Interview with Gordon E. Moore

Scientific American

September 22, 1997

PART 1 THE BIRTH OF THE MICROPROCESSOR

Imagine being in the room to witness the genesis of an invention that would profoundly change human civilization. Gordon E. Moore was there; he supervised the project 25 years ago that put the first "computer on a chip."

As the co-founder of a Silicon Valley startup known as Intel Corp.. Moore has seen feat after feat of technical prowess sustain a geometric growth in computer power so regular you can practically set your watch by it. The growth is so predictable and important, it has been canonized as a law–"Moore's Law." It was Moore who noticed as early as 1965 that microchips were doubling in circuit density (and thus in their potential computational power) every year or so.

Moore retired as chairman of Intel in 1989. In 1990, he received the National Medal of Technology from then-President George Bush. Now 68, still works several days a week in his old cubicle at Intel. Recently, Moore took some time to speak with Scientific American's west coast editor, W. Wayt Gibbs. Their wide ranging discussion is presented in four parts.

In part one, Moore describes how he narrowly missed studying nuclear bombs instead of microchips, how he helped found a multibillion-dollar company, and how he snookered Japanese investors out of rights to microprocessors. Part two recounts the discovery of Moore's Law and his take on the formidable technical obstacles it is beginning to face.

On October 6, look for part three, in which Moore describes the technological tricks that Intel engineers are developing to keep the computer revolution humming. And in the final section, which will appear here on October 13, Moore will reveal his predictions for the next 10 years of computing.

When you were younger were you more interested in science than in engineering?

Yes, from the time I was in junior high school I decided I wanted to be a chemist. I didn't quite know what a chemist was, but I kept it up and got my Ph.D. in physical chemistry. My first job out of school was to do basic research at Johns Hopkins University's applied physics lab. Then William Shockley [co-discoverer of semiconductors] caught up with me.

So, tell me about Johns Hopkins. Were you working on weapons research and military research?

I wasn't in weapons research. In the applied physics lab there was what they called the research center, which was essentially doing basic research on . . . well, on whatever we were interested in, almost. I was looking at the shapes of infrared absorption lines and some spectroscopic study of flames.

Why were you dissatisfied with that job?

First of all, the team I was with was breaking up, and my two bosses were moving out. I started calculating the cost per word in the published articles [that emerged from the lab] and decided that at $5 a word, I wasn't sure that the government was getting its money's worth. I didn't know whether anybody was even reading them.

I decided I really wanted to get closer to something with practical application. I was looking for some technical stuff that would lead to a real product. I didn't know quite what.

I interviewed at several places, one of which was the Lawrence Livermore Lab out here, and considered thermonuclear devices as a practical application. That's where Shockley got my name, actually. They made an offer to me, and I decided to take the position of Inspector of Nuclear Explosions.

That's what the job involved?

Essentially. Shockley got permission to go through Livermore Lab's files of all the people to whom they had made offers or who had turned them down. He thought he needed a chemist in his new operation; chemists had proved useful things in his group at Bell Laboratories. So that's how he came up with me. I had no background whatsoever in semiconductors. I did at least know who Shockley was, because I had heard him talk in Washington.

So that background wasn't necessary for the job?

No, there weren't an awful lot of people around in those days who did know much about semiconductors.

What happened with Shockley? That didn't work out for very long either, did it?

I was there for about a year and a half. We fiddled around, trying to make some devices. Then Shockley changed direction. When I first went to work for him, he was thinking of making a transistor. But then he decided he wanted to make a rather obscure device called a four-layer diode. Mainly it was just that while Shockley was a technical genius, he really didn't understand how people worked very well. He stirred up things internally.

A group of us finally went around Shockley to try to get that straightened out. We went to Arnold Beckman, the source of his funding. We thought we were making considerable progress. But Beckman finally told us that Shockley was the boss, and we'd just have to learn to live with him.

Sounds like a mutiny of sorts.

Well, it was, I mean we had burned our bridge pretty badly. So after doing that, eight of us felt that we really had to leave and go someplace else. We didn't want that kind of situation there long-term. We were later called the Traitorous Eight and a variety of other things. We founded Fairchild Semiconductor.

Now there was already a Fairchild making other things, right?

Fairchild Camera and Instrument supported it. To be more complete, we sat down with a copy of The Wall Street Journal and went down the companies in the New York Stock Exchange to see which ones might like to start a semiconductor operation. We identified something like 35 companies. The investment bankers that we were dealing with, one of whom was named Arthur Rock, a Harvard business school graduate at the time and a senior partner, went and contacted all the companies we identified, and they all turned it down without even talking to the group.

Then they caught up with Sherman Fairchild, who was the founder of Fairchild Camera and Instrument and also of Fairchild Aircraft, and he liked technology. He introduced them to the person who was then running Fairchild Camera and Instrument. They sent a representative out to talk with us and decided to support us in founding a new company.

So now you had your own company, yet that didn't work out perfectly either.

There was a lot of on-the-job learning there. It was quite technically successful, and we developed a fairly large business out of it. It became a division of Fairchild a few years later when they exercised an option to buy it out. Our division was something like 30,000 employees and $150 million. So it was a fairly successful operation.

But then in the parent company there were some changes that I never have understood. They fired two chief executives within six months. They fired the first and put in the person with whom we had dealt most. And then he was out, and they started trying to run the company with a three-man committee of the board of directors, while starting to look on the outside for a new president.

