INSTITUTE OF ELECTRICAL & ELECTRONICS ENGINEERS

CENTENNIAL TECHNICAL CONVOCATION

October 8-9, 1984

Tuesday, October 9, 1984

Computer Evolution

Session II

Franklin Institute
Philadelphia, Pennsylvania

SecrePhone

RECORDING AND TRANSCRIPTION SERVICE

Administrative Headquarters
2874 Limekiln Pike, North Hills, PA 19038
(215) 885-3815

New York: (212) 425-7520

MR. J. L. EVERETT, III: We turn now from microelectronic technologies to computer evolution. Our speaker this morning is Dr. Ralph E. Gomory. He is Vice President and Director of Research for IBM and heads up his firm's research activities. But principally he's responsible for the research laboratories at Yorktown Heights, New York; San Jose, California; and Zurich, Switzerland. He's also a member of the Corporate Management Board.

Dr. Gomory graduated from Williams College, studied at Cambridge University and received his Ph.D. in mathematics from Princeton in 1954. He served in the Navy, was Higgins lecturer in math at Princeton before joining IBM in 1959 as a research mathematician at Yorktown Heights. In 1964 he was made an IBM Fellow, a rank conferred on a small number of scientists and engineers by IBM. In 1970 he was named Director of Research and was elected a Vice President of IBM in 1973. He was named member of the Corporate Management Board in March of 1983.

Dr. Gomory is a member of the National Academy of Sciences and the National Academy of Engineering.

He's a fellow of the American Academy of Arts and Sciences and of the Econometric Society. He's a trustee of Hampshire College; chairman of the Advisory Council of the Department of Mathematics, Princeton University; and a member of the Advisory Council, School of Engineering, Stanford University. He's a director of the Bank of New York and IBM World Trade Americas-Far East Corporation. He holds an honorary Doctor of Science degree from Williams College and was awarded the Lancaster Prize of the Operations Research Society in 1984.

It is my pleasure to present to you Dr. Ralph E. Gomory.

(Applause)

DR. RALPH E. GOMORY: It certainly is a pleasure to be in the presence of so many less than perfect devices.

(Laughter)

As a complex biological system, I want to tell you I belong to a school of people very pessimistic about predicting the future. I think things vary a great deal in their predictability. For example, the motion of the planets is notoriously predictable. But if you pick up a stone standing on a hill and throw it down the hillside, you don't know where it's going to end up -- too complicated, too bumpy, too rough, too dependent on detail.

You know when it comes to predicting the course and impact, especially the impact, of future technologies, I think it's the second type. I don't know how to do it and I doubt that even biological systems like those represented here today know how to do it. So I will confine myself, except for a few moments perhaps of exuberance when I get carried away, to talking not so much about prediction but about some of the forces which are currently at work and the outcome of those forces acting in our society I will leave to your imaginations with full confidence that you will reach a variety of different conclusions.

Now one of those forces that's at work and that I want to talk about is the development of the technology of smallness. Now I'm just using unfamiliar words to describe something whose present day manifestations is, first of all, the semiconductor industry. But I think one can say that the reason why computers make so much progress in their interior workings -- and I will distinguish in this talk between their interior workings and the problems of getting things in and out of them -- is simply due to the fact that they are only bit handlers. All they do is make and compare or occasionally add bits. As long as you can read it and write it, a small bit is as good as a big bit; and if you can make it out of less silicon, it is cheaper and usually faster. So there is a motivation to make things small.

In response to that, you can see the evolution of ever better lithographic devices, ever cleaner areas, and the increasing problem of keeping those areas free of tiny particles which have the capability of disrupting the functioning of very small devices, perhaps soon the exclusion of people who seem to kind of emanate small particles all the time from these areas. So we are creating this technology of smallness in the area of semiconductors.

I guess I should show my ritual slide here. Just so you understand we're talking about the old familiar things, this is supposed to be, when it appears, a 64K static RAM. But since this is kind of a ritual slide, maybe it won't matter if it doesn't appear. There

.it goes.

But I think the thing I want to convey to you is that the technology of smallness is more than the familiar semi....could we have some lights, sort of the half-way arrangement. My seeing you is more important than your seeing the slides if we have to make that trade-off.

(Laughter)

A second aspect of making things small that is very, very important for computers, though perhaps less familiar, is the problem of fabricating the interconnections or packaging. If chips are to talk to each other -- and I think that will be necessary for quite a while -- they have to somehow transmit their signals over wires, and those wires need to be very small. We'll try to get another slide up here.

So the problem of chip interconnecting really comes the problem of laying the interconnecting wires down between the chips, getting them up and down to the chips, and somehow building an underlying structure, the package, in this case the first level package, the one that comes first, under the chips that's capable of supporting those wires. Now this in itself is an enormous task. Today for high speed computing those structures, the first level modules, are themselves let us say 30 layers of ceramic with wiring in between, all those layers being needed to allow the degree of wiring required to interconnect the chips.

