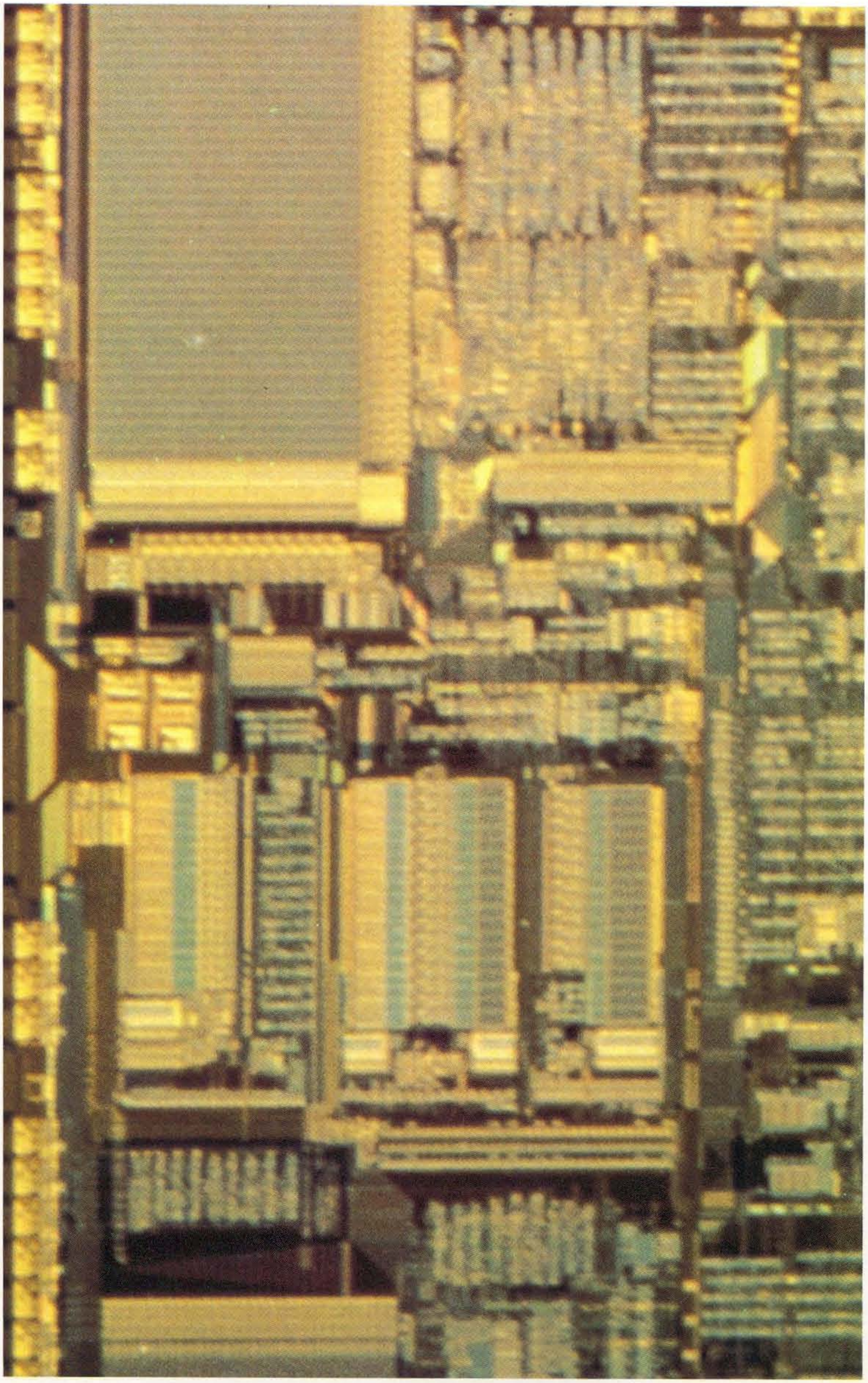


Have You Used Your 4 Million Transistors Yet This Year?

Future trends in microcomputing

At left, detail from the Intel 80386 microprocessor, one of the most wanted chips today. The image is blurry because a photographer's lens is not as good as the quarter-million-dollar ones used to print the pattern on the chip.

More electronics are manufactured each year than existed in the world at the beginning of that year—10^{15} transistors this year, according to Gordon Moore, chairman of Intel Corporation. That translates into a million transistors per person in the developed world, or 3-4 million per family. So your family will have to consume 4 million transistors this year alone, says Moore. Next year it will be 8 million, and 16 million the following year. Although some of your transistors have found homes in your automobile, microwave oven, and TV set, most have been gobbled up by the microcomputing industry, which has mushroomed from nonexistence a decade ago into a $20-billion business today.

What the future holds for a field of such phenomenal growth was the subject of four public lectures at Caltech last spring, "Future Trends in Microcomputing," a series that the Institute hopes to continue next year. Organized by Barry Simon, Caltech's IBM Professor of Mathematics and Theoretical Physics, and Professor of Theoretical Physics Geoffrey Fox (who is also associate provost for computing), the lectures brought world leaders in the fields of microcomputing, as well as overflow audiences, to Beckman Auditorium on campus.