Robert Noyce [a co-founder of Fairchild Semiconductor and later of Intel] was the logical internal candidate, but he wasn't too enthused about the way that was going. I could see that the company was going to change quite a bit, because they went outside for someone.

So I decided I'd rather leave before than after the changes. And I'd been a bit frustrated with what I was doing. I was running the laboratory, and it was getting harder and harder to transfer stuff to manufacturing.

The walls were growing higher and higher?

Yes, the wall grew higher and higher. The more technically competent the manufacturing people became, the less willing they were to accept the things that had to be done to the technology coming out of the lab. So, out of the convergence of those things, Bob decided to resign, and I decided that I would, too.

Were you two pretty close at that point?

We'd been working with one another since the early Shockley days. He came to work for Shockley on Friday, and I arrived the following Monday. At Fairchild, at the beginning we were peers, then he became general manager of the division, so he became my boss for a long time.

Legend has it that Bob typed up a simple one-page business plan for Intel on his home typewriter. Is that actually true?

There is a copy of it downstairs. It says absolutely nothing. It is completely and utterly vague. Our plans were a little more concrete than that suggests!

The idea we had for Intel was to try to make complex integrated circuits. The problem was that when you define complex integrated circuits, they tended to be unique–you know, used once in a computer system or something. And we saw semiconductor memory as an opportunity to make something complex and sell it for all kinds of digital applications.

So that was the first thing we went after. That was really the basis of our business plan. And then we were looking for other products that had the same kind of characteristics, namely complex chips that could be made in large volume.

The idea of the electronic calculator was really just getting going then. We started looking for a calculator company that we could deal with. But we were a little late–all of the calculator companies you'd heard of had already tied up with a semiconductor company.

So we caught up with Busicom, which was a Japanese start-up–a peculiar operation in itself, not very well financed. But they wanted to build a business in scientific calculators, and they were looking for a semiconductor partner. They came in, actually, with all of the logic done for their family of calculators, something like 13 chips with considerable complexity.

They had done all the design work?

They had done all the design work on those. We had a small engineering group, and most of the people were up to their eyeballs in memory circuits, so we didn't have a lot of engineering to put on something like this. To do 13 different complex custom circuits was far beyond what we could tackle.

Then one of the guys we had looking at this, Ted Hoff, looked at what they were trying to do and told us that with a general-purpose computer architecture, he thought he could do all of their calculators. Beyond that, I remember him suggesting elevator controls and traffic light controls as two specific things–this would be a general-purpose controller, too.

His real insight was seeing that this could be done with about the complexity of the MOS [metal oxide semiconductor] memory circuit we were making then. So the idea of a single chip computer was something to talk about in the industry, way in the future, a "someday" kind of a deal. But Ted saw that we were at the point where we could actually do that.

Was his insight to see that you didn't actually need as many elements as the industry guys had thought you would need or that in fact you had enough elements now?

Maybe a little bit of each. He'd been working with the old DEC PDP-8, which was a relatively hardware-efficient computer, and he knew it very well. So he knew some of the techniques used to make things hardware efficient. And he just recognized that with about the same complexity as the memory chip you could make a simple processing unit. That cut the project down to bite size because you had one special chip to design, instead of a dozen or so, and a couple of memories, which were just variations of what we were doing anyhow. So this knocked it down to something we could actually try.

Then our problem became selling this to Busicom–making them throw away all of the design work they had done and start all over with a little start-up company here in Silicon Valley. I remember we faced that meeting with a lot of trepidation. We had the chief technical guy from Busicom visiting us with one of his engineers. We went in there and gave a pitch on this, expecting a lot of push back, but he said, "Fine, we'll do it your way."

Not at all, and more than that he didn't offer any alternatives. He immediately agreed, which really shocked me. I thought we were going to have a really tough selling job.

Did you already in the back of your mind think: "Gee, this is a great way to boost our business–really to fund an RD project for us, because we can then crank out different versions of these things and sell them to others?"

Well, initially we were looking at it as a way to get into the calculator business. And we knew it had potential beyond that because Ted had pointed that out first. But it was just another one of these chips that we could make in fairly large volume.

In fact, the way the development got done, Busicom paid a portion of the development costs and therefore owned the rights to the design. So in the beginning, we weren't able to sell it for these other applications.

Then Busicom was meeting a lot of cost pressure in the calculator business and wanted to get lower prices for the chips. We started shipping these chips in early February 1971. Busicom wanted lower prices, and we wanted higher volume. We negotiated a deal to give them lower prices if we could have the rights to sell this chip for other applications.

So we got the rights for noncalculator applications by giving them lower prices on the things they had. And then eventually, when Busicom got into deeper financial trouble, we essentially gave them back their $65,000 and got the rights to the chips back for all uses. So the Japanese initially owned all the rights to microprocessors, but sold them for 65 grand. In retrospect, it was kind of like the purchase of Manhattan [from the Native Americans, for $24].

I've read that the 4004 [Intel's first commercial microprocessor] took nine months to design and create. Do you remember how many engineers worked on that?

In those days all chips took about nine months. It wasn't very many; something like four. In fact, one problem we had was we didn't have a team ready to take on the design right away. We had to go out and hire some more people. I don't remember the exact timing, but one of the Busicom engineers, named Shima, was due to come over here to meet with Intel and check our progress.