It's necessary to have very, very smooth surfaces. And this notion of smoothness I think is very closely connected with that of smallness. If you have a rather coarse wire, you can have a rather bumpy ceramic surface and you can lay the wire down on it and the bumps will make the edges look a little ragged but it won't part the wire. On the other hand, if you want to continue making progress in miniaturization, you must make concurrent progress in the smoothness of that surface. And this opens up the question of how do you make, let us say if we stick to ceramics for a moment, super-smooth ceramic, which opens up the question of how do you make the little particles that make up the ceramic of a uniform size and how do you make them fall together in sort of crystaline structures or something like that that end up being very, very smooth.

Another aspect of miniaturization which is not strictly semiconductor deals with the making of disks. Disks are perhaps not the most glamorous element of computers but, as Andy van Dam was reminding me last evening, you spend most of your time in practice waiting for the information to come off the disk. And disks today, of course, are an enormous industry.

And again, the need for smallness to make progress by making bits small dominates the technical problems. What you're looking at in this slide is the head on a disk. Disks, if you're not very familiar with them, are basically like phonographs. They have an arm which swings out to some position over the disk and then tries to sense the magnitized areas of the disk as they whip by underneath it. They pick up that signal.

Now the way that you make progress in disks is you make the bits in the surface (the magnitized areas) smaller and smaller which unfortunately means that the head must fly closer and closer in order to pick up their signals since we do not have any way of beaming magnetic fields. And the coil structure in the head has to get smaller and smaller. So what you're looking at here is a coil structure in a head which is made by semiconductor processing techniques. So miniaturization processes reach out again beyond the area of semiconductors.

To understand how demanding the smoothness and smallness issues are in this area of computing, I might add that today's heads are passing over the surface, or as you say flying because they are actually supported by air, at a height that is a fraction of a wave length of light. We use microinches so you're talking about flying heights of 7 to 12 microinches, something like that. Now microinches are, you know, about 40 times smaller than micrometers or microns, so curiously enough you are dealing here with dimensions smaller than those typical of semiconductors.

To get some feeling for this, we always use this analogy which is that if you were to blow the head up so that it was the size of a 747 aircraft, you would then envisage this thing flying over the surface (the surface being the disk surface) at a height of a fraction of an inch. You can see -- and at full speed of course -- that it is very important not to have any bumps.

(Laughter)

Again, we encounter the need for smoothness if you want to have smallness. And smallness, of course, is what you want when you're dealing with bits.

Now one way of ease these technical problems of disks which are such an essential part of computers would be if the 747 didn't have to fly that close to the surface, if it could just rise up a yard or two. With magnetism you don't see how to do that. But if you substitute for magnetism a laser beam so that the 747 can shoot little holes in the surface or cause something to happen on the surface, you relax that difficulty. Of course, you instead create other difficulties which are the technical difficulties which are associated with optical storage in its various forms, of which one, for example, is, well, if I'm going to write directly into the surface with the laser beam, first of all, I want to write directly and make changes in very small areas because that will be the bit size which is essentially the determinant of its economic viability, and then I have to bring about some sort of a change in the material, preferably reversible so that it isn't a write-once phenomenon. A candidate phenomenon is phase change, that is that you'll crystalize the material. But then in order to erase it and rewrite it, you have to uncrystalize it. So you run into a host of material problems having to do with the creation and subsequent abolition and creation and abolition perhaps of a billion times of very small crystalline areas.

Thus, what we're seeing is that the desire for information processing is creating a whole host of scientific and engineering challenges having to do with smallness, smoothness, problems having to do with the fine structure of materials. I won't recite the litany of semiconductor problems per se because I think they are fairly familiar. And even before all these things run out of gas, which I think they will, they will have created a vast array of tools, fabrication ability, materials understanding, which will probably not be limited in its applicability once created to information processing.

For example, one interesting development which is perhaps indicative is the use microfabricated structures for the separation of uranium isotopes. This may sound a little different. It is. But basically once you know how to make very tiny structures, it becomes possible basically to make very tiny metal nozzles -- very tiny meaning that the radius of curvature of one of these things is perhaps 3 microns. And then if you shoot gas around this curve, which is a very tight curve by normal standards, the separation effect for the isotopes is enormously intense. And indeed you can build a whole cascaded sequence of these since they're so tiny. All together, they have the advantages then of being a cascaded sequence and of having very intense separation power.

So microfabrication techniques, once created, may extend in many, many ways. Some of the other techniques also have their own odd possibilities.

I talked about using laser to produce a phase change in material. Some of the processing desires in packaging encourage you to use lasers to promote, say, electrodeposition at a particular spot. Now electroplating in the presence of a laser beam is different from electroplating without out. Therefore, you can create, so to speak, spots of metal by the use of a laser beam.

. So again this notion of doing things very, very locally and understanding in some detail what's happening there, in fact this notion of certain kinds of lasers used in certain ways is a natural for eye surgery and in fact is being explored in that realm.

So I think one of the things that we are going to see is a continual pressure to learn how to microfabricate driven by the information processing -- a small bit is better than a big bit notion -- but having once created a sort of life of its own, as if you had created a steel industry or something of that sort.

