THE ENGINEERING OF THE VAX-11 COMPUTING ENVIRONMENT

The VAX-11 architectural design and implementation began in 1975 with the goals of a wide range of system sizes and different styles of use. While much of the implementation has been "as planned", various nodes (eg. computers, disk servers) and combined structures (eg. clusters) have evolved in response to the technology forces, user requirements and constraints. The future offers even more possibilities for interesting structures.

ETHERNET AND THE FIFTH GENERATION

Gordon Bell
Vice President, Engineering
Digital Equipment Corporation

In the Fifth Computer Generation, a wide variety of computers will communicate with one another. No one argues about this. The concern is about how to do it and what form the computers will take.

A standard communications language is the key. I believe Ethernet is this unifying key to the 5th computer generation because it interconnects all sizes and types of computers in a passive, tightly-coupled, high performance fashion, permiting the formation of local-area networks.

HOW THE JAPANESE HAVE CONVERTED WORLD INDUSTRY INTO DISTRIBUTORSHIPS -- CONCERN NOW FOR SEMICONDUCTORS AND COMPUTERS

Gordon Bell
Vice President
Digital Equipment Corporation

Abstract

We all must be impressed with the intense drive, technical and manufacturing ability of the Japanese.

As an island with few natural resources, and only very bright, hard working people they have set about and accomplished the market domination of virtually all manufactured consumer goods and the components and processes to make these goods (i.e., vertical integration). Currently the U.S. has a dominant position in computers and semiconductors. However, there's no fundamental reason why the Japanese won't attain a basic goal to dominate these industries, given their history in other areas and helped by our governments.

On a first visit to Japanese computer/semiconductor companies, universities, and a government R&D laboratory, I found them relatively open. This was in contrast to my former experience as a computer science researcher with their one-way scientific interchange and being an information sink. Perhaps their openness is because they are so far along with good products, and their position so secure. Their competence, hospitality, and "apparent openness" made me quite fond of them; but I now fear them more than ever.

Forty-odd reasons are given in the form of "feelings" to support this domination conjecture. No solutions are given, assuming a distributorship is basically unhealthy (as shown in the 3 island formulation). The reasons are formed from my observations, but like the Japanese, taken freely from other sources with an attempt to make a better, more complete end product for industrial and government users.

ID#0257
9/7/78
September 22, 1983

Dr. John Wakerly
David Systems
45 Cabot Avenue
Santa Clara, CA 95051

Ms. Christina Champion
Computer Magazine
10662
Los Alamitos CA 90720

Dear John and Christina:

I'm delighted to accept your offer to Keynote the Spring Compcon in San Francisco. Enclosed is a vitae and photograph for Computer Magazine.

The title for the talk is tentatively,

Understanding Evolution To Leverage the Leverage.

Civilization has always been concerned with building tools to leverage intellectual processes. Although a few tools are revolutionary, nearly all are evolutionary. Virtually all revolutionary tools (machines) fail, usually for simple reasons. What are the heuristics for success (and failure avoidance)?

I'll be in the Bay Area the week of October 17, and will call you about a meeting to further discuss the topic.

Sincerely,

Gordon Bell
Chief Technical Officer

GB8.13
LOCAL AREA NETS,
DISTRIBUTED PROCESSING AND THE FIFTH GENERATION

Gordon Bell

Vice President, Engineering
Digital Equipment Corporation

Local Area Networks are a natural evolutionary form of computing which have come into existence to form the Fifth Computer Generation. Economies of scale for processing have disappeared, but other forms of information processing (transducing, storage, communications, specialized processing) are still expensive. The high cost components are necessary and even desirable to share when all costs (especially including user time) are evaluated. Whether computing is personalized and independent, shared in a group or provided by a central service will vary with the user and organization. Local Area Networks offer the greatest flexibility in deciding where and how to process information.

OBSERVATIONS ON GENERATING COMPUTER GENERATIONS

We've implemented thousands of species of the computer in a few, basically evolutionary technologies. These technologies mark the generations. The evolutionary process is cyclic and includes the technology, the architecture and implementation of species, followed by use which in turn generates increased demand for better technology, permitting evolutionary new computer structures.

Since new generations spring from new technologies and often different people, a new generation most likely follows the time-worn path of early pioneers. New generation builders tend to relearn the same lessons about technology limits, architecture, its evolution including the "wheel of reincarnation" for specialized functions, multiprocessors, etc.

DISTRIBUTED PROCESSING AND LIMITS TO ITS GROWTH

Gordon Bell
Vice President
Engineering
Digital Equipment Corporation

Abstract

Invariably, the computer has been complex enough, yet structurally simple, to effectively utilize semiconductor and magnetic storage density improvements for both increased processing and lower cost providing widely available computation on a distributed basis. From an end-user viewpoint, it's hard to imagine a saturation, despite the fact that a leveling off has always been predicted. The new fifth generation and successor generation, VLSI-based computers, will stimulate use even further providing computation for what is analogous to the fractional horsepower motor.

It is worth trying to identify and then examine the factors which might limit growth: the basic technologies? the complexity of the design, given the planarity of semiconductors? either the lack or forced existence of standards? defacto constraints (e.g., communication and TV formats)? the imagination and skill of the intermediate designer/applier who must cope with the far greater complexity brought about by larger systems and programs? inability of the final user to cope with greater complexity? and possible rejection to being suplemented with so much information and information processing?

GENERATING COMPUTER GENERATIONS

Gordon Bell
Vice President, Engineering
Digital Equipment Corporation

Abstract:

The computer and pre-computer generations are marked by an identifiably new machine structure, physical technology, the basic needs for computation and actual machine use. There were four pre-computer generations and four computer

generations.

Thirty-six lessons have been observed, so far, and will be described and illustrated.

INNOVATION ISN'T THE PROBLEM - THE JAPANESE HAVE CONVERTED U.S. INDUSTRY INTO DISTRIBUTORS BY EFFECTIVELY UTILIZING AVAILABLE IDEAS

Gordon Bell
Vice President
Digital Equipment Corporation

Abstract

The island of Japan, with few natural resources and over 100 million people, virtually dominates world production of manufactured goods, including the components and processes to make these goods. The Japanese have progressed from domination of low-technology simple commodities to complex manufactured goods. The United States still holds a dominant position in the production of computers and semiconductors, but the Japanese plan to dominate these industries. Unwittingly, U.S. industry, government and society continue to aid the Japanese. Many reasons support this conjecture, each one providing a lesson.

TALKS /12

MINICOMPUTER ARCHITECTURE

Gordon Bell

Vice President, Engineering

Digital Equipment Corporation

& Professor of Computer Science

& Electrical Engineering

Carnegie-Mellon University (on leave)

Abstract

Minimal cost computer designs (i.e. minicomputers) are predicated on using technological cost-performance improvements which occur at an annual rate of 25-30%. New applications are thereby feasible with the decreasing costs.

A significant number of minicomputers are manufactured in which the cost is constant (or rising), thereby providing more performance (capabilities).

The higher performance machines "take" their characteristics from the larger, general purpose computers (e.g. floating point arithmetic, multiprocessors, cache memories and memory management).

The origin and evolution of the minicomputer will be discussed with regard to technology and applications.

THE PDP-11 FAMILY AND VAX-11/780 FOR A LARGE VIRTUAL ADDRESS

Gordon Bell

Vice President, Engineering

Digital Equipment Corporation

& Professor of Computer Science

& Electrical Engineering

Carnegie-Mellon University (on leave)

Abstract

In the eight years the PDP-11 has been on the market, more than 50,000 units in ten different models have been sold. Although one of the system design goals was a broad range of models, the actual range of 500 to 1 (in price and memory size) has exceeded the design goal.

The PDP-11 was designed and first implemented to be a small minicomputer. Its first extension was to a bigger physical address, memory segmentation for multiprogramming and for higher performance. This part of the talk will briefly reflect the experience in the design process, comment on its success from the point of view of the goals, and its use of technology.

The main presentation will be on the VAX-11 architectural extensions, including: goals and implementation.

PMS Structure Changes with Technology and Use

Gordon Bell
Vice President, Engineering, DEC
Professor, Computer Science, CMU

October 19, 1976

ABSTRACT

The basic semiconductor logic technology, cables and other constituent parts have caused the computer to change dramatically in the past; the change will likely continue.

The migration of control to be with the physical entity being "controlled", encourage networks and other multicomputer structures. Also, the final user is architect in many instances.

The evolution will be discussed, together with some of the protocols used in the architecture.

CONSIDERATIONS ON THE DESIGN OF A PROFESSION BASED SYSTEM

Gordon Bell
Vice President of Engineering

The computer science community is interested in building effective personal systems for the professional (especially computer scientists). Market forecasts abound for the Office Of The Future, and we build many point products such as Word Processing Systems, Data Entry, and Electronic Mail toward this end. If office workers e.g. professionals are to use computers in a significant way, computers must be both more useful (i.e. do more things) and easier to use (especially without programming). As a professional technical manager, one finds this frustrating dichotomy between need and knowledge that a computer could be used if it were only easier to use.

The goals and functions of such a personal system will be described. The main goal is providing a computing environment that is a pleasure to live in. It is important to build this system now for use within our environment in order to research the hard problems that these systems will create.

LESSONS FOR REJUVINATING EXPERIMENTAL COMPUTER SCIENCE

by Gordon Bell

Vice President of Engineering, Digital Equipment Corporation
Professor, on leave, Department of Computer Science and
Electrical Engineering, Carnegie-Mellon University

Although Babbage failed to produce the computing machines he contracted for, the indirect results to manufacturing and the fact Scheutz did succeed with the Babbage desing, were worthwhile. A significant number of innovations in computing have come out of a healthy, experimental computer science environment.

There is a proposal to NSF to significantly increase the amount of equipment in order to stimulate the generation of computer science knowledge using this equipment for experimentation. Senator Stevenson is proposing a National Technology Innovation Act to address the general problem of stimulating the generation of technological ideas that can be useful to various segments of society. The Japanese have a very good process for both stimulating the ideas and managing the flow of ideas.

It is worth looking at the various past situations, together with the several resource allocation proposals in order to rejuvinate experimental computer science.

STANDARDS: THE BASIS OF THIS GENERATION

Gordon Bell

Chief Technical Officer

Encore Computer Corporation

This generation is based on a compleley product fragmented industry that is stratified by levels of integration. Entrepreneurial energy is a major driving force. Short product gestation times and the rapid evolution require formal and de facto standards. What are the goals (product targets) and constraints (the standards)? What are the roles of the various organizations at the various levels of integration?

Foreword

On 11 November, 1980 John V. Atanasoff presented his work on digital computation at a Pioneer Computer Lecture at The Computer Museum. I urged him to write a fuller account and told him I would be honored to write a foreword. This is the first real account of his work outside of his August 1940 manuscript (reprinted in Randell's book) and 1338 pages of testimony in a Federal court trial.

The paper is important because it:

. is a primary source and, as such, its value will only become apparent with its use by historians. It should be valuable in the understanding of how science and technology develop, in general, and how the computer was invented, specifically.

. provides insight about people and organizations. For example, the controversary on the number base pervaded organizations for many years, and he turned out to be right in selecting base 2.

- documents his inventions of many important concepts in digital computation, especially the notions of serial computation and regeneration for memory, which he called jogging. Regeneration is the basis for delay line, drum delay line, Williams tube, and charged coupled device memories.

- gives an insight into how Atanasoff himself thinks, how he approaches ideas and problems. For me, it provided an inside view of a creative and brilliant person who provided significant ideas on computation.

Now, I urge you to read it.

Gordon Bell

20 February 1983

GB8.2

10/15/83 Sat

Dear Fellow Friends of JVA:

Please let us join you in honoring this great computing pioneer on his 80th birthday.

His ideas we take for granted about using binary arithmetic, storing information by regeneration which he called jogging and is the basis of today's random access semiconductor memory, processing information serially and fast division are almost 50 years old, and have been right.

From the moment we heard about the ABC at the Computer Museum in November 1980, we believed that others should know the story too. Over the last few years John and Alice have worked at writing about the invention of the first electronic digital calculator and we look forward to finally seeing it in print. We also applaud the movie as another way to tell the story.

We regret not being present on this occasion, but look forward to viewing and preserving the movie.

Sincerely,

Gordon Bell

Encore Computer Corporation

Gwen Bell, Director

The Computer Museum

Q. How long have you been at DEC?

A. I came to DEC in 1960. Before that, I was at MIT. I had gotten a Master's Degree there in 1957, went on a Fulbright to Australia in 1958, then came back and started down the Ph.D. route. But I really wasn't that much interested in a Ph.D.

Q. You were at MIT when you met Kenneth Olsen and the others who founded DEC?

A. Yes. There was a computer called the TX-0 at MIT--the first transistorized computer--and I started using it on a speech research project. The people who designed the TX-0 were the same ones who had founded DEC in 1957. I was working on some circuitry associated with that computer, using DEC products. They'd just built the PDP-1 when I came aboard in 1960 as the second computer engineer.

Q. You preferred DEC to the academic life at MIT?

A. I had pretty much decided that I didn't want to work as an engineer because of my experiences as a co-op student for a large corporation. I had wanted to be an engineer since I was old enough to know what they were. But practicing as a co-op student and my mental image of engineering just didn't match. I thought I'd <u>have</u> to go to academia for more freedom, responsibility, and more interesting work.

Now we're faced with trying to keep the engineering environment here from becoming like it was in the places where I worked as a student.

Q. You left DEC in 1966 to come to CMU. Why?

A. Yes. I worked here from '60 to '66. I did a lot of design and engineering. I was in charge of computer design. DEC was at about $15 million in sales then. We had more computers than we needed from a market standpoint, and I was feeling burned out--as a manager, and I wasn't designing things. I wanted to learn and do projects again.

That's a period I see many of our engineers go through here. It's the dual-track/dual-ladder problem--wanting to be fundamentally technical, to do design, to do projects as an individual, and, at the same time, wanting or feeling that one has to be a manager just to get more leverage on the projects one wants to do.

Traditionally no company is supposed to work this out very well, but I think we are. We try to keep both paths open. We try to have comparable positions as both management and individual contributing engineers.

Q. Why did you come to CMU instead of going back to MIT?

A. I had met Newell, and Perlis. Also Everett Williams, Head of the Electrical Engineering Department, influenced me to come. Carnegie had really put together the first computer science department. Ivan Sutherland, a Carnegie graduate aid head of ARPA (Advanced Research Projects Agency of the U.S. Dept. of Defense) also suggested CMU.

Q. Was the CMU experience what you'd hoped it would be?

A. Absolutely. CMU's a great place. Compared to other state and private universities I'm familiar with--it seems to be the right size and scale. I thoroughly enjoy the student-faculty interaction, general atmosphere and learned more than during any other period of my life. Writing the book with Allen Newell was especially rewarding.

Q. Your work at CMU was, of course, the computers. It's been said that by the end of the twentieth century, every major enterprise of science, goverment, education, and industry will rely on them. What does that mean in terms of how managers are or should be trained at GSIA and other business schools?

A. I think there's still a long way to go. many of the business schools, CMU, Wharton, Harvard, Chicago--they all have their own departmental PDP-10's. But they're just scratching the surface of what they could do. It all pretty much revolves around the difficulty of using computers. The students do solve some problems on a canned basis. They're given a program to run, and they put some numbers in. Computers are used for building simulation games. Students learn a little bit about programming--perhaps just enough to make them dangerous. Then they think they know what computers are all about because they've written some one-page BASIC programs. I think they need to build some large, non-toy systems and solve non-toy problems where they write reasonable programs.

The business schools have to change a lot in this area, and I don't know how they're going to do that. The computer science departments could help if they became much more strongly involved with the application in business schools. So much business school activity is in the industrial organization that that's a good application area. And I think every science or engineering discipline needs a target user to really understand systems design.

Q. You're implying that there are problems inherent in the use of computers by people who don't know as much about them as they should?

A. We get management reports that are absolutely impossible to deal with. The computer's made that possible. And it's probably made organizations grow bigger than they might have otherwise simply because you can keep your hand on a lot more information. You'll never be able to test that as a hypothesis. But we're far away from being able to have any kind of understanding or control in an organizational sense because of the way a lot of the reports are done. The touch part is putting machines within a human organization so that you can get more out of that system, not strangle it with paper.

The way the computer ultimately has to go will really be as something that supplements human information processing.

Q. Has the problem been compounded by the fact that computers have changed so much in recent years?

A. Right. Many managers still think everything has to go through one central place. That's sort of like having the Electronic Data Processing person be in charge of every telephone that's put in. Sure, you can do that. But is that the right way to build or run an organization? That's saying that person has to be <u>so</u> bright, because he's really setting and controlling the organizational structure.

But computers are being built differently now. Soon they'll be in every telephone, in every typewriter, in every copying machine, in every mechanism--and we all will be interfaced with very many of them in various ways.

Q. In another vein, how do you keep in touch with what's happening at DEC's 23 manufacturing plants? Do you do a lot of traveling?

A. Not as much as I should. I get out a couple of times a month. Unfortunately I haven't visited all of the plants. I should. It's too stimulating and frustrating because it causes me to push for changes in products, the plants, and the engineering process.

Q. Your work is mainly here in Maynard?

A. Yes. Every manufacturing plant has engineers responsible for the product flow in that plant. Design engineering is here and the design engineers do a tremendous amount of traveling.

Q. Do you rely on information sent to you by computers from the various plants?

A. Not really, although we use it extensively for message switching. Also, I don't stay that coupled to manufacturing. To really know what's going on I have to be more direct by visiting the plant or talking with somebody who has been there. The financial numbers tell superficial kinds of things. The whole notion of control should not be so oriented to financial numbers to the exclusion of other metrics. Mostly what's being controlled is not an item that is easily assigned a dollar value. In several years when classical engineering control theory gets into business schools and then can be taught and also learned, some change may be possible.

There are two basic ways to control: the input and the output. Business schools seem to teach controlling (actually just accounting) the input. For instance, one might control spending into the library. But whether or not that's at all right, one has to look at the output and its value. Controllers don't report on how many books get checked out or other transactions because these are non-financial. Much of the output of an organization really is non-financial. For a drafting room, the response time and the drafting they do per dollar are the control metrics.

Q. Do you try to control what's happening in your plants from your headquarters in Maynard?

A. That's manufacturing's domain. When we set up our first remote plant years ago, our Puerto Rico operation, we started out with just assembly there. All the controls were from here, all the ordering. Parts were taken down, assembled, and then testing was done back here. And from a control standpoint, it was not very satisfactory because there were long delays in the information channels. Now plants control their own raw materials and are self-sufficient.

Q. How do you spend your time on the job?

A. I try to avoid going to meetings, with little success. I try to work by phone, and I like to walk around and visit the projects--contribute to them and critique them. Sometimes I visit customers to see how other people use our computers--at Kodak, Dupont, and other customer sites. Applications are all different at each site in terms of what their problems are. I also try to spend a fair amount of time on product issues. If I look at the problem list that I deal with, it includes everything from defending the engineering budget, to supporting a product that I feel particularly strongly about, to hiring. I like products. So I worry about strategies, advanced development, research and things we ought to be doing that we're not.

I worry some about space and that people, space and capital and equipment are in balance. I worry about keeping all the processes going. Essentially we've got a whole zoo of processes that have to be kept on-going. Traditionally, in a high growth company, these are the things that you ignore until there's a disaster because all the effort is on product strategy, hiring and training.

As a hobby, I like to write about computing. This was training I got at CMU. Now, one of the engineers and I are doing a book of readings with several original papers on the engineering aspects of our computers.

Q. Do you pay close attention to your competitors?

A. Sure. I'm always worried about the competition. they can come from everywhere. IBM, the semi-conductor vendors, all the people the semi-conductor vendors put in business, the existing institutions, the Japanese.

Very often the only thing our engineers understand, however, is market feedback or competitive feedback, and I detest our behaving that way. The minute you start responding like that, you're building a development process that's going to produce obsolete products. I think that's always the biggest danger when you get bigger. Keying off of a conventional marketing structure is the biggest worry that I have. And DEC traditionally hasn't done that. It doesn't mean we don't listen to the marketplace. But the minute you start responding, then in fact you're gearing a process that is going to be about eighteen to thirty-six months behind--or whatever the gestation period of a product is.

Q. Are you concerned about what IBM is doing with minicomputers?

A. IBM has such vast resources to do things that by and large it sets its own standards and goes off and does what it wants. Everybody has to couple in with them in some way. We have to be able to communicate with their machines.

Q. Are you working on tieing machines together and what standards exist?

A. In the network area, standards are very desperately needed. Yet the standards work didn't progress rapidly enough so people could key off them. IBM seems to have made some mistakes in the network area that we think will be too limiting. They may move the users around so much that it'll be too hard to track.

When we went off on our DECnet system and standards, which allows users to build a variety of networks connecting our computers to each other or to those of other manufacturers, we felt we knew networks. Still, it's turned out to be harder than we thought. We are successful now.

Q. What about standards for computing?

A. Historically, I believe computers have evolved rapidly because the government has been an intelligent and demanding user. They have not designed the systems by specifying standards. The government really bothers me in terms of the way it seems to now want to operate in the standards area. One particular standard that I was very much opposed to is supposed to reduce their disk acquisition costs. I worry that it will impede technology and cause higher costs too?

Then, too, on something like the networks, communication protocols should have been standardized before now. That one's trailing, and we're facing some real problems as a result. It's like having a bunch of private telephone exchanges, each with different signal levels so that every phone network needs a converter if it wants to communicate with another.

The government, together with the telephone company, should have set the standards. I don't think it would have been political, and it would have saved an incredible amount of time on everybody's part. We'll ultimately have to do that anyway.

Q. The computer industry isn't regulated by a government commission. But you do have numerous regulations to deal with?

A. Sure. there are product safety regulations, guidelines for power suply efficiency, radiation, noise--I could go on and on. We've got the government pulling at us. And I don't want us to get into a relationship like that which exists between the FCC and the telephone companies. Little happens outside of that structure. That doesn't mean the telephone company hasn't changed over a long period of time. They've improved service a great deal. But computers have evolved rapidly over a short perod of time. Looking at our indicators on an exponential basis, for the short time we've been in business, things have changed rapidly and a great deal. What was important a couple of years ago isn't going to be important in two years, just when the standards begin to come out. The government standards process can't deal with these very rapid exponential changes.

Q. Looking at computers in a general way--are they overrated?

A. Not at all. Historically we'll look back and say that they really started being used about 1975. It will be similar to the Industrial Revolution but more significant. There will be a clear line of things that have changed and were totally impossible without machines. People's lifestyles will change. I don't know what the revolution will be called at that point, but the computer is clearly the root of it.

Q. Will there be a number of computer-related inventions?

A. Much will be keyed off of supplementing existing information processing. On the other hand, we'll be doing things that we couldn't have done otherwise. For instance, having a robot in the room that's smart enough to know when there are people in it and controlling the lighting and heating accordingly--and doing other trivial tasks. Doing all the things that no one can train his children or wife or himself to do will be possible. Computers can take over a number of chores that are in the resource control domain.

Q. How will they be used in the near future?

A. In all kinds of ways in the communications area--for message switching, for all the office automation. I think computers will come to be used widely--simply for communications and text preparation, storage and transmission (e.g. electronic mail). More and more people will start to do local, totally distributed processing. I think most all conventional tasks computers perform will move to a totally distributed form to be associated with each organizational entity.

We have a word processing system here in my office. We moved from a big machine to this. My secretary loves it because she's not dependent on the large machine. Also, I can type memos and messages myself. It's got processing associated with it, and there's much that it can do that we use to go to the large machine for--report generation. In fact, we can operate and do a lot more control now. We keep a list of all the projects we track, for instance. To get that kind of thing done from the corporate data base is virtually impossible. So the task is to get some of these things down into the organization where people feel comfortable. There an organization can operate the way it's operated before, but more efficiently.

There are a lot of reports that are generated, a lot of files. Everybody's got file boxes on their desks, or a list of things to watch and do. All of that they can do now with these word-processing machines, and they'll do them informally. The centralized system person always says "I'll maintain all the files and all the reports that everyone wants in the whole organization. I've got this one data base, look how great it is. There'll never be any wrong information in there." But the problem is the timeliness of the information. Also large, central data bases are very difficult and expensive to build and maintain. Unless people are keyed in or have a terminal into that data base all the time, it's not very useful and it's generally wrong. You can't get the response you need in terms of kinds of queries and formats.

Q. Modern science fiction has utilized the computer extensively. The computer is the bad guy. Is there a danger associated with the computer?

A. It's not a bad guy. But it can be an instrument of bad guys. The notion that you can have a machine monitoring all of the communications in the world, processing all of that communication and filtering it--well, I've never tried to compute whether that's possible or not. But right now--and as far into the future as I can see--it feels impossible. And even if it weren't, there are very good security devices that we <u>do</u> have right now. You can put a personal scrambler on a telephone that you carry around with you, if you're worried about that sort of thing. Technologically we can deal with security problems. We can have secure communication channels.

Then there is this whole business of records. I don't think that presents an insurmountable problem, either, of course people have to be a lot more careful than they have been with information. The risks are no greater, however, than they have been.