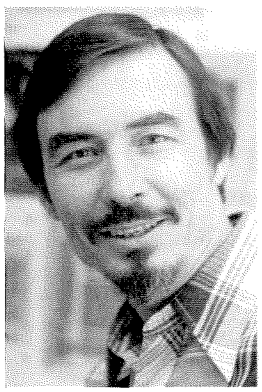
Besides Moore, the speakers included Benjamin Rosen, Carver Mead, and Philippe Kahn. In 1957 Moore co-founded Fairchild Semiconductor, which built the first integrated circuit; he and Bob Noyce then went on to found Intel, which invented the microprocessor in 1971. Moore and Rosen are both members of the Caltech Board of Trustees, and they, as well as Mead, hold Caltech degrees. Rosen (BS '54) remembers that when he was a freshman, Moore (PhD '54) was his chemistry teaching assistant. "He gave me a D," Rosen said.

Rosen co-founded Sevin Rosen Management Company, a venture capital firm with large investments in the microcomputer industry. He is also chairman of the Compaq Computer Corporation, a corporation that grew to have annual gross sales greater than a billion dollars, faster than any other in history; he was a founding director of the Lotus Development Corporation and is, as well, a director of Borland International, Inc., Bestinfo, and Quarterdeck Office Systems.

Mead (BS '56, MS '57, PhD '60), now the Gordon and Betty Moore Professor of Computer Science at Caltech, built the first workable gallium arsenide transistor, and his contributions to the theory of quantum tunneling were essential to the invention of the integrated circuit and the microprocessor. He is a well-known innovator (and textbook author) in VLSI and is currently doing pioneering work in neural networks.

Kahn, an immigrant from France, founded (and is currently president of) Borland International, Inc., which produces the popular software programs SideKick, Quattro, and Paradox. His innovative software, sold at discount prices, is challenging the industry giants. Kahn founded Borland in 1983 with $5,000 out of his own pocket, because all the venture capitalists ("including me," Rosen admits) turned him down. Rosen describes him today as "the most outspoken person in the industry."

Benjamin Rosen



Carver Mead

Gordon Moore

Philippe Kahn

"If the same kind of progress had been made in the auto industry over the past seven years, you'd go a million miles per hour and get half-a-million miles to a gallon of gas."

In the beginning of his talk, Mead traced the development of the microcomputer industry from the introduction of the printed circuit board ("the first big step") through the invention of the transistor in 1947, and then the first integrated circuit, made by Fairchild in 1959. "None of us saw this as the beginning of a revolution," noted Mead. "It is characteristic of great inventions that most people—even those working in the field—notice them only when they are adopted."

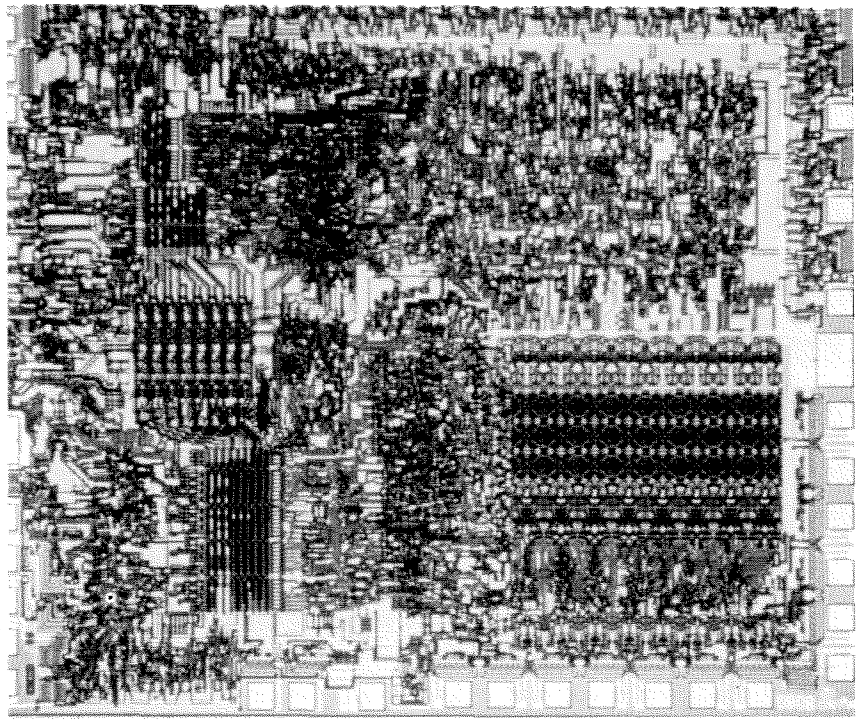
Rosen, in his talk, also described the "hypergrowth" of the personal computer industry as a revolution. That revolution was created out of three ingredients, he claimed—"technology, entrepreneurs, and money." But, as successful as this revolution has been, the microcomputer industry has still penetrated only 20 percent of its potential market, according to Rosen. Apparently we haven't all consumed our 4 million transistors this year.