We had just hired Federico Faggin, who ran the design team, a week before the guy got here. So he arrived, and it was obvious that we hadn't done anything! Shima ended up staying over here as a Busicom employee and then later worked for Intel.

It would be interesting to contrast the manpower that went into the 4004 with what went into, say, the Pentium II.

My recollection is that about four engineers worked on the 4004 as well. Now to design one of our chips we have more like 400 engineers, often spread around several different sites. And today it takes more like four years. It's a much bigger deal.

PART 2
MOORE'S LAW

From being part of a closely knit team that created the first "computer on a chip" 25 years ago to co-founding and shepherding Intel Corp. into a semiconductor powerhouse, Gordon E. Moore has remained a dominant figure in the development of the modern computer.

In 1965, Moore noted that the number of devices on a microchip (and hence the potential power of a computer) was doubling each year, and he projected that trend would continue for 10 years. Thirty years later, that geometric growth, now canonized as "Moore's Law." remains the fundamental economic force driving the computer industry.

Recently, Moore took some time to speak with Scientific American's west coast editor, W. Wayt Gibbs. In this second section of their four-part interview, they discuss Moore's Law and its implications.

We've all heard of "Moore's Law" that computer power doubles each year. Let me get a little more detail. I've seen different dates for when you came upon this observation, 1964 and 1965.

It was 1965. I published it in the 35th anniversary edition of Electronics magazine.

Do you remember when you noticed the trend?

When I was writing the article! The gist of my article was really that integrated circuit technology is going to make electronics cheap. I was trying to drive home the fact this was the route to low-cost electronics–it wasn't at all clear that that was true. Most of the integrated circuits made so far had gone into fairly expensive machines, like Minuteman missiles or something. They were just starting to make commercial inroads.

I happened to plot the increase in complexity and saw that it was doubling every year, so I extrapolated that for 10 years. I extrapolated it from about 60 components to about 64,000 components on a chip–a pretty long extrapolation. I was just making the pitch that the cheapest way to buy a component 10 years later was going to be as one of these very complex chips. It tracked that curve better than I ever could have imagined.

People make all sorts of long extrapolations today, and a lot of the time they only half believe them. Did you honestly believe that it would last 10 years?

I didn't really have any feeling for the precision of it. Really the precision wasn't even important for the argument I was trying to make. At the end of that time I dug back into it and what had happened, and in 1975 I gave a paper updating the thing, and there I tried to be more precise as to what had contributed to the progress we had made. I predicted we were going to change from doubling every year to doubling every two years, which is kind of where we are now. I never said 18 months, in spite of that being quoted in literature quite often!

Plotted on semi-log graph paper this does make a beautiful smooth line, but that smoothes over many difficult engineering struggles that have occurred along the way, doesn't it?

Yes, it sure does. In one respect it has become a self-fulfilling prophecy. People know they have to stay on that curve to remain competitive, so they put the effort in to make it happen.

In my view, this was the best thing I ever did to the Japanese semiconductor industry. Once they understood the progress of DRAMs–one, four, 16, 64 [megabit]–they could multiply by four as well as any of us. Then, for the first time, they really had a fix on where the industry was going.

Before that, the industry seemed to move in more or less random directions, which didn't work well in the Japanese top-down corporate culture. But once they had a road map of where the industry was going, they became very formidable competitors. And even now, people look at these curves at the semiconductor industry association and essentially turn out road maps for staying exactly on the curves we have been on. They just try to get the industry to commit the resources to be there. So each of the individual participants tries to get ahead of that curve.

To consumers, this almost seems to be a law of nature; it just happens by some magic. They don't necessarily see the tremendous engineering efforts that have to go into knocking down obstacles each time.

Yes, there is a phenomenal amount of R&D work involved in this. This year we'll spend about $2.5 billion on R&D; it was about $1.9 last year. And we represent only about 10 percent of total industry spending on research. So it's up to $20 to $30 billion a year in R&D. A big, big investment.

Does it seem to be getting harder or easier to knock down these obstacles each time we move from one generation to the next?

I get farther away from it each time, so it gets easier for me. Technically these are phenomenally challenging problems, and the things we used to do relatively casually now take teams of Ph.D.s to advance the technology by an equivalent amount. But the amazing thing is that we've been able to do this for almost 40 years without running up against a barrier that really stopped progress.

Eventually, we may run out of gas. We are subject to the fact that materials are made out of atoms, and things like that. We're getting down now into some places where the atomic nature of matter starts to be a concern.

Let me run through some of the most commonly cited obstacles facing continuing growth in chip density, and let me get your opinion on each. For instance, the cost of smaller features, switching to excimer lasers: is that a big problem?

The cost overall of a modern fabrication facility keeps going up, but it hasn't proven to be the barrier people thought it would. At one time even talking of a billion-dollar factory would generate concern that only a couple companies could build them. Heck, now we all build them.

A couple of them are over $2.5 billion. There's one in Albuquerque that's $2.7 billion total investment. We have another near Phoenix that is not full yet, but by the time it's full will cost more than $2.5 billion. It's absurd! They're continuing to go up, but there doesn't seem to be a shortage of capital that would impose a limit.

But the margins are decreasing, right? Aren't the costs of the plants going up faster than the returns you can get from them?