Within the computer, miniaturization can be regarded as a dominant means of progress. However, once you have to deal with human beings at the interface.... all that computing goes on inside but eventually you have to do something with it. You have to make it visible on the screen or you have to print with it. You have to somehow deal with the human scale and show things that are visible to the human eye. At this point, it's harder. That is there's no simple high road of progress like miniaturization once you reach the human scale.

However, sort of along the lines of what was said last night, if something is desired intensely enough, it tends to happen. Because there's enormous progress in information processing within the machine, enormous demand is created for progress in printers and displays, not to mention more unconventional schemes for input and output.

In a variety of ways, I think we can confidently predict in the immediate future the improvement of displays. So certainly I think in ten years we will all have high resolution displays which I will blandly describe as 5 to 10 million dot displays. They may be flat, if flatness matters for other than portability. For instance, you may want it to lie down rather than stand up stiffly in front of you. You may want to poke at it, so you'll want it to sense so that you will be able to poke at the screen with its high resolution and give it a few bits of information. Color should also be a part of this picture because it's a way of conveying information.

There's enormous challenge, which is all that information being carried around inside this thing, somehow you have to get it across to the person. And that

.gateway, which is what the display is or the printer, is very worthwhile to improve. The technical capabilities for doing that are really there.

The same forces drive printing. You do see a proliferation of new printing technologies. Certainly impact printing will continue. Electrophotography is a tremendous and major printing technology. But we also see, probably mainly driven by new demands on printing, the emergence of a whole host of other technologies such as ink jet, such as various forms of thermal. The new demands being, gee, I don't care if this thing goes 30,000 lines a minute, which in fact some printers do, what I do care is that it goes fast enough for one user, I do care that it's quality printing because I have to read it, not some anonymous person reading it for me, and I'd like it to be quiet because it's sitting right next to me. These are the demands that spawned, for example, ink jet. And again, because you may want to convey a lot of things, both beautiful font and high resolution for pictures, you need an all points addressible rather than a character printer.

There's progress then in these areas sort of as a second, as a derived progress in my opinion, from the smallness. The demand is created by smallness and so on.into the interface.

Now let's assume that all this progress takes place. Ten years from now with these forces acting, I think we can safely predict that your ordinary mass produced work station or personal computer (the equivalent of today's mass produced, I'm not talking about advanced) will be a powerful engine, will have, let us say, a single chip microprocessor in the 10 to 20 mip range, which means as large as today's largest ordinary machines. You will store 16 megabites of memory. You will have a beautiful CRT or other display of high resolution. And you will have a high quality printer, probably removable optical storage of two gigabites and half a gigabite probably of magnetic recording technology.

What I mean by a beautiful printer, which we don't take for granted today but we will within ten years, is this kind of thing. And this is the quality of printing that can be done today with....this happens to have been created by an ink jet. It could be created by other means. The challenge is only to get this thing economically viable. I'm sure we'll manage that.

In addition to these microprocessors -- and I've simply described to you what's on your desk driven by one micro -- there will be large systems, 100 or 150 mip systems employing all the rest of this technology that I've described, that is to say chips with a powerful interconnect system, and this all jammed into at most a few cubic inches. This is where you start to see that there are difficulties, many difficulties. Aside from the limits already described at the chip level, the business of jamming all this stuff into a few cubic inches, which you must do because of speed of light considerations -- you can't afford to wait around while the signal travels from chip to chip -- will be a limiting factor on the speed of very large multichip machines. So I think that in the range of, say, 150 mips or give or take some factors, that will be difficult to make progress beyond that. Difficult -- I use the word difficult because these are engineering barriers but very difficult ones, to get all that power in and to get it out.

Will that power suffice for the future? Carver Mead has already given you examples of things that call for far, far more than that. I think there are tons. And I'm going to give you a list of driving forces which are interesting to contemplate in themselves because they will be forces driving computers, but in particular in most cases call for the creation either of special purpose machines or highly parallel machines. And I will use the term highly parallel to cover them all because machines can be parallel in many, many ways. Some special purpose machines are merely parallel machines that are parallel at a very low level. So you can hear new architectures if you want to when I say highly parallel or special purpose.

Here are some of the things creating that need.

Number one, believe it not, conventional applications. All those machines that do data processing. The demand for the large machines, the mips, is growing faster than our ability to make uniprocessor (single processor) machines for conventional applications. So already the commercial machine is slowly evolving into a multiheaded machine all by itself.

Secondly, the desire to have a high degree of reliability or availability which is most easily obtained by duplication in the processor and in the storage elements, again, drives you towards multiheaded machines.

Thirdly, the promise of wholly new and exciting applications, whether these be very large scientific calculations which people are already doing and are building special purpose machines. One of the machines that we are building in one of our research labs is a special machine to do quantum chromodynamic calculations, or more precisely do one.

Along the lines of what Carver said, it's relatively easy to build a special purpose machine, and it tends to be successful. If you name the problem, I'll give you the machine. And it will go like a bat out of hell. Not to be confused with the problem of devising a "general purpose parallel machine or multipurpose parallel machine," these being much more difficult and much more speculative endeavors.