Several of the speakers offered comparisons of "then" and "now" to illustrate just how far and how fast the microcomputing industry has ballooned. Moore compared "IBM's top-of-the line personal computer for 1987 to a big mainframe computer like the IBM 370, model 168, top of the line in 1975. The PC has four mips (million instructions per second, a measure of computing power) instead of two—twice the power at 1/34th of the price. If the same kind of progress had been made in the auto industry over the past seven years, you'd go a million miles per hour and get half-a-million miles to a gallon of gas. It would be cheaper to throw your Rolls away than park it downtown in the evening. I thought that was a neat analogy until someone pointed out that the Rolls would be only six inches long and two inches wide."

"Look at the six years from mid-'81 to mid-'87," said Rosen. "Memory chips went from 16,000 bits (the fundamental information unit —a binary 1 or 0) to 1,000,000 bits, a 64-times increase. The microprocessor word length (the number of bits it can handle per computing step) has doubled; floppy disk storage densities have gone from 160,000 to almost 1,500,000 bytes (a group of bits, usually six to eight, that represents a text character or a processing instruction), an eight-fold improvement. The Winchester hard disk, introduced in '83, held 10 megabytes (million bytes). Now you can get 300, a 30-times improvement. And microprocessor speeds have quadrupled. If you multiply all these factors together you get an absolutely specious figure of merit, so I can categorically say the personal computer is 122,880 times better than it was 6 years ago."

Mead provided another comparison: "The cost of a chip today is about the same as the cost of one of those individual transistors we used to solder onto circuit boards. Yet the capability represented by that chip has gone up by a factor of more than a million. The Industrial Revolution, which substituted fossil fuels for human and animal power—and gave us smog and urban waste and all the other good things about modern society—gave us, in terms of getting from the East Coast to the West, or printing a book—an increase of a factor of about 100."

Taking examples from his own experience,



The first commercial microprocessor, Intel's 4004 chip, built in 1971, contained about 2,200 transistors. It addressed 9.2 K of memory designed for arithmetic applications or control functions.

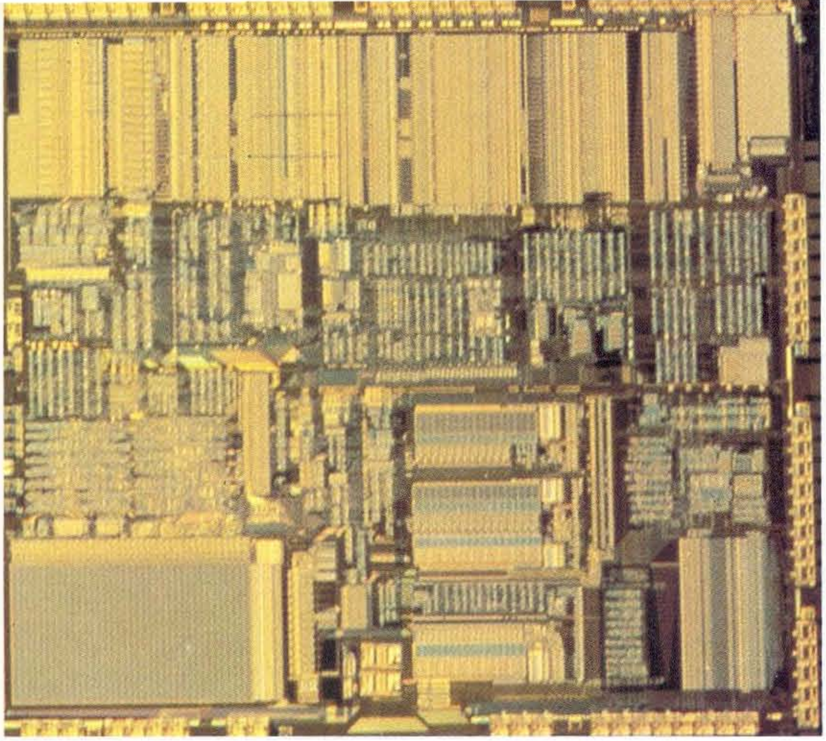
Moore contributed some insight into how this explosion happened. "As we've learned to pack more and more electronics on a given area of silicon, the standard chip becomes increasingly complex. The technology may exist to make something even more complex, but if the design costs dwarf the manufacturing costs, it will be cheaper to build your system from simpler products. That's why Intel got its start making memory chips—it's a universal function.

"This leads to the idea of a transistor budget—the maximum number of transistors you can put on a chip and still manufacture it economically. So how do you use the budget to make practical devices? Early in Intel's history, a Japanese company that wanted to make a family of calculators came to us. (There were hardly any Japanese semiconductor companies then.) They had designed some 13 logic chips, all quite complex and far beyond our ability to undertake. One of our engineers, Ted Hoff, suggested that he could get all their functions using a general-purpose computer architecture and some stored programs, and went on to point out that the same chip could be used for elevator control, traffic-light control, and a whole bunch of dedicated logic operations. And that was the origin of the microprocessor.