Not for us; our margins aren't decreasing in that respect. The memory chips, the DRAMs, do go through cyclical periods that depend on demand-capacity relationships. DRAM prices have come down 80 percent in the last year. That clearly makes a huge difference in the profitability of the DRAM makers; they're not turning out five times as many chips as they were. So right now they're in one of these dips, but history can be depended on–it'll come back.

There was a terrible dip in the '81 time period, a terrible one in the '84-'85 period, and then it came back strongly after that. In '91-'92 there was a big drop, but it came back again, and now we're in a dip since last year.

It's the nature of the product that you've got a huge fixed investment in the plant–and more important, in the people who run the plant, all the engineers and people in the place. Once you've got that and demand goes down, it's very tempting to look at the incremental cost of making one more memory. You are faced with the choice of shutting down the plant or running it and filling it with something more than incremental cost but less than total cost. The industry has always opted to run it and sell the chips at lower prices.

And it's been a very elastic market. The memory in your PC keeps growing–the memory in everything keeps growing. So if you just run it, and wait awhile, the elasticity of the market has always bailed the industry out. So the common response is not to shut down plants but rather to run them, sell them for what you can get, and wait. The knee-jerk reaction of this industry is: if there is any problem, cut the price.

PART 3 ADVANCING THE TECHNOLOGY

In 1965, Gordon E. Moore noted that the number of devices on a chip (and hence the potential power of a computer) was doubling each year–and projected that out 10 years. As astonishing as it seemed, that relentless progression held true. His initial observation is now known as "Moore's Law." As co-founder and chairman of Intel Corp., Moore himself had a lot to do with proving out his prediction. Each leap in power required new technology that shrank the size of circuit lines so that ever more devices could be packed onto a sliver of silicon. In the 25 years since a team headed by Moore produced the first true microprocessor, Intel and its competitors have been able to pull off the stream of technological breakthroughs needed to sustain the computer revolution.

But how far can semiconductor technology go? As microcircuit transistors shrink from microscopic to nanoscopic dimensions, is Moore's Law about to run out of steam? In this third section of a four-part interview, Moore speaks with Scientific American's west coast editor W. Wayt Gibbs about the new advances in design and manufacturing that the industry is betting on to provide faster and more complex chips for the next generations of computers

While computers continue to demand more and more memory–at lower and lower prices–many experts believe that we are reaching the limits of optical technology to etch ever-smaller circuits. Some people are worrying about qualitative jumps in the increase of fabrication costs due to having to move beyond optics. Can Moore's Law survive the transition?

Moving beyond optics is a real challenge. We keep pushing optics further and further. Frankly, we've done it further that I ever would have imagined. There used to be conventional wisdom that a minimum circuit-line width on a microchip of one micron was the limit that we could do optically. Now we can do a quarter micron. The next couple generations–0.18 microns, probably 0.13–it looks like we can do optically.

Beyond that, life gets very interesting. We have three equally unattractive alternatives, maybe four. I don't know quite how it's going to go. There's been a lot of effort spent on x-rays. X-rays were said to be the technology of choice at half a micron; now people hopefully predict using them at submicron levels–at 0.13, for instance.

But it'll probably get tougher–it's the nature of the mask. X-ray photolithography requires one-to-one shadow imaging. In optical lithography, we make the pattern a lot bigger than the device, then project it down. Well, the one-to-one mask problem is extremely severe, particularly for x-rays, because the mask layer has to be thick enough to absorb the x-rays.

What you end up with, if you look at it under a microscope, are tall, skinny features. They are much taller than they are wide, at these dimensions. It is very hard to make the mask perfect enough and then to do the precision alignment. So while a lot of work continues on x-rays, some of us have lost our enthusiasm for that technology.

Then there's the idea of electron beam writing. This can be used to make the small features. But it tends to be relatively slow. As you go to smaller dimensions, the total distance the beam has to travel to make the pattern keeps going up. The slowness gets emphasized by finer and finer dimensions and more complex structures.

Now the industry is looking at ways to get around that by using electron beams in shapes other than a pencil beam–to write with squares, rectangles or whatever depending on the feature you're trying to build. Worst case, we will be able to make a layer or two of some very fine structures with an electron beam and then use optics to add on structures that are not so fine. That way you can still make very small transistors where you need them. That doesn't get you as far as you would like to go, but it gets you some of the advantages. So that's kind of a fallback position.

Another option that we think deserves a very good look is using an intermediate wavelength, between x-rays and the ultraviolet light we use now. This has been given the name of EUV, for extreme ultraviolet. It used to be called soft x-rays, but x-rays have gotten enough of a bad name that it's called ultraviolet now. This is a range of wavelengths on the order of 13 nanometers.

Versus what wavelength range for x-rays?

Well, this is really soft x-ray, but x-rays are typically down at more like 13 angstroms, an order of magnitude smaller. Actually around 30 angstroms is where the x-ray work generally is done. Anyhow, at 13 nanometers, .013 micron, at that range you can still make mirrors. They're not easy–you have to coat them with something like 81 layers of masking material. And with current materials, reflectivity is only about 70 percent.