So the possibility of massive scientific, massive engineering calculations, for example, calculations of design automation for which we have also built special machines to simulate the logic of computers. The design cost was something which Bob Noyce mentioned. And one of the ways you tackle that in turn is by building a special purpose machine for that very demanding purpose. And again, if you have one special purpose, you can build a machine.

Another challenging area is artificial intelligence about which I'm sure that you've heard a great deal. I'm not doing this in the right order, but that's all right -- we'll get to it. Here we go. For example, this is a very simple example of what's called an expert system.

Supposing you have a data base of facts about airlines -- you know, what flights go where and that sort of thing. You may want to ask it a variety of questions, preferably with minimal programming. One of the things that you can store is a number of facts about the system which are other than data, which are simply rules about the way it works.

For example, if a flight goes to the destination on the day that you want it to, then it is a possible flight for your purposes -- a very trivial remark. Or if the flight is possible and if it offers a special reduced fare and a few other conditions, then it's one you might be interested in. You can write down a string of rules like this. Then if you ask a question like what flights go to Paris on the day I want to and give me an APEX fare, you can start going through the rules. That is, first of all, you take the first rule and see what flights listed in the data base are leaving on the desired day and have Paris among their list of destinations. Then you can take that restricted list of flights and test them against your other conditions.

This kind of processing is a very trivial example of what's called rule based system. I just want to point out a couple of things about them.

It opens up the possibility of asking many different questions about a set of data provided that you have built in the proper set of rules.

Second, it's very computation demanding. This trivial example doesn't show that, but you will normally find that they end up doing an awfully lot of relatively blind search. In other words, in some sense this is another extension of the way we make progress in programming which is we do less and less art and use more and more power. However, given the progress of machines, that is fundamentally a right way to go.

We have an expert system in one of our labs that parses English and it goes at 20 million instructions a sentence. So it's a very demanding user of machines.

This is the same thing over again only written in prologue which is one of the languages used for this kind of thing. You can as a first approximation think of the fifth generation effort in Japan as the development of parallel and special purpose machines to do this kind of work. There are a lot of other things on it like natural language and so forth, but I would say this is the technical core of that effort.

Now we'll go back to speech. It's remarkable that we have gotten into a loop.

(Laughter)

Let me just say then that another tremendous user and driver for mips and for mips that probably cannot be supplied by the ordinary machine is the user interface. A couple of things.

One is the sort of unconventional IO, for example, speech recognition. And you've already seen the speech recognition slide on our way to other things.

(Laughter)

A tremendous mip burner. The best estimates that I can make on the subject or to do any kind of reasonable continuous speech is in the many hundreds of mips. Those estimates are very suspect because to do continuous speech is an ill-defined term. With what error rate over what vocabulary in perhaps a limited range of discourse, or you name it. The main message though is that kind of thing, trying to recognize speech, another enormous mip demander. Similarly, the understanding of handwritten input which is also I believe feasible. I regard both of these things as feasible. The understanding of natural language, another enormous mip burner.

And something which I only want to mention very briefly is the direct manipulation of objects in place of programming. I'm going to put it that way although it's a somewhat exaggerated thing. Programmers really used to deal with a conceptual structure which was the machine. Memory. They talk about memory locations. When you write instructions, you put it in a memory location, sort of a mental picture. I'm dealing with something that's got memory locations and registers, blah, blah, blah. There's been some simpliciation of the mental model. The memory, let us say, has become infinite, the registers have become invisible. You just imagine you're putting things in the memory registers. There are new departures, new models that you manipulate instead of the machine which the machine holds up to you and you work with.

For example, the desk top image in which you do things by pushing around on a desk top things that look relatively real. Like you put a piece of paper in a wastebasket. Actually what you do is a mixture of symbolic processing and moving images. But I think that's a significant input/output thing that will be very demanding again in mips because there's so much to be done to take that enormous power and translate it into usable form. That transformation, whether it be through speech recognition or maneuvering objects, is another tremendous demander of mips.

All of this then leads to the study of unconvential architectures and especially parallel because it's basically out of parallelism, one level to another, that you get the increase in mips for your special purpose. It's nontrivial to try and develop the proper parallel machines.

One way, I think, is fairly straightforward and Carver has given us examples. There are others. You've got the problem, I give you the machine. But the notion of a general purpose parallel machine is a much more elusive and difficult things. So there will be tremendous needs and pressures to create an understanding of parallel algorithms far more than ever needed before, to create software to run these parallel machines.

The interconnecting in the case of discrete or multiple discrete processors will emerge as a discipline of its own. It's not simply a question of hanging processors on a bus. The interconnect structure will probably be critical and will have a great deal of structure itself. It will be more than a simple switch in all probably. The need for mips then will not be satisfied by the conventional machine. Those pressures will create new architectures, a whole new family of progress is needed.