Q. Does someone who aspires to running an organization have to know computers?

A. Yes. If for no other reason than to have some notion of what a process is. But whether they can really effectively understand how machines will diffuse into organizations and be used is questionable. So many mechanisms of how processes work in organizations, all the informal communications paths, how an organization performs its functions, aren't very well understood. And computers point out the lack of understanding because all activites for machines must be so explicit. Machines can force a rigor that I think is necessary in would-be, cloudy headed, future managers.

Q. Why should the average person bother to learn about computers?

A. Simply so he can get along and understand the world. You have to have some way of relating to what the world is today rather than just writing letters that say "Your computer screwed me."

It really burns me up the way people put in systems and then use the computer as a scapegoat. "My computer did it to you," they'll say. That's nonsense because the organization (usually just one person) is respnsible in any event. They'd better have a process in place to sort and cope with the input.

I get furious when I get a bill for 00 cents, or a check for 00 cents. It's simply unnecessary and at best a sloppy program which is permitted to exist by some sloppy, wasteful manager. And then sometimes people will get dunning letters and threats, and when they write or call to complain, and ask how can things be so absolutely screwed up, they'll be told laughingly "We;ve just installed a computer, and you understand what that means."

This is totally absurd. What it comes down to is that people have found a new scapegoat, something they feel everybody can relate to and understand. Somehow there is a notion that people weren't involved; it was a machine that did a dumb act. It really irks me. If you think the computer is causing the problem get it thrown out!

Q. Do you have any advice to people on what their attitude should be towards the computer revolution you predict?

A. People shouldn't worry about it. They should relax and enjoy it. Machines are (or should be) friendly, fair and basically helpful. I think it's going to be fun. It'll all come in a basically innocuous way, driven mainly from the economics of everything. It can'be stopped, especially as long as organizations are operated so much on purely economic metrics.

CHALLENGES IN GENERATING THE NEXT COMPUTER GENERATION

Even before the Japanese told us about the Fifth Generation, Computer Generations have been of interest--what they are, why they happen and especially the next one. This fascination surrounds a computer structures taxonomy essential for The Computer Museum and understanding computer evolution. The Museum must have a way to contain and segment ideas: by generations and by information processing structures. Observations from our past and present will help in creating our future.

WHAT IS A GENERATION, now that we need one?

A generation is simply the convergence of:

need (in this case--threat of military and industrial annihilation) which frees resources;

technology, science and ideas to build from; and

organizations to build

new computing structures.

Finally use will confirm a generation after the fact.

The whole process is like a cyclotron and a generation is one or more trips around. The concept to "do a machine" is injected into the accelerator at some stage... I'd like this to be needs driven to a large extent. Technology is the first stage, architecture and design are down stream, followed by the actual construction and manufacture. System software further accelerates the electron. Algorithms and use with critical evaluation, which we often ignore, provide the final stages... and of course by now, the particle has gone around once. And now it is ready to be accelerated again and attain the critical energy level necessary for real use or for going around again. For many generations, going around twice constitutes a new generation. The first time around a new structure is formed, and the second time around it is made more useful and gains market acceptance. Clearly the Personal Computer (PC) was like this: the very first PC, the LINC, now in the Computer Museum, cost about $40K in 1965 but not until 1975 with the microprocessor was it viable from a market perspective. The PC actually took about three trips around to reach the high energy level characteristic of a Generation. The Apple and IBM PC characterize the second and third times around the ring. Now a trip around takes less than 2 years. This process is highly evolutionary with all parts of an industry acceleration.

Richard DeLauer, of DARPA, claims the U.S. is working on the Nth generation, and I believe that the Fifth Generation is already cast, even though the Japanese are laying claim to it.

WHAT IS THE NEXT GENERATION?

Last week Alan Perlis spoke at The Computer Museum, and in passing gave a number of his pearls:

"If a computer understands English, it must be Japanese."

My concern is that the Japanese have already won the race to the next generation. In the past, no one was interested in a race, contest or game. In fact our strength was the independent, uncoordinated inventors of board games, physical skill games, simple intellectual games like Chess or complex ones like Go. Now as a guerilla warfare army, we've been drawn into a contest where we seem to be forced to compete and where we have no knowledge of the rules. We have no notion of how to pick teams or whether the game is played with teams or individuals, and whether more or less resources count. In the midst of all this, many forces are moving people among institutions.

The Japanese evolutionary approach to engineering and their leverage of the world's research has been impressive. They understand the notion of long term processes and learning from the past. We also can learn from the past.

Observe how the Japanese understand this notion of generations and evolution. The concepts of AI and AI workstations have existed for years in the lab. They started with a DECsystem 10 and are making the very best workstation hardware they can to execute LISP and Prolog at a factor of 10-20 times the large system! In parallel, they're working on significant real applications and trying to develop the engineering discipline. Finally, they'll use and evaluate their applications and workstations in order to go around again at a much higher performance level. They plan about two more evolutionary cycles by 1990: use with critical evaluation, re-architect, build, deploy, then repeat the use and evaluation stage to start around again. The important thing is to start with use NOT revolutionary new structures!

In a recent talk, Mike Dertouzos of M.I.T. says there are 4 ways to beat the Japanese in the forthcoming race:

1. $100-200M to develop high speed computers with AI functions
2. an open policy toward foreign workers in industry and academe
3. tax credit for long range and in accord with national policy
4. careful re-exmination of antitrusts to permit consortia

He also argues for foregoing the traditional short term gain at the expense of long term R and D.

The above 4 points do raise questions:

1. Where's a reasonable plan that would spend $100M in a coordinated fashion? Won't a large budget just serve to swap a fixed set of people from place to place? Postwar university research has been run as independent, decoupled projects. Can we change to a more coordinated, directed approach that a new generation requires?
2. Although we have been successful and probably need an open door policy, is this really a relevant success factor?
3. Where is a national policy or plan? R and D credit may just go right to the bottom line of a corporation to increase earnings. Similarly, few corporations are equipped to do either credible or useful research. Even Advanced Development can be a conflict because few managers understand the differences between product development and product enhancement, let alone concepts of basic and applied research. There is poor understanding of these activities and we clearly can't manage the flow of ideas through the stages. The Japanese are masters at moving world research into products.
4. We have several consortia but they have taken a long time to establish; antitrusts may not be the issue.

Now, we must learn from Japan about how to define, establish and then execute projects. The Japanese Fifth Generation effort appears to be 3-5 years ahead of us because they understand large scale, long term interacting processes and they have a plan that started in 1980 and based on the world's research.

For example, in contrast to the Japanese directed and evolutionary approach, we have many projects aimed at designing and building revolutionary machines at various universities to exploit fine-grain parallelism. All, violate the historical notion of evolution since they start with a structure that looks interesting to build with new technology and not science, programming technology or a problem.

They all involve incredible personal commitments. How many revolutionary computers can we really afford to build to completion with analytical use? Are we prepared to run these 10 year, very high risk experiments?

WHAT IS THE FIFTH GENERATION WE'RE ENTERING?

The need is intercommunication.

The technology is VLSI which permits powerful microprocessors and Local Area Network interconnection.

The technology permits building: Personal Computers, powerful personal workstations, multiple processors for fault tolerance and performance.

WHAT IS THE NEXT GENERATION?

The emphasis is on Artificial Intelligence applications with voice and natural language communication, built with VLSI and ULSI and predicated on a high degree of parallelism. Furthermore, the new structures will supposedly be revolutionary!

For any generation, we need a clear view of the target and the problems standing in the way. Although fuzzy, the Japanese appear to have a view and an approach. Finally, the notion of revolution is not consistent with a next generation.

Only a handful of real AI applications, including "expert systems" are in operation. A future predicated on parallelism is equally risky based on past results. Thus, a computer can't be evolved unless a model of use exists. Revolutionary machines usually fail even though they often provide useful by-products. Breadboard of the real structure usually operate in a previous generation. Do these structures now exist?

THE PLAYERS

Even with all the caveats participation is required by everyone provided we can have a more focussed approach. Even a guerilla army needs some leadership. In the past, DARPA has indirectly provided this leadership and science for industry in the form of timesharing, speech understanding, graphics, packet switching, and most recently VLSI.

Universities played the key training and scientific discovery roles in the past. The university role is vital because the science of parallelism is underdeveloped, ULSI is too hard and we have little understanding about communication with people.

Jay Forrester, who headed MIT's Whirlwind and invented the core memory, made several comments on building machines in Universities that still hold today:

"Experimental equipment merely for demonstration of principle and without inherent possibility of transformation to designs of value to others does not meet the principle of systems engineering".

This lesson should govern building new experimental machines: Unless a machine provides about an order of magnitude more power to the individuals who may use it than is available to them, there will be insufficient pull to attract users and test the basic idea. In other words, don't build toys. However, building experimental systems today appears to be even more difficult than in the past.

WHAT CAN BE LEARNED FROM PAST EXPERIMENTAL MACHINES?

Table I shows several university-based computers in the first-fourth generations. Nearly all were useful in training engineers and scientists. Some machine not only were especially useful, but in addition trained users and provided insight into various algorithms.

Harvard's Mark I played a role in the search for the computer. The main architect, Howard Aiken was not particularly gracious in acknowledging IBM who actually designed and built the Mark I, which might be considered an impossible to build machine were it not for IBM's impressive engineering. The later Marks weren't near the state of the art, and none were as influential. However the most important by-product was to train individuals who have influenced computing.

Columbia was influential when Wallace J. Eckert got IBM to build the SSEC computer, a first, pre-computer generation machine composed of relays and vacuum tubes and using many of the techniques derived from the Mark I.

At the University of Pennsylvania, ENIAC was the truly revolutionary machine because it provided several orders of magnitude more performance than the Marks or the Bell Labs relay machines! The stored programs concept came from the ENIAC. The work lead to EDVAC, IAS, the University of Illinois' ILLIAC I, and then indirectly to the computer industry.

MIT's machines were evolutionary in structure, but revolutionary in technology with Whirlwind. Later on the TX-0 and LINC were also successful and influential. TX-0 took about a year to design and then was in use over 10 years. The well engineered, state of the art circuits were the basis of starting Digital Equipment Corporation.

A machine rarely pioneers more than one aspect of computing: current technology, architecture, on application (use).

UNIVERSITY OF ILLINOIS

Illiac I was built in the IAS and von Neuman architecture. The real contribution was the detailed circuitry and logic that permitted copies to be made at various laboratories. The machines had a long life resulting in contributions to knowledge and use about software and applications.

Illiac II, a transistor circuit-based machine, operated three years after the start of the second generation.

With a new architecture, new circuits and logic, the machine, was completed three years after significantly better industrial machines, e.g IBM 1401, 7090, CDC 160/1604 and DEC PDP-1, were available.

Because there were so many new, risky parts, conservative and obsolete technologies were selected (i.e. germanium versus silicon transistors, discrete wiring versus printed circuits), creating an unwieldy machine. Furthermore, asynchronous logic and a small memory were used to further slow down the system. Although the machine was designed to be a very high performance computer, the industry moved past Illiac II and hence the notion of building experimental machines at universities was squelched for sometime.

Illiac IV came out of the Solomon project described in 1962. Illiac IV, a truly revolutionary machine, was put in service in 1975 and operated at 250 million operations/sec. with a total of 64 parallel processing elements controlled by a single instruction stream. A memory hierarchy for the processing elements of 1 Mbyte (RAM), 2 Mbyte (core), and 139 Mbyte (fixed head disk)--clearly violate Amdahl's constant of 1 byte of memory for each instruction per second.

Dan Slotnick, the designer of the Illiac IV commented:

"Most machines come about through evolution and that's counter to the notion of original research which is supposedly the basis of university rewards." The activity of building a machine for study entails major engineering; this too can conflict with the emphasis on science.

"I'm convinced that universities can't and shouldn't build machines. There are too many ideas, too much democracy and too little discipline. I used to have to stop the flow of ideas on interconnection every week when we were designing Illiac IV. There is also too much bureaucracy. In a state university it takes 90 days to get an IC."

Larry Roberts, who headed DARPA then, claimed that it was absolutely clear that the machine should have been done with TTL and not ECL technology. "People complain bitterly, but in the end, conservative technology seems to work out better." (This is what I like to define as a tradeoff of instructions per second versus instructions per month. Not getting an operational machine limits its life and delays the essential purpose of the original design--which should be to understand if the structure can be useful!

The contributions of Illiac IV were mostly as by-products even though it did operate as the world's fastest machine for some problems until the Cray 1 came into production. A number of people began working on parallelism at Illinois and elsewhere. The fast semiconductor memories that resulted from the effort were essential for all machines including the Cray 1. Illiac IV may have stimulated TI's ASC, CDC's STAR and the CRAY 1.

CARNEGIE-MELLON UNIVERSITY'S MULTIPROCESSOR

CMU's machines, designed to obtain experimental results about parallelism were more evolutionary and had more side-effects for a smaller cost than Illiac IV.

Multiprocessors are intriguing to an engineer because performance is obtained by replicating a simple design instead of massive design. We must understand them so they can be applied to real use. Furthermore, multiprocessors represent another form of parallelism whereby multiple instruction streams operate on multiple data streams.

Multiprocessors were studied at CMU in the late 60's, and Bill Strecker's 1970 thesis computed the performance for p processors accessing a common memory of m modules. This main reference work for multiprocessors was rejected as the first paper on these structures because it wasn't relevant at the time. There have been dozens of subsequent theses and papers during the last 10 years which embellish the model, and all reference the Strecker thesis.

Today's research on switching structures which focuses on thousands of processors and memories seems to be completely irrelevant because we have no evidence that over a few processors can operate in parallel on a single problem. In fact, in a recent visit to the University of Illinois one researcher stated to me that he wouldn't work on a project of only 32 processors if it couldn't be extended to 1000! This prevailing attitude which focuses on the exotic completely masks the more difficult job of building and what may be the impossible job of using such a machine. Thus we have a paradox: we have no real demonstration or understanding that more than 10 processors can be used effectively; on the other hand, the researchers who must provide this fundamental understanding have no interest in developing an understanding because of the focus on finding switches for several thousand processors.

The issue is not the switch performance, nor finding exotic switching structures, but simply: getting on with finding out whether multiprocessors can work together on a single problem. This is a combination of architecture, system software, language and algorithm design. I believe that if anyone can demonstrate that a small scale multiprocessor of say 10 can work routinely in production, we can extend this to a large scale multiprocessor of 100 and then to 1000. Note, the Japanese Fifth Generation project is predicated on parallelism.

C.ai, a multiprocessor with 16 processors for AI research which had a one gigabyte, very high bandwidth memory called C.ai was proposed in May 1971. C.mmp, a much simpler design, was in place using 16 PDP-11 processor modules in August 1971.

The project had two goals: a capability based Operating System based on changing the PDP-11 and to examine the use of multiprocessing. The addressing problem using the PDP-11 became a major issue and problem.

The project is well documented including what was learned in Wulf's book on Hydra. Maximum speedups were hard to obtain. It is unclear why C.mmp wasn't used enough for applications, but on the other hand we know that any machine must provide more computation than is available by other means in order to be attractive for users. By 1978, the CMU computing environment had more machines which were easier to use. Ironically, not everyone on the project learned about the small address problem when they went on to design the Intel 432.

Cm* is a set of computer modules which permit building a medium scale multiprocessor of 50 processors in an open-ended fashion.

Cm* was an evolution of C.mmp, and foresaw the notion of functional multiprocessors that is used in Intel's Multibus. Cm* used the same C.mmp operating system concepts. Even though any processor could access any memory, there was a preference to a local memory, or to other processors within a cluster of 10, and finally to memory outside the cluster. Thus, the machine is problem idiosyncratic because the access time varies whether data is in local, in the cluster or external to the clusters. The structure of computation and data with respect to particular physical structures is being understood using Cm*.

Significant work is needed before these machines can work together harmoniously without extensive hand tuning of programs. The evolution of Cm* from C.mmp paid off.

For Multiprocessors, the progress has been slow. In each generation, there is renewed optimism in the concept. In the mid 60's with large computers and mid 70's with minicomputers, I felt multiprocessors were the best way to provide more computation. Now since the smallest unit is the very high performance processor with the characteristic that the smaller it becomes, the faster it goes, multiprocessors must be an important way to increase performance. Also note that many companies are finally offering multiprocessors in all product ranges: supercomputers (Cray X-mP, Dennelcor), superminicomputers (ELEXSI), and micros (Synapse).

Maybe there are reasons why multiprocessors have not been used appreciably:

. we always find a simpler way using technology or instruction set (e.g. vectors) to provide high performance

. engineering has been too conservative

. operating systems and languages haven't supported or encouraged them

. too many other ideas are present and in use

no market because users may not be able to program them/no product to test a market

With the advent of several commercial multiprocessors, it is critical for universities to become involved in use and providing understanding.

Human organization theory doesn't seem to help the work on parallelism except in an anecdotal fashion. More than a decade ago, Melvin Conway wrote that people build computer structures like the human organizations they know. This explains why n people build n-pass compilers; IBM build hierarchically structured protocols like SNA; ARPA has to have a store and forward net independent of its users; Digital builds democratic (anarchic) structures like Unibus, Ethernet, DECnet and multiprocessors.

If we could use human organization theory it might shed light on parallelism from structures that are connected together in particular ways. It might also explain, like humans, why its difficult to get more than six processors to work together--unless totally top down directed with clear goals (like, capturing a beach or hill). For now we need to concentrate on the general case of multiprocessors because it has the ultimate in connectivity via the memory. Slow or restricted networks such as LANs, trees, hypercubes, etc. can come later after we understand the general case.

Some general observations about the experimental machines:

1. nearly all the machines were useful in training computer engineers and scientists

2. early machines tended to be built for use, not understanding and hence had longer lives.

3. machines with long gestation risk a long life or being useful due to obsolescence from competitive approaches

4. later machines tend not to be used because other production machines are available as such their contribution has to be to science

5. unless a machine is used, its contributions to understanding algorithms won't happen

6. a single machine is unlikely to pioneer more than one aspect of architecture, technology

7. revolutionary machines such as Eniac, Illiac IV and C.mmp provide understanding and by-products that are more important than the use.

Given the concerns, but yet the need for research, there are several ways to do this work:

1. cheap labor of graduate students... brilliant, but unpredictable. Not recommended unless the machine is easily assembled from well-defined industry standard modules or computers!

2. professionals within the university which create a second culture. This structure is somewhat difficult to manage and unstable, but essential to building a system within a university. This is what has been done at the CMU projects.

3. jointly with a company. A hardware/software split may be the right division of labor whereby universities do software. This was used in the PC generations. Why not do it again? It's being used at CMU with IBM for products. The Japanese companies build machines for the various universities, especially at the University of Tokyo.

4. as a separate company outside the university and fueled by venture capital.

Contrary to popular opinion, I believe large amounts of money will cause excessive swapping of people and further perpetuate the erroneous economy-based notion that money can be traded off for science ideas, and talent! The money comes from two sources:

1. The government. This acts to simply churn the small number researchers moving them from place to place. The nice effect is to raise everyone's salary.

   Since these research projects are large, they require professors to be good project managers in a university environment designed for teaching. Large projects diminish freedom. By becoming managers, the reaction after a few years may be: why work at sometimes lower pay, lack of freedom and without adequate engineering resources. This provides a target for industry to scoop up kernels of the nation's "seed corn."

2. The Venture Capital world which draws people from established industries and academe into often mundane and low tech products because of the risk. For example, one high tech company started up in March and were shipping your generic 68,000-based UNIX product in 9 months, the standard gestation time. A company of 4 recently, built one board and assembled a UNIX product. Others build NOTHING but merely assemble.

Many believe that entrepreneurism is the way to beat the Japanese because it unleashes such an incredible amount of focussed energy--but I wonder if the Japanese are going to feel threatened by 123 different kinds of 68,000 based workstations! On the other hand this system has funded really creative technology, e.g. Amdahl's Trilogy Corporation.

Today the result of several recent dissertations at Stanford has been a chip, program, algorithm or system capable of starting a company. Recent examples at Stanford include Clarke's Geometry Engine the basis for Silicon Graphics; the Timing Verify of Widdoes/McWilliams, the basis of the Valid Logic Company; and the SUN terminal, the basis of SUN Microsystems. So finally people can rapidly progress through the cycle from freedom to fame to riches.

I don't know what the final answer to the question of how do we continue to generate interesting generations. But it is clear we've got to get organized. Or in the words of Pogo, "we have met the enemy and he is us."

GB8.22 - 12/19/83 Mon 17:21:19

12/20/83 Tue 11:03:16

Thoughts stimulated by the Poduska/Apollo 13 December 1983 talk:

Apollo... has gone for quality people who have a very good understanding of the industry. The only thing to stop them in getting to $1B by 88 is IBM, when it gets the workstation... it seems determined to get. DEC is clearly no threat even though there are a few tiny islands of competence in a sea of incompetence.   DEC is the source of people... just as Honeywell staffed DEC in the 70's.

This is another clear indication of a generation: the new structure surpasses the old, a new industrial layer is formed and the high output people leave an organization to build a new one. Honeywell folks left to come to DEC. The good folks of Prime, DG and DEC now leave to form the next generation based on Micros: Masscomp, Apollo, Sun, Sequent, Dataflow, etc.

IBM... which I believe was based on the OEM market models of the old DEC. IBM also learned about university interaction from DEC. IBM is fundamentally structured around Independent Business Units which are entrepreneurial centers. IBM has quality folks who can manage.

PRIME... all fouled up and probably irrecoverable until it gets a new head. Options are to be a marketing company. It's technology is nil. Henson's experience of service was wholly inadequate for running a company. Prime's success came from unique products in the early days which allowed building a total oranization. Finally, its success was its sales force being incentivised and in effect being the entrepreneurs of the company.

DG... looks quite good! The new IBMers, Chapman and Miller are impressive and have brought in a deeper understanding of the industry, management, computers, quality, manufacturing, and technology. The power has been distributed in an entrepreneurial fashion. DG will grow at a substantially higher rate than DEC or Prime, but not at Apollo's.

DEC... is unworkable and headed nowhere. There's no overall control system given the destruction of the Product Line structure which, in the past, provided a product check on engineering AND MANAGED the field (because sales was not incentivized). Now, no one manages the field sales. The company lacks entrepreneurs, and top management lacks basic understanding of products, industry, and quality (as say compared to DG). The large functional organizations (sales, engineering, manufacturing and marketing... whatever that is now) place responsiblity squarely and solely at the top! It can now only prosper by divisionalization of some sort. It is too large to be a functional organization like it prospered on. Lack of checks on Ken as President, CEO and COB together with a nil board means more stockholder suits... I'll bet Caldwell will leave the board soon. There is clearly no successor to Ken.

<>slide of product segmented/level of integration stratified
This generation is based on a large set of product segmented industries that are organized by levels of integration strata. Entrepreneurial energy fuels the individual companies of the product industries.

<>Venture_Entrepreneur_Cycle
Using the basic instincts of fear and greed, entreprenurial energy and venture capital greed and fear drive the company formations... these supply the critical needs that free resources in what now appears to be a never ending cycle like a perpetual motion machine. The release of this energy and capital can only come about if the basic technology supports new products. This generation has been on a long, steady role since the development of the microprocessor and may continue for awhile as long as technology progresses.

<>Slide of 4G
This generation, which is called the fourth, independent of whether we call the next one the fifth, sixth or simply the next one is based on standards. Note it is an evolutionary one based on evolutionary semiconductors especially the microprocessor and large semiconductor memories, magnetics and communication technology. Use is evolutionary too, providing much more access to interactive computing by using Personal Computers instead of shared systems. This wide use creates a a massive interconnection problem which the generation must solve. This in turn demands the use of all kinds of Local Area Networks and the necessity of standards. Like other generations, we won't understand the real nature of

generations as measured by useage until 10-20 years from now.

Previous products provide the critical goals for what are mostly evolutionary products at lower cost and much higher demand. Demand for minicomputers doubled each time the cost was reduced by 20%. Occasionally, revolutionary products emerge based on new technology, but these are surprisingly rare. Though semiconductor technology evolves rapidly, measured by speed and density, it is evolutionary.

<>2 slides of structures made possible by the micro
(AEA/Rodgers)
Unlike the previous generations where the processing element and memories constituted a large fraction of the size and cost, with the microprocessor, these parts are a comparatively small part of a system. Thus, many more structures are possible.

Previous generation products are the goals. For example, most business plans start with the goal of building a better VAX at a lower price using one of the three microprocessors or they combine micros to build a higher performance or higher availability machine at the same price. The final section of the plan lists a VAX 750 as capital equipment. This is the ultimate form of capitalism. I feel like the capitalist who sold the rope to hang himself. As a startup now, my PC at Encore Computer is a VAX 730 and most of the Encore companies have 750's.

To beat the VAX is an important goal, but constraints are also important because they help focus. Standards are the constraints for finding the target or at least defining the class of games or race courses we run. Defacto and industry standards allow the statification by permitting incredible parallelism in the development of products. In this way one can build new systems quickly by assembling lower level standard components quickly.