This is actually technology we developed for Star Wars [the Ronald Reagan era anti-missile program]. We're thinking that this is potentially a lithography system that will take us as far as the material will let us go, a long ways from where it is now. Intel is actually trying to get an industry consortium together to support the research on this to see whether it really is practical or not. Then there are things like focused ion beams. Again, that has the resolution possibilities but also a lot of problems.

But if EUV works, we have to go to a completely reflective system, because nothing is transparent in that range. You have to have a reflective mask instead of a transparent mask, which is an absolute change in the technology. You have to have a vacuum system. Everything has to be completely enclosed with inert gases to stabilize the material. You have to have a new resist system, something that will penetrate it enough at that wavelength. So there's a tremendous amount of engineering involved in making this work.

Is it clear that in principle at least, all these elements exist and will work?

It's clear that the optical things do. Is there a resist that has the desired characteristics? I don't know, but I suspect there is. People can make x-ray resist, people can make UV resist. It'll take a lot of fooling around, but somebody who really knows their organic chemistry well will come up with something.

Another roadblock that is sometimes cited is memory speeds, bottlenecks happening outside the processor that prevent it from running at full capacity.

This is an interesting deal. That used to be the case. The processors were quite a bit faster than memory, and that's what led initially to the complex instructions [used in Intel's CPUs]. You want the computer to do as much as it can with the stuff that's there. Then when semiconductor memory got up to the same order of speed as the processor, that's when the idea of RISC [Reduced Instruction Set Computing] processors came along.

With RISC, you can go to memory a lot more often and do a lot of simple instructions. Now we're going back to the situation we had before, where the memory is quite a bit slower than the processor. I guess that would swing the scales back toward making complex instructions.

That's something one has to live with, but what has happened in the meantime is much more dependence on cache memory, which is built into the microprocessor itself. And the cache memory does run in the same range of speeds as the processor. On chip you can fetch data from memory every cycle; off chip you can get there every couple of cycles. And the effectiveness of cache memory is pretty darn good. So that gets around most of the problem.

Do you expect to see more processor real estate devoted to cache, then?

That's one alternative. But if you look in our Pentium IIs, what we've done is to jam a lot of cache memory in separate packages right up against the processor. So we have some on chip, and then we have a lot more just off chip. We think at least for now that is a better compromise.

Does this give you an intermediate speed between completely on-chip memory and separate DRAM [Dynamic Random Access Memory] chips?

Yes, it works as a level 2 cache–that is, a two-clock-cycle cache. On chip you can still stay with one. But two isn't bad compared to going off to DRAMs, which requires tens of clock cycles.

What about synchronization problems as clock speeds rise and chip and die sizes stay large?

That's something that requires a lot of attention. This isn't any area that I am expert in, but our people don't seem to be that concerned about it. In primitive circuit boards, keeping the clock signal consistent across the board was a problem. But there you had pretty significant dimensions. With the chips you can bring the clock in at a lot of different points, so you can keep it pretty well synchronized. It requires good engineering.

There are some–such as Ivan Sutherland and Robert Sproull at Sun Microsystems–who maintain that once you get into a gigahertz range it's going to be a real engineering headache to try and keep the clock signal synchronized everywhere.

A lot of things become headaches. Power is at least as big a concern. If you just let these things scale–you make the chips bigger, you make the frequencies higher–then you make the capacitance per unit area higher, since you have scaled everything. In two generations of technology, say from half micron to quarter micron, that's two steps down, to 50 percent of the starting size. When you look at the trends of making bigger chips, with more complexity and jacking up the clock speed, if you don't do anything else, the power goes up something like 40-fold.

If you start with a 10-watt device and go up 40-fold...the darn thing smokes! It'll keep your lap warm, all right. So that is an area that really requires a lot of attention. And, of course we've handled it to date by lowering the voltage. But you can only go so far on that. So power gets to be a real problem when you get up into these high frequencies.

Is power a limiting factor?

You're kind of in a multidimensional box, and that's one of the dimensions you have to worry about. I suspect that clock skew will be another one. These are tough problems that require a lot of attention, but we have a lot of workpower working on them, too. Exactly when it will end up limiting us is hard to say.

We clearly have a long way we could go before we get into trouble.

Would you give me your opinion on some of the technologies that are seen as most likely to help extend the life of the current procession of computer technology? How about phase shift masks: do you already use those in your manufacturing?

We keep avoiding them. Phase shift masks allow you to go to smaller dimensions with a given wavelength. They get very complicated to make when you go to a kind of random layout like you have on a microprocessor. It's easier to use them on memories. But if we don't have a shorter wavelength, it is the kind of thing we'll have to use to do the 0.13-micron generation with 193-nanometer excimer lasers.

So it sounds like you think they're going to be used eventually, it's just a matter of time.

I think it's likely we'll do something like that. We've done things sort of like that all along, although we weren't clever enough to call it phase shift masking. For years if we wanted to print a rectangle–if you just made a rectangular mask, the etched pattern tends to have rounded corners and look like a pillow due to diffraction–so we would just put little spikes around the corners of the rectangle to balance it out so that it printed a square. That's really a phase shift mask.

How about adding more layers to the chips?

More layers are something we do now without much concern. Going from one to two was tough, two to three was difficult, but five to six–piece of cake. A technology has come in there that is really amazing. This is the idea of chemical-mechanical polishing of the top surface. The problem used to be that as you went through more layers, the polishing got all screwed up. You'd get mountains and valleys and undercut levels, and things didn't work well. Now between putting down every layer of insulator and metal, we polish either the top of the metal or the top of the insulator flat. So we're always working on a flat surface. And that has really been a breakthrough technology in allowing multilayer structures.