In this direction, some of the most demanding users of this new power will be software itself. The creation of software is one of the most complex things that people do at all. It is a somewhat maligned subject because of its proximity to hardware. Hardware makes progress at an extraordinary rate because of the smallness issue. We can make progress through making things small. Most technologies do not have such a magic formula. Automobiles don't and almost anything else you think of. Software is in the ordinary category. Its misfortune is that it's sitting next to something very unusual. So I think we should stop demanding of software that it make the extraordinary progress of its neighbor. It is ordinary. It's its neighbor that is extraordinary. Software benefits, of course, from that hardward progress and it will continue to. It will continue to develop as an engineering discipline of its own. All these better interfaces, all this power will benefit it and it will develop its own engineering procedures. So the software generation procedures will become both more disciplined and more perfect. Andy will be talking more about those things.

Finally, I cannot forebear saying a few words about robots. Those of us who were brought up on H. G. Wells remember the clanking mechanical monsters that did such wonderful things or such evil things, depending on what story it was. As a child I wondered why it was we couldn't build those when I could see all around us equally complex machines. And after a lot of thought, I reached the conclusion that something was lacking. I mean if I imagined myself building it, there was no way to direct it.

When I became the Director of Research for IBM in 1970, I found a very strange situation, which is I had labs full of people who were working on intelligence but no mechanical motion. So we started a robot, and today we have a small robot business.

But I do remind this audience that the essential ingredients are there. The ability to power mechanical motion has been there since the industrial revolution. The ability to do a great deal of thought-like work is here today. It's relatively straightforward -- everything is straightforward in the perspective of 100 years -- to equip these creatures with sensors and with vision. So robots and mechanized production are definitely coming.

Let me summarize.

One of the forces at work, I think, is the creation of a technology of miniaturization which will continue to enable us to make progress for some time, but once created will have its own consequences. We will, by the extrapolation of the doctrine of smallness, have machines of enormous power and by building on the interfaces of great ease of use. Nevertheless, as we approach the limit of these machines, parallelism and special machines in all their forms will become necessary and challenge us in many ways -- scientifically, algorithmically, and all the rest. Robots are a real possibility because the ingredients, physical and of thought, are present. This combination of forces I think is hard to imagine doing anything but having a most profound effect on the world.

Thank you very much.

(Applause)

MR. EVERETT: To discuss this most interesting paper, we have Dr. Andries van Dam who is Chairman of the Department of Computer Science at Brown University where he's been teaching since 1965. His research has concerned software in general and computer graphics, text processing, mini and microcomputers, microprogramming, and most recently personal work stations. He has been working for over 15 years on the design of computer books based on high resolution graphic displays both for teaching and for research.

Dr. van Dam received a B.S. degree from Swarthmore College in 1960, M.S. and Ph.D. from the University of Pennsylvania in '63 and '66. He's a member of Sigma Xi, IEEE Computer Society, ACM, and the Computer Science Board. He helped to found and is an editor of Computer Graphics and Image Processing for the past decade, and since 1982 has been an editor of ACM's Transactions on Graphics. He was also the co-founder of ACM SIGGRAPH in 1967. The book, "Fundamentals of Interactive Computer Graphics," co-authored with J. D. Foley was published in 1982. He has co-authored and authored some 60 papers on contributions to the technology.

He is on the Electronics Systems Board of Gould Incorporated, on the Scientific Advisory Boards of Metagraphics, Intelligence Systems, and Cadre, and is a consultant to such organizations as Exxon Research and Engineering, and IBM.

It's a pleasure to present Dr. van Dam.

(Applause)

DR. ANDRIES VAN DAM: Thank you very much. Good morning, ladies and gentlemen. It's a really a privilege to be in the company of so many luminaries but it's also a bit frightening and it puts me in mind of a story of this engineer who had been practicing due diligence for all his lifetime and finally went to his just rewards and was met, as is the custom, by St. Peter at the gates of heaven. St. Peter introduced him to the general system and said, "We have this little ceremony at night to introduce the new members of this organization. We have you tell a little bit about yourself and sort of what the most important event in your life was."

The engineer thought about this for a moment and decided his engineering career certainly wasn't relevant here. He said, "Well, I am a survivor of the Youngstown flood, so I guess I'll talk about that."

St. Peter thought about that for a moment and said, "Well, it sounds all right to me but remember Noah will be in the audience."

(Laughter)

Now you know how I feel.

I want to recap for just a few minutes and then talk about my favorite subject, computer graphics, talk a bit about software, a very small about AI, and finish up with my business which is education.

As far as our baseline is concerned, we have hardware magic and things are only going to get better. For the first time ever, we can say that to a first approximation, there is enough computing power to go around. And we're talking here about embedded computation in the home, in offices, in factory automation, in laboratories. We're using computers for computer aided design, manufacturing, and so on; and you'll hear more about that today.

In many ways personal computers in particular have become a commodity item, much like calculators. We buy them, they're obsolete in six months to a year, we throw them away, and we buy something new. Yes, they will be as pervasive as the telephone.

May I have the first slide.