"That was in 1971. The 4004 chip had about 2,200 transistors, right against the limit of our transistor budget then. It was a 4-bit microprocessor addressing 9.2 K (thousand bits) of memory (on another chip), designed for arithmetic applications or control functions. Since it had a 4-bit word length, there were 16 potential instructions you could give it. As the technology developed, we added the 8080, which had about 8,000 transistors. It was an 8-bit microprocessor, alphanumeric-oriented, aimed at data-processing applications; it addressed 64 K of memory and was actually the basis of the first personal computer, as far as I know. There was a machine called the Altair that came as an 8080 and a bunch of stuff in a kit, and you assembled it at home. But the 8080 was still mostly used as a dedicated controller—it wasn't big enough to be really reprogrammable, like a stand-alone computer. I'm talking about Intel because I know the data, but the trend is true for the other manufacturers as well.

"And the budget kept growing. Two years later, the 8086 had about 30,000 transistors—over ten times what the 4004 had. The 8086 had a 16-bit word length, and addressed one megabit. It was big enough to separate the data interface from the central processing unit, so it could walk and chew gum at the same time. With 16 bits, it could receive some 64,000 instructions, plenty for high-level programming languages. It was fully reprogrammable, in other words. In fact, the 8088—essentially the same chip—was the processor IBM chose for their first PC.

"Next came the 80286, with about 125,000 transistors. It addressed 4 billion bits, I think, enough to use the high-capacity hard disks that were just coming out. Plus it had multi-user capability—different programs could run simultaneously, and its hardware kept the data for each program separate. It's the basis for the



The 80386 is a full 32-bit processor, with 64 trillion bits addressable, designed to use multiple operating systems simultaneously.

"We think the 80386 will pass the 80286 by 1990, and by 1992, it will be dominant. It'll be the chip for all seasons."

IBM PC-AT and all its clones. A lot of the budget had to be used for compatibility. The 80286 had to run all the software written for the 8086 and 8088. So the increased budget went for compatibility, performance, and memory management. The current step, the 80386, is a full 32-bit processor, with 64 trillion bits addressable, designed to use multiple operating systems simultaneously. The rest of the budget went to increased ease of use and compatibility."

The 80286 was introduced in the IBM PC-AT at a four percent market share. "In three years' time," Rosen pointed out, "the 80286, with no operating system or applications programs designed specifically to take advantage of it, but simply by being faster, took over 53 percent of the market. And the 80386 has had a much faster start. We think the 80386 will pass the 80286 by 1990, and by 1992, it will be dominant. It'll be the chip for all seasons."

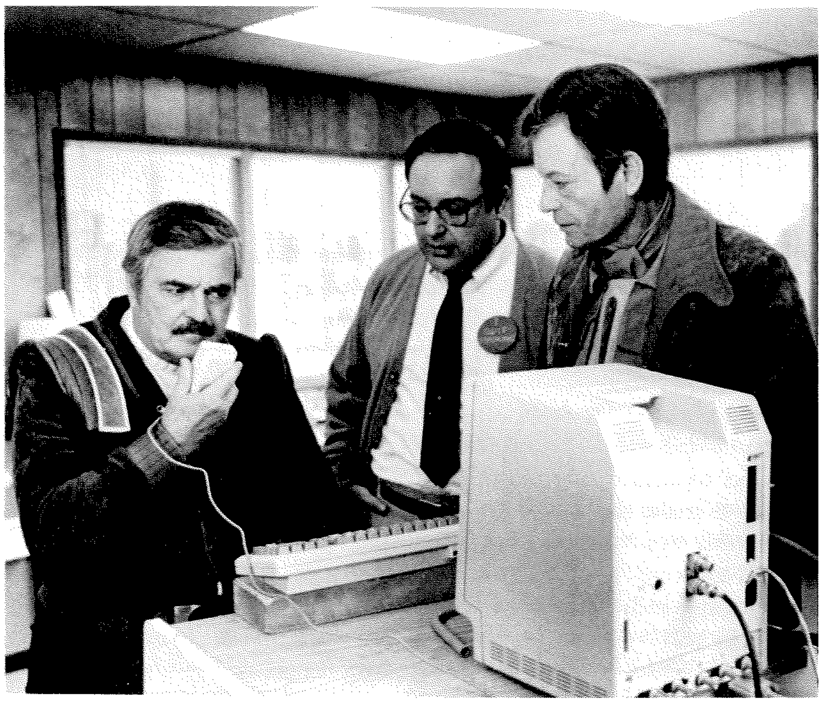
But the chip designers are racing on ahead. Said Moore, "If we do a linear extrapolation, in 1990 we'll have 2 million transistors per chip—about 7 times the 80386, and in the year 2000 we'll have about 50 million. What features might we put on a 2-million-transistor chip? Faster execution. And you can add a lot of memory on-chip, so the machine isn't always waiting to get information from memory chips. You could add a floating-point arithmetic processor, which consumes some 70 or 80,000 transistors—only a couple percent of the budget. We could add a variety of other dedicated processors. It will have a lot of parallel processing capability, and hardware fault tolerance—redundant circuits built into the chip. And a lot of the 2 million will go to compatibility—why abandon $10 billion worth of existing software?