How exactly do you polish them?

We have a great big lapping machine with some goo on there–chemical-mechanical, it's called. They use slurries that also react somewhat chemically with the surface. It's not just grinding. But it gives them a very flat surface. The end result is, we put five layers on top of each other and then ask the design engineers, "Would you like another layer of metal?"

Do you think that trend will continue, that chips will get even taller?

I think it will, yes. I think that's one of the real levers we have to work with.

Bigger wafers are coming, right?

I'm afraid so. Again I was a skeptic there. I convinced myself that we'd never go above the 200-millimeter wafer. The reason was, I argued, that the cost of material was going to become prohibitive. But the people who are going to supply it seem to think they can do it. Now, I haven't been at a silicon crystal growing facility in years. They must've learned something new since I was there.

Does it require an entirely new crystal growth technique, or does it just involve refinements of what they use to create 200-millimeter wafers?

It has to require something different, because the crystal hangs by this little seed. And the size of that seed has to be pretty small because you have to squeeze all the imperfections out of this seed before you start expanding. The limit to the size of the crystal you can grow used to be determined by the tensile strength of that seed: how much weight could you hang from it.

That was why I argued that you couldn't go much bigger. As you increase the diameter of the silicon crystal and keep the weight the same, you have to decrease the length by the square! So an eight-inch-diameter crystal might be about 18 inches, and I could see a 12-inch crystal only a foot long. Then it takes longer to get out to the full width of the cylinder from the pointed top, and longer to get back. You need a thicker saw blade, so you've got to cut thicker wafers. So everything went in the direction of saying you get far fewer wafers out of a 12-inch crystal than an eight-inch one. I thought that would be a real limit.

Now somebody must've learned how to go in there and grab the crystal and keep it growing, rather than support all the weight from the seed. That didn't used to be possible. And I don't quite know what they are doing, maybe they're getting away with short crystals. But somehow or other, the people who have to supply the silicon seem to think that 300 millimeters is okay. That being the case, the industry will build to 300-millimeter wafers.

Will it go to bigger die size, do you think?

Those are kind of independent variables. We could fit a lot bigger die on the 200-millimeter wafer if we had to. That depends partially on the field of the lithography tool. We don't like to have to stitch fields together. But the economics of that thing are limiting that as much as anything. We sell area, we sell real estate. And we've always sold it for about a billion dollars per acre of silicon; a bit less for DRAM, a bit more for microprocessors. But when I first started out in business, we sold it for about half that. And the problem is, if you let the die get too big, your costs get all out of whack. So, if you're limited in how much a particular market will pay for your product, you've got to limit the area also.

Assuming that the trend will continue for the next 10 years, what do you see happening with all those extra cycles? What are we going to do with that power?

That becomes an interesting question. Fortunately, the software industry has been able to take advantage of whatever speed and memory we could give them. They taken more than we've given, in fact. I used to run Windows 3.1 on a 60 megahertz 486, and things worked pretty well. Now I have a 196 megahertz Pentium running Windows95, and a lot of things take longer than they used to on the slower machine. There's just that much more in software, I guess. But one application that I think we're not to far away from is good speech recognition. It's dangerous to predict that, because it's been the application that has been five years away for the last 25 years. But I think that within the 10-year timetable that we're talking about, it ought to be generally available.

PART IV THE FUTURE OF THE COMPUTER

From leading a closely knit team that put the first computer on a single chip 25 years ago to co-founding and leading Intel into a semiconductor powerhouse, Gordon E. Moore has been a dominant figure in the development of the modern computer.

Recently, Moore took some time to speak with Scientific American's west coast editor, W. Wayt Gibbs. In this final section of their interview, Moore gazes into his crystal ball. Here is what he sees coming over the next decade.

You've said that good computer speech recognition will be realized in the next decade. Let me press you a bit on that, because there are certainly skeptics who would argue that the problem with speech recognition is not so much lack of fast processors and memory but lack of understanding about how to put grammar into a computer and how to parse complex sentences.

There is a lot of algorithm stuff going on there, too. I was at Cambridge University for the hundredth anniversary of the electron earlier this year and looked at what they were doing. They gave me a newspaper article to read, and I read the article [to the computer], and the machine took about four times as long as it took me to read this, but it did a pretty good job of recognizing what I read.

This was a case where it certainly wasn't trained on someone with an American accent. So it was quite significantly speaker-independent. I didn't even read it anything initially to get it started. It uses a lot of context-related things in order to recognize speech. It calculates what phrase fits in this kind of a sentence.

So in that respect it was looking at grammar, language structure and everything. And I read it continuous speech, no isolated words or anything. I haven't played with one in quite awhile; I think a lot of them you can teach individual words, and you can train them. But this looked to me like a fairly significant advance since the last time I had looked at modern speech recognition system.

Do you think that computers that are able to take dictation, like this machine, by themselves will be a significant breakthrough, or that we'll have to wait until the next step, which is computers being able to understand speech?

Well, if the Cambridge approach is the one that happens, these may not be that far apart. That system is recognizing speech in the context of a complete thought. It's recognizing which phrase fits into a particular kind of a sentence structure. And it even selects two or three different choices in places it misses, so that you can go back in and pick out the one you wanted.