I intend to explore, in a somewhat rambling fashion, the nature of standards, and ask that the next COMPCON be completely devoted to standards because they form the constraints necessary for both evolutionary and revolutionary technology, processes and products. The conference would ask

- . what kinds of standard and what are needed for the future -- without constrating creativity
- . what are the goals and constraints for various standards
- . what are the responsiblities and behaviors of various organizations, manufacturers, professional organizations, governments, academe
- . how do standards form, ie timing
- . when do you leave things alone, when do you evolve, and when do you throw out and start a new path

My obsession about standards lies in a belief that the lack of standards at various levels of integration has been both costly, is simply non-productive and impedes technological progress more than any other factor. At least 50% of our efforts seem to go into supporting redundant work. One of the questions such a conference might examine is the economics of supporting multiple standards.

We can observe the effect of having a good standard in the case of the IBM PC. It came at a propitious time-- concurrent with a processor capable of accessing a megabyte of memory, the 64K chip, widescale availability of 5" floppies, and just prior to the availability of 5" winchesters. The great progress or rather explosion in software came about because people could work on applications instead of reinventing and porting operating systems for various hardware idiosyncratic PCs.

A guiding principle should be:

either make the standard or follow the standard!

On the other hand, a caveat about following de facto standards:

be prepared to react quickly and follow!

Those who follow the IBM standards might observe that IBM does change as in the 360/370 transition... ask the Amdahl corporation about this. We will soon see a repeat of this as the next IBM PC obsoletes a current product by providing a fully upward compatible product, but offering more capability. One can fairly accurately guess about the characteristics by looking at the 286 architecture.

Setting de facto standards such as the Unibus are important, because they are first and establish the way for succeeding product generations. To form an officially approved standard in 1970 for interconnecting computer components to form minicomputers just wouldn't have had any takers. Today, everyone recognizes the importance of the bus in building computer systems. Every company, consortia and many academics try to get one more standard bus or feature to ride the bus specification. Having too many alternatives such as the set forming 802 simply delays work on building networks and distributed computing systems.

<>Guidelines for standards

Given the desirability of standards, let's look at some heuristics governing them.

The first rule is to have someone (person, persons, a company, several companies) responsible for defining, implementing and caring for the standard. This is called responsiblity. Preferably the standard has to work in order for the individuals to be successful. Let me cite Ethernet as an example of this. All of the companies needed it: Xerox and DEC as the backbone of their produt strategies, Intel to sell chips. Rarely will we see such an important interconect need as the LAN. How it is implemented is moot-- the modulation and topology whether busses, rings, trees, or centralized switching is quite irrelevant except that it completely consumes us and hence we can't work on building the systems.

An equally important part of making a standard based on existence proofs is the ability to test conformance. This is another responsibility of the sponsor.

A standard must be real. The best way to insure reality is that it has been implemented before designing it with a committee who are sure to make it unimplementable.

Unfortunately, when engineers get hold of a particular implementation, the temptation is to look at the implementation as a template, throw out the old, and extend it... not just use it in an upward compatible fashion.

Again, Ethernet is a good example. It took almost ten years to get a standard called 802.3 after the original Ethernet operated. The upgrade provided almost a factor of 4 performance improvement, but the delay in starting the understanding as to what is really wanted in Local Area Network was quite long. We should have simply used the old one to get more real experience. Here, a guiding heuristic:

if you haven't lived with a new computing
structure, use an
arbitrary structure in order to get the
experience before trying
to design the ultimate system or standard-- the
standard is much
less important than its existance (will return
to this on UNIX)

While we're on the notion of reality, it is occasionally useful to have models by which new standards may fit such as the 7 layer open systems interconnect model. Here, again we might invoke the rule that implementation is a necessity. Had this happened, there might only have been 4, 5 or 6 layers.... or even 8. Unfortunately, every real implementation that says it uses the 7 levels uses the levels like one uses a metric ruler to draw on 1/4" squared quadrille graph paper. The lines on the graph paper serve only as reference lines for the infinity of figures that one can draw using the ruler. About every 2.5 inches the two scales line up pretty well.

This brings up the notion of the necessity for having a sparse set of standards for two reasons: First having too many standards is like having NO standards at all. The current plethora of LAN standards, including various digital PBXen, which I also call a LAN, is a good example of too many, with no basis of experimentaion. Second, a standard is hard to specify it every detail: I consider the Unibus to be a good standard. It specified a way of interconnecting a whole set of different kinds of parts, not just a pair; furthermore it showed the way for this generation of buses and the future generation of micros. Yet, it took about 8 years after the bus was in operation to have a really complete Unibus specification... even though hundreds of engineers had designed hardware to attach to Unibuses. In this regard, the standard should be understandable in various levels of precision.

A finally role of the responsible organization is evolution. With expoential change in virtually every dimension of computing, changes are necessary. Ideally, the domain of a standard is specified a priori so that one knows when it should be extended. Many standards, such as Fortran, live longer than the sponsor thought or intended them to. As a result, ad hoc extensions occur because everyone makes extensions and no one is responsible. It was felt that Fortran, now about 25 years old was dead, so why evolve it or work on compilers for it? It turns out that many use it and it does pay to really work on it, but that's the final line in this talk.

Finally, standards should be timeless, and failing the test of time should simply remain static and hopefully then disappear. But they rarely can or do.

<>Levels of integration in this generation
Let's get specific by looking at critical standards associated with various strata and product segments. About eight levels of integration form the strata, half of which are hardware. A given level has many product segments, with a given organization ususally excelling in only a few groups. That is an organization has culture and cost structure that constrains its behaviour. Contrast this with the complete vertical integration in the first generation where a computer company designed and manufactured circuits, peripherals, systems, operating systems, languages and applications. Standards provide clear constraints for building products within a given strata and segment such as the Spread Sheet Industry. For example, it is enlightening that data format standards have evolved for these various packages built by different software companies.

SILICON WAFER LEVEL
Rarely do we think of the Silicon Wafer as a level of integration. It is certainly not a well publicized or documented level since processes have been tradtionally been the jewels of a semiconductor company. It is realitively safe to predict that in our fairly near future, perhaps even the real next generation, many of the systems will be a single chip with up to 1 or 10 million thousand transistors. Of course many or even most chips will continue to be "standard" or combinations of "standards" such as microcomputers, peripherals and memories all integrated on a chip or even a large chip.

The creative products will come from the use of silicon using the so called silicon foundry industry that Carver Mead advocates. A good example is a product like the Silicon Graphics IRIS, which uses a ___________dozen? 40,000 transistor chips which Jim Clark calls the Geometry Engine and computes at the rate of ________ Megaflops, which is roughly equivalent to the power of a _______________ computer. One can invision hundreds of these sorts of systems which operate on all kinks of pictures, voice, and mechanisms.

We are nowhere near being able to realize such a scenario with today's state of the silicon foundry, mainly due to lack of standards in foundries and CAD systems. Standards are needed for the various approaches whether gate arrays, standard cell or fully custom chips are used. Let me list a few interfaces:

high level system descriptions

specifications of structure and behaviour, including simulation at all levels

physical information for processing wafer masks (such as CIF)

control for foundry processes, especially if processing steps become optional

chip test, including automatic generation of test data

chip assembly including bonding and multi-chip bonding

We must target the development of standard interfaces to languages and datbases that are communicable via networks. Agreeing on these interfaces doesn't limit the competitiveness or creativity of a given company or product, it simply means that users don't have to spend all their resources in converting among different formats or worry about being locked into a corner. Standards would let users mix and match different CAD systems in a completely flexible fashion. This would still let every vendor build their own editors, timing verifiers, simultators, design rule checkers, etc. but a user could interchange data among the various systems. The market would expand much more rapidly because users could buy without fear of being trapped into a particular system or format. This is completely analogous to the pre-Cobol / pre-Fortran where every manufacturer was pushing different languages. Everyone got sick of the situation and rebelled by designing COBOL.

When going to the foundry and testing folks, the user is faced with an equally fuzzy and perplexing situation regarding the characterization of a process including testing.

To clarify: this is a message to the foundries, CAD companies and failing that to the users to specify what they should be demanding.

STANDARD CHIP: MICROS, MICRO-PERIPHERALS, MEMORIES
The first rule of standards, having a responsible organization is critical and not well understood by all semicomputer manufacturers. Since the Instruction Set Architecture is the bottom level of integration that includes substantially more hardware in the form of busses, boards and systems and goes on to include operating systems and languages, the responsiblity of the semicomputer manufacturer is quite large! I'd like them to acknowledge this responsibility.

The microprocessor is at the root of most of our redundant efforts. A micro's life seems so incredibly predictable, following a time worn path with respect to its ability to accesses memory. A recent article in Computer Architecture News suggested that there are about 20 measures of word length. Only one counts-- the amount of directly addressable memory. Of course there are a few embellishments like data-types when considering performance. In 1976, having lived through a moderate amount of hell in terms of trying to extend the 11 and well along on VAX, it was safe to warn future designers of microprocessors. I certainly did in two papers. They didn't listen.

Unlike semiconductor process evolution, all users are dragged along as one evolves a simple stack idea that started out in a Datapoint terminal, went on to become the 8008, the 8080, the Z80 (by another company), the 8086, 186, 286 and more. As someone who has sponsored using many of these parts, I have been able to relive computer evolution for a third time... and frankly, this is boring as hell!
In the late 50's the folks at the University of Manchester, using Mercury, Ferranti's version of their second machine, developed a system that allowed users to

treat both primary and secondary memory as one. By 1962, the University had a breadboard for Atlas operating with a 27 bit virtual address. Atlas also had a number of other ideas that people continually rediscover, for example, in the last issue of CAN someone reinvented Atlas' Extracodes. Let's call Atlas the 0th time through because it was a university machine and there were only a dozen papers written on it, the critical one was repubished in 1971 in Bell and Newell but it was in the UK, and Ferranti only sold a few.

Having erred in a similar fashion on DEC's early minicomputers by designing two computers which had only 12 bit addresses that immediately had to be extended to 16 bits, I architected the PDP-6, the forerunner of DECsystem 10, with a 20 bit address in 1964. This was concurrent with the 360, which though having a larger physical address, only really implemented a small address... that's why the two versions of the 370 came into existence 10 years later with 24 and eventually 32 bits of address. The DECsystem 10/20 and the 370 eventually ended up with 32 bits of address, complete with paging, just like ATLAS, but about 15 years later. The mainframe was the first time through.

As the PDP-11 came out in 1970 with the goal of solving the minicomputer addressing problem by having a 16 bit address, the first customer demanded a physical address extension to 18 bits. Eventually, the virtual address got to 17 and the physical address to 22. For many years DECs engineering spent thousands of hours trying to figure out how to address more memory. Users spent much more time encoding programs in small memories. In 1975, we finally gave up and built the VAX 32 bit architecture with an embedded PDP-11. Other minis followed essentially the same path for the second time around, but most were on the east coast.

The micro was born on the west coast with the 4004 and 8008 concurrent with the extensions to the 11. These had 12 and 14 bits of address, hence why I wrote the paper on the 11 about addressing in 1975. The leverage of doing it right the first time was very high. Subsequently, the 8086 was extended to 20 bits and most recently to 24 bits of physical and 30 bits of virtual address. It is ironic that information on addressing didn't travel from California to Oregon where the 432 was developed, but then again Oregon didn't become a state until 9 years after California.

Motorola's saga is similar. National took the high road and simply copied VAX without violating the patents. Another tragedy. If an exact copy could have been made several billion dollars worth of software could have been made available! And many resources could have been freed for doing something creative or otherwise productive. Finally, with the micro we have everyone going around three times. The saga is not yet ended as we understand the ramifications of greater than 32 bit address spaces.

There is an equally tragic story about an architecture called CFA, for Computer Family Architecture, which is the defense department's version of VAX. This time, they could have used an exact copy of VAX. Won't our enemies just use US standard micros, and get the parts and software at least 5 years earlier?

With shifts in relative speed and sizes of on chip registers, cache memory and control memory, it looks like a return to simple, CRAY type load/store architectures which are implemented without microcode may perform much faster than architectures oriented to processing the data-types of high level languages such as the 360. Since these so called reduced instruction set architectures trade off microprogram control complexity for compiler technology, it would be well to find and use a single one rather than continual evolution.

New architectures, especially those which have gone along well travelled evolution paths, have cost computing at least half of our resources. The glib answer of using C and UNIX to obtain machine inddependence is deceptively simple and errorneous. A compiler for C or a compiler written in C is only a starting point for a product...

not the end. An architecture pervades virtually every part of a system and database.

When an architectue should be copied, evolved or thrown out and started over is fundamental to the notion of standards because of the tremendous user program and database investments. Let's understand it.

BOARD: BUSES FOR VARIOUS PERFORMANCE, APPLICATIONS, ETC.
The board level is similar to the Instruction Set Architecture story, except that busses have longer lives than specific instances along the evolutionary life of an architecture. For example, the various species of the IBM channel buses are now 20 years old and will no doubt continue for another 20 years in their current forms, even though many of the functions that a peripheral might perform could be handled in the same amount of hardware as that required to interface the bus. The Unibus is almost 15 years old.

The IEEE is in the business of blessing these buses and I don't understand the politics of this process. One manufacturer already has an adequate unibus-type standard to build multiprocessor and multicomputer structures. Does the IEEE support a bus indpendent of whether there are any riders? How many more do we need or can we afford?

LANs and LANCs ANOTHER KIND OF SWITCH
<>Ethernet, the Unibus of the Fifth Generation
While on the issue of busses, this is a fine place to discuss another important switch, we now call a LAN which is used for interconnecting computers and terminals in a local area. This slide is one I used exactly three years ago. Several of us from Intel, Xerox and Digital including Bob Noyce, Dave Liddle and myself presented the case for Ethernet as a standard and to show that we were committed to use it. It was useful because I wanted to convince all of the engineers working on the project of its importance. I attempted to show the need for the bus for building new, distributed computer structures or clusters of computer. Let's call these structures LANCs.

<>Unibus,
Note that this computer structure and the LAN/LANC are nearly identical except they are seperated by 15 years. The unibus is used to build a single computer from constituent information processing components such as processors, memory, communications equipment and terminals. It was designed to travel about 15 meters. Ethernet, or rather IEEE 802.3 allows a user to build LANs and LANCs. It was designed to travel several kilometers.

Digital needed a LAN to interconnect computers into a network and to be able to interconnect terminals to computers in an open ended fashion. I was receptive to using Ethernet as the wheelbase when Bob Metcalfe, its inventor, proposed the standard and consortia to built it. At the time, two experimental LANs were operating within DEC. While Ethernet was proposed as simply a network interconnect the main motivation was a bus for the evolution of two types of clusters: first, a single shared mini or large computer would gradually be decomposed into functional server components; and the proliferation of PCs would require intercommunication into a cluster formed by aggragation.

The key reason for the standard was to allow us to get on with building LANCs, which only a few organizations understand experientially. To reitierate, to propose a standard, one should have lived with it for awhile and really understand it.

In retrospect, getting anyone outside of the three organizations involved may have been a mistake... had we simply built the bus, and offered it as a LAN standard, the process would have been done quickly. Furthermore, instead of engaging in debate about something we knew little about experientially, we could have simply designed and implemented it 2 years earlier. What appeared to happen was that no one knew they needed a LAN, but when they found out one was being proposed as a standard, then everyone had a design to try.

The IEEE tried to help with inventing 802 and now have .3, .4, .5 and .8.   .9 is needed for PABXen and we'll soon need a second digit to add to the new proposals--still LANs and LANCs don't exist to any degree.

802.3 was allocated for the CSMA/CD type, or Ethernet. Since others would like lower cost LANs of this flavor, then several folks took the basic idea and built fully incompatible versions. Alternatively, the same energy applied to cost reducing Ethernet would have made everyone win.

Of course, one would like to have some sort of LAN on broadband, using a token bus technique, so 802.4 was assigned with only 3 incompatible versions.

Another early kind of switch, the ring came out of early work at Bell Labs, Cambridge and other places. Prime built such a ring, and when these folks formed Apollo, took this basic religion with them when they moved 15 miles north. Because rings usually require some form of central controller, IBM grabbed the ring, hence 802.5.

Since one can obviously use fiber optics for building LANs we require a fiber standard, 802.8.

802.3 can be transmitted on standard orange or yellow Ethernet cable; for others who like a simpler installation and will give up distance, RGU 58 can be used if you call it cheapernet Bob Metcalfe calls it thin Ethernet; Codenol has a fiber optic system using the same scheme; for those who like broadband there is a modem. The purpose of all these media is to be able to get users to build LANs and not to wait for what is really quite an arbitrary choice of media, and one which only delays the critical use. Surely someone could take the controller / transceiver interconnect and build a transceiver to operate on a ring structure. I hope this could also be used to adapt to high speed digital PABXen as they become available.

While we're on switching, the forthcoming high speed PABXen will permit the same function as the LAN, and hence should come under the 802 perview. It is imperative to have conformance at the higher levels. Can I suggest 802.9?

These alternatives for standards to switch information at a modern, computer data rate versus a scheme that evolved from the Morse code allows everyone to avoid working on the essential problem of building networks and evolving into clusters. Again, the multiplicity of standards delays the introduction of structures at least five years!

The glib answer to multiple or no standards is gateways. However building gateways is often about as easy as having a single train that can travel on different gauge tracks. It's fine when you reach steady state, it's the transition among track sizes that kills you.

ELECTROMECHANICAL ASSEMBLY: DISKS, I/O, POWER, ENCLOSURES

The evolution of small disks and tapes has been very impressive. I remember meeting Al Shughart at the start of Seagate when his greatest concern was making sure of a competitive second source with the same interface and form factor, which in effect creates a complete industry. This is the same formula that he used in creating the floppy form factors, standards and industries. The standardization process might be understood by these examples.

We also see the effect of edicted, blind standards when looking at the issue of keyboard thickness. The IBM PC's keyboard is designed to pass a particular national standard, but has little legs that are raised that make it comfortable to use. I've never seen one in use without the legs.

OPERATING SYSTEM: COMMUNICATIONS (eg. WAN, LAN), DATABASE, SCREEN

In 1966, a user could have a 300 baud Teletype using a phone line. By 1980 the speed of the common dial-up line had been raised to 1200 baud. This amounts to a a performance improvement of less than 10% per year, and I believe the connect cost rose. This is not the kind of improvement we're used to in computing.

During this period, through computer controlled switching, telephones have gained improved functionality. All of the telephones systems are incompatible with one another at the user level beyond the plain old telephone. All have a relatively large manual and training to get back to the capabilities we had with multiple button phones. The new phones are not user friendly nor do they pass the ease of use test nor are they particularly helpful about adding or deleting the appropriate one except to say that what you've just dialed in is wrong. If a system is knowledgeable enough to always give you the same error message, then it should simply always fix the error.

We have turned a large part of our future systems development over to AT&T for one of the key interfaces by adopting UNIX. In fact, it is the kernel of the system, just above the hardware. When the Justice Department was playing God, why wasn't UNIXCO separated? Maybe it's the only money making part, especially at the price AT&T charges for royalties. Given the simplicity of UNIX, it would seem completely appropriate to install a venture capital offices around the various Bell Labs in order to extricate and form an independent, responsive company to evolve UNIX. Are all you San Francisco based venture folks listening?

The UNIX phenomenon illustrates some principles of standardization and I'm sure we can learn from it. Like all operating systems, the only people who really love UNIX are its parents and those who only grew up with it. This is a large set. It also illustrates a recurring theme of standards:

in order to make forward progress, one has to regress for awhile

UNIX evolved along these lines:

. UNIX came from a reaction by Thompson and Ritchie to MULTICS, the large, joint MIT Bell Labs project of the late 60's. It was written for a DEC mini and evolved to the PDP-11 in the early 70's.

. DEC didn't give away operating systems to universities-- especially the source code; UNIX was essentially free.

. UNIX is by most measures a very simple operating system, to do anything useful requires other programs such as database access, special communications, programming, etc. Students and faculty could understand all facets of its internals and use. It was written in a high level assembly language, C, and as such could be modified. It was an excellent pedagogical tool. Universities embraced it and trained many students with it. A built-in market.

. UNIX evolved to be used on other computers by being transportable. A team of people could carry it to another computer system, provided a C language compiler was available. This was something that early high level languages were supposed to do, but never quite succeeded with due to extensions for calling the operating system. In turn, this created the notion that it might someday be possible to have a complete system that was machine and manufacturer independent. Users like this idea.

. Chip makers and system builders who had no means of building software were able to get a system relatively cheaply. Thus, we have more support and the beginning of a standard. The semiconductor industry knows about standards.

. Much work is required to have a system that supports 80's computing concepts. This is why I worry about the control in a bureaucracy. The extensions include:

.virtual memory. This function was worked out about 5 years ago and has been in operation for at least 4 years in the verion of UNIX for VAX called 4.1! Recall the notion of virtual memory was only invented 20 years ago.

. special functions for real time and transaction

processing. UNIX is being extended and adapted in incompatible ways by diverse organizations.
.human interfaces that are competitive with the PC. UNIX grew up in a timesharing world using dumb terminals. Windowing and fast interaction are critical.
.multiprocessing. With the micro, many companies started up to extend UNIX for multiprocessing.
. networks. Given the origin of UNIX in a communications, we should demand modern communications capabilities.
.fully distributed processing across a LAN to form LANCs. The Universisty of Newcastle, Berkeley and several companies have all implemented incompatible systems for fully distributed processing. Berkely 4.2, which is being distributed is a good starting point.

UNIXco must take the responsibility commensurate with their selling of UNIX as a standard operating system. The notion of a standard is good. But it must be evolved more rapidly than any single manufacturer. It can provided there is parallelism in the development using multiple organizations. If UNIXco is the single company doing and blessing all the extensions, we have simply subsituted multiple competitive companies with a single, behmouth! The system has to be evolved in a reasonable, not ad hoc fashion. I think this is the most serious problem we have in extending computing today.

LANGUAGE: INCLUDING EXTENSIONS TO APPLICATION LANGUAGES
With the very strong concern regarding UNIX, C is a weaker concern. C is at the heart of applications portability. It's time that C be treated like a serious language, complete with standard.

<>picture of ACM september with Japanese and US going toward 5g
All the languages could be enumerated with concerns especially ADA, are important for this next generation. LISP has been proven to be useful for Artificial Intelligence applications. Like the Japanese, I believe

these applications may be the basis of the next generation.

LISP was defined about 25 years ago by John McCarthy while at MIT. I was so enamoured by LISP 20 years ago that I put the critical primitives into the hardware that ultimately became the DECsystem 10... still about the fastest LISP computer. LISP branched and created many dialects. One path went west via BBand N to Xerox, creating INTERLISP. Many dialects evolved from the original MIT LISP: MACLISP, Zetalisp, NIL, SCHEME, TLISP, Portable Standard LISP and now Common LISP the later two are vying for standards status. Virtually everyone who gets inside a LISP compiler or interpreter creates a new language. These languages are incompatible with one another and thus one can't benchmark, or use common techniques to bootstrap extend the language in a compatible fashion. Much work surrounding LISP is to make applications development easier. But given the number of dialects and the number of extensions to make development easier, I wonder if anyone is working on applications. The effeciency for normal development is 0.5 due to redundancy. This is high for AI applications because there is no standard base.

To reiterate, in order to get on with the business of applying AI, we need some way of sharing information across the various different languages called LISP. A serious standards activity is long overdue.

In fact, the Japanese were so confused about LISP that they totally gave up and went to Prolog.

Having extolled standards now for sometime, there's a downside. A standard provides an interface or target by which systems can be compared. Recently, the Livermore Laboratory kernel becnhmark codes expressed in 25 year old Fortran, were run in Japan on the Fujitsu VT100, VT200 and Hitachi 810/820. Using very good vectorizing compilers, all machines ran at a rate of over 2 times a Cray XMP. There is virtue of understanding the old and evolving it.

(DRAFT) COMMENTS ON

(CONVENTIONAL, NUMERICAL) SUPERCOMPUTER DEVELOPMENTS

DRAFT NRC/OSTP BRIEFING DOCUMENTS

Gordon Bell
Encore Computer Corporation
15 Walnut Street
Wellesley Hills, Massachusetts 02181
617-237-1022

SUMMARY

I believe the report greatly underestimates the position and underlying strength of the Japanese in regard to Supercomputers. The report fails to make a substantive case about the U. S. position, based on actual data in all the technologies from chips (where the Japanese clear lead) to software engineering productivity.

The numbers used for present and projected performance appear to be wildly optimistic with no real underlying experimental basis. A near term future based on parallelism other than evolving pipelining is probably not realistic.

The report continues the tradition of recommending that funding science is good, and in addition everything be funded. The conclusions to continue to invest in small scale fundamental research without a prioritization across the levels of integration or kinds of projects would seem to be of little value to decision makers. For example, the specific knowledge that we badly need in order to exploit parallelism is not addressed. Nor is the issue of how we go about getting this knowledge.

My own belief is that small scale research around a single researcher is the only style of work we understand or are effective with. This may not get us very far in supercomputers. Infrastructure is more important than wild, new computer structures if the "one professor" research model is to be useful in the supercomputer effort. While this is useful to generate small startup companies, it also generates basic ideas for improving the Japanese state of the art. This occurs because the Japanese excel in the transfer of knowledge from world research laboratories into their products and because the U.S. has a declining technological base of product and process (manufacturing) engineering.