"What benefits will the user see? Simplified networking, improved graphics, and I hope they'll be a lot easier to use. Or we could put a whole simple computer system on one chip, greatly reducing the cost of a run-of-the-mill microprocessor."

Rosen also had some predictions for the near term: "Fortunately, John Adams didn't close down the patent office in 1799, although he wanted to, feeling that everything to be invented already had been. In the next five years or so, I don't think there will be an increase in word length; 32 bits is absolutely adequate to meet all our needs in memory addressability and in the speed you need to communicate with disk drives, printers, video displays, and so forth. You will, however, see this basic 32-bit architecture, whether it's Intel's or Motorola's, go up in performance at lower cost. You will also see lots more co-processors—graphics co-processors, more advanced math chips, better input/output processors. Chip memory and disk storage will continue to grow, all at a much lower cost per bit, of course, and with faster access as well. Displays are going to higher resolution, and I think you'll see flat, low-power, color displays for portables in a few years. Further miniaturization—in a few years, a ten-pound portable with more functionality than any PC today."

For the year 2000, according to Moore, predictions are much tougher. "What do you do with 50 million transistors? That's two hundred

"Can you imagine an office environment where you have 50 people talking to computers?"



Talking to computers is a matter of course in the 23rd century, but it's not so easy in the 20th, as Chief Engineer Scott discovers. In this scene from *Star Trek IV: The Voyage Home*, **he is unsuccessfully trying to speak to a Macintosh, using its mouse as a microphone.**

80386 chips in one. It's a mind-boggling amount of electronics. We could put every function we've ever built to date on one chip. We'll definitely put a lot of software on the chip and increase its parallelism.

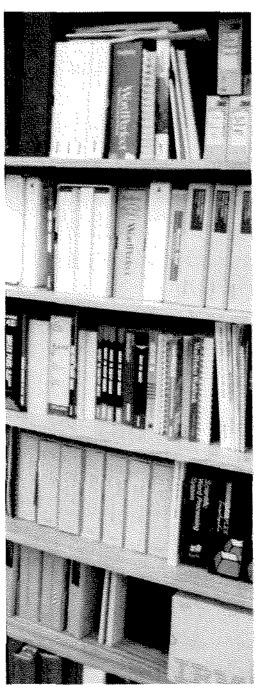
"The user's benefits will include speed: desktop computers that execute billions of instructions per second. Systems interconnection will be very easy, resulting in local and global networks and instant access to data at a level we can hardly conceive of today. A lot of the budget will go to the human interface. I hope to never open a manual again after the late '90s. And I hope a lot of the artificial intelligence functions really come into play."

Artificial intelligence has been slow coming to PCs because they haven't had enough horsepower, Rosen claimed in his lecture. "AI is a hog. It requires a lot of speed, lots of chip memory, and lots of disk storage, but now with the third generation of personal computers we finally have the hardware to go with the software's requirements. AI includes pattern recognition, handwriting recognition, expert systems, machine intelligence, natural language use, and speech recognition. When we get a system that recognizes continuous human speech, regardless of who's talking, we'll be able to dispense with the keyboard. The keyboard is a big impediment to anyone who doesn't use it frequently."

Kahn and Moore were less sanguine about the imminence of voice input. "The user interface—how the user gets information into and out of a system—is a surface," said Kahn, "and the depth of what's available in the computer lies underneath. There's an evolution going on from DOS-type (computer prompt and command input by keyboard) to graphical, a set of pictures on the screen and a pointing device (a light pen or a 'mouse') to select the function wanted. Some people want to get to a natural-language interface, so you can talk to the computer the way we're talking now. Can you imagine an office environment where you have 50 people talking to computers? And what if someone calls you on the phone while you're talking to it? It's like a videophone—do you really *want* people to see you on the phone? (I've got a phone in my bathroom.) So it's an interesting proposition, and the technology will exist to do it, but do you want it all the time?"

"I think it will be well into the next century before we're really comfortable with voice input instead of the keyboard," Moore said. "The keyboard is really pretty efficient—if you know how to type. Maybe we'll have to teach

"People love their favorite software, and why shouldn't they—they spent nights *learning it, and the last thing they want to do is to have to learn something else."*



typing in computer science classes."

Rosen saw user-friendliness as an important factor in the industry's future growth. "The 20 percent of the market we have now are the easy sales—the people who want to be first on the block to have one," he said. "How do we get the other 80 percent? It seems contradictory, but we need high-performance computers to attract low-performance users. You've got to make the machine less complicated on the outside by doing more work inside. You need software that's more intuitive and easier to learn—and with consistent interfaces between user and machine, graphics-based, so you can dispense with the manual. People don't read manuals anyway, so you might as well get rid of them. The Macintosh has done a lot in this direction."