That approach clearly is the one that's going to have the impact. I think, though, that even a machine that would take really good dictation could have some fairly significant use. There are still a lot of people who would like to use computers, but who are intimidated by keyboards. Being able to talk to the machine would help them quite a bit.

That is the one incremental capability that I can see that I think would have a rather significant impact on the way people use computers and would open up the whole next step towards the day when one can carry on an intelligent conversation with a computer. That may not be in 10 years. But I'll bet that certainly within 50 years and probably within 20 you will be able to have a conversation with your computer.

It's an interesting example because there are a lot of linguists working on this problem who are not entirely certain how to encode common knowledge and the kind of things that you need to make sense out of simple language into a computer. So in many respects, it's as much a software problem as it is a hardware problem.

It is. It is more of a software problem. But having the very capable hardware there gives them a lot of opportunities to derive other ways. Playing chess is a software problem, but you still need a mighty powerful computer to do it in a reasonable amount of time.

The state of software engineering is not as mature as the state of what you do here at Intel, and it doesn't seem to be progressing quite as fast either. Do you think the difficulty of designing very large, very complicated software might be a factor that limits demand for very fast, advanced computers?

I suppose it is, but the industry seems to muddle along okay. I will admit to not understanding why software is fundamentally different than the kind of hardware we do. One of these processors is also the output of hundreds of very bright engineers, who are working together to see the whole entity at the end, and we develop techniques so we can predict when things are going to get finished. We still have errata, but nothing like software.

You rarely miss your ship date by a year, as Microsoft has in the past.

No, we don't. It used to be that every once in a while we'd pass a threshold where the old techniques didn't work anymore. The last time we did that with processors was the 386 generation. It just took forever to get that up to the point where it was a shippable product, iteration after iteration.

But then we went through an extensive effort improving our tools, and we have a very large ongoing investment in tools to try to keep them up to what we require for the processor we're working on today. So we've learned how to run projects like that, and we can predict pretty well when the things are going to come out. I don't see why software isn't potentially subject to the same kind of control.

Dan Hutcheson [president of VLSI Research] has mentioned in conversations that he and others in the industry worry about the design tools–the software that you use for designing these incredibly complex chips–and a scaling problem there. As Moore's Law continues its march, the simulation and modeling tools that you use to design and test the circuits are struggling to keep up. Might that create problems?

We're doing better now than we used to. Now we know how to work, and we're paying a lot of attention to it.

So for your purposes they are keeping up.

Yes. We all would like to have more, of course.

Here's another factor that might conceivably limit the impact of high-performance chips on computing as an industry: computers seem to be becoming more communications tools than calculation tools, so what about bandwidth as a limiting factor?

Bandwidth is a real problem in general–although not inside a company like Intel, where we've done a pretty good job connecting our computers. Processing power can substitute for bandwidth to a significant extent. For example, you can send tolerable video over ordinary phone lines so long as you have enough processing power to compress and decompress the images.

So in some respect they're complementary. But I look forward to the day when we all have gigabit pipes coming into our houses. There's obviously a lot of industry work going on to try to make that available. It looks to me like it's going to come from a variety of different directions. Some of it will be the cable industry supplying it, maybe some of this DSL [digital subscriber line] stuff is going to actually come to pass.

Yet it's pretty clear that since communications bandwidth is infrastructure-dependent and so costly, we're going to see it grow more slowly than processing power, right?

A lot of the basic stuff is already out there. The fiber backbone that exists can carry an awful lot of stuff. I used to think that it was principally a switching problem. I didn't realize how much that, now that it's all digital, you can do without any real switching.

I think it's being slowed down most by regulation. If they were driven by competition equivalent to that in our industry, things would go a lot quicker. Of course, such opinions are often the case of not understanding the other guy's problems.

The growth of processors and memory has over the years enabled a huge number of applications in industry and business that just could not have happened otherwise: controlling equipment and so on. Will further order-of-magnitude increases be applied to solve those kinds of business problems? Or are they probably going to get shunted into things that you suggested, like user interface and broadening the appeal and the use of computers, rather than solving problems that currently computers are too weak to attack?

I think it's going to move in a whole bunch of different directions. This industry is at the point now where some variety of specialization becomes increasingly likely.

You see that already. There are industries that are crying for more computing power–drug design, for example. They want to model how molecules fit together, which requires a lot of computing. If all you want is a word processor and a spreadsheet, you have more than enough power now. We're looking to see what kind of applications on a typical desktop in a business would benefit from higher performance. And we haven't identified really any very general ones yet. It's easier to identify home applications that require higher power.

Home applications such as what?

They typically tend to be the multimedia type of thing. Image processing and getting a good video is still something that's much more attractive to the home user, typically, than it is to the business user. Although that is not necessarily going to be the case forever.

But there is a point at which your video is the best you can get through whatever size pipe you have connected to the house, and you aren't going to get anymore because compression is limited by mathematical theory.

Okay, you will always lose something in compression, I guess. But there is still a long way to go before we get that far.

You mentioned specialization. By that do you mean specialization in hardware devices?