The problem of organizing experimental research in the many projects requiring a small laboratory (Cray-style lab of 40 or so) to actually build supercomputer prototypes isn't addressed; these larger projects have been uniformly disastrous and the transfer to non-Japanese products negligible.

Surprisingly, no one asked Seymour Cray whether there was anything he wanted in order to stay ahead. (It's unclear whether he'd say anything other than getting some decent semiconductors and peripherals, and to be left alone.

Throughout the report I attempt to give specific action items, and the final section on HOW TO FORWARD gives some heuristics about projects together some additional actions items.

I have commented on the existing report, using its structure because of a personal time constraint. Hopefully, my comments don't conflict too much with one another or are too vague. If they are, I apologize. I would like to rewrite the report to make it more clear and concise. Or in the words of somebody: "I wrote a long letter because I didn't have time to write a short one".

(COMMENTS ABOUT THE) INTRODUCTION

The second two sentences are fallacious and unfounded; from them follow faulty conclusions. Supercomputers aren't that fast today, nor are they increasing in speed rapidly over the last decade. The report lacks substance and detail, e.g. it doesn't differentiate between MIPS, MOPS or MFLOPS and the notion of peak and average. Note these data:

	DF	LLL	Av	Pk	Min	Year	% / yr. increase
Cray X-MP	33	53	150	3	83	8%	
Cyber 205	25	40	80	2	82		
Cray 1	18	38	83	3	75?	32%	
FPS 164	1.3				84		
7600	3.3				69	52%	
6600	.4				64	(base)	
Fujitsu VT200			132	190	5	84	
Hitachi 820		100	240	4.2	84		

DF Megaflops- Dongarra's Double Precision LINPACK
LLL Megaflops- Livermore Kernels of 14 as of Jan. 84

The above data <u>should not</u> be used for conclusions without more basic understanding; it is all I had immediately available. If the Crays run at a much slower then the above average rate, averaged over an entire day, then this would strongly argue for simple, cheap 10 mip machines to front end and offload everything that can't run in a highly parallel fashion.

The committee was very unclear about what kind of operations are desired. Is it having:

.the greatest MIPS for just a few problems and national prestige?

. a much larger number of MIPS for researchers who now get by sharing a Cray?

. or is it simply having some reasonable fraction of a Cray at a much lower cost?

In general, I took the problem to be one of national prestige and having something that computes faster than anything else. On the other hand, if it's to provide lots of effective cycles, I would urge us to terminate all existing, complex architectures used for building microprocessors and to make available a simple, very fast, hardwired processor such as Hennessey's MIPS or Patterson's RISC chip but with floating point and memory management.

It is quite likely that the basic approach to multiple pipelines to increase M (in SIMD) is risky when you look at delivering either more or the most cost-effective operations. Given our poor understanding of multiprocessor for parallelism, much work is needed in order to get anything reasonable out of a multiprocessor, let alone a multi-processor, multi-pipelined machine. Based on large differences among peaks and long term averages, much basic and applied work in compilers needs doing now; as such research is required.

ACTION: Data suggests there are at least these problem areas:

. understanding about existing machine performance,

. fundamental work on compilers to utilize current pipelined machines (especially for non-floating point work), and

.alternative machines and structures to get around what appears to be poor utilization of expensive resources. (Here, I think several startups may be addressing this.)

ACTION: Given the Fujitsu and Hitachi are IBM compatible and should perform very well for a more general load, particularly ones requiring a large virtual memory, I believe we should urge the National Labs to take delivery of one of these machine at the earliest possible time in order to proceed with this understanding. Time should be available for computer science.

RISKS IN PREDICATING THE FUTURE OF SUPERCOMPUTERS ON PARALLELISM:

While I concur, the report is unconvincing because results to date are sparse. Note:

<u>Existing Pipelined Computers</u>. It would appear that fundamental work is still required in order to design and

exploit these computers, especially when multiple pipelines are used.

Real, Experimental Machines. The only, experimental evidence for parallelism (that I'm aware of):

. C.mmp and Cm* multiprocessors at CMU showed that many problems could be solved giving near linear speedup, but NO general results were obtained; Several multiprocessors are entering the market (Dennelcor, Elexi, and Synapse), and many more are coming, based on the commodity micros. Clearly the Dennelcor machine should have produced some useful results to demonstrate parallel processing; I know of now.

. Manchester's Dataflow machine works for a few "toy" problems that were laboriously coded; I am unconvinced that general purpose Dataflow Machines will provide high performance-- i.e. be useful for supercomputers. I am completely convinced that it will NOT be cost-effective. Dataflow structured hardware may be the right way to control signal processors! It may be possible to use a Dataflow language to extract parallelism for pipelined, multiprocessors and multicomputers --- but alas, NO ONE is working on what should be the first thing to understand about dataflow!

. Fisher, at Yale has a compiler that can exploit the parallelism in array processors; He is continuing, by building a machine along these lines which he believes will provide parallelism up to 10 using a single, wide instruction to control parallel execution units. The work is convincing--and he may have a reasonably, super, computer.

. IBM built the Yorktown Simulation Engine and showed that logic simulation can be run with a special purpose multiprocessor oriented to simulation; and

. Fox and Seitz built a 64 computer Hypercube which has been used for various physics applications. This looks extremely promising because the machine hardware is so trivial. Larger machines are in progress. We need to understand its general applicability.

"In Progress" Machines That Promise Great Parallelism. These include:

- MIT's Connection Machine being funded by DARPA and built at Thinking Machines Corp; This is a fascinating SIMD machine that has 64K processing elements with extensions to IM. While originally designed for AI, it appears to be suitable for arithmetic calculations.
- Systolic Array Processors; Several machines are in progress, including one by Kung. It is unclear whether a systolic organization of a dozen or so pipelined processing elements can either be controlled (programmed) or have a rich enough interconnections structure for more than a few applications.
- MIT Dataflow Projects; The whole dataflow area needs review.

Inoperative or Poor Experimental Machines. There are at least twice as many machines which yielded either poor or no experimental evidence about parallelism. Some are published, but few describe the failures so that others may profit from their mistakes. Some that are continuing should be stopped to free valuable resources!

Conjecture Machines. There are at least a factor of ten more machines that are irrelevant for anything other than tenure and miss-training graduate students.

Especially distressing is the work on large scale and ultra large scale multiprocessors with thousands of processors because we have only sparse data and no understanding now of whether multiprocessors really work. Resources are needed to work on both the general problem and specific applications involving a dozen to a hundred using existing machines. We can always build a 1000 processor system if we can find out that they "work."

(1) PURSUE ALL DESIGNS LIKELY TO SUCCEED IN ANY BIG WAY
This simply is and can not be implemented. We have two cases:

- our potential talent is being wasted on examining structures that look interesting because they can be built using VLSI; and
- we are not working on the structures that must be built and understood, or those which we have but don't understand well enough to apply broadly.

Poor Work. There's probably no way to outlaw or manage poor work, but funding for it could be stopped. The only reason to worry this is that there's so much real work to do! I would like to take a budgetary "chain saw" to cut tree, grid, and other partially connected structures, as well as banyan and perfect shuffle switches etc. that claim to provide anything useful for computing. None of these have either systems software or applications understanding behind them; they are only interesting because they may some day be buildable and are publishable. This work (similar to associative memory research) yields about 10 to 20 micropapers per research dollar with absolutely no use for any future (10-20 years) timeframe. The work can be easily re-invented anytime, by anyone and usually is in every 5-10 year increments.

Potential, Good Work. Supporting a major supercomputer project within a university or government laboratory across hardware, software and systems applications has shown to have been impossible. A major, large project of this type requires on the order of 30-40, focussed, well-led researchers and engineers. The machines are important to build; universities have many of the "right" people to build them but lack leadership, hardware and software engineering discipline, skills and facilities to build them. Companies have few people with the vision (willingness to accept risk), or ability to do much of the research to carry them out. A combination of the two institutions is somehow needed. The IBM-CMU Development Laboratory is one interesting experiment for building large systems. Also, Entrepreneurial Energy, released by Venture Capital may be an alternative way to carry out these projects... but Venture Capital alone is very unventuresome.

Great Individual Researcher or Small Team Work. Universities are incredibly cost effective for building systems where a single professor or group can work on a project with a dozen or so students. The work on non-microprogrammed processors (RISC and MIPS), Cal Tech's Hypercube, the SUN Workstation forming SUN Microsystems, Clark's Geometry Engine forming Silicon Graphics, the LISP Machine as the basis for Symbolics and the LISP Machine Company, Scald as the basis of Valid

Logic, etc. are all examples of this kind of work.

Nearly ALL of the great ideas for modern CAD on which today's VLSI is based seem to emenate from the individual professor-based projects (Cal Tech, Berkeley, Stanford, MIT ... Mead's VLSI design methodology, the silicon compiler, Supreme, Spice, etc.). This software has either moved directly to use (eg. Supreme, Spice) or been the basis of a startup company (eg. Valid and Silicon Compilers) to exploit the technology.

ACTION: I would like to limit poor work, fund the great work in small projects where results and people are proven, and find someway to address the large projects where past results have been almost universally disastrous and poor. It is essential to get small projects surrounding large systems; these are likely to produce very good results.

(2) GET DESIGNS INTO THE HANDS OF USERS NOW
ACTION: I concur. We need to immediately engage in working (experimentally) on parallelism at the systems software and applications levels right now, using real, existing computers. Both multicomputers (the Seitz-Fox Hypercube) and multiprocessors (Dennelcor, Elexsi, Synapse) can be placed in universities almost immediately to start this work.

(3) ACCELERATE COMMERCIAL DESIGNS INTO PRODUCTION
Several methods can be used to accomplish this provided there is anything worth producing:

- .Great, Individual Researcher doing seminal work (works well)
- . Cray-style Laboratory (untried, except by Cray)
- . Large project in small scale research environment (typical, but poor)
- .NASA-style Project (multiple, interconnected projects) (used effectively by DARPA for very well-defined, focused projects... requires a prime contractor)
- . Consortia of multiple companies or universities (current fad)
- . Industry-University Partnership (on premise or dual labs) (Could be effective, provided universities permit them.)

The committee could have examined these alternatives and developed some heuristics about the kind of projects that are likely to be successful based on real data about past work.

(COMMENTS ON) SPECIFIC TECHNOLOGICAL AREAS

The report examines the constituent technologies for supercomputers in a less than quantitative, friendly, fashion. The US has only one, unique resource for building supercomputers, Seymour Cray; hopefully the ETA Lab of CDC will be a backup. Without him, supercomputers wouldn't exist. In order to provide backup to the well funded, well organized, super hardware technology based Japanese efforts, much fundamental AND applied work is required, to be followed by exceptional hardware, software and manufacturing engineering.

CHIP TECHNOLOGY

U. S. chip technology available through conventional semiconductor companies and computer companies outside of IBM doesn't appear to be relevant to supercomputers. Chip technology lag with respect to Japan is increasing because all major Japanese suppliers are working hard across the board in all technologies, including significant efforts in sub-micron research. Note:

. basic CMOS for RAM and gate arrays; Japan is several years ahead because suppliers were slow to make the transition from NMOS to CMOS. America's only serious gate array supplier, LSI Logic is Toshiba based.

. high speed circuits based on HEMT, GaAs and conventional ECL; The Japanese continue to increase the lead in today's ECL gate array circuits, and they continue to build and describe the highest speed circuits (ISSCC).

. state of the art, microprocessor peripherals; While not directly relevant to supers this does indicate the state of the art. Many of the major chips are designed in Japan such as the NEC graphics controller for the IBM PC.

. conventional microprocessors. These are dominated by U. S. "Semicomputer" manufacturers quite likely because the Japanese are unwilling to make the investments when the leverage is so low. These architectures are clearly wrong for today's systems. All manufacturers need to abandon their current architectures! This would provide much more scientific operations than any supercomputer effort.

. Computer Aided Design of VLSI. This area has been developed by U. S. Universities. The programs move rapidly across all borders, creating an even more powerful industry in Japan. This work aimed at small systems could be extended for supercomputers.

ACTION: It is heartening to see real research being carried out at the chip level now by Berkeley, MCNC, MIT, and Stanford. Unfortunately, all of this work is aimed at lower cost systems. A U.S. supplier of high performance chips for supercomputers is needed.

PACKAGING

Packaging is vital for supercomputers. Cray's creative packaging has been in large part, the reason why his computers remain at the forefront. IBM is able to fund the large "Nasa-style" projects for packaging large scale computers, but it is unclear that this packaging is suitable for building supercomputers. It clearly cannot be used outside of IBM. Hopefully, Cray will come up with something again.

ACTION: With the demise of Trilogy's Wafer Scale Integration, we have lost the possibility of a major lead. If it is important to have Wafer Scale Integration, we should encourage Trilogy to work with the Japanese. If we are concerned that Cray's next package is inadequate, then an effort should be considered.

ELECTRO-OPTICAL TECHNOLOGIES

The report omitted this important area. This offers potential both for computation and for interconnections.

DISK TECHNOLOGY

The report fails to acknowledge the fact that the U. S. is only leading in the production of low cost, 5" winchester disks. A recent, innovative U. S. designed disk is in the process of being transferred to Japan for manufacturing because U. S. manufacturing technology is lagging.

ACTION: The Fujitsu Eagle provides the greatest real density. Only a few universities such as the University of Minnesota are doing fundamental work in magnetics; the Japanese have graduate students in these labs. We simply need research in magnetics to regain the lead coupled with a major effort in manufacturing engineering.

ACTION: It was heartening to learn that a very high speed optical disk would be available in the next few years. However, one should point out that the current optical disk was invented by MCA, but had to be taken to Europe and Japan for manufacture. My skeptical guess is that it is just a demonstration, like the vast number of past demos by large, military contractors whose main goal is funding, not science or products. If, indeed there is a "breakthrough" optical disk, then we should make every effort to support and exploit it.

CURRENT AND NEW COMPUTER ARCHITECTURES

The first statement of this section claims:

> "During the next five years, new types of machines based on large-scale parallelism ... thousands ... will appears along side today's high end computers ... these machines will progressively supplant present designs in some market sectors."

This statement finally destroys any of the report's remaining credibility through wild optimism. Although I strongly believe in an emerging class of multiprocessors and multicomputers using possibly a few hundred processors, I don't see this as a serious alternative to supercomputers for general purpose computation in the next 5 or 10 years.

The section on New Parallel Supercomputer Designs leaves open the door to magic whereby with a small amount of intellectual work, one obtains vast payoff, leaving the "competitor" surprised. Unfortunately, any design of this type that emerges quickly is easily replicated, and either doesn't work at all or works for only a limited set of applications. There are no easy roads to success. All the parallel machines are going to be tough to build and program. Selected, special purpose function machines can be useful and should be encouraged as an alternative path to understand generality. For example, a dataflow controlled multiple (array) processor may be able to deliver vast amounts of ops or flops, but for selected functions.

SPECIAL PURPOSE COMPUTING

The ability to rapidly construct systems in silicon may be the best way to provide cost-effective solutions to a wide range of problems. This process, I call VLSIzation is coming along nicely, but should be better understood and extended to high performance technology.

COMMUNICATIONS TECHNOLOGY

LANs will cause higher speed LANs to be required. With the widescale availability of Ethernet, hopefully other standards that operate an order of magnitude faster will be forthcoming. These will require significantly better interfaces between the link, processor and various levels of system software. In affect, this is "spatial parallelism" or distributed processing.

Clearly we need an upgraded ARPAnet to carry large files and videoconferencing.

ALGORITHMIC AND SOFTWARE ISSUES

This section of the report underestimates the need for changes to languages to express parallelism, including the possibility of using a Dataflow language. Also, neglected is the possibility of using expert systems for organizing problems for parallel execution... but in order to accomplish this, we need much more experience.

EFFORTS ABROAD

We need to couple into the British for ideas and fundamental advances, and could couple to the Japanese for semiconductors, other hardware technology, manufacturing, and engineering.

ARTIFICAL INTELLIGENCE ISSUE

This effort needs to be co-ordinated so that the fixed resources capable of building fast systems can be effectively employed toward either or both numeric and/or symbolic problems. It should be pointed out that the Japanese manufacturers are well along in having both fast and low cost LISP Workstations.

HOW TO MOVE FORWARD (Bell Heuristics)

In general, I believe it is important to:

1. Narrow the choice of architectures that are to be pursued. There are simply too many poor ones, and too few that can be adequately staffed.
2. Fund only competent, full-time efforts where people have proven ability to build hardware and software systems. These projects should be carried out by full-time people, not researchers who are servicing multiple contracts and doing consulting. New entrants can spend a year or two to demonstrate competence by actually building something!
3. Have competitive proposals <u>and</u> projects. If something is really an important area to fund in the first place, then have two projects with forced intermediate progress information exchange.
4. Fund balanced hardware/software/systems applications. Doing architectures without user involvement (or understanding) is sure to produce useless toys.
5. Recognize the various types of projects and what the various organizational structures are likely to be able to produce.
6. A strong infrastructure of chips to systems to support individual researchers will continue to produce interesting results. These projects are not more than a dozen people because professors don't work for or with other professors very well.
7. There are many existing multicomputers and multiprocessors that could be delivered to universities to understand parallelism before we go off and build really large multi's.
8. It is essential to get the Cray X-MP alongside the Fujitsu machine in a computer science setting in order to understand the two approaches and also to work on the parallelism associated with multiple processor, multiple pipeline machines.
9. Build "technology transfer mechanisms" in up front. Transfer doesn't happen automatically. Monitor the progress associated with "the transfer".

ELECTRONIC GENERATION

950, the computer era had been established: at least seven ons had announced their intent to build computers -- Zuse AG, Elliott Brothers, Ltd., J. Lyons and Co Ltd., UNIVAC, and IBM e ERA 1101 was on the marketplace. (Science Museum, 1975)

stry itself and its leaders had been changed by the ical advances of the war period. Goldstine states:

In my opinion, it was Thomas Watson, Jr. who played the ole in moving IBM into the electronic computer field. When he out of the Air Force in 1945 his experience as a pilot had ently convinced him of the fundamental importance of ronics as a new and prime technology for our society. He fore exerted considerable pressure on IBM..." (Goldstine, p. 329)

COMPUTER

LGP-30 - Librascope General Precison Computer (X14.81)

Word Length: 31 bits, including a sign bit, but excluding a blank spacer bit

Memory Size: 4096 words

Speed: .260 milliseconds access time between two adjacent physical words; access times between two adjacent addresses 2.340 milliseconds.

Clock Rate: 120 Khz

Power: 1500 Watts

Arithmetic element: Three working registers: C the counter register, R the instruction register and A the accumulator register.

Instruction format: Sixteen instructions using half-word format.

Technology: 113 vacuum tubes and 1350 diodes.

Number Produced:

320-490

First Delivery: September, 1956

Price: $47,000

Software: ACT I (Fortran type compiler)

Successor: LGP-21

Achievements: With the Bendix G-15 the first of the desk-sized computers offering small scale scientific computing. Revolutionizing the computer industry with the potential for low-cost distributed processing.

The Maniac

"The Maniac", Los Alamos Scientific Laboratory, 1957, Color, 3/4" videotape, 29 min. running time (V5.81).

This 1957 production describes the MANIAC computer's architecture and operating principles for a general audience. The Los Alamos-designed machine features cathode ray tube memory and binary-coded-decimal input by punched paper tape.

COMPONENTS

LOGIC MODULE

Deuce Arithmetic Logic Element, English Electric Co, 1955, Gift of Professor Murray Allen, University of New South Wales (D4.75).

IBM 650 Logic Module, IBM, 1955, Gift of Professor Murray Allen, University of New south Wales (D12.75).

G15 Logic Module, Bendix Computer Corp, 1955, (D109.80).

READABLE & WRITABLE MEMORY

WAVE STORAGE

CYCLIC

Mercury delay line.

Mercury was used to propagate an acoustic wave and hold information. A two meter tube held about 1000 bits, with a delay time of approximately one millisecond with a bit separation of about one microsecond or two millimeters. Early computers such as the Pilot ACE, EDSAC, and Bureau of Standards computers used both long and short delay lines.

<u>Deuce Mercury Delay-line</u>, English Electric Co, 1955, Short register, 64 bit, 64 microsecond delay line. Gift of Murray Allen, University of New South Wales (D3.75).

ELECTRIC CHARGE

RANDOM

<u>Maniac Electrostatic Memory & Williams Tube</u>, Atomic Energy Commisssion, 1949, Gift of Dale Sparks, Los Alamos Laboratory (D214.80).

MAGNETIC FLUX

RANDOM

<u>Illiac 54x128 bit Core Memory</u>, Gift of Clifford Carter, University of Illinois (D19.75).

? <u>RCA Selectron Tube-from JOHNNIAC</u>, RCA, 1950, Gift of John Postley (D215.80).

One of forty RCA Selectron tubes installed on the Rand Corp JOHNNIAC Computer in 1950. The tubes constituted the 256 word 40-bit memory that operated the machine. In 1954 a 4000 word magnetic core memory replaced the tubes.

? <u>Mark IV 64 bit Magnetic Shift Register</u>, Aiken-Harvard, 1944, Gift of Bob Trocchi (D6.75).

March 23, 1981

Arthur W. Burks

University of Michigan
Department of Computer and Communication Sciences
105 South State
2076 Frieze Building
Ann Arbor, MI 48109

Dear Art:

Thanks so much for your slides. Do hope you are planning to dub the ENIAC film with a sound track for the Museum -- at our expense. If this is impossible for you, perhaps you could do a script that we could then prepare. Whoever speaks, we won't have a professional media-type.

Could we set a firm date for your Museum lecture next winter?

Thanks again.

Sincerely yours,

Gordon Bell
Vice President,
Engineering

GB:swh
GB2.S4.7

Selected University-Based Computers

to Results (in addition to
l engineering and scientist
e training)

7 Use,

6 Use,

7 Use, stored program, Electronic Computer

6 Use, circuits , core memory, real time and interactive computation, proto for SAGe system

7 Use, proto for 6 others

Use, transistor circuits,

large core memory

3 Asynchronous logic, design too conservative

3 Use, parallelism (algorithms), accelerate bipolar memory develop, stimulated competitive approaches

7 Parallelism, Intel 432 proto

6 Parallelism, Multibus-type structures,

1 -

GB8.21

GENERATING COMPUTER GENERATIONS

DEFI mark generations of computing devices by four factors:

one - an identifiable new machine structure

two - a

new physical technology,

three - meeting new needs, and

four - a new level of use.

Generations are evolutionary, with family trees that can be followed. A revolutionary change -- such as the computer itself -- marks a discontinuity and the start of a new set of family trees.

1945 marks the end and beginning of two eras. Up to that time -- with few exceptions -- only calculating devices were made. From 1945 several centers were established that had the technology and developed stored program, general purpose computers.

MODELA simple model of the process includes all four factors. The "perceived needs" by society generally are at some level above that of the existent technology. Man's imagination seems always one step ahead; and newer, higher level needs are always on the horizon so that we are locked into a continual quest. Technology is defined as the way that groups provide themselves with the material objects of their civilization. The lightbulbs floating between the levels of technology and perceived need represent the ideas of inventors and knowledge from science. In each generation, a number of isolated ideas precedes the actual project that upgrades the technological level and begins to fulfill the perceived need. Prior to the ENIAC project, Charles Babbage, John Vincent Atanasoff, and Alan Turing had the major ideas for computing machines. After ENIAC the whole process of generating computer generations began.

OP RATEThe project pipeline starts with discover, goes on to prototype construction of the principle, construction of some sort of working system, manufacturing, evolution of manufacturing processes, enhancement, possibly hybridization, and finally, most likely, replacement. This pipeline may extend through several generations -- and replacement may not take place until several orders of magnitude improvement can be realized. For example, many second generation computers operated through the fourth generation. Replacement often takes several orders of magnitude improvement, especially in periods of rapid change.

GENERADuring the 400 year, 10 generation period from 1600 to 2000, the technological change is roughly a factor of 10**12. Using the product of processing rate and the memory size to measure computing power, then the computer has evolved almost 20 orders of magnitude since stone-based manual, single register devices supplemented fingers and toes for counting and arithmetic.

TAB IThe four factors marking generations are listed for each of the pre-computer and computer generations. The generations are named for the predominant technology of the time. The key invention for each generation, thus precedes that generation in time. For example, the simple mechanical calculators of Pascal and Leibniz were invented in the "manual" generation from 1600 to 1800, then produced in the mechanical generation when the technology could support manufacture ... and hence use. Similarly the basic electro-mechanical inventions were made prior to 1890 when this generation starts. For each generation, the perceived needs, actual machines, and typical uses are listed.

ABACUSThe abacus, a simple calculator, started well before my categories of pre-computer generations. It is such a good idea and simple device, that it has been claimed to be invented in Egypt, the Roman Empire, and China.

.Lesson One: If it is a good idea, then everyone will take credit for it. The original Chinese abacus represents up to 15 in a digit with a combination of 5 and 2 beads; It is similar to what computer engineers invented several times and call the bi-quinary system.

SOBOBANUltimately the Japanese refined the abacus, first using 5 and 1, and then 4 and 1 beads for lower cost and faster operation.

Lesson Two: Any basically good idea can be evolved.

