescence: part of the job

With the rapid movement of electrotechnology comes the threat that the EE will be left behind. Today, most EEs try to keep up technically by reading the literature, taking courses in new fields, and going to conferences when possible. Nonetheless, many older engineers are considered by their employers, even if not by themselves, to be out of date. The problem is not, according to Robert Fano, professor of EE at the Massachusetts Institute of Technology in Cambridge, that engineers are unable to keep up with their technical specialties, but rather that entire technical specialties die out. "The engineer whose background is not broad enough to switch successfully to a different specialty, no one needs his skills," Dr. Fano said. Today, a technical spe-



AT&T Bell Laboratories

Transistor inventors, 1948, *William Shockley (seated), John Bardeen (left), and Walter Brattain*

cialty may last for only 10 or 15 years, and so "it is impossible to have a career based on one specialty," he asserted.

Dr. Early of Fairchild took a different view of obsolescence: the major problem for the practicing engineer is not becoming outdated as an individual, but rather being caught in an organization that becomes outdated as a whole. "Organizations cannot adapt because they have too many people there who think the same way," he said. "You can't take a bunch of people who think in one way and get them to think in another way without a new organization and new leadership. Individuals can adapt fairly easily, but only if they're not with too many of their old buddies."

The question of obsolescence seems to parallel the questions of broad scientific background for engineers that were raised in the 1940s and 1950s: those engineers who do not have a solid grasp of the underpinnings of their specialty run the risk of being left behind when that specialty changes. The rapid pace of electrotechnology is the source of this danger. Do any of the products of electrotechnology help in the solutions? It is not clear. Dr. Fano of MIT noted that his organization has been active in producing video-taped courses that can then be viewed by thousands of engineers across the country at minimal cost to the engineers or MIT, and IEEE satellite broadcasts on particular topics have been moderately successful in attracting interest. Nonetheless, as long as engineers and their managers take a particularly narrow view of the nature of electrical engineering, the problems of technical obsolescence will continue. What has not been widely recognized is that, as the pace of technical change increases, the engineer's job changes from knowing how to do a particular thing to constantly innovating and doing new things. "Today, industry is living on innovation, and engineers are expected to be creative *en masse*," Dr. Fano asserted. The demand for innovation not merely from a few engineers but from all of them, he said, "is unprecedented. It's changing the picture very rapidly."

One of the most obvious consequences of technical obsolescence as electrotechnology advances is the mass hiring and firing of engineers as particular projects are initiated and then come to a conclusion. If matters remain unchanged, further increases in the pace of technical innovation may make EEs' jobs even less secure than today, exacerbating the paradoxical situation where industry cries out for trained (as opposed to educated) engineers in particular specialties while those trained in other specialties find themselves underutilized and in dead-end jobs. The failure to retrain engineers from specialties that have gone out of favor also places a burden on engineering educational institutions to be up to date not only in the basic science they teach their students, but in the detailed practice as well, since that is what industry demands. University equipment has long been considered inadequate to train engineers in current techniques, and electrotechnological innovations will exacerbate this situation for the most part. (The low cost of some newer tools, such as personal computers, makes it possible for universities to supply their students with them in large numbers, but other technologies, such as integrated-circuit fabrication, are far beyond the resources of virtually all universities.)

One consideration for the practicing EE is that the fast pace of technological change appears to mandate as fast as possible a shift into engineering management, where detailed technical knowledge is not as crucial. Some observers might argue that, as computers and technicians take over technical detail work, all electrical-engineering jobs will in some sense become management. Others, such as Mr. Haddad and Dr. Fano, take strong exception to the division made between engineering design and management. "Until you get into management, you're not really doing engineering," Mr. Haddad said, pointing out that today a single engineer working alone cannot accomplish much at all, and so all engineering work except for the most detailed technical designs involves management. "A major part of engineering is organizing complex tasks," said Dr. Fano. "The responsibility of the engineer is to get a job done, and if that job requires a lot of people, then management will be part of it."

Synergy

Another possibility is the restructuring of electrical engineering in ways that recognize the constant changes in the field without either making obsolete the EEs who acquire deep knowledge in a particular field or relinquishing all the actual work of implementing products to technicians and computer-based systems. The use of computers to design further computers is one example of how the same kinds of activities the EE values can be carried out at a higher level, so that, rather than exercising creativity to build products directly, the engineer exercises creativity to build tools that can then be used to build products. In some cases, this progression of metaengineering may go even further, as in the development of computer networks and methods for communicating quickly and efficiently by means of them, where engineers build tools to speed the development and debugging of other tools, which are then used to build products.

◆