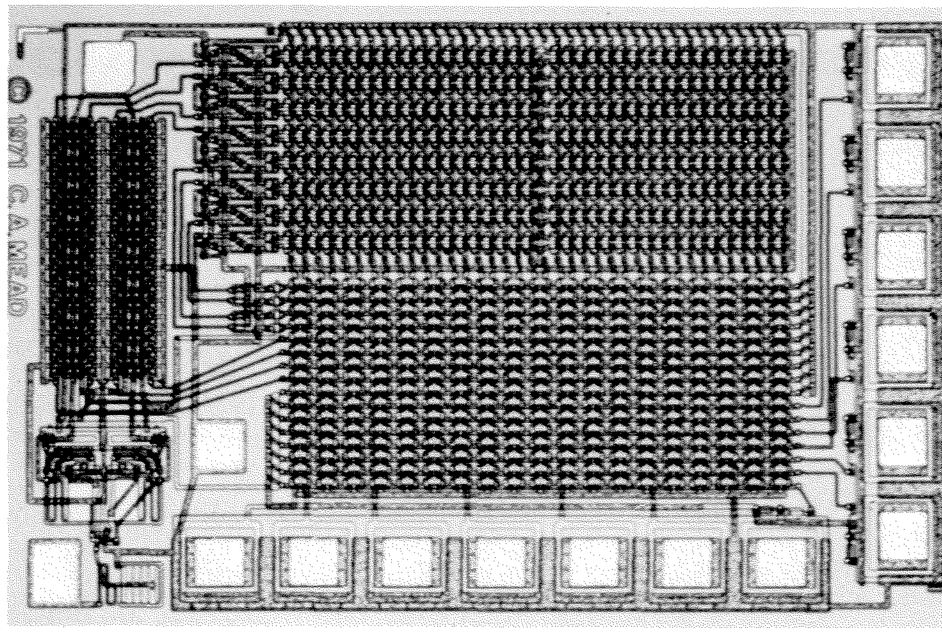
Kahn had a different viewpoint: "Graphic interfaces are not necessarily easier to use. It is easier to get into something, but running it may not be trivial. There are some Macintosh programs now where you have to press SHIFT, COMMAND, SPACEBAR, and move the mouse down to make something happen on screen."

As for future software, Kahn predicted that the main categories—word processing, spreadsheets, database managers, and communications packages—will not change much, "but there will be all these new tools, like AI and parallel processing, to do them with."

"The next thing in word processors will be to make them 'habit-compatible,'" forecast Kahn. "If you like to do things one way, why should you have to learn another way just because some software publisher thinks it's better? The machine should learn to work the way the user works, not the other way round. It's called ergonomicity—the human factor in design.

"People love their favorite software, and why shouldn't they—they spent *nights* learning it, and the last thing they want to do is to have to learn something else. Commands have to be logical and intuitive so people can remember them or figure out how to use them. People were screaming at a product like WordStar, saying it was difficult to use, but at least it had logical ways to remember things. Some of the more 'modern' word processors don't—there is no logical way to remember that pressing SHIFT ALT F4 does whatever it does.

"Software will have to get faster. People *hate* slow software. We all know we're going to die, and we have better ways to spend the time we have than sitting in front of a screen reading, 'Please wait while I process this command.' But the wait is going to get worse if we're not careful, because the processors and architectures we'll be using in PCs for the next several years are single-processor, single-memory-bank architectures managed by multi-tasking operating systems. Which means that instead of one program having all the hardware's resources, you'll be running several applications at once, swapping them all in and out. Accessing rotating (hard) disk storage is the PC's slowest function, because it's mechanical. And you're sharing the processor's time, too. One big, slow application, like sorting a massive mailing list, will penalize the whole multi-tasking system. So software engineers will have to write smaller, faster code.



The first chip designed by a silicon compiler was created by Carver Mead in 1971. As the complexity of chips increased, designing them became more and more difficult. When the silicon compiler became commercially available, complex chips could be designed in days instead of years.

"We all know we're going to die, and we have better ways to spend the time we have than sitting in front of a screen reading, 'Please wait while I process this command.'"

Craftsmanship will be even more important than it is now, perhaps crucial."