SOR/CALThis 1979 Casio calculator/soroban is ideal in several ways: low cost storage of a second number is provided; simple operations can be done traditionally and more rapidly on the soroban; users can be gradually trained on the new machine without losing any traditional computational capability; the market is larger; and a culture is preserved.

Lesson Three: Compatability is important for a transition machine.

BONES/SORThe beginning of the pre-computer generations, 1600, is marked by the development by Napier of his bones or rods. Based on the table look-up method these can be thought of as the first pocket calculator. Napier, who is more famous for inventing logarithms, knew the value of the bones when he stated that they permitted the calculation of multiplication to be "free of slippery errors."

.Lesson Four: Over the years, a hand-held, personal, general purpose computing aid has been a perceived need. Now people just want greater computational power in their pocket.

LIEBNIZIn the sixteen hundreds, Leibniz designed and built one four-function mechnical calculator. He had a vision of the use of the machines calling them his "living bank clerks". The one he built worked in principle, but not in fact. The mechanical technology for constructing it did not exist.

DRUMTwo hundred years later, in the mid nineteenth century, Thomas used the stepped drum principle of Leibniz in conjunction with a simple system of counting gears and an automatic carry in making a commecially successful machine.

THOMASFor all this, Thomas' machine met with skepticm from the scientific establishment. In 1849, SCIENTIFIC AMERICAN wrote, the Thomas machine "is said to be one of the most astonishing pieces of mechanism that has ever been invented, but to our view, its complexity shows its defectability."

TATESBut subsequent manufacturers streamlined the machine and continued to make and sell them into the twentieth century.

.Lesson Five: Ideas come a generation ahead of their practical application and use.

.Lesson Six: Three kinds of people can be identified in the process: IDEA GENERATORS, UNDERSTANDERS OR EXPLAINERS (usually academics who can often be wrong if looking to the future and not the past), and ENTREPRENEURS, who form and maintain the industry. Seldom does anyone ever cross over; and as we read some biographies many people become bitter over this factor.

BABBAGEBabbage must be looked at as an idea generator. The idea of the computer can be traced directly back to him. In the mid-nineteenth century he was determined to build machines to calculate tide and navigation tables for the navy.

DIF ENGHis own ideas were racing far ahead of the technology. He left his first machine, a difference engine, barely half-finished and certainly not working, in order to build a better, more powerful machine -- an analytical engine.

SCH ENGLater Scheutz took Babbage's ideas and built a working difference engine, for which he received some acclaim, to the annoyance of Babbage.

.Lesson Seven: Don't be concerned if someone else takes your half-finished idea and perserveres to make it work. ... or even gets rich manufacturing it.

LOOMBabbage himself freely used ideas of others. The Jacquard card-driven loom gave him inspiration for program storage sequencing machine control.

BAB CARD.Lesson Eight: Freely borrow ideas and technology from other mechanisms and disciplines. The converse is also true: Don't keep industrial cliches around.

HOLLERITHThe card which was the savior of the 1890 cenusu became so tied to some corporations approach to computing that they could see no alternative methods for input or output. When the 80 column card was on the way out, true believers in card computing invented a 132 column card.

PANTOGRAPH.Lesson Nine: Beware of the dinosaurs that are created as a last gasp to extend a dying species. Somehow I continue to see larger and larger beasts created on a small bone structure (or architecture) just when a technology should be let go. These drain resources and quickly become extinct.

ANALYTIC Babbage's ideas provided computing with a significant goal for over a century. Although there were periods when Babbage's work was not commonly known, enough people knew what he had attempted so that that knowledge was never totally lost. He was certainly never acclaimed during his lifetime to the extent that we acclaim him today.

.Lesson Ten: If you have the foresight to provide a century-long goal, you can't expect acclaim during your lifetime. Few people will understand you.

MARK INot until the forties, when Professor Howard Aiken at Harvard who used Babbage's analytic engine as a template for his Mark I was the machine begun to be understood. IBM funded Aiken and also worked on another version of it, their Automatic Sequence Controlled Calculator. That is, from what I can unravel these machines are versions of Babbage's analytic engine.

ANAL ENGBut Babbage never finished the analytic engine now did he clearly and cleanly annotate it. Scholars are still unravelling and learning from Babbage's notations. At least Babbage was a prolific writer and did speak and write about his machine designs. The primary effect was not for people to steal his ideas -- but to applaud Babbage as an interesting thinker.

VON NEUAt present, among the computer historians there is controversy over the roles of Eckert, Mauchly, and Von Neumann on the idea for the stored program computer concept. When Von Neumann joined the project group the machine design had been set but virtually nothing had been written down, although meetings were recorded via wire. Von Neumann started to take minutes and wrote these up in consolidated form, as a document that became distributed as THE EDVAC report carrying his name. This report gave VonNeumann what some people think of as an exalted role as a computer pioneer. But the EDVAC report like Babbage's papers and books were probably were critical to the development of others ideas, than the actual built machines.

.Lesson Eleven: If you want your ideas to be used and understood, then clearly document the design intent and the details.

BABBAGEWhen Babbage was not trying to push back the limits of technology, he was trying to generate funds from the government, friends, and various agencies. He tried everyone's patience by not completing any projects or producing any results but promising the "fantastic" if only monies were available for the next machine.

.Lesson Twelve: If building an operational machine is important, then it takes three ingredients:

a steady supply of funds,

useable technology, and

the machine design.

Two of the three is not enough. And having only one of the three -- only the machine design as Babbage had -- dooms a project to failure.

TABLE IIBetween 1833 when Babbage was working on the analytic engine and 1945 when the first large scale computers operated, the technological base was consolidated, and the war created a super-need to generate large-scale funding. In Britain, Turing was involved in electronic cryptography that provided the impetus for some of their early computers, especially the National Physics Laboratory Pilot ACE, and the Manchester machines. The four U.S. efforts that I'll discuss were funded by various parts of the war department. These are the Harvard Mark I designed by Howard Aiken with Clair Lake of IBM doing the engineering; the Bell Lab machines under the direction of George Stibitz and Charles Andrews; ENIAC under the leadership of J. Prespert Eckert and John Mauchley with John VonNeumann consulting; and Project Whirlwind with J. Forrester at the helm.

.Lesson Thirteen: Although large-scale funding was made possible, these first generation, path-breaking machines were driven by individuals; later and larger machines needed larger teams to design and build them. As a corrollary, most of the early computers were developed by a pair of people: an idea-generator and an engineer.

MARK IDespite the fact that the IBM engineers who built Harvard's Mark I had good backgrounds in relay technology, the design was fundamentally a copy of Babbage's mechanical machine, with some electromechanical control. It had 23 digits, 72 numbers for primary memory, other storage and tape for program control. Operations varied from a one-third second add, 6 second multiply, and 12 second divide.

MARK IThe machine took about 8 years to develop and ran from 1943 until 1959. It was the last machine that you could hear the mechanisms working. There was some controversay as to whether Mark I was worthwhile. Comrie stated, "It is disappointing to have to record that the only output of the machine ... consisted of tables of Bessel functions. . . . If the machine is to justify its existence, it must be used to explore fields in which the numerical labor has so far been prohibitive."

AIKENAiken estimated that the Mark I was equivalent to 100 dest calculators. He predicted, "If all 3 to 4 machines currently under construction worked, it would saturate all conceivable need for computing."

.Lesson Fourteen is clear: be careful about predicting the ultimate computer. With every computer, new applications emerge commensurate with exponential machine population and capability growth.

MK I PrAiken went on to build advanced versions of basically the same machine: The Mark II, a relay computer in 1947, and an electronic machine in '50. The ballistic benchmark took 12 hours on hand operated calculators; Mark I , 2 hours; and the 1950 machne, 15 minutes, the same as the differential analyzer. The grand ideal of Babbage had climaxed.

BTLIn contrast to the Mark I, the first Bell Labs computer made excellent use of the available relay technology. Although similar to the 1920 Torres calculator, George Stibitz produced the 1939 machine design independently.

STIBITZStibitz built within the design constraints and concerns of Bell Labs. Their perceived need was to build a highly reliable machine using known, working technology, that is a lot of telephone relays. The 1939 machine was the first calculator that could do complex arithmetic and operated via Teletypes in an interactive fashion. It could also be operated remotely and in a shared fashion with the first demo run between Dartmouth College and Bell Labs in New Jersey.

BTL 5The concern was relability with exhaustive checking and diagnostics. Stibitiz' approach was to use what was available to solve the problems at hand and get the job done. The 1944 Bell Labs machine had these specs: a .3 second add and 1 second multiply for seven digits. The machine ran 20 minutes on the ballistic benchmark.

BUSHIn the thirties, differential analyzers were widely considered as signficant computing devices. Both Vanevar Bush and George Stibitz were skeptical about the activities surrounding the building of the ENIAC at the Moore School.

ATANFrom our understanding, they were unaware of the work of a physicist at Iowa State, John Vincent Atanasoff who described the difference between analog and impulse, i.e., digital, computation...and probably invented the phrases analog computer and impulse computer. He did invent the notion of direct digital computation, and considered that differential computation was a blind alley.

MONROEAtanasoff was driven by the needs of his physics students in solving 34 simultaneous linear equations using electric Monroe calculators. He had reduced the solution of partial differential equations to an interactive set of linear equations.

ABCAtanasoff worried the problem of building a calculating machine through the early thirties, and in 1937 specified a serial computer with a serial regenerative memory using an electrostatic drum.

DRUMThis electrostatic drum is the only part of the machine that is left. With Clifford Berry doing the engineering, Atanasoff built a working machine that operated until 1942. During that time he gave a paper on it at the AAAS meeting and fully shared his ideas. In 1940, Mauchly visited Atanasoff for three days, saw his machine and looked at his circuit diagrams.

E & MAt the time of Mauchly's visit, a differential analyzer at the University of Pennsylvania was serving as the model of computing. Eckert and Mauchly worked on mechanical control for improved function generators, but they clearly understood limits of mechanisms and analog computation. Although Eckert claims to have invented the digital differential analyzer, the idea was abandoned because each order of magnitude required an order of magnitude increase in speed, something that Atanasoff had previously described. Eckert and Mauchly had the dream of a large-scale significant computing endeavor and ultimately, Herman Goldstine funded the project to compute firing tables for the Army.

ENIACThe magnitude of the effort and the skepticsm of the scientific establishment (except for Von Neumann) led to a number of delays, with the result that the machine did not run until the war was over. ENIAC with a 200K hz clock was roughly 500 to 1,000 times faster than relay machines. Adding took .2 ms; multiplying 3 ms, and division of 10 digits 30 ms. It had 20 accumulators and three function tables of 104 values. It held temporaries in relays with card i/o.

Reliability was an issue on ENIAC. It contained 18,000 vacuum tubes each with a predicted 500 hour life. Nevertheless, Mauchly was unconcerned. He reasoned that even if ENIAC only ran a few minutes it would accomplish more than they slow relay machines. Fortunately, the 500 hour life was underestimated, otherwise the exponentially increasing repair time for multiple tube failure would have bootstraped the machine to its death, that is, if it ever lived.

ENIACThe results are mixed as to ENIAC's reliability. For example, all problems were run twice to insure accuracy. Franz Alt, commenting on the 40 plugboards and extensive cabling, estimated that the overall effective rate was five percent uilization. Goldstine used a different metric, observing that there were only three tube failures per week, giving a tube failure rate of about one million hours that was achieved by derating the filament and plate. Thus, even considering the amount of time the machine ran, it would still be 25 to 50 times faster than the relay machines. The fact that such a large system ran was a tribute to significant engineering, mostly on the part of Prespert Eckert.

MET TreeEckert in the historic tapes produced by the Science Museum, London, describes how the stored program computer came about. Various priced memories were available and Von Neumann coined the phrase "memory hierarchy." The ENIAC team speculated that it would be very difficult to determine how much memory should be avilable for various kinds of data, functions and programs. This led to the notion of a common memory pool. But it couldn't be implemented because primary memory was not adequate. Mercury delay lines, magnetic drums and storage tubes were subsequently developed. The mercury delay line holding regenerated shock waves is the exact dual of Atanasoff's electrostatic drum holding regenerated electronic charge.

The effort surrounding ENIAC led to the stored program concept as embodied in the EDVAC draft report written up by VonNeumann. The EDVAC was, of course, to be the successor to ENIAC. Eckert and Mauchly, like Babbage, throughout their careers in computing were often after the next better, faster, more powerful machine before they really realized the full potential and had all the bugs out of the machine they first built. This holds in reading their history in developing BINAC and then the UNIVAC series. Eckert and Mauchly left the Moore School and the ENIAC project during the infancy of the machine; shipped the BINAC without really making it work and following through on the idea; and never lived with their systems long enough to make them great ... the next machine was too enticing.

.Lesson Fifteen. Don't only design the machines, document them, build them and then use and understand them.

EDSACMaurice Wilkes, who took the summer course on the ENIAC at the Moore School, returned to Cambridge University and built and programmed the EDSAC. He kept on with this successful venture which included the invention of microprogramming, but that's another story.

MANCHIronically, the first operational stored program computer did not result from the ENIAC project or the EDVAC report, but from the work of Sir Frederic Williams, the inventor of the electrostatic storage tube, and Professor Tom Kilburn. This was the Manchester University Mark I, the prototype for their large-scale machine, MADM, built to test the electrostatic memory. The efforts at Manchester produced five innovative, influential machine designs.

ENIAC.Lesson Sixteen. The computer revolution cannot be marked by one machine, one person, one idea. A number of concurrent machines, each built incorporating a small number of changes laid the groundwork for the computer era. Atanasoff built the first electronic digital calculator introducing the notion of direct, serial, binary computation using a regenerative memory; Aiken trained a large number of programmers on a machine using Babbage's program-control concept although the technology was conservative if not reactionary; Stibitz, attending mainly to obtaining a reliable number of operations per month, built the first machine providing computation; Williams and Kilburn added a new storage device; Wilkes inaugerated micro-programming; and if any single team was at the fulcrum of the revolution it was the ENIAC group, stressing speed in operations per second.

As a corollary, the machines that have tried too change too much at once, have not triggered new generations. The Babbage machines and ILLIAC IV are in this category.

DEC6205A more recent example comes from my own experience designing the PDP 6 about 1964. This bit slice module is my memento. We thought we could change everything, that there would be little risk in doubling the circuit speed using a new mechanical packaging technique placing connectors on both the front and back of the modules in order to get the requisite numbers of pins; specifying a new architecture with a megabyte adress when everyone else was at most 256K; organizing a flexible structure that would permit building a large multi-processor in an evolutionary fashion so that we could build subsequent machines on the same base; presenting a straight forward interface which as a side effect probably started the whole idea of third pary vendors at Stanford, and predicating the design on timesharing -- a concept that was just being breadboarded at BBN, MIT, Stanford, and SDC.

PDP 6No wonder only 20 PDP 6's were made. But the team stayed together and gained experience for the PDP 10. I would have hated to say to customers at the time that we were selling them an advanced development effort for what turned out to be much of our own, and others, interactive computing. Thinking of the 6 as a breadboard, probably the main mistake was not changing the packaging more to avoid the mechanical problems that were not worked out until the PDP 10 was built. By changing the package so that wirewrap was permitted, a fundamental new technology was used allowing computers to really be mass-produced and not handcrafted. This was the key to the formation of minicomputers and the explosion of the computer population.

PDP 8.Lesson Seventeen: In making revolutionary changes make sure that every aspect of technology is covered. A better, longer life tube was critical for the ENIAC, and the seemingly trivial issue of sound mechnical connector mounting was critical for minis.

.Lesson Eighteen: Prototype development provides a way to reduce risk. In the first and second generation of computing, many pioneers assumed that they would build the computer itself. Unfortunately all the lessons can be taken to a illogical extreme -- I often feel that we take too long on the prototypes and tests today.

MARK I .Lesson Nineteen: The original justification (that we call need) for funding and developing a machine is often different than its use and contribution. Harvard's Mark I is not recognized for the intrinsic value of solving Bessel functions, but for training a number of the people who became leaders in computing: Bob Ashenhurst, Gerrit Blauuw, Fred Brooks, Grace Hopper, and Jerry Salton. ENIAC had limited value as a computer, but it proved that computing could be done electronically and the summer course, offered at the Moore school, led a number of people to design computers. In contrast, the Bell Labs relay machines really did computing but did not have much effect beyond that.

ENIACENIAC was a great catalyst because its combined innovations were revolutionary.

.Lesson Twenty: Totally new ideas, often coming under highly skeptical criticism of the establishment, are needed in order to change the direction of future generations. ENIAC provided this. Although evolutionary changes in relay technology may have resulted in the same performance as ENIAC in terms of operation per month, ENIAC's high speed in terms of operations per second permitted revolutionary use.

ENIACBut ENIAC also had a number of real problem areas; the plugboard programming which was based on its differential analyzer predecessors was unreliable. This provided a drive for the stored program concept.

DEUCEIn my own experience, I've found that an adversary design has often created an extraordinarily strong driving factor for change. This module from DEUCE, a machine that was derived from Alan Turing's Pilot Ace of Britain's National Physics Laboratory, is a memento of a year I spent programming the machine. In 1958, when I started to work on the Deuce it was programmed in punching row binary and I was driven to write an assmbler that provided symbolic programming using three addresses. The program allocated instructions to positions in the delay line in an optimum fashion. This assembler may be the first one-level store machine using the 8 K word secondary memory and 320 primary memory as one. Because of my experience writing absurdly complicated programs for DEUCE, strong internal goals for computer architecture were fixed in my mind.

.Lesson Twenty-one: Adversary designs and the poor use of technology can provide the definition of a need that is useful in determining evolutionary designs.

.Lesson Twenty-two: Nearly all mechanisms that appear in computer hardware structures start with software implementations.

MK & ENI've been able to draw a number of these lessons from comparing the properities of pioneer computers. If there had been learning between the different efforts, then the computer revolution might have happened faster, . Mark I could have used relay technology and some of the design techniques developed fro the Bell Labs machines; Bell Labs and ENIAC could have used some control mechanisms of MARK I avoiding the large tube counts.

.Lesson Twenty-three: When working in a new area, determine other pioneers and keep abreast of what they are doing. In reading Pam McCorduck's book, MACHINES WHO THINK, the impression is given that communication between the leaders in artificial intelligence enabled the field to advance as rapidly as it has. Furthermore, this sharing has not taken away from the credit of any individuals.

WWI think Whirlwind was the most significant computer of this period because the design team investigated other machines and a varieties of technology. Then they designed a machine to solve a significant real time, interactive, and control problem. Every other computer built in the forties was either oriented to arithmetic computation or data processing. The original task of project Whirlwind was to build an Aircraft Stability Control Analyzer, requiring real time simulation of an aircraft. This need constrained the problem in three ways: reliability, accuracy, and speed. Over 100 simultaneous equations, with an accuracy of .1 percent, had to be solved at a 10-20 herz rate, forcing a parallel organization.

BUSHOriginally, the program was conceived an an Extension of Bush's work on analog and differental analyzers, with the project starting at MIT's servo-mechanisms lab. As the work progressed, the transition from analog to digital was based on a suggestion by Perry Crawford who worked for Bush. Crawford's ideas based on his 1942 thesis on digital computation were critical to the decision of both ENIAC and Whirlwind to become digital computing projects.

WW DIAThe MIT team, led by Jay Forrester, investigated the efforts of ENIAC and EDVAC. They made two unusual design decisions for the period. The serial approach was ruled out in favor of going to a parallel computer. They also moved from the 40 bit word length convention to a 16 bit word. To a large extent the word length was chosen to gain speed and accuracy within the size and cost constraints.

WWAt both the University of Pennsylvania and at MIT, the administration and the design teams tangled over the "value" of building a computer. Eckert and Mauchly put a high value on the economic potential of computers and insisted on holding all the patents themselves. This ultimately led to their leaving the University. Forrester at MIT was not interested in building his own company. He was interested in sound engineering practices, stating, "Experimental equipment, merely for demonstration of principle and without inherent possibility of transformation to designs of value to others, does not meet the principle of systems engineering." MIT never got into the computer business -- but the Whirlwind did provide many businesses with proven designs and trained engineers.

AC Mod.Lesson Twenty-four: Build real things, not toys. The Whirlwind modules were taken verbatim by Burroughs and by ERA for the 1101, and the machine itself was built by IBM to serve the SAGE system. ENIAC was the breadboard for the UNIVAC machines. These real, engineered efforts at universities were significant spurs to American industry, the economy, and computing. In contrast, the Harvard Marks and Atanasoff's machine were toys for training graduate students.

WW TUBESJay Forrester, concerned with highly reliable, real time computing, knew that the estimated tube reliability of 500 hours had to be increased by several orders of magnitude. An outside review prodded at the gradual failure mechanism of the tubes and led to marginal checking. By understanding the tube failure mechanism, the manufacturing process, and introducing marginal checking, reliability was raised to five million hours. In fact, the Vacuum tube IBM AN/FSQ7 sage computers, that should be known as Whirlwind II except for the stuffiness of IBM, are still in service.

.Lesson Twenty-five: Question the technology suppliers, solicit outside reviews, and pay attention to all the details. As Mies VanderRohe--the most pristine architect/engineer said, "God is in the details."

ELECTRoIn the late forties, everyone building machines was searching for a reliable primary memory matched to the machine speed. The two Mhz clock and 50 K ips speed using MIT designed Williams Storage tubes cost $1K/1K bit/month. Impressive, but expensive. Searching for a better solution Jay Forrester started to investigate using magnetic cores. At first they used wound magnetic tape Deltamax cores.

Cer CoreThen the beautifully made, but little understood, ceramic cores were found at Philips. According to Forrester, the manufactuers claimed that they could not be used for storage and theoretically this was true, but it didn't stop Jay Forrester from trying ceramic cores and succeeding.

.Lesson Twenty-six: Don't be undone by theory, especially if the art is much ahead of it. Forrester commented, "This is an example of where the art was substantially ahead of the theory. Cores worked and could be made by trained ceramicists. Years later scientists understood how and why, but for many years production of ceramic cores was a materials art."

CoreMIT's University Research Corporation did not see fit to patent the core because they considered its commercial applicability would be neglible. Forrester got MIT to patent it, and to his chagrin (and probably many others) kept many patent lawyers in business for years. He stated, "The Patent effort and litigation took about 1000 times the effort of the design. It took six years to convince industry to use the core and then six years to convince them they hadn't invented it." In this case, IBM lost the suit against Forrester and MIT, but they still will not readily admit it. I was recently told that IBM invented the co-incident current core memory; An Wang and Jan Rachjman of RCA also claim it. It was such a good idea at the time, everyone wanted the credit, just like the abacus. Significantly, the idea did come from the university environment where openness across disciplines and cultures are much more likely to occur than in industry.

.Lesson Twenty-seven: The role of the universities continues to be critical for generating new computer generations. President Killian who was at MIT during the early electronics boom stated that it was the mix of three things -- teaching the bright young undergraduates who were free of preconceived ideas, the drudgery of the graduate students plodding on theses, and faculty consulting to industry that made the daisies -- the new ideas -- bloom.

WWIt is hard to look back to 1948 when there weren't any computers for students to use at universities. One of Forrester's reports gives some feeling for the frustration that he felt. It stated, "If a high speed computer capable of 1 K to 20 K operations per second were sitting here today, it would be nearly two years before the macine were in effective and efficient operation. One would be caught totally unprepared for feeding to this equipment problems ... this represents one-half of the vicious circle in which an adequate national interest in computer training cannot be developed until the equipment is actually available."

.Lesson Twenty-eight: Understanding and training about a revolutionary new device requires the device. The problem is still here. The 1979 Feldman report argued for funding for equipment for experimental computer science.

Sage CFortunately for all of us, Whirlwind was built and had probably the first operating system for real time processing. It was used in demonstrating the SAGE air defense system using real time input from radars whose information was transmitted via phone lines. It also had the first crt's and light pens. In addition, Whirlwind was used for at least two purposes not conceived in its design but that fell out of it:

the first computer speech research and Linvill's work on sample data.

.Lesson Twenty-nine: Build in generality, because the system may be used for something entirely different from what it was intended.

Barta BWhirlwind occupied this entire building in Cambridge. Yet, I like to think of it as the first mini-computer. It operated like one -- that is, it was personal -- even though the programmer had to walk into and not up to the console -- and it was interactive. Just a little bulky. Yet it is easy to understand that in the early fifties, the engineers were anxious to try out transistor technology in order to significantly reduce the size from that of a building to a room.

TX-OWhirlwind, with all its investment in operating programs, was dicarded for the TX-0, the first transistor machine, originally designed to test transistor circuitry and large memories.

TX-0It was 18 bits, and quite impressive -- so much so that you can see the Japanese had already taken cognizance of it for speech research. Not I was, and still feel, surrounded by the Japanese. (I'm leaning over the machine with my hand over my mouth -- holding in our secrets.)

PDP-1The people who designed TX-0s circuits started DEC, first building logic modules using the basic circuits and then the PDP-1. Much of the software investment in both Whirlwind and the TX-O was lost in making progress.

.Lesson Thirty: Don't be too hasty at throwing out a previous set of technology especially if there is significant investment in software.