I just want to show that Bob Lucky's request for the Dick Tracey style of wrist TV and computer has to a first approximation again been met. Not this -- this is ENIAC (Phonetic). How can you come to a celebration here without showing ENIAC? So the power of ENIAC but in a much smaller package. Here you see on the wrist (I happen to have it here) something that contains a Z80 microprocessor, certainly more powerful than ENIAC and indeed a 360-50 that we helped install at my university in the mid-60s. It has 8K of read only memory and 2K of random access memory. And that 2K can be downline loaded with programs and data through that little cord that you see coming out of the side which, for those of you who know about these things, is an RS232 port which you can attach to your personal computer. It has a touch sensitive screen. It's all points addressible, can show text and graphics, and you can key in. It has a little tiny editor built in so you can update your appointments and your calendar and things like that. That's only the beginning. I really don't think that the wristwatch TV is that far away.

In the profession, of course, we are seeing the increasing dominance of computer technology. In Spectrum Beyond '84, the June issue of Spectrum, there is a subtitle entitled "EE Specialties, Computers Are Pervasive." And in the 1984 January Technology issue, of all the articles which talked about all the various subdisciplines in EE, I believe there was only one in which the words computer or microprocessor or displays did not appear. So it's everywhere.

Now one of the most exciting things, switching to my second comment, is that we're moving away from a preoccupation with raw power to what I think of as refined power -- power really in the service of the end user. So we now hear terms like user interface and user friendliness. What we're really seeing is a liberation from text and the proper introduction of pictures. We see two dimensionally and we have a long history.... that's the Spectrum issue. I've got to get my slides in sync here.

We have a long history of seeing things represented two dimensionally, integrated text and graphics, the introduction of pointillism which of course is mirrored electronically. That's how we produce color on either screens or on the printer output, the magnificent output that we just saw. When can I have one of those, by the way? That's really an amazing piece of progress. And from relatively simple and crude drawings like this, we go to what we can see today on a utility machine like the Macintosh being popularized by Apple and lots of other manufacturers will be producing these in quantity as well. What's interesting about them is that they're starting to do away with all that ugly typing and dependency on textural specification of commands and data.

(Laughter)

And going back to hieroglyphics because they're self-disclosing.

So the icons that we just heard about where we delete folders by moving their icon into the wastebasket, the icon of the calculator, the clock, and of course integrated text and graphics. Notice the Greek font on the left hand side. Through the magic of all points addressible displays, we can create anything we can think about, including more complicated pictures not yet available on personal computers. This took about four hours of VAC 780 time and about the same amount for that. Here is a close-up. Now these systems were produced first with solid modeling creation systems -- that's how we create the world -- and then an image synthesis pipeline which takes pictures of that just produced world.

That's a picture made by the Lucas film folks which they worked on for months and months and which took unbelievable numbers of hours of computer time. So I just show that picture as a challenge to all the hardware people saying, we can use up any amount of power that you'll be creating for the remainder of this decade and beyond -- no problem. We want effectively a CRA MPX per pixal (Phonetic)....

(Laughter)

....in order to be able to create images of this complexity.

Well, it isn't the image creation that really concerns us at this point because here we're just talking about the need for raw computation. What we're really concerned about is that we don't understand yet very well how we create models of objects that we want to take pictures of. As I said, a picture like this and even the simpler pictures that I showed before are the products of many, many operations and months of work. What we really still need is a fundamental understanding of object modeling and object construction.

In the words of Allen Kay, we don't just want pictures that look good; we want pictures that are good because they are based on what I might call a deep structure, borrowing a word from AI and knowledge engineering, a real understanding of how we mathematically model fundamental components and what sort of operations we use to put them together being studied in solids modeling systems, which are still very much in their infancy. We need a lot more work on those.

We also need a lot of work on animation and things like body movement and the modeling of nature, both objects that occur on factory floors and objects such as this real nature.

Now pictures aren't a be-all and end-all either. We have lots of other multimedia possibilities. Sound is starting to come into its own. Segmented speech recognition is beginning to be here commercially. Sound feedback certainly is very important. We have some nice laboratory demonstrations of force feedback, and we're slowly but surely working on what you might think of as a total sensory environment.

Some of the things are a little far away. Eye tracking, for example, although it's been done in a lab, still really isn't a commercial reality. Brain wave monitoring, some of that is being done for helping paraplegics read and specify commands, but again it's still a long way from being a commercial reality.

These are the kinds of things that we will be working on during the next decade and into the next century. So we definitely do want to be multimedia -- text and graphics and sound and natural language in a variety of ways, pictures out and pictures in even. Naturally you'd like to be able to manipulate three dimensional objects with your hand somehow, and people are working on touchy-feely gadgets to let you do that.

But that is still a long ways away.

By the way, once we have something that approaches an understanding of the deep structure of objects, we can then begin in earnest on a very interesting project which we call automated authoring at Brown, and we have some very early prototypes of something that combines AI expert system technology with very fancy computer graphics image synthesis. For certain kinds of standardized pages, certain kinds of standardized phenomenon, you'd like to be able to work from a problem description to a set of pictures accompanied by appropriate text and sounds and other media which explain something that needs to be explained to a student, to a maintenance and repair technician, to a physician, whomever, whatever. And I think there is a very major industry in learning to create those kinds of multimedia presentation with the aid of computer knowledge.