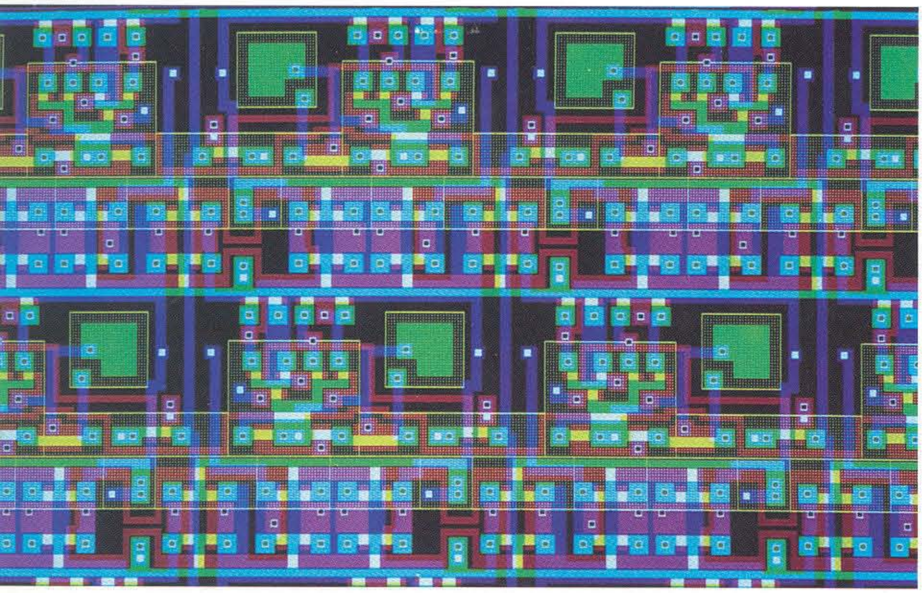
All the speakers noted the disparity between hardware and software development. Moore is convinced that the semiconductor engine will grow as long as the market holds, "so go use your 4 million transistors this year. But will the software to fuel that engine in the year 2000 be ready? Look at the PC-AT, which is basically the 80286. It has all the functions for multiprocessing, but none of the PC software uses it. It's just baggage—unused six years after the chip came out and probably eight years after the software people knew it would be there. I don't see the software catching up. Chips are growing exponentially in complexity and improvements in software are nowhere near the same rate."

Even Kahn agreed that software has to catch up. "Technology will help us build better software," he said. "Software people will have to master technology like AI and apply it where needed. The last thing we want is for software to become sloppy because we're just playing catch-up. There's no point in having more memory and faster processors if it just goes to support software that could be, and should have been, streamlined."

"New software stimulates hardware growth," said Rosen. "Look at the 8-bit, 16-bit, and 32-bit microprocessors. The 8-bit, starting with, say, the Apple-II in 1977, didn't take root until the first applications software came along that business professionals could use: Visicalc, two years later. The same with 16-bits. IBM introduced the PC in 1981. It grew moderately in '82, then explosively in '83. Lotus 1-2-3 was introduced in January of '83, and that was the first application that took full advantage of the 16-bit architecture. Now we have the same situation with 32-bits. Compaq led this generation with the DeskPro 386. But we haven't seen the application yet that will make them really take off. Nothing takes full advantage of the hardware. What is the new 1-2-3, the new Visicalc? If I knew, I'd invest in it.

"It's interesting that most software comes from smaller, more entrepreneurial companies. Compare this to the hardware situation, where IBM, Apple, and Compaq together have a 77-percent market share, and about 100 companies are fighting for the other 23 percent. It's like a supermarket shelf—there is only room for so many brands of toothpaste, and those that can't get on the shelf die. Not too long ago there were so many PC companies going out of business that it was said the industry was entering a new chapter, Chapter 11. To start a new PC company today is impossible. No one will fund a company to compete for unavailable shelf space. . . . But the software companies, such as Microsoft, Lotus, Ashton-Tate, Borland, WordPerfect, Autodesk, have all started off very small and have stayed largely software-only. The big hardware manufacturers have either not tried, or have been very unsuccessful at creating PC software. I think creating software is a discipline that lends itself better to a small, dedicated company than to an appendage of a large manufacturer of iron. And software companies can still get started today, they're still fundable by the venture-capital community."

The emergence of standards has been critical



Mead's silicon retina, which contains a 48 x 48 array of these cells on a tiny chip, mimics the processes of "slippery, gooey neurons." The colors in this computer represent the semi-layers, which perform the neural processes of the retina's layers of cells.

to developing software as an industry in its own right. According to Mead, "Standardization has unleashed a wave of innovation in software—many bright, innovative people were able to concentrate on individual applications. That development has had an immeasurable effect on the economy. Although it doesn't show up in any of the standard measures of productivity, it has allowed us to do things we couldn't have imagined doing previously."

But it wasn't always that way. "In 1980," said Rosen, "there was a Tower of Babel in operating systems. There was Apple DOS, which didn't talk to TRS-80 DOS, which didn't talk to Commodore PEP/DOS, which didn't talk to IBM. Then, when Microsoft's MS/DOS came out in late '81, we had a *de facto* standard, at least for business users, for quite a while. This had a galvanic effect. . . .

"That's changing now. Microsoft will release the last version of DOS, Version 3.4 this year. DOS is going to dominate the business market for at least another two years, until OS/2, released in the last two months, kicks in. OS/2 is an IBM/Microsoft joint release. There are over a thousand new OS/2 applications that will be on the market by 1989. OS/2 will probably pass DOS in 1991. In the meantime, Macintosh has become a force in the business market with its operating system, another standard. But even though there's no longer a single standard, each is large enough now to attract software developers and other support companies to it, ensuring they'll all have strong growth in the coming years.