EDSACIn 1949, only one month after EDSAC was operational, Maurice Wilkes perceived the value of a series of computers sharing the same instruction set. He stated, "When a machine was finished, and a number of subroutines were in use, the order code could not be altered without causing a good deal of trouble. There would be almost as much capital sunk in the library of subroutines as the machine itself, and builders of new machines in the future might wish to make use of the same order code as an existing machine in order that the subroutines could be taken over without modification."

TX-0TX-0 had an inadequate word length for accessing the 65K word memory that the machine was designed to test. Every three years the cost of a given size memory declines by a factor of two. Thus, each generation the machines have inadequate address space expansion to move to the next level.

.Lesson Thirty-one: Every three years another address bit is required throughout the system to address various memories.

Minis.Lesson Thirty-two: A general purpose machine, including a language, should be designed for orderly extensibility, especially in address-size, or in the case of language machines, data types, otherwise the past machine will have to be emulated in successive generations because of perceived software investments.

Given that I've opened the issue of building successor machines that are compatible with or build on the past, I feel duty bound to state a lesson that RCA ignored and the Japanese eventually learned.

.Lesson Thirty-three: If you copy a machine, do it exactly -- not just closely. The test has to be that the software, including all user data and files can't know the difference between the original and the copy. Furthermore, if there is a desire to attract and then entrap a given set of user to your machine (or language), then build it compatible with extensions that other machines don't have which your users will feel duty-bound to use.

.Lesson Thirty-four: Getting the right standards at the right time is essential. If a defacto standard exists, such as the IBM channel and Unibus, let it be. If a standard is needed, then go all out to create it so that others can avoid the hassle of having to invent in an area that will generally make work. Alternatively anarchy can reign until IBM makes an ad hoc decision, and then it can be accepted in a de facto fashion. I hope the forthcoming standard based on Ethernet will permit communication systems to be built.

TAB IIIWhereas early computing technology was marked by a change in the basic phenomena, now, it is a refinement of the semiconductor phenomenon. The end of the fourth computer generation is marked by the number of semiconductors on a single silicon chip. The fifth generation microprocessor with a single computer on a silicon substrate has emerged. The sixth generation will be limited by the time to make and refine a design and to find the next collection of ideas that generates the new structures. The estimate is seven years, which is also the time taken to get a factor of one-hundred times increase in the bit density on semiconductor memories.

With the fifth generation or perhaps near the end of it, we may see the beginning of the end of the computer as it becomes part of more of our goods. Soon, cams and levers in typewriters will disappear as we form all electronic typewriters and make the transition to all electronic transmission, storage and transduction of information. This later step is just a matter of time unless we find out that there really is an infinite supply of energy for transmitting us and out paper

$ vsAnother positive feedback cycle exists for continuing to supply machines at a constant cost with increasing performance because the existing user-base metric is cost/performance or productivity. Given a substantial investment in costs of operations, increasing performance at the same cost gives the highest overall increase in productivity.

New computers and use evolve in three different ways providing three lessons:

One. Holding costs constant, improved and cheaper technology allows increasing performance and evolving use; hence

.Lesson Thirty-five: The current economic mechanism favors evolution of machines in order to aid short-run productivity for existing users.

Two. Holding performance constant, a new structure can be developed based on decreasing costs. In this case use will simply become more widespread.

.Lesson Thirty-six: These machines are based on new technology -- which will be old generation computing by the time they are on the market. As such, the machines are likely to make the same errors and go through the same evolution as their predecessors. They do evolve more rapidly than their predecessors because of the elastic nature of the market and because of the numerous design templates.

Three. Developing a new, larger structure with new uses emerges because of free resources. New technology permits change based on increased component reliability, speed, and density. Price always seems to be constrained to about $10 million and the achievement of overall system reliability.

.Lesson Thirty-seven: New uses come out of free and available resources not out of a computer system that has high throughput. This may explain why batch processing disappeared.

TABLE IVAs I look forward, two goals could force the evolution of computing. These are energy self-sufficiency and economic self-sufficiency through production. Regaining a number one position in overall science and technology might then be a fallout.

.Lesson Thirty-eight: The last one. It's all in the timing. Change when the technology is obsolete. Don't make the same mistake over and over again. Deviate when to make a significant gain but don't necessarily throw out or ignore the old. It's really alot like rafting on a fast river -- you've got to go with the flow and do some fancy paddling to get through the rapids right side up.

GB2.S1.31

```
00  BURT  DECGRAM ACCEPTED  S 8508  O 42 11-APR-81  16:24:26

*****************
* d i g i t a l *
*****************

TO: see "TO" DISTRIBUTION                  DATE: SAT 11 APR 1981
16:17 EST
                                           FROM: GORDON BELL
                                           DEPT: ENG STAFF
```

EXT: 223-2236
LOC/MAIL STOP: ML12-1/A51

SUBJECT: THE 5TH GENERATION: A TRANSITION AND OPPORTUNITIES

It has become increasingly clear to me that we are unaware of the change taking place brought on by the Fifth Computer Generation, beginning in 1980+. The problem appears first in Microproducts and low end part of 16-bit Rack and Stack business (11/23 and 11/24), and may be symptomatic of our difficulty in successfully building a competitive, low cost computer. We have many transitions to make in the way we design, build and measure (see EDN, 4/1/81) our products.

In the fifth generation, a whole set of new standards come together (conspire) to bring about a new way of building products. These threaten the way we have done business and force transitions.

HISTORICALLY, we have these generations:

1 1950- vacuum tubes, computers simply formed: IBM and Univac introduced the so called mainframes in the million dollar range.

2 1960- transistorized machines (cdc 160... bread board for the 8, pdp-1 the departmental mini emerges, the 7090, IBM tried to build a supercomputer)

3 1966- the mini (PDP-8) is introduced as a systems component, timesharing emerges, the 360 enters scene to become the

standard for mainframes (plug-compatible everything), CDC (Cray) built the 6600 to establish the $4M+ supercomputer market.

4 1973- micro emerges with lots of standards, mini standard somewhat clear (it's the 11 and the Unibus allows standardization, competition, etc.), departmental timesharing (RSTS) emerges and takes off, Unix emerges

Some historical observations:

A generation is the confluence of:

New technology (VLSI > 100K transistors/chip), NI and other local area network standards, winis and floppies, UNIX and modern typed and structured programming languages

A new computer structure (the large address, high speed micro that performs beyond the speed of previous minis), allowing large systems to be built more easily by combining the hardware in a linear rather than combinatorial fashion.

Need. This is hard to tell, but it would seem to be economic driven based on office costs, transportation costs, information costs, and trouble-shooting for health and machinery maintenance. Organizations drive people to supply more information.

Use. Can only be seen after the fact... in '90.

NEW COMPANIES establish the standards that are the basis of new generations (note from Ron Smart's charts, the big revenues are in the >$250K market that use the IBM standard. CDC established and dominated the supercomputer market... albeit very small. We established the departmental <$250K machine, but weren't able to grow as rapidly as the market, hence others came in. Likewise there is a similar situation based on the micro. (Could the situation have been different if we had licensed the 11?)

There are several ways to respond to the transistion opportunity:

support the current customer base even though the standard is somewhere else (Honeywell, Univac, ICL, CDC... although they also entered the OEM hardware business);

create the new generation, which is fundamentally impossible in an existing organzation whose first goal, like ours and probably IBM's, is an organization that lives forever; or

a hybrid between the two. Some support of the past while moving to respond to the new standard.

THE FIFTH GENERATION MARKET:

CRAY dominates the supercomputer market.

370 is still standard with locked in base and many plug-compatible suppliers... relatively low growth rate although prices coming down to enter the mini market.

Micros with mini performance (and claims of micromainframes), providing hardware at zero cost by multple vendors. The high performance machines and completeness of peripheral chip sets at low cost enables a very wide range of systems to be built that are cost effective, from personals up to 11/70's at less than 20K sell price. Notice these standards are not ours and are different from our historical and intuitive understanding:

Multibus (adopted by all Semicomputer Companies) as an IEEE standard to form a commodity industry of modules;

peripheral standards based on wini and floppies together with Multibus permitting commodity peripherals;

Ethernet, enabling Clusters so they can form large systems in a bottom-up, distributed fashion;

IEEE floating point for lower level interchange of data and more important, algorithms and programs;

UNIX, including C, Pascal, structured Fortran, thus getting away from ISP dependencies; Any engineering group that looks at its investment, and incremental productivity in assembly code knows they must abandon assembly language programming.

ADA as a potentially revolutionary language by subsuming both the machine and operating system, although file system dependencies still could get in the way. This would allow totally linear build-up of software based on the work of others rather than specific start overs.

Personals based around chips on various standard boards and using the Z80, Microsoft BASIC

DIGITAL TODAY, seems to be convinced our customer software is worth much (as opposed to cost lots) and our bus standards (with some commodity suppliers) are still fine for the 80's. The situation:

U/Q with CTI and BI emerging as hardware bus base;

RSX-11 S/M/M+, RSTS, VMS, and UNIX which, though supportable are written in Macro, also much other system software that has been evolved, but still written in low level language (like Macro and BLISS); No use of higher level languages except in P/L's.

customer base in macro, and DEC unique BASIC, DIBOL, other higher level languages; We don't even teach Pascal in Edu Services. Those customers in our proprietary languages are probably anxious to move to modern, structured programming languages.

microproducts being built in communications, personal computers and terminals being coded in Macro. These appear to cost more in production, they are very expensive to design because of the Macro code and because many go beyond the various address spaces. Our competitors are using faster, cheaper semicomputer company components without the limits we have.

Given this situation, we should be concerned about how much revenue can safely be based on these past investments. Note the PDP-8 was used for quite some time after the 8/A was built, although it isn't clear that the 8/M wasn't adequate. Note that the WPS program is somewhere around 100K instructions, and represents an investment of about $10M, or $100/instruction, which seems in line with DOD costs. The incremental cost to add to this is probably 5 times greater. I believe the replacement cost, using a higher level language would be no more than $1 Million, or $10 per instruction, using a modern language provided the architecture of the program is sufficient (doesn't break of its own weight or exceed the structure of the computer it uses).

BOTTOM LINE

Commodity hardware bounds us both at the high end by the IBM 370, and at the low end as described above. We have been doing a great deal to not be entrapped, but clearly much more is required... everywhere! Notice that many of the issues are within engineering, but others do cross the company.

ACTION

What are some tasks and risks in our directions in software? in sofwtware engineering? in hardware engineering across the range? in busses? in systems? in applications? in make/buy?

What are your views about what to do in this transistion?

What are you planning to do?

"TO" DISTRIBUTION:

PETER CHRISTY ENG STAFF: BARRY JAMES FOLSOM

LLOYD FUGATE WARD MACKENZIE JACK MACKEEN
AVRAM MILLER OPERATIONS COMMITTEE: BILL PICOTT
HERB SHANZER STEVE TEICHER MIKE TOMASIC
LARRY WADE

GB2.S5.63

GORDON BELL AT STRATTON V (1980): WHERE DO WE GO FROM HERE?

The conference so far has generated a lot of - certainly stimulated me through a fair amount - I've prepared a two- or three-hour talk based on all of that (laughter) which I'll attempt to give in 50 minutes because my university training still makes it impossible for me to talk more than 50 minutes. This is actually - I guess, a few sort of final remarks before I go into this final talk, which is essentially - there's - do I have to answer that? - I want to run without interrupt. Sorry. Later (unintelligible) systems off. The video was really an experiment. We wanted to do that. We want to make this some way of getting much more participation here, and sometimes I think it would have been nice - we could have - maybe sometime we will have a distributed STRATTON in three or four sites with all of the things, including the workshops. I want the Colorado people to know in fact at 6 o'clock a lot of people were up here, too, from the night before, usually. (laughter) That's why the camera detected a lot of sleepy faces from time to time. I want to apologize to all the people who would like to have been able to be here but weren't. Notice - I would like to have had a lot more Product Line participation, Marketing participation, Programming, because programming is really what PBS is all about, and in particular, one crew of people that are really important to PBS are people who do technical documentation and already we're going to have a special conference that's being set up in Tewksbury with some people who complain about the way I talk about documentation. So it's going to be an n-on-one where (laughter) they'll give - they can - that we can interact about this because, hey, documentation is really important for these systems and finding a way in that people can use things, so I want to apologize that I don't think that we work that issue here at all. It's vital and we're going to go out now and work on that as something that has come up repeatedly as being important. Okay. Now, the three talks.

One, I just couldn't help, during this thing, of doing a little bit of sharing (unintelligible) and it may come across like an educational seminar, but it in fact comes across because I think I have a perspective - there are a few things that it is - that you do get by meeting regularly with people across functions, and I probably do that more than anybody else, just because of the meetings that I hold with the Marketing Committee and the Operations Committee and customers and things like that. So I've got to figure out how you can get into some of that, but certainly there's a bit of that I'd like to share - some reality and education here.

The second talk is really on realizing the passive - if in the past I've been able to set goals or help set goals because of some perspective I may have, why, that often ends up - I'm not trying to design all the systems, but by helping to set goals, hey, it provides a focus for people to work in, and I know that's not popular at DED for managers to have anything to do with establishing goals. It's supposed to bubble up and somehow the managers are to be able to have a process to let you decide which way - how to do your own thing. I would like to focus a little bit where you do your own thing at, which way - your own thing we go. So I'm going to interject where I want to go, and that's called goals. And then finally there's - got a tremendous frustration of the wide diversity of what - when we talk Profession-Based Systems - an incredible array over cost, size, depth, you know, many, many dimensions - physical architecture and want to start sharing some of that taxonomy. I've been playing with taxonomies throughout the last three or four days. I'll sort of share some of that with you too.

So, on the first part then, let's go on this, which is - I guess the - in fact, the reality - I find that we're really decoupled from the users because of what I hear, namely, that in fact - hey, folks, we're already building a Profession-Based Systems now. Now, the question is, "How are we building?" I heard four methods come up in this. Let me characterize them. Let's call it the Marketing Textbook approach. And we saw that in a flow-chart form. It's called the ask 'em/tell 'em approach, namely, you go out and you do some stuff, and then you provide a thing and then (unintelligible) a feedback that was described. And that you don't take them out of captivity, but you try to figure out what to feed them and maybe they'll die, maybe they'll eat the food. (laughter) Market survey is absolutely straight - straight conventional marketing text. I don't think it works worth a damn in - for complex things. I don't think that's been the secret of our success anywhere in and can't be except in conventional things where you ask them and it's COBOL 79 and it's clear because that standard is there and they know that that's what they should say, and you go back and do it. Now I heard the other one - let's call it the zoo approach, which is bring them in and let them tell us. We'll interrogate it and we'll build them and ask them. That's essentially just moving the animals in and try to find out a little closer what to feed them and see if they - but, in fact, when they come in they really aren't animals any longer. They take on a lot of our flavor and they can't run. They can't do any of the things that they used to do. And in short they're not professionals any longer. They absolutely have disappeared as professionals in that environment. They are - you know - that's how - in fact, well, I'll say, that's how we get a lot of marketing people when we're going to "go in the insurance business." Somebody might have sold insurance once for their father and so they came in and now we're, by God! we're in for an insurance business. So there's the zoo approach. Those animals are probably the most dangerous animals because they're in captivity and they can't do any of the things they wanted to do before, and they're asked to be translators. So - and then there's another form of that which is the zoo plus a genetic mutation form where we try to turn the animals also into an insurance-agent programmer, and so you essentially - you've got this new animal. But in fact you may get a great programmer out of him, but you probably have lost his view as what is means to have been an insurance broker. I don't think Tom McIntyre will say right now that he's a physiologist. I don't think he'll claim he's a physiologist any more. And we're glad to have him as a software engineer. And in the fourth - I think those may - that one may - conceivably might work in some limited thing. And then the fourth point is essentially: leave them alone in their environment. Do some genetic mutation of them. They've got to do it themselves. If you look at the MITech - MIT - result on where are complex instruments and tools built, God dammit, it's clear. The users have to build them. You give them as much help as possible with tools and I think one of the hidden tools we've got to supply is some programming languages that are non-sequential programming languages and Charle' - we saw one in Charle's slides - the Demo didn't quite make it, but if you know what that problem is all about, man! that's probably one of the most impressive pieces of work - being able to - he went up and changed the line on a 1040 and the whole goddamn thing changed and there wasn't a program sitting back behind there. He had expressed those in relationships. Man! that's impressive! And that's not programming like we used to know it. That's the way to do it, probably. Getting rid of sequencing: sequencing is only natural if you love computers, and I - the world, I don't think, thinks sequential. I mean, think of all the people who can't, who, given a watch - I mean, I got to meet a guy who used to be vice-president of an insurance - a retired guy - came over at Christmastime and said, "Hey, I got this watch and I can't set it. I wanted this electronic watch and I can't - and man, that was before it had four buttons. I think there were only a couple buttons. I mean, if you threw the Seiko manual at him, that guy never could - I mean, literally never could set a Seiko watch, and I must admit that I did have to refer to the manual a couple of times (laughter). But I had to put it in terms of a (state?) diagram (laughter) and I got it on one (laughter) page - I had to figure out how it worked and then it was obvious what the thing did. But in fact, trying to read it from the manual was a bitch. (continuing laughter) Now, who we're doing it for now is - here are some of the product line - here are the results we've got now and I'll just go over the approach. There's kind of the ask 'em/tell 'em approach, but in fact we have a number of them in captivity and we're trying - we're doing it - we're providing that. That both - and the column on the right-hand side is - this is at the levels. And level is an important thing; how do you provide it is an economy of scale. That's both for personal- and group-level kinds of things through the 78 and the 248 kinds of systems and on larger systems. Hey, we've got a group which is improbably using the approach number - it's the - actually, I think it's approach number one: it's the ask 'em/tell 'em approach, which is the professional typesetting. We're doing that now, and we believe we can do it because we use the ask 'em/tell 'em approach for newspapers, and we're in that business. That's a hard job; those guys have made money in their - in that - in the typesetting/graphic-arts thing, but boy! it's a real bear, and it's perseverance that has to carry - or has carried that effort. I don't think that they would advocate that's the way to do - if you had to do every profession that way, forget it. Then, you go down into the physical and natural scientist kind of things, there I think we really are pretty much sticking with a very low-level tool. We're really not addressing that, although we do sell it there. And who's doing it? It turns out the users are doing it, and library-sharing is the vehicle by which they're doing it. Now you go in to the engineeer, educator, and small-business persons - there's a methodology that we use there. Hey! all of that stuff - we got tools there and they're all done, pretty much without exception, buying out. Now: the question is, "How do you buy out things?" and then - here's a process that we are going through now which is user's development. It's probably the only way. And it's really through OEMs who are or have been users and in general we are not. And the process is really one of evolution. And it's an evolution (unintelligible). We look at success in our business, providing language capability, this is evolutionary. It isn't after the animals go headlong into something. So here's the process I'm advocating, namely, improve the languages, in this case DIBOL(?) and, by the way, let me say that in fact, if I were to program in some of these application languages, it would probably either be in MUMS (?) or DIBOL(?). Neither of these are popular, for some reason, by these groups. Simply because they're the best languages. They have the right data types already, and if you like low - if you like to look at (unintelligible) to program, you'd love PASCAL and FORTRAN and lower-level languages like this. These things know about the data types that you're working in, and I don't know why we don't use them for that rather than going down into these very low-level programming languages. Certainly it's got to be well beyond the implementation-language levels that we have today. But I think the most productivity is going to come out of that approach. But first off, I want - have advocated seeing an evolution of DIBOL to make it a little more robust in terms of allowing users to do things like set alternatives, set parameters, state alternatives to the parameters, and then eventually express limited algorithms, so that, in fact, you've got an algorithm for computation of income tax, and the neat thing about certain kinds of these programs we're talking about - they go in and they flow through sequentially for all records and don't tell people about programming and loops. So I think we can - the world can - we can get these people into processes for starters not telling them about things called loops, which aren't natural things to most people. Okay, select a profession market-size. Go out and do the regular marketing kind of stuff that one does: how many people are there, and so on. Now here's where I, being chicken, and probably as a corporate officer, begin to say, "Hey, find a program out there!" I would simply go out and buy a program, and by God, there are a lot of them out there. We've got a lot of users who've done their own thing and - a lot of OEMs providing these services now - and go out and pick that program up. And then, what do we do? We probably want to do it. It probably isn't up to our standards. We don't think it's got the quality levels. We don't think it can be enhanced. But go and do it by testing in QC, enhance it, document it, and sell it. Let's - and then we've got some information to work on. And we aren't starting from a PASCAL or a file system or some very low-level thing. Now, I think we want to sense additional - sense what's happening somehow, not sure how you do it. Probably by getting a bunch of users to come to meetings like DECUS and say, "Gee, what do you want?" "Well, I couldn't - they changed the income tax kinds of things and you'd better figure out a way to give us a new option there somehow, so I think by having groups combine or by networking - lots of mechanisms to do sensing here. And then, what would I do? I would then do it like we do now, Goddammit, within - by evolution. Namely, I want to reimplement the thing using the requirements. I want to keep the documents constant, probably I have to - unfortunately, I have to keep the file system constant because these users have now got a lot of information on those old file systems. I then do a better job of reprogramming the thing, if the program by this time, as it keeps being refined, is going to get a little bit smellier, or creaky(?), and may burn out. So, I think this to me is probably the approach that I think is going to yield success in there. And I think, in general, "Gee, a lot of us may not find that challenging." Because we're not working from raw technology, we're not working through this other process, but boy!?r I think it - I kinda like it, because I think it'll just make scads of money. To increase the Engineering budget. (light laughter) And that's really what drives (chuckle). (laughter)

Okay.

Now I want to go on - that's kind of a piece of overall perspectiv?re about how I think we're doing - how we've done it, a?rnd how I think we're going to do it in some of these insurance offices, the dental, the small-business kinds of markets. Now I want to go into, essentially, these goals. And if I look back on past STRATTONs, well, first off, I could't get to Lorrin to find out what STRATTON I was. But in fact, on II we had VLSI - was the focus, and I think we've made results since then. Mass Storage was III - hard to judge. I think we've made progress there. IV: Distributed Data Processing - I think we did a lot when we focused on Distributed Processing. I think the HYDRA stuff has come out there on the DECNET Version 3, 4, and the direction there, I think. The Interconnect is clearly what we were beginning to focus on last year. And then one of the side benefits is - we had a Workshop last year and really went at the RAMP issue. And I said, I remember I guess it was Steve Rothman and, I think, maybe, Dave Cane and Bob Stewart, all said, "Look. We know - give us a cookbook of techniques. Don't tell us how to design it." And I said, "Let's get a good cookbook." And in fact I've seen such a cookbook. Mickey Smith wrote it. It looks very good as far as I'm concerned, in terms of coverage and approach. But anyway, you're going to (unintelligible). Okay, fine, okay. But I hope you'll read this book in terms of - it must have been (unintelligible). But anyway, I think that there were lots of side benefits of interaction that I can identify with. In fact, this one - it seems to me we've gotten some ideas about the next ones. There are a lot of candidates for sensing I'd like to know - well, maybe next year - have we cleaned up our act? I'd like to have a follow-on. But I certainly intend to devote a substantial amount of time to cleaning up our act now. And then, clearly one that's come out of this is the Manufacturing/Engineering team aspect. That's really - really needs a lot of work. And, so whether that's the next one I don't know - there ought to be a lot of ideas for new ones. So, essentially, let me - in the goal sense - I think - what I want to do is go over some of the goals, which is, essentially, I want to clean up our act and, in essence, get back to basics. Not BASIC. (laughter) That doesn't call for everybody going out and building their own BASIC version (laughter) because we've only got two or three, three or four, how many? five? six? We'v gotten it down somewhat. I know we've gotten it down by one. So we're back. (unintelligible) But basics, as far as I'm concerned, are what's going to follow. This root level, either finish or finish starting, the Interconnect, SUVAX, VENUS, SCORPIO. There are projects here that I think we want to, above all, get those things - get - that's the goals I have, getting this root-system level - I'm sorry, DBMS-32, some of the database work. Let's get the whole operating system language hardware, these bases done, including the small systems stuff. I think we know what they are.