Yes. For a lot of business applications, one of the most important things is being able to control them centrally. There's the NetPC proposed by Larry Ellison–this tackles the issue of total cost of ownership. We have different views there. Here we think that the network is still the weak part, so you should put a lot of stuff out there on the terminal if the user wants it. Larry would like to have something of his on the server and a bunch of dumb terminals out there. And, at Intel, we obviously have somewhat parochial interests in this matter, too.

I think we will see some of each. But my personal view is that we will rely very heavily on smart clients, because we have all gotten used to owning our own resources, and I think we will want to hang on to that. There will be some specialized applications where you won't want to give the people on the end any control at all. But most of the cost of business computing in a big company is in the problems associated with controlling what gets in the system. And there is going to be a tremendous interest in some kind of control–being able to load software onto your computer and troubleshoot it from a distance.

Now, if something goes wrong on my computer, I have to make a phone call, and somebody has to come up here and fiddle with the keyboard. That's ridiculous. The right way to do that is to let them take control of my computer from some central location. That kind of network control is something we can do now. It's just a question of getting it in place throughout the organization.

Taking specialization out a bit more into the future, there are quite a number in the in-

dustry who predict that although PCs won't go away, they will probably be supplemented, perhaps by putting some tasks into put into much more specialized devices. We already each have probably three or four computers scattered around our desk in various forms. But there might be lots more specialized for particular tasks. This, they say, would make them easier to use because they would have specialized interfaces and also would make them more convenient and powerful because they would be more portable.

Possibly. The last thing I want is five different interfaces. Although I guess I have that if I have five different programs.

Right. Let me put that another way. You can imagine having, five years from now, an incredibly powerful machine that is the equivalent of a minisupercomputer today. Or buying a bunch of chips equivalent to today's Pentium or Pentium Pro chips but that are performing smaller, simple tasks with great intelligence. So Moore's Law could push everybody toward buying the latest, greatest high-end chip, but it could also, by making the chips we now have much cheaper, start a whole new market down there. What do you think about that possibility?

It's not the chip that determines the cost of the machine, its everything else that goes with it. If the chips were free you could only shave a few hundred bucks off the price of these things.

I think it is likely to go in several different directions. If someone can identify special-purpose things that really fill a need, that's fine. In some respects, the WebTV is that kind of device, when all you want is Internet access from your family room. It has a simple interface and is a relatively simple machine that lets you do one task. The general-purpose machine has tremendous advantages and the disadvantage of complexity. It will be interesting to see how some of these things play out. They're awfully hard to predict.

Complexity seems to have been rising in general purpose machines. Is the market for lower-power chips embedded in specialized devices a target for you?

Oh sure. It's a different kind of a market. We sell a lot of embedded control processors, mostly simple ones.

But those also get more complex with time, presumably.

You know, not much. Once they get in an application, you don't often need to increase the processing power. Even in auto engine control, you've got plenty of computing power in the 16-bit chips. You get very slow growth, but nothing like the PC market.

The PC is a terribly complex device. It needs a simpler interface.

Do you think we're on the right track for getting to simple interface?

I'm not sure. I feel the same way about my TV and hi-fi sets–all these damn remotes. I don't use them often enough to learn how, and I get so frustrated I could throw the thing through the set. My wife gets even more frustrated than I do. But I'm in a different generation; the kids who grew up with this stuff seem to find it much easier. You probably are a lot more relaxed about it than I am.

One other thing I wanted to get you to prognosticate on is parallel processing. Intel has built supercomputers that use fleets of processors. When we hit the wall at the end of Moore's Law, as eventually we will–whether it is a firm wall or a squishy one–could this be at least a stopgap solution until the next generation technology is ready?

Oh, I think it is an ultimate solution, actually. Whatever you can do with one, you can do a lot more with several. And a surprisingly large number of the real-world problems can be done on a parallel machine. We've dealt mainly with a research consortium down at CalTech, and they found that the class of things that can be done faster on a parallel machine is larger than the class of things that cannot. All physical modeling and the like splits up fine.

The problem is that those machines are so expensive. A lot of people would like to have them but spending millions of dollars for the state of the art for a year or two is not something many research organizations can do.

What about parallel processing on a much smaller scale–instead of massively parallel, just dual- or triple-processing?

That's no problem; servers are that way now. Workstations are or will be. We'll even see some of that on power users' PCs, I'm sure. Our new processors are rolling out the capability of doing that pretty easily. But it requires special software to take advantage of it.

Is that something Intel is working on?

Not the software; we let other people do that. Typically the UNIX systems can exploit dual processors, Windows NT does to some extent. The typical PC operating systems do not.

Is that because it is very hard to do so, or is this a chicken-and-egg problem?

It certainly is more difficult and complex, and the operating system gets bigger when you add that capability. But it is a way of getting more power out of the machines. We do a lot of other things to get more power–a typical engineering workstation here has 250 megabytes of DRAM, for example. And anything you can do to squeeze some more performance out is worth doing. In fact, our workstations all work together like a huge parallel computer on big problems at night. Then during the day they revert to single-user systems.

So, if over the next decade the generations between chip designs do stretch out, this would be one way to sell more processors, wouldn't it–put two or three in every machine?

Could be. Of course in that case the cost of computers would rise linearly with the number of chips. That is typically not the case when you replace one chip generation with a more powerful successor. They tend to come down the same price curve, but just offset in time.

Good point. Thanks for your time.

ROCK EDITIONS
Valparaíso - Göteborg