Let me switch to the next subject now which is software, software factories. Software is the gating function. Hardware is here. Software is still coming. Software in many ways is the Achilles heel of the computer industry. It costs too much. We have unreliability as an endemic problems. Things are not extensible. Things are not maintainable. These are well documented problems, and I won't dwell on them.

In a sense, we're coming out of what you might call the age of naivete, an age that Allen Kay again characterized so nicely as, "People's ideas were that there was a software fairy who simple sprinkled some systems dust on hardware and magically the system as a whole would work. We now know that tremendous labor and tremendous investment is required, and we're going back to the old 80/20 or 90/10 rule. Ninety percent of the costs are beginning to be in the software rather than in the hardware."

What we are seeing is much more sensitivity to the needs for integrating hardware and software. And indeed, some of our engineers are finally beginning to get integrated education in both those very important and intersecting skills and fields.

Speaking about skills, because of software and its slow evolution, we are finally beginning to see some skill displacement. Cobalt programmers are fortunately becoming a glut on the market, and there is now a tremendous undersupply of people who, for example, know unex (Phonetic) and other modern operating systems.

This particular issue I wanted to show you a slide of because it deals with the previous undercapitalization of software people and the current recognition that what we need is a capital intensive software technology. Peter Wagner describes in a set of articles here tools and methodologies that finally will begin to have some impact. We're moving away from the shoemaker's children syndrome whereby to have a terminal on your desk five years ago was considered somehow not necessary whereas hardware engineers were working with 100 to 150 thousand dollars worth of capitalization for their hardware CAD work stations. That's changing. Because of the cheap availability of high powered work stations, software people are now getting them on their desks and we're developing software environments which are very rich tools for producing good software.

Peter talks about capital intensive software technology in terms of modules and how we build them and how we reuse them and the notion of abstraction. We're learning to get away, as Ralph said, from preoccupation with memories and registers and low level details and are talking in terms of the high level abstractions and metaphors that suit our particular problem. That again is highlighted in that very nice "Beyond '84" Spectrum issue that I recommend to all of you.

The second article is about programming in the large where Peter talks about life cycle paradigms and application generators. Now application generators and fifth generation application languages and things of that sort really spell one bottom line, and that is that we're taking software development out of the hands of typically unschooled application programmers and turning it over to end users who fill in forms, who program by example, as in query by example and office by example developed at IBM, and basically say what they want to do and then have the program figure out for them how to do it rather than having to specify or have specified for them in minute detail all of the basic operations.

The third thing that Peter talks about is knowledge engineering and that, as far as I'm concerned, is the real growth industry and the real new frontier at point. So that brings me into a few tiny layman's observations on artificial intelligence. All the experts are here so I want to stay away from this topic.

(Laughter)

Mostly my feeling as layman is that there is a bit of hype at this point.

(Laughter)

And the field as a whole is a bit oversold. We are, of course, in this field, as in all others, going to move with a series of plateaus. We make rapid progress and things stay the same for a while, another burst of creativity and energy and a lot of new things happen. We need to spend a lot of our resource on the problems of artificial intelligence because there is tremendous mileage there. That's where we're finally going to realize the potential of computers as intelligence amplifiers and not merely the takers away of drudge labor for routine data processing calculation and plotting and things like that. That we're doing fine today. What we're really not yet doing anywhere near well enough is providing what we all want -- the X assistant, the programmer's assistant, the physician's assistant, the lawyer's assistant, and so on. And these expert system programs are now being developed. It's very difficult. The easy things are easy to do. The hard things are always very hard to do.

We're talking about often taking knowledge that is implicit -- decision making skills that we as imperfect biological devices do so well -- and trying to put them down on paper so that we can put them in the machines, making explicit and tangible that which is implicit and intangible. And that is a really tough intellectual task. It takes many years, and we get systems of hundreds of thousands of rules probably in the future for a realistic system.

Speech recognition, natural language understanding, vision -- all beset by problems of ambiguity and context and deep knowledge that is required in order to resolve them. If you want to be just a little humble, for example, about how difficult the vision problem is, go to the Franklin Institute exhibition on visual and optical illusions and see how good the eye is at seeing things that aren't there. And then figure out how we are going to make computers see things that aren't there but that probably ought to be there. Very, very difficult problems.

And finally, robotics. We're going to hear more about that this afternoon. Well, my challenge there is build me a robotics system that can tie that silly bowtie when I'm trying to get into my tuxedo. When you guys have figured out how to do that, then you're really going to be able to talk about good robotics. The problem of picking up parts on the conveyor belt is a lot easier than that one of recognition and manipulation.

Raw power. Computing power is going to help a lot with many of these problems, as Ralph mentioned just a little bit ago.

Let me wrap up by talking about education. I think the computer has a tremendous capability as the tool par excellence to help us with modern education. There is no royal road to mathematics or to anything else, but the sugar coating that Bob Lucky asked for yesterday can be used effectively to help motivate, to help us visualize complex and abstract phenomena and to customize education to our particular needs, our skill levels, the way in which we like to get information, the media we like to use.