Now, the next area - here's what I mean by the basics, which are the generic level. I want to master this level within the next year-and-a-half products - products or product breadboard - or products that are reasonable through tasks. What does that have in it? We've dwelt on that a few times, which is essentially among other things, a virtual terminal capability to other systems. That's been left off a couple times. I want to get it back. To make sure it gets back on. So they will communicate to the rest of the hierarchy. And right now we've got to get the word-processing stuff straightened out. I want high-quality, compatible WPS in the 8 area and be able to evolve the 8. Right now the 8, as far as I'm concerned, the structure of that thing, the number of versions of it, isn't in a form that we can evolve and enhance on it. And, in short, we're going to get killed, continuing work based on that current base. So we've got to get to a point where we can pack the market, given that we've got a piece of hardware coming out there. We're going to get creamed. Stand-alone - 11. Hey! Right now I look at being able to track the thing - we're just not going to be able to track it in the long run, based on PDP8, the PDP architecture. Hey, and that's hard for me to say, because I invented that architecture about 1963. It's sort of - and so it's kind of like a first love or something. But for the kinds of complexity and systems we're building it just doesn't happen from a software standpoint, even though we can build cheap hardware it's not good enough. Then we've got to get on and get that capability on multi-terminal, probably on RSTS and VMS, right now. So I want those systems all up and running, compatible, high-quality. And then, getting that, we've got to get the EMS in there with the WPS compatibility, probably on RSTS and VMS again. So we've got to bring our EMS work in and have an EMS - a working EMS product - let me say it's got to be running, breadboard, internally, probably this time next year, for at least three months. So on the engineering network we've got to have this thing running. I guess that essentially what it amounts to is DECmail. And then, certainly a file cabinet for the documents and the forms and the thing that I'm thinking there - probably if we limit it to VMS, then I've got the candidate and I want to - I think you know what that is. Phone management: some - a few words got left off here - essentially a probably a breadboard for phone management. I don't - in its full glory, I don't think we'll have that running a year now. So I'd say a phone management, a breadboard, some office procedures, some of these things, some breadboards here, and then certainly a voice breadboard. So, of the last three, probably are breadboard kind of items, not running basics - running "as part of the generic base." Voice-out: some level of voice interaction that's appropriate to the technology that we have. I think there's a clear amount of technology there - let's do it. So essentially here's - I guess this one really should have been that other one - I think probably QBF is the way we ought to go there, in order to get these - to be able to deal with these filing-cabinet kinds of things, and have that integrated in there. Having it integrated in with the other syntax and use. Now there are a bunch of things that essentially are understanding issues. I think we need to understand what features should be versus time. Hey, we're not going to be able to get it - I want to go for completeness next year. There are a lot of things we can't get in by then. Then let's have a planned thing of, Gee, two years from now we ought to be here, and four years from now we are out there. So essentially a set of goals - that's going to help - that's going to determine exactly what we should be working on in the advanced development and the research domain. So I want a lot of the advanced development to come out of product direction that we should be going to. And then the other one is essentially some understanding which is one-bound(?). Count it back up one.

This last one is Understanding and - I'm adding this new word, which is Cost Of Use, not Cost of Ownership. Jim Bell came up with a nice metric, I think, which is set for the professionals that we're dealing with. If you save an hour a week, that probably is enough to justify the system. So all we have to do is find things that can save us about an hour a week to justify the systems that we're talking about. There are issues of understanding: what does it really mean to be personalized. Now, it's clear to me that we can't personalize these things. This is the dual(?) of why we're in such a mess in the networks domain because every company and every network group is personalizing their protocols. And by this personalization effect, as you build systems you get the cost product of every protocol that has to go in every system. And therefore nobody can ever talk to anything else, and nobody is ever complete, because you can evolve, and personalization is an issue that has to be treated very, very gingerly. You can't build a (TICO-based, TM-mass, QBF?), and so on, with that syntax, because when you go off on a particular path, then it all has to personalize within that domain and the connotorial effect on every one of those modules just can't happen. Certainly the PMS structure processor memory switch: how are the boxes, both the - let me use PMS much more loosely - the PMS - hardware PMS and software PMS - connected - how are the things packaged, that is, agglomerated in different things - and what's the structure - the PMS structure - for each size? What's appropriate for the kinds of things that we're doing there? We desperately need something - something in there - because there are so damned many options and, in fact, we saw Al Shugart give us a new option. God! the last thing I wanted was a new option. I'd like to point out, in fact, that with Shugart's correlation, we think - I've had a model until recently that in fact the world was pretty much driven from semiconductors, certainly, but in fact the sizes - systems sizes - semiconductors drive that, but in fact it's disks that drive the packaging structure. And hey, the net effect of that floppy or probably the mini-floppy was the creation of the whole word-processing industry. If you really want to get right down to it, Wang probably is based on Al Shugart. You know, they can say, "Boy, they were fantastic marketing - boy, wasn't that great? Software and man, neatsy RTs and all of that and great management." But yet it was a piece of hardware there that in that sense created that whole industry and they probably - at Wang - let's not tell them. They probably don't even know that's why it happened. But at least my simple model says that's probably the reason why Wang exists. Up until then they were making little desk calculators that you could do addition with, poorer that a program and all kinds of other local things. Now, I want to go and then, probably another thing - some other comments here - this is not in terms of the - summing up the goals - essentially I think using the generic level will get a bunch of users ready for the professions specifics. So let's do them as a - it's not a bait and switch - it's a get-'em-in or - let's - oh, it's the Trojan - I don't want to use the word "Trojan Horse," because that's a good - that's a security number. It's like the Trojan Horse. It's - it's - you know what I mean: it's not the Trojan Horse that's - which is a well-defined concept in security, but it's like that, where we sneak this thing in and people think they're getting a free ride to do all their communication and take word-processing report generation, all the kinds of nifty things that way, and when it's in there they're going to find out - "Gee, why isn't that thing doing this?" And they may try to do it, and that's one way to do it, or they may get us to say "Hey, just put up for a few forms for me and let me do some of the stuff." But that's probably - no , that's clearly the way it's probably going to happen in the profession. And I say, "Stay the hell away from this for a year." I don't want to hear any more nonsense about professionals. Besides, we're a bunch of amateurs and amateurs have more fun, anyway. And we can talk about professions in another year. So, essentially, here, in order to talk about this in another year, maybe we want to understand some profession structure and rude size and cost each could use a function of time, so that in fact what a professional does is really an economic question of "Is this thing going to help him or not?" and to do that you've got to look inside the profession structure, and it's all that mundane marketing crap - you get (unintelligible) and you can have a good linear programming program go (unintelligible) big data bases and professions will pop out at you. Gee, there are a lot of those guys in - you know, we can sell all the dentists in Detroit, or something like that, with one package. But let's get some understanding there. I'd say maybe we want to understand some design approaces and perhaps build a tool that will help them build some of their things, and then perhaps design and understand one profession in addition to the professional systems programmer that I claim we're probably designing our systems for now. If I had to take any profession - if I were programming, by God, you know, that's the one profession I'd really want to serve. It's a self-serving thing, but, in fact, it's probably the right thing for us to address, because it addresses productivity and ease of use, all the things that are important to our environment. And - hint! - don't go outside and look for animals to capture or to bring in, whether they're friendly or we want to keep them or whether - let's use - find some inside. There are enough interesting-looking animals inside to use. The zoo's big enough. (light laughter) And maybe here's one - I don't know - just for an example - design engineers at DEC, perhaps including packaging engineers - I look at all of the stuff coming out. First off, they do the generic capabilities - I have an ulterior motive about having (unintelligible) people who do packaging to use this equipment. I think that, in fact, a lot of these things were their - really - they felt a little stronger because they were using that - things like noise and cabling - by golly, they might get a little more attention. Because there's nothing like having to use your own stuff. Right now, we're really worried about schedules, so I put PERT in there. Everybody needs that. And then I - I'd like to hold all - right now, we're getting a much more complex environment. I'd like to hold all of the DEC standards. I can't think of any reason why that is on microfiche. I can not! It really is - this is a moral for us not to use computing for these particular kinds of things. I mean that whole process, that ability to use this. John Holman, I hope - Bill Tays, God dammit, get that stuff on there now. (laughter) Independent of whether you're professional or not, (laughter) use - I'd like to have query, answers (unintelligible) program (unintelligible) - that you can probably steal - don't do any programming either (laughter) - and can let a bunch of people go on and interrogate that and do some stuff like the HYDRA system. And then if we go beyond that essentially, I like to go to do printing of that and microfiche publishing directly of those kinds of things. But I guess I've got another fear here - I'd like to be able to check some designs against some of these database things - gee, I'd like to know, "Have I violated a given - DEC standard 30, for example. We ought to be able to do a little bit of that kind of thing - maybe - let me put - that one, I think, is a little bit hard. So, let me not - this isn't - besides, this wasn't a gauntlet, anyway. I'm - I think this is an idea that we might do, except one we really ought to do. That's just one that happened to come to mind when some people were talking. Now, (give me some time - twenty minutes - okay - it's going to go quick) I was - for example - looking at, say, the physical system with the -

This is the third talk. Sorry.

We need some measures and understanding about use in order to do what we're doing even at the generic level, and to provide systems everywhere. And I, frankly, am worried, because a physical system - I interviewed a few people, namely system designers, builders, system manager types, some product line engineers, some software engineers, some disk engineer designers and builders, some disk manager types, and my informal survey here reveals no knowledge of file size, RSTS versus features, number of users, and what set of programs are run, who wrote them, and how they relate to one another. And, you can say, "Boy, are we (unintelligible)" That just goes to show you how good our users are in taking what we provide and doing something with this. And I think probably that we've got too many - I don't know - I don't understand why this is. I think we get security in having so many people around and you figure out - it's the distributed database problem. It's me - if you can - somebody has probably got that knowledge. I think that knowledge is in the organization. And therefore, there's a certain security that comes with having knowledge in the organization that you don't - it is - you don't have to know anything because somebody knows it, and all you have to know is who to ask. And, with so many people - I don't know, I tried and I don't even know who to ask. I was really disturbed on that, particularly as I was trying to dream up a system there and I just couldn't get any sense at all, because of, really of lack of knowledge of how that - what that thing really was and how it was used. I think it's easy - not easy to get, it's going to be tricky - but we certainly have to know a lot of that. So that kind of prompted a thing which says, "We want to go for some definitions and measures." I think this is really another set of goals - or it's a tailing in of the goals, which is - certainly I'd like to publish a glossary with accompanied taxonomy. I want to really go back and relate to the price bands.

I want to go into some of the - these are dimensions of the taxonomy - certainly the physical structure dimension, the PMS structure dimensions, software levels, and the program structures dimensions, which are really extended PMS structure kinds of things. So a way of talking about these things - I don't think we've got a very - we can't talk about these things with each other right now. I can't do design - I mean, I try to do design with Tom Orr and just for a minute throw out a thing out there and he says, "Oh, you can't do that because of that," and it turns out it's a piece - because it's all based on a piece of folklore that he happened to have gotten from a marketing survey that somebody - whether it's a natural constant or a - but we don't understand - you know, we don't understand some of this stuff. So we've got to have these dimensions so that we can talk about various alternatives here. And then, essentially, this is one like that - I said before, in terms of the - of a particular structure, and that is, we certainly - the kinds of things - one reason we've got a problem talking is, we're talking about systems - to me, my profession-based system costs thirty - sells for thirty thousand - I think we can - oh, I don't know whether we can get it out there for thirty thousand dollars or not, but it's a bargain if it produces the results I think it should produce. Thirty K is really quite cheap for one of these workstations, because if you look at some of the workstations that we have in a large KL-10, it's a forty-fifty-thousand-dollar workstation. When you put a KL-10 - central KL-10 - divide the number of actual users of that system in some of these design stuff, we're spending fifty thousand dollars a terminal. So thirty thousand - if we can come down to thirty thousand dollars for some of the things and get some benefit, that sound - to other people, they - Avram and Ken can talk about their professional-based system and they're only a factor - their dream is only a factor of ten apart. And in cost - and then to Tom - we heard a three-hundred-dollar - was his - he's got a system there that he's trying to build for three hundred bucks. And so we've got a factor - easily a factor of a hundred that we're talking about. So no wonder - and let's assume price - you can't always assume - that price has some relationship with capability, but say it did for the moment, then we're talking about a factor of ten or a hundred in terms of what the capabilities of these systems. So I'd like to get some understanding of how we justify these systems, because I'm into that a lot on the case of, say, EMS. Why do we want more EMS terminals? Does it really make it more productive? And - for a select group of you, I'll let a secret out, that - I asked about, "Should we have touch-tones in the Mill?" And - I won't - please don't respond to me on EMS - oh, do it, do it anyway - but we wanted to find out a little bit about that, and see how do you go arbout - I wanted to know how you justify some of that. How do you go through the analysis of that? And, by George, I got a couple different points of view. Alan Kotok had a point of view. Mitch Kur had a point of view that Alan Kotok knew, of course, that he would have, and then Peter Christy has a different point of view, and then I like BJ's point of view. God dammit, I'm sorry (unintelligible) all this bitching - let's just do it. So, anyway, that was an exercise in how do you go about understanding this. What I've been trying to do is understand that. So if we look at - here's a set of - for those of you who don't - either haven't been exposed to - let me offer a notation - it's called PMS - as a way of describing things - I'm not going to talk about it here - this is just to refresh your memory as to what the words really - what the characters really mean. It has the advantage that you can type it on typewriters, and, forgive me for not putting boxes around all components, because everybody knows that components have to have boxes around them. The chemists didn't know that when they have their molecule diagrams, and I wish we could have EMS without boxes. But, forgive me for doing it here. But, look at some of the alternative structures that have evolved over time. In the beginning, people had their tertiary memory on cards, their file systems and - this double dotted line is a communication link - it was called "walking upstairs" - dumping it into a transducer, a card reader, putting it into the computer - and maybe having a secondary storage for programs and for mag tapes, and you walked off with - well, one of the things that you walked off with was paper - isn't - don't have it there. In DEC minis what we did was essentially - we put all that stuff in a room and you walked up to a terminal, which was connected to the computer and - these single dotted lines are - that's one integrated system, not going through a communication line, and that computer was connected with some kind of a memory that was both secondary and tertiary file, some - initially, it was tertiary, that is, a paper tape, and ultimately it evolved to mag tape and DEC tape, and in fact DEC tape was such an ideal tape. So LINC was an example of that kind of very, very simple structure. Well, then we got the bright idea of - it was called time-sharing, and that says, "Hey, keep - put the terminal with the user, go through some communication links, the three dot - the ellipsis means what you think it means - there are a bunch of those terminals. They're connected to a computer and there - that computer has both secondary and tertiary memories. So you keep your files there and you get rid of all that old problem." Them we built the 78 and we put the terminal and the computer together in a package, and then we have a link to the secondary/tertiary file store and the double link there is a communication link because it's got to communicate with other systems. And then the PDT says, "Hey, that's a dumb idea! You don't put the terminal with the computer. You put the terminal alone and you put the computer with the mass storage, so we went down that route. And tried - and we put them all - the terminal goes through a cable to the computer and the secondary/tertiary store and then the double dotted line for communication goes there. And now, Al Shugart, and Jesus Christ! what that allows you to do - that really frustrates me - you can - there will be - if we could get our competitors to really play fair, we could really do well. (laughter) And you know what's going to happen? Everybod's looking at that damned - that 5-megabyte thing - they're going to stuff the computer and the 5-megabyte in the terminal. That's unfair, because everybody knows either you go the PDT route or the 78 route. And what that does is get a lot of cost-reduction out of there. It gets a lot of the file storage. It gets rid of a lot of paper because - say I take a megabyte of that and run the last megabyte - I just simply allocate that - let's call it paper - and - which is four hundred pages of paper, and I simply scroll there. So I've got - I've now combined my paper input and my wastebasket in one unit. (laughter) No. I want to say - that wastebasket - because if I ever want to get at something again, I simply scroll back down that paper. So I've got the last four hundred pages of garbage that this thing has - well, of these words of wisdom that have come out of this system, or what have you. So, it really provides a very neat system. The trouble is, that's not - I've got a couple of other systems I'd better introduce here - and the other thing is - the small floppy that Chuck found - gee, that's got some interesting possiblities too, because then you can take the - do the same thing. And if these guys hadn't built these small units we couldn't stuff it all in one box and get that low attendant cost, we would continue selling what we have and we wouldn't have to do any work and we could think about profession-based systems rather than having to do something. And it's a lot more fun. We wouldn't have to interact with Manufacturing so much and gee, we could be philosophical. But we've really got to get back to work. (light laughter) And then this link here, I think, is - the possibilities there are to - oh, by the way, this one kind of should be up there, namely, that that's to - this is only a secondary storer, in our parlance, right now. It may end up to be a tertiary storer if it turns out to be so reliable and you can think that that's as reliable as the paper you have around, or it's not going to burn(?) - because things happen to paper, too - you spill coffee on it, and all kinds of other junk - and the alternatives that we do there is simply put that link into a tertiary storer to hard copy and other systems for shared use. We've totally changed the structure of the system. Not like the one we've ever had before. We've never seen an animal like that in our hardware zoo. And, I don't know, that may be an alternative, and if you looked at the di.. - if you looked at Xerox's stuff upstairs - that (?) - that's their model of the world. Those guys are doing it. And if they - let's not tell Xerox about Shugart - can you - Shugart probably won't ever talk to Xerox about that. There's probably no communication out there in Silicon Valley at all, and

(light laughter) we can simply do all of our - continue all of our product planning based on the fact that those two guys will never get together. And the Xerox thing upstairs, which is a 5-megabyte hard disk, non-removable disk - they don't know what we know, that you must have removable media on the thing. We've got a history of removable media. There's no way that that system will work that they built a thousand of up there. All of those users can't be right (laughter), because after all it was a laboratory thing anyway, and when they really have to face the hard world instead of giving those to the White House and places like that, they will find out that it's no good. And so, what we've got to do in 1985 is go back to where we think the world is, which is - hey, let's go back and - it turns out - by God! lo and behold! it's exactly where we were twenty years ago. We've got a bunch of terminals connected to - through communications lines - to a computer and secondary storage and tertiary storage, with a couple of dotted lines that go out as communications options because these things have to talk to one another. So, in a sense, maybe there's nothing new under the sun. The thing that is new is, in fact, neat packages like Tom Orr has, which this is all really really quite small - it goes under a desk - and we have made a lot of progress, because in '65 we could barely get it in a room with air conditioning. Now we're sitting there and it easily fits under a desk with a hell of a lot more capability. So that's one way of looking - just wanted to throw that out as a way of talking about systems, namely - a little of the grammar - when you don't put lines between them, that means they're all in the same box. When you put a line between them, they're connected by a cable in different boxes. Anyway, it's a slight bastardization of PMS, but that's what happens when you take something from the academic world and try to apply it in a real-life application. And I think this allows this to go and brainstorm and look at a lot of things, maybe in a non-threatening way, and then look at what the alternatives are and what you can do with each of those. Now there's a taxonomy here that I've been pushing, not very successfully with Manufacturing, but it really has to start within our shop, which is really a packaging system type, and - I hate numbers but in this case - it's really based on scale and modular index, whether something is either, that is, the size, plus whether it's modular or integrated and that's an important thing because, again, when I talk with Manufacturing about "How do you build something?" somebody will talk about an 1134 as the epitome of the way to build something and somebody else is talking about MINC, and we have no way of focusing. Conversation about how to manufacture something is just a nightmare, because in some person's mind it's a terminal and in somebody's it's "How are we going to connect the cables on a big hydra structure?" And so, let me offer this one, which is Type One. Why say - why I want to bind it - when I bind something, it's putting a number on it, and that's about the worst thing you can do in my world, is actually assigning an index to it. And hand-held is Type One. It's an integrated system. Type Two is either a fixed or - I don't know - Two-A is a portable terminal like we saw with Field Service thing - those are integrated things. Type Three is really stackable. It's a modular and we saw Type Three, really a lovely thing, up there by Tom Orr. And if you like the lightweight version, you'll like the industrial design. The non-(?) version and the lightweight model, but if you've ever tried to compute on a bunch of styrofoam, it ain't that easy - you know - and the signals propagate somehow. There's no paper required, and - but, in essence it's a - Tom did an embodiment of the modular stuff that in fact - with the neat white box. The next version of our stuff certainly is going to get down in the white boxes, so we have to make the styrofoam versions first, but we've got the bigger versions now as things that are real, and they look - that looks like a neat thing. I think, whether you stack them vertically or horizontally matters. I think you can - probably there's - I don't know - maybe they have to be vertically stacked - and sort of Three-A is whether they're either a bench or a table or a desktop; B is whether it's on the floor - sort of a bottom-up design; and C is whether you put it in a cart as you did the MINC thing. And then Type Four are the rack things that you somehow bolt it in - and those are integrated systems, too. These, by the way, are the ones that, in a Manufacturing sense, are probably giving us the most trouble, because you're trying to pull a lot of different units that don't have as clean interfaces as one needs and make those all work as a system. And so, when we talk with Manufacturing, the world - somehow there's a view that everything is Type Four. My view is we're going to Type Twos and Type Threes and then - and we've got Type Fives, which is really collections of cabinets with big disks, and they are integratable. And then Type Six, which is the hydrastructures, collections of computers. So, this is another taxonomy. I'd like tro keep us saying, "Gee, are you talking about a Type Two system or a Type Three system?" So we are able to focus how our thinking went, rather than one person's three-hundred dollar terminal and somebody else's, well, three-million-dollar system. So it's a taxonomy I like to - let me skip past it - this one's a little messy - there are five - I think only five things matter, it turns out, but let me not get into that one. That's - how do I say stop? - okay. Now, let me turn the interrupts back on. Sorry.

Any questions?

I overloaded the channel or (laughter) or the tummies are underloaded. Probably it's tummy underload rather than information overload. (laughter)

VOICE: We have a scheduled one-minute break here.

GORDON BELL: Oh, there's a scheduled one-minute break.

VOICE: Go ahead, Mike.

(pause)

(END OF TAPE)

GORDON BELL AT STRATTON V: PBS ALTERNATE STRUCTURES

I made the mistake of not having GIGI-generated slides, and as a result they penalized me by only showing them at half-scale (laughter). Which I think is right: if you don't keep up with the times, you get the shaft. (laughter) But on my second talk on Friday, I will have GIGI-generated slides. They will be generated during the course of the next two - next three days. Somehow, Jan has said, "Oh, this is..." - STRATTON is a meeting that she says is mine. Well, I think it's not mine. I think it's clearly yours. I'm really impressed with the - with what's been achieved here in - I guess in many, many dimensions: the quality of the presentations, the demonstrations, just what we've gotten done, and in fact, it's so good that we're going to schedule a STRATTON in some of our plants so that we can get some of our products shipped that we have promised and scheduled (laughter), so there's going to be something called a Distributed STRATTON associated with each product now, in order to meet some of the deadlines that we've got. But seriously, I think that it sort of shows what deadlines will do, and also just what a lot of money will do in terms of going after some advanced development and having the right set of goals. It's exciting that we can now have some - we're seeing the signs of having better products, probably spaced longer in time, so it's really counter to what some of the arguments have been. I think we'll really make out a lot better that way than having a lot of products that are obsolete when they come out and I see some signs here of having some very good advanced products when they come out and I think that's a better approach.

Actually, this is three talks in one: one on Distributed Processing, which has to do with the - really, the root of Profession-Based Systems - because, like Dick's theme was balanced, I want to balance Dick's balancing by saying that to me the professional systems that I think are going to be exciting in the - probably in the near term and then even over the long term - really are predicated on a significant amount of distributed computing and we get two concepts mixed up here in professional - when we talk about the professional systems: one is low cost and personal - and the idea of having profession - and I don't think - I think we ought to keep those things separate. Not that we shouldn't strive for low-cost systems. But I think we should strive for cost-effective systems. I want to make sure we keep those ideas separate. The other thing is to - just to give some of my thoughts on Profession-Based Systems - and then the third talk is really on some of the human engineering details that I think are important about personal - or when dealing with people who have to use these systems which I - some slides I took last week of some of the systems that I use. Okay, the first slide.

So, really these are the - notice the - I hope you can - some of you can read them. They can be read in all of the remote sites because I told them, "Cameras, zoom in on those." I can read them. I hope you can. Fundamentally, that it - that the profession-based system and - I'll say for large organizations, because I think that's totally different than the retail products store kind of problem where you're selling small systems to a small organization. That, fundamentally, there's a hierarchy of computers, including personal computers, that is, computers associated with an individual. That's sort of the first talk, and - which is really cleaning up a lot of last - the tail end of STRATTON last year, which is where we left off about a year ago.

It's - the second talk is really on the first - I'll call it the first level of profession-based systems. Let's not worry so much about the profession - each of the specific professions - say for the next year or two. Let's get what I call generic applications dealt with, namely, things that everybody has to do in a large organization; that is, handle text, handle graphics, handle filings, handle communications, and in fact, Charlé's system there, I think, is probably one of the best I'v seen as - that may be the right kind of terminal for the - for dealing with this set of problems.

And then the third talk is really on "Hey, let's worry about some of the attention in terms of human engineering and the cost of a capability." Now, I want to coin something which is - which - I don't know - I haven't got the right buzzwords yet. I'm not a marketing person. So we'll - we need some marketing people here to tune the new concept out. But it's beyond cost of ownership. I want to introduce a thing called Cost of a Capability or Cost of Providing the Service. Because I think we all only focus on the cost of ownership and what we really ignore is what is the total cost of use. So maybe it's cost - let me say it's Cost of Use - it includes the cost of ownership but in fact the real dominant cost is having somebody sit there and look at the thing. So here goes -

These first level of generic capabilities, goals I'd like to have for 1985 - I want to probably revise those upward, given the SUVAX, because it looks like we can get there, but I just needed something to feed my imagination a little bit. And certainly, at the word-processing level, full-page, voice-input graphics, profession-dependent archives at document parts and all documents - that is, the ability to retrieve parts of documents and put them all together in a sensible way, not by having to call it out by with a programming language or by commands but by some other means, something - probably using something like the knowledge-based systems to generate how you put those together. User typesetting: I want to be able to do my own typesetting, to generate my own slides and that should be part of the capability - I don't want to go through the middle man of having to have other people generate slides.