"I'd like to show you what happens when you take the long view of technology. Almost without fail, when technology changes, the leadership changes too. If you look at calculators, the big names in electro-mechanical calculators were Frieden, Marchant, Victor, Monroe—where are they now? They aren't. We have Hewlett-Packard, Casio, Texas Instruments—a new set of players. Components—vacuum tubes were led by RCA, Raytheon, GE, and Sylvania—they're barely participants in the semiconductor industry. Or computers, as we've gone from mainframes to minis to PCs. Look how sleepy almost all the mainframe companies, with the exception of IBM, were as we went to minis. Look how sleepy the minis have been as we've moved to PCs. Or in software—look what happened to Visicalc when Lotus 1-2-3 came along; WordStar once had almost 100 percent of the word-processing market, and then WordPerfect and 20 others passed it by. There's a lot of inertia in business, and I think it behooves all of us to remember that today's complacency could well become tomorrow's obituary. It's not that some domestic or foreign competitor obsoletes us, we obsolete ourselves. Only those companies that keep innovating, keep pushing the state of the art, survive."

Mead looked at the reason behind the inertia. "Breakthrough technologies come from a direction not foreseen by the existing industry or predicted by the analysts. This may sound like an amateur taking a potshot at the professionals, but that isn't my intent. A breakthrough technology, by definition, is not part of the existing culture that's established in companies; great inventions come out of left field. And there's a corollary to that observation: New technologies are adopted last by the companies that need them most. That's because they're not part of the culture that drives the company. Therefore, they won't be seen because they're contrary to what was successful in the past. Once you've built a successful culture, it is difficult to see your environment in a new way.

"Fortunately, we have an entrepreneurial system. We depend on the innovations of the citizens of a free economy to keep ahead of the bureaucrats, ahead of the people who make a living by controlling and planning. In the long term it is the element of surprise that gives us the edge over much more controlled economies. I think this must be true in any industry that is driven by the intellectual insights that make possible entirely new ways of doing things."

Innovation is alive and well in universities as well as industry. In his lecture, Mead described some of his work at Caltech on custom chips designed for specific applications. Using a VAX for the electronic synthesis of high-quality music, the simulation takes 600 times longer than real time. "So we designed a chip whose architecture was specifically crafted for this task. One such chip simulated the instrument in real time. That chip had the effective power of about 600 VAXes."

Mead has also built a chip that simulates the neurons in an animal retina (*E&S*, June 1987). "It does a fantastic amount of computation at a level that can't be done by a supercomputer. (Those slippery, gooey neurons are at least a billion-fold more powerful than our biggest supercomputers.) I believe that building silicon chips that compute analogously to our carbon-based nervous system will be the next fundamental step in electronics," Mead predicted.

"There's a lot of inertia in business, and I think it behooves all of us to remember that today's complacency could well become tomorrow's obituary."

As for the outlook for U.S. competitiveness in the microcomputer industry, the speakers were optimistic to varying degrees. Rosen concluded his talk with four observations. "First, the microcomputer industry was created in the U.S. because of our unique entrepreneurial technology sector—our tradition of individualism going back to the first homesteaders, of people willing to take chances. There is no stigma attached to failure here; you can always pull up stakes and try something else. That's less true in Europe, and in Japan it's very hard to fail with honor. Second, after 10 years, the U.S. still leads the microcomputer world, and by a wide margin, if I may be chauvinistic. I think we're likely to continue, both in hardware and in software, for many years to come. Third, microcomputers are the fastest growth industry ever, with lots of room still to grow; and finally, I think that microcomputers are going to become the dominant part of the entire computer industry in the 1990s."

From the vantage point of a chip manufacturer, Moore seemed a bit less enthusiastic. "It's of significant concern that most of our memory chips are now built overseas," he said, "as our systems manufacturers found out in the current shortage. The U.S. now produces only a couple of percent of the world's D-RAMs (dynamic random access memory chips). The dynamic RAM was the product that got Intel going, and we dropped out several years ago because we couldn't see a return on investment there, with the Japanese in particular just pouring money into market share. Once you lose an industry like that, it doesn't come back. It's not just a case of getting incrementally cheaper—I think right now we could probably make D-RAMs as cheaply as the Japanese can. But it has to get to the point where you can see that lasting for a long period of time, and I don't see that. The Japanese are reinvesting in vast amounts of capacity because of the present shortage, and next year they'll probably catch up with demand again, prices will plummet again, and we'll be very glad we're not in the D-RAM business. So we're going to have to get used to our D-RAMS and a lot of other components coming from overseas. A tremendous interdependence is developing. We can expect the Japanese to be major competitors, and we'll continue to see some loss of our chip market, especially as Japan is now a larger market for semiconductors than the U.S., and the Japanese have a tremendous advantage serving that sector. But they also have a very significant disadvantage serving the U.S. market. We have made some progress in trying to get the competition to be more fair than in the past. If we could get free trade, we'd be happy—it's never been free trade. But I don't believe they're going to put us out of business."

Mead, reporting from the thick of the creative end of the business, claims that "there is still plenty of innovation in the electronics industry. We don't need the feds to bail us out. We're doing just fine. There is as much innovation and creativity in this business now as I ever have seen, and there are numerous directions for us to travel in the future."

"The future trend in microcomputing, I think, is eliminating the 'micro,'" said Moore. "Increasingly, microcomputing *is* computing." □