We have to learn about algorithms. We have to teach people about algorithms in the small and about systems and managing complexity in the large. We have to get sensitive to the proper teaching of dynamics, and I claim computer graphics again is the tool with which to illustrate dynamic phenomena. Slides and blackboard presentations and viewgraphs don't make it. The dynamics of process is what we really need to understand. So let me show you a couple of examples, chauvinistically chosen of course from Brown University, in experiments in electronic books in the classroom.

This was a conceptualization.Bob Sedgwick came back with from his stint at Xerox Palo Alto Research Center, and this is what we actually built. There are 60 1.5 megabite half-mip machines in that classroom. That's more computing power than I could dream of ten years ago on the campus as a whole, and it's all here in one classroom networked together so that I can broadcast from my instructor station at the front and have all the students watch and interact with the demonstration during the course of classroom teaching.

So for example, I teach binary tree manipulations. On the left we have programs overlaying each other as they're called, on the right various graphical representations. They're synced together so that as we single step through the program at a speed controlled by individual students, we walk through the graphic representation as well. Students are asked questions, and they interact with the material during the course of the classroom.

We've used this technique successfully in differential equations, differential geometry, introductory modules, neuroscience, in art color theory; and we're trying to branch out to a variety of other disciplines on campus.

(Laughter)

We think it works very well with a few exceptions. We're very excited about the potential of showing dynamic phenomena and interacting with them during the course of the classroom and are looking at a number of extrapolations and fields in which computers have not been traditionally used.

Let me stop there by saying that I really believe one of the key phrases used by an ACM conference many years ago which has stuck in my mind. The past is prologue, and we ain't seen anything yet. I'm glad to be alive in this century to see the monumental progress we're all making.

Thank you.

(Applause)

MR. EVERETT: In this world of artificial intelligence, there is a way you can distinguish between a human being and a computer. The computer stays within its time limits.

(Laughter)

We're running a bit late. But nevertheless, we'll take a couple of questions and I'll ask you to keep the coffee break down to ten minutes instead of five minutes.

Would the two speakers please come forward and sit at the table. Who would like to ask the first question on computers for the next hundred years? Bill.

MR. C. WILLIAM HARGENS: Dr. Gomory, you went rather quickly over that lovely colored still life. At least two of us are interested in it. How long did it take to make that, and say a little bit about how it's done.

DR. GOMORY: It's a color ink jet printer. That is to say the printing technique is firing little drops of ink out of a nozzle. Where you want the ink to land, you fire and then you have a pause while you move the head and then you fire again. So the drops are fired through the air. This calls for a certain degree of control. The way you get color is multinozzles. And it was a multinozzle color ink jet printer.

MR. HARGENS: How fast?

DR. GOMORY: I don't think I can quote you that. Nor would it be meaningful since it's a lab robot basically, a lab printing robot.

MR. HARGENS: An hour?

DR. GOMORY: Oh, no, nothing like that. It's a regular printer speed kind of thing -- pages per minute.

MR. EVERETT: Yes, sir.

MR. ELLIS RUBENSTEIN: Joseph Weisenbaum has recently observed that one problem that engineers all know about but don't like to address is the fact that nobody understands the programs well enough to be sure that they will work even under the most controlled situations like the shuttle launches. And he says that our society is getting so tied up with our programs that we are looking at a tremendous disaster possible in the future. I wonder if anybody would like to address whether engineers can seriously do anything about this kind of a problem or should, and if there's any way to address that.

DR. GOROMY: I seem to be given this one. Well, I don't know. I can't comment really on the quote since I haven't read it.

But as far as the content, I would just say this, well, programs are error prone, people are error prone, complex systems are error prone. But that's not news. What we've always done is not done systems that must function right and are beyond our capability to make function right. In other words, there's nothing that's forcing you to do something that will only work if it's perfect. Don't. Don't do it.

I mean imperfection has always been with us and I see no reason why we should expect disastrous results in the future from what we've lived through very successfully in the past.

DR. VAN DAM: I'd like to make a comment on that as well. I think if you look at the way programs have been hand crafted up till now, it's amazing that anything works at all. The stuff is put together so poorly and using such poor tools and such poor discipline that you can't really call it an engineering profession. It's becoming an engineering profession. We're beginning to learn how to do the things the way engineers have always done them if they work properly -- top down design, step-wise refinement. And we're also beginning to learn how to prove things correct as we go. Right now the proofs in order to be practical are still kind of informal, but there is a science, a mathematics of formalism, of trying to build programs that are demonstrably correct. Sometime within the next 10 to 20 years, we may have some breakthroughs in doing that.

The main thing right now that we really need to do is to educate students on how to do things right. That's one of the things that we really put a lot of emphasis on in our undergraduate curriculum. When you do things right, you reduce the chances for error. And managing complexity is part of what you ought to be teaching.

MR. EVERETT: We'll take one more question and break. Without a question, we will break. I will ask you please to bring your coffee back in here as quickly as possible. We will reconvene the program in five minutes, plus or minus two.

(Laughter)

(Coffee Break)