Filing cabinet: yes - very good electronic filing cabinet, ultimately searching on the contents. So be content with searching for key words that you imbed in a document but then ultimately we really want to be able to go in and look through the document. The mail systems: certainly we want voice there, computer-conferencing to happen, and then I'll say personal videoconferencing in sight. A pretty good idea of how we want to do videoconferencing by 1985. Communications: certainly all the interconnections with the other computer companies which we - which I think our official party line is called INTERNET. PACKETNETs, which are the public network message - message networks - and then I'll say the old nets or the non-computer nets: phone and TWX and other institutions that have networks that we have to be able to interface to. We want the system to be able to deal with that. So, to me, that's the sort of first - that's the level that I think we want to be operating on. That's essentially the FORTRAN, the cost-enter(?) systems by the 1985 area.

Yeah.

Q: Gordon, are you proposing that we recognize voice and turn it into English?

A: No. On that one, I believe we're limited by our ability to sell - to provide systems by whether or not we have voice input. I'm proposing that we have some combination of voice/text recognition so that you can correct on the fly as you dictate. We've got a Demo, I hope, that shows what I mean by that, made by an unbuildable machine right now, but I believe that the speech recognition - with proper feedback, I think the speech recognition capability we have now at an isolated word will let us do voice dictation, because I really see that as the limit of use now. There's a whole set of culture - of people who are not going to type. Either they can't type or they will never admit that they can type. Let me say, my peers, for starters.

Yeah.

Q: Recorded voice: do you see that as well?

A: Recorded voice?

Q: Recordings of the voice.

A: Oh, I think that's an interim stage that we will deal with - voice answerback so that the thing will sit there and behave as an expensive phone answering machine.

Q: I was referring to actually dictating and sending that coded voice to some of the personal office(?)..

A: I think voicegrams are an - that's an interesting interim way to go. I think we've - clearly we've got to do a bunch of experiments here to see whether that's better - or getting the whole message up at once. I really think it depends on what the thing is. The whole - this was the point I just made on whether - is it personal or shared? And I want to point out that everywhere, with the exception of disks, the - all the economy of scale is disappearing. The only time you get better cost-per-something is with large disks. In terms of keyboards, they're hard to share - to make a great timesharing keyboard that we all play on (light laughter) or even a primary memory or a processor. Processors don't cost anything. The primary memories are getting so that they don't cost anything. We're limited by a tube and a keyboard that that are already fixed in cost that we can't share. So the strategy is one that we've outlined that Dick has talked about, essentially moving through - we've been providing very general tools, namely with these kinds - I'll liken it to the fact that we've been in the lawnmower business by providing wheels, gasoline engines, wood, and it's simply up to the users to build their own lawnmowers. You can build lawnmowers and cars and everything else with what we provide. But I think we're going to move into a much more general - from a general to the specific generic, and then go into the much more personalizable things, which is probably beyond this. So the problems are clearly distributing and sharing programs - programming and data among this network, and then the other problem of simply how do you use the stuff, because I really - I come up against that every day in the systems that I use.

Here's where we were - I wanted to sort of report how we've come over the last year.

This is roughly the slide that I put up a year ago, which was - this is the environment that we're heading to, that we're building. I'm happy to say that the Interconnect program under Dave Rodgers and George and Bill Demmer and Gary(?) has really come a long way. This was a - I'll say a virtual network last year and now I see signs of it being a real network this year. We are - we've got a lot of the details fleshed out. We start with the top level, the central-sited computers, and in fact, in my - can I have a pointer? - introduction, I describe these - this is also in the handout. So, I believe that computing will continue along the lines that it has today of the central-site computers, these local group-level computers, that is, a computer assigned to perform the function of a group, which turn out to be mini-size computers, and then going down to the professional personal computers, and personally I've been interested in how do you take a program and move that around dynamically through that network, or even how do you take any kind of a program, even on a fixed basis, and have it work with any of the next levels. So I think that migration and cooperation among the various levels in getting the right kind of operating well. Jack Gilmore's slide of trying to follow those - essentially at those optimum points is the name of the game here, because these really represent the three curves that he had. But the impressive thing is that we've got CI coming, we've got the NI that's just about to be - an agreement with Xerox is going to be announced in the next week or two, in terms of providing coupling within an organization, and then down at this level we're working on the communications. And then, of course, you see SUVAX here as a beginning to see an inkling of what one would provide there, plus, of course, all the proliferation of everybody building a personal computer for something.

This is roughly the slide that corresponds to what I have in the handout in terms of what are all those levels going to do. The only one that I really forgot was in fact the fact that what I think the central computers are going to provide more than anything else are communications among computers, and the communication-oriented services, like a central mail facility. Strangely enough, I forgot that. I think that these machines, in a sense, are going to be out of business in many environments and just be in a holding pattern, if people try to get off of them. They have to be there because the data is enmeshed in the computation and these big COBOL programs and you probably can never move it from there, and so there's - this is job security for many centers and I don't know - you know - but I don't think it's going to grow. But clearly it's going - nothing here is really - these are all specializations rather than economy of scale. The only thing that I think is really special, that you really want for the central facilities to provide, is the archiving of a lot of the file stuff, because this is where it really costs to have individuals be their own filing clerks, becuase they lose the data and they don't worry about backup, and again the cost per byte there is in the right order. The group level machines: right now I believe these are the most cost-effective because they are the best matching of what does the group do for that - there's a group function, like a design group or a word-processing group, and you get cost-effectiveness by having only a single program or kind of a set of programs for that collection of people. And with processing power, what it is - this is kind - you get enough performance here, and then you're really doing a very good matching of needs to resource. Right now these clearly win in almost every dimension when you - if you can cluster a number of people around a single function system, they win, and in fact, this is why many computers have come in. Now what we see is, in fact, these guys coming in saying, "Gee, why do it in a cluster or why do it centrally? Just give it to me and let me do it at a personal level, having computed now or - having computed and still computing at all of these levels, this one has its set of problems, which I want to - which we'll get into on the last - my last talk, because we tend to thing of them as a panacea, and what happens is we all end up spending time doing all of the things that somebody else used to do for us, probably more professionally and more cost-effective. And the ultimate in this is: take all the computers away from everybody, give them all TI calculators, and have them programmmed in octal or decimal, which is what you program in in those hand-held calculators, and look at what the costs are for that. Very cheap to buy but the cost of ownership is the most. So I think these are the functions that we'll end up doing at the personal level facilities, and the SUVAX is kind of my ideal of - slightly modified, of course (light laughter) - it doesn't recognize that several people may want to use the same personal computer, for example, and that your personal computer isn't small enough to take home, and you need a port into it, but it's a good start. I like the - the resolution on the tube's(?) just right.

The - this is a slide Terry gave me which is really the status of the Interconnect at this time, which is - the other - the first slide on its - actually, upside-down - with a bit of information added. The low - the terminals are here, small systems here, mid-range systems and the large systems, and what this shows is the various ways of interconnecting these various systems. This thing - this symbol is a CI symbol, which says all of these computers are tied together through CI. There's some - all the other slides - all the other colors, which are a little bit hard to read. I won't go into - but, in essence, these various systems are connected together all through an NI type of structure. It really is - with the proper transformation, it can be shown that the two map(?) ones - I'll leave that to Terry in his talk. But I'm convinced that they are the same and that we are building this distributed computing system. And I think this is where it's at; this has really been our strength, and we are, particularly the software engineering network is, really a testament to the fact that this stuff works.

Now I want to go back into my system, and this is one of the central systems that I compute on. This is the input communication link that we - this is why NI(?) - come in through the telephone to PK1, in through some patch panels, and hope that I didn't get cut off, lots of scopes to map everything to everything else. If all else fails, there's a T-bar here which takes all the communications on one system and throws it to another system. Then it goes into all the modems of the thing, and then finally it gets to the computer. So we think - we happen to think that the computer is the most central part of this, but in fact, it's really kind of peripheral to all the other boxes of equipment that - actually this isn't the system - the EMS system - this is a thing called RCS, which is sort of an electronic torn-tape system (laughter), which allows anybody to talk to anybody else. This is the EMS in PK1. But you do need RCS, because RCS is the only way you can get from one EMS system to another EMS system. So it acts as a nice transmitting unit. But - so, here we're seeing one of the central services that we all need and have to compute on.

This other slide that should be about twice as big is the engineering network that in fact exists today of having all of the sites from Phoenix to the Mill, Hudson, Parker Street, Reading, Tewksbury - Reading isn't on yet, but Merrimack - and having them all tied together through a DECnet link, so that - in fact, here's another system that one computes on, that have access to - through - a group-level machine today, and can go in to several of these - I guess I have an account on the Corporate Research computer, and one somewhere else - can come in - the whole system isn't shown - and can compute in a true - at a group level or through this central thing. Here are the details, unfortunately, all of the links shown.

Now I want to get into - sort of, this is the second talk: What is the PBS system that I'd like to see - this sort of 1985 - by 1985, having it out there. The physical system is: a processor, a primary memory of a megabyte, 100 megabytes of fixed memory, no removable media - I don't want to mess with the file problem - I've got floppies that I can't find my stuff on now, and I don't want to have cartridge tapes that I can't find things on too. (laughter) So, I want to get them off to somebody else who will take care of my files and not lose them. I believe we need two to four CRTs per personal machine, because there are a bunch of persons that happen to use this personal system, which is really a database. Black-and-white or color monitor, high-resolution black-and-white *a la* the SUVAX - I think that's the right thing as long as we can make it so that it can either be this way or that way and display full 8 1/2 by 11.

Q: Gordon?

A: You got it? Yes?

Q: One of the things that's bothered me is (unintelligible) SUVAX structure.

A: Right.

Q: (unintelligible)

A: No, they're all going to be mapped out. (unintelligible) large organization they're all going to be networked.

Q: (unintelligible)

A: What?

Q: (unintelligible) small organizations, too.

A: Yeah, but they're not going to sell them. They want - unless it costs two dollars they're not going to sell it. And let them beat their brains out trying - they can't cost-justify anything to their clientele. I want to build computers to people who really understand productivity and the minute that you get...

Q: (unintelligible) you don't have to work in that mode.

A: What mode?

Q: (unintelligible) you don't want to have to do it. You want to have to take the (unintelligible) When you're in a large network (unintelligible)

A: Right. Yeah, if somebody can figure out how to deal with the floppy problem why, great. I mean, that's...

Q: (unintelligible) work for Digital (unintelligible)

A: Oh, I'm not going to outlaw them. I mean, we did (laughter) - no, I mean I...

Q: (unintelligible)

A: Oh yeah, I just think it's a crock anyway (laughter). I hope we don't have any on our systems. No, I think people will get past that. I think the research so far in these kinds of machines has shown exactly that, and it certainly agrees with all the experience I've got in terms of the personal machines, of - get rid of that removable medium. The system has to know about that, and why turn us into (unintelligible)?

Q: (unintelligible) removable media...

A: Yeah?

Q: ...and they're so dead(?) that I don't want to have on the central system, because I know there's guys who can get at it. It's my data; I want to be able to keep it quiet.

A: Oh, for you people who really have got all those secrets (laughter), you'd better have personal media. Maybe you should never tell the computer, either.

But - and then, certainly, letter-quality printing, and being able to print what we see, so that means if we've got color we need a color printer(?) somehow.

A telephone dialer, a phone answerer, a voice output, and I somehow left off the voice input - something to really deal with the telephone. I really don't like the telephone. And I want this - the personal machine to really deal with the telephone. Certainly a link to the systems of the same type, and the ability to do things across systems. That may include actually both tele- and computer-conferencing, and videoconferencing. So the ability to share pictures across the network. And at the 10-megabit rate it seems to me that we get. We can transmit pictures on the thing, and particularly at the cost of current television cameras and the low-cost, low-resolution cameras that were coming, this is the kind of thing that we should have.

Yes?

Q: (unintelligible) why do we have to bother with an LI. I mean, the telephones (unintelligible) communicate over the other networks.

A: Well, there's - I'll talk - I've got some slides about why you have to bother with the telephone company. There are a couple of reasons why they are important.

Q: (unintelligible)

(laughter)

A: There is this other world out there. But (unintelligible)

(laughter)

And then, I want a link to the central system for filing, printing, typesetting, slide-making, distribution of documents for people not on the system, and probably accepting documents for people not on the system, some ability to get stuff into the system. And then, certainly, electronic mail systems and other systems, so that they're providing the kind of capapbilities that (unintelligible). And then, video and voice I/O - that's kind of the physical system that I'd like to see us head toward. So, if we were sitting here in '83 - '84, say, then, having this same thing and having the next level of voice I/O and having communication among the various systems, really having the systems know what we're saying and the ability to share...

Q: (unintelligible) micro... (unintelligible)

A: Micro - microfiche? I think microfiche is an absolute crock. I - you know - I hate the stuff - I mean - I - until they invent a zoom lens for microfiche, it's totally useless. Every time I - talk about compatibility - if you - how many people use microfiche to any extent? Do you ever worry about compatibility? Do you have any problems with compatibility? Do you mind turning the little different knobs? (light laughter) I never have the right lens - I guess I (unintelligible) microfiche from outside.

What?

Q: Assuming that it's better...

A: Assuming that it works, that it's better? I don't know. I just see it as an out-in-left-field kind of thing, that external documents will come into for a while, but there's got to be a better way. If we could get the data in magnetically, then why do we want to muck around with the microfiche? I just find microfiche very, very painful. Accessing and compatibility, and another thing around that you have to tolerate. I think we can have - as an organization, we can use microfiche for specs and stuff like that. That I like it for - DEC standards - to collapse that many DEC standards into that much space. But to further clolapse it, put it on a disk, I think we'd be - we might be better off looking now on microfiche internally - having some of the stuff that we use microfiche for internally, having that on the database. Having direct recall from that, from the mail system or something else. I think it's - I hope it's an anomaly, because it just doesn't feel right in this (unintelligible). Because the system doesn't understand it, so it means every time we deal with it it's got to be translated, and that's going to a pretty high cost. These are sort of the functional levels that I was - at least, the way I think of the profession-based systems. We've come from an era of which - let me call it the profession-based system route - that is, hardware-operated systems, languages, networking and databases, are given. And I think we're just getting to that point. As soon as we have DBMS-32 and a good relational database we will essentially have the roots to build with. And there'll be extra languages, and we'll argue vehemently whether an ADA will surpass PASCAL, and whether everybody should turn off to APL, and how much (unintelligible), but I don't think it's - those are critical decisions to going beyond that. So there'll be the roots. And then, this next level of modules are the ones that we really have to concentrate on the next few years: generic modules for communication. As, dealing with text, for filing, electronic mail, these office procedures and forms filing: the tickler file, and (unintelligible) file, processing, and then these interfaces to these other systems. That's something that I think everybody needs. I think even LDP needs it. I mean, I think that those people communicate pretty much the way we do. I mean they use graphs, I think. The last time I was a scientist I remember doing that. But engineers also - actually, engineers and scientists use the same display forms and even the same natural language. So it's conceivable that we can make a text-processing system that a large number of users (unintelligible). I think even accountants use English to communicate, when they're not using numbers. But that level is something that we're into now. Now we move up a level, and we get into a whole bunch of - let me call them generic professional - general professional discipline modules. These are the ones - Engineering is one such discipline. So the best analogy: this is a Dean, this is a Department, and this a Compartment. And namely, that Engineering, Electrical Engineering, and then RF Circuit Design, so that - there's a lot in common with that, but as you go more and more specific, then this commonality disappears. It all started from Calculus and then worked down to a specific set of laws governing different physical behavior. And the same holds, whether it's in the School of Business or School of Arts, or what have you. But this is what I think - this will come after we've dealt with this problem of communication. Essentially this is just another (unintelligible). I'm really using that for office automation. Definition - this came out of a report. In essence, it

> says "If the evolutionary use of word processing and electronic mail to improve office productivity through a bunch of different kinds of - really substitutions - we're substituting for typewriters, for torn-tape systems, for common carriers, for paper files, for keypunches, and so on. And then, the office of the future is a - I was amused - I (unintelligible) this out because I think it's in the definition. So it's a use of equipment which allows drastic restructuring of the office work among a different composite working force. And I paraphrase - I read a long report - it wasn't that long - three pages - they were trying to describe what Office of the Future was, because I'm always looking for a definition of something. And, in essence, paraphrasing it: if a secretary uses the equipment, it's office automation; and if we all use it, it's Office of the Future. And - oh - (laughter)

This is - now this - this is the third talk. (laughter) This is (laughter) - we finally - I was looking around for - in terms of - Charlé's got the person he wants to design for. I wanted to find me an average man (laughter) and so - yeah - and so - it turns out we finally got the blueprints (unintelligible) the way everything matches. My boss normally gives me a hard time, because when I say, "Hey, we want to put this capability," he says, "The trouble with you is, you can't - we can't base it on your thing, because you're not average. And so this is a proof that I am (laughter) (applause). Now we've got a bread(?) prototype, so when I say - when you say, "Hey, who are you designing for?" you say, "Well, the average person." Well, boy! am I going to come on strong, because I found him. (laughter) Okay. So, the average person last week - I went through the average person's computing last week. This is a typewriter that the average person has (light laughter) and it's bought because it was so pretty, actually. Every time I buy one of these things - it's an Olivetti, and I did it just because it's a classical thing - it'll - it's actually for the museum. It's mine, but it - but - the interesting thing about that, or I guess I'd talk to Dick last week about this, but the trouble with Olivetti typewriters, they have the - well, aside from Italian design (light laughter) they - you know, they really look nice - but then you sit down to play on them, and they really feel crappy. And fortunately, Olivetti is dominated by designers. There are no engineers there, and so we're safe. (light laughter) That's one extreme, and then you go to, essentially, TI, which is essentially - there are no designers there; there are only production engineers, and if it isn't cheap, it's not good. And I think we've got a wide open market. All we have to do is make a (unintelligible) look nice and have it work. That's the unique market. (laughter) And I think we can get that market. All we - just - only two things - but Olivetti right now has got the design market and TI's got the <u>schlock</u> market for cost. But I think the market we want.

This is what the average person's word processor looks like in an office environment. Note an IBM typewriter over there, that still has to exist. There are two modems, because the 1200-bod modem and the 300-bod - 1200-bod is 1200-bod only and the 300-bod is 300-bod only, and never the two will be compatible with one another, and I happen to be on two systems, and it's simply a matter of - I'll show you how you change from one to the other. It's a very simple (laughter) operation. And then - no - and note all of - oh, and this is - people worry about storage of (laughter) - can they have the whole phase(?) Notice the two-drawer filing cabinet and the printer up there. (laughter) That's where the printer is stored. It occupies a predominant altar in my house, and - I will - believe me, when people say, "What do you want - what about hard copy?" I say, "Oh, God, I would like to do anything to get rid of hard copy," because it - I just can't stand it looming over me like that. (laughter) And besides, does anybody know - when you turn that printer on, it adds about 6dB at 125 cycles. And - oh, the other measure I - the average person - in my house - actually, the noise level is under the sound meter, so it was less than 25dB. Then, when I get everything tuned up and turn the radio up it runs up to about 40dB, then up to 43dB with the printer off, and then 48dB. But that's on the A scale, and if you flip over to - look at the cycle bands, there's 62dB at 125 cycles. So - we may get by with the right kind of - we'll get our lawyers to work in Germany to make sure that we can pass the test, but it just ain't very comforting and so probably what I would like to do is move the printer somewhere. And, sure enough, the cables are long enough to deal with it.

Documentation is really an important part of (laughter) of systems, and where do - and there's, of course, lots of room to store documents. This is my filing system for documents. There's a convenient shelf provided under the word-processing system (laughter), where I throw everything, and in the last resort, if I can't do it by trial-and-error, I will get the information out of the manual somewhere. There's only about - there are five manuals there, and about, oh, five hundred pages, and that usually only takes me ten minutes to find the stuff. So, documents aren't really cost-effective. Oh - here's how you change the modems: you simply get up behind (laughter) or pull the thing at your own risk - pull the thing out because you're afraid a few of these cables are a bit fragile, unscrew that EIA connector there, move it over to that, and then there's a little knob back here, conveniently located (laughter), that you can deal with by a TV repairman. (laughter) A mirror, but I don't happen to have that option. But it would be nice to be able to change the speeds, particularly in that environment. So, Ken has his slide of Engineering or Marketing. This is mine. Note: here's Average Man sitting with knees crouched at the terminal. And all of these places of little tags on there: that's the ready reference manuals to deal with the three or four systems I use of tight dollar signs - well, we don't have any dollar signs and things, but log-in, remembering what the passwords are, and project numbers and so on, on the various systems. And then, also, all the various protocols of - there are only three different mail systems and they are all, of course, totally different, with how you speak to them. And then, this is getting enmeshed in the system here (laughter) - there are a couple of phones associated with these terminals, and normally the phone wires are entangled in this desk chair. This is me reflecting at the terminal (laughter). If you - I don't know - do you all often reflect at the terminal? (laughter)

VOICE: I glare at it.

You glare at it. (laughter) Well, it glares back. I wasn't glaring; I was reflecting that day. Actually, it was - oh, and things are better when you go to a shared system, because you don't have to take care of it any more, and that's the best thing about the shared system, except there's - the thing on the - the modem on the left is the GANDALF switch that can go to several systems. There are a couple of other - there's 300-1200 mod modems. Then you simply get up from your desk about 50 yards away, grab the terminal, go over and dial this thing in a very convenient cost-effective way, and - it got its cables too, of course. This is an advertisement for NI, in case people did - I hope it's better than this, but - this is a 248, full-house, 4 modems, 4 printers, and 8 terminals. So there's a hell of a lot of cables coming out of that (unintelligible). And then - now, here's a nice - now, this is why we need the telephone company. I wish you could see this a little better, but that black ribbon cable about that big is simply - now in the Mill you can get away with it, where we don't really control the esthetic qualities, namely, Field Service can come in with hammers and all kinds of units and put in this very black, wide black cable and run it up and down over the ceilings. I have - there's a lot of offices - ugh! - actually, I went to IBM, Armank, actually, a week ago, to look at some of their historical stuff, and - I just - they had this wonderfully - designer award-winning building there, and with people around that looked like IBM people running around. (laughter) And you didn't want to touch them - nice robots, and (laughter) they - you know, they wouldn't let us install a system like that, where with this big black cable which you simply tack up on the ceiling and run down the thing (laughter). So why we need the telephone company is to install all of this stuff, because we really don't know how to install cables like this. This was our - and this is a kind of a lesson in compatibility: here's the phone thing. At least it comes up under - at the floor. There's the big black ribbon cable that goes into the printer. This was our last aborted attempt to deal with terminals. That was still left on there. This is a four-wire phone jack with - but that's probably the one we would want to use if we'd had the phone company install it. It wasn't phone company installed. I mean, the phone company installed it where you can't see it. In our case, the guy installed it where it's the most convenient for him to put - he doesn't want to get too close to the ground, because if you're trying to put screws - wood screws - in, you don't want to do it down there. It's at user-level height. (laughter) And that didn't work, and we've now switched to another system. But frankly, I would like to go back to - this is the back of - Dick Schneider says that you're not supposed to do that - put coffee cups out - and then you also - there's a tilt mechanism here (laughter). It's two books - that way. And then we switched to a new jack back here. I want to switch again to another jack. I want to switch the phone company jack. Really want to get there. So that that way we let the phone company install all the cabling in the machines, and if you don't like that, you can go to Radio Shack and buy these little four-wire pinjacks and connect the stuff, if you've got to do it yourself. But certainly having 15 - having the EI connector, and having that go over to the central machine - that's just not the way to run the terminal. So let's go to phone-compatible interconnections.

Mary Jane reflecting at her terminal. (laughter) That's - and in fact we both - I want to indicate - there are several of us that use the same system and - I want to put that plug in. Here's a document that was edited. This is a printed document. Notice the big block letters, and then there's some symbols down here. And now, as you map that into a word-processing system, that's all you can see, so this is a pitch for a full-page graphics, and being able to type all the special characters and fully general stuff. This is a thing I've learned to which is - almost learned to do, using EMS and talking on a speaker phone at the same time. (light laughter) So I found out that I can - or listening on the speaker phone, I can get two channels of input but it's hard to do two channels of output. When you get two channels of output going, this is what happens: the guy on the other end has this dazed look on him. This is Dick, while I was talking to him, because there are these lapses of - when you get the two channels mixed up as you're(?) typing (laughter). But it does say that you can use two channels simultaneously. And then - I was away for a while and the trouble with EMS is that I came back and there were 24 unread memos staring me - that was (unintelligible) for a day or two, and then - but that didn't take away from leaving the office with two briefcases that night. So somehow we haven't learned to deal with the problem. Electronic mail doesn't solve - doesn't reduce all of the other communication. You still get paper mail.

I didn't go into some of the other things I've learned about the EMS system or the various systems in terms of just the speed that it operates, and the fact that if you're on electronic mail, that it's probably more cost-effective to print out the messages, not - if you're only running at 300-bod then have somebody else - have the - have all the mail printed out for you, then do it on paper and then have it rekeyed. That's better than sitting there looking at mail coming out at 300-bod. It just ain't cost-effective to sit at 300-bod, and certainly it's one of the things I've learned over the last few months. You really have to be running at 1200-bod for this to be cost-effective. There's a good - satisfactory - or satisfying effect of being able to press carriage return and having - knowing that you've deleted something and it's gone away. It's like throwing something in the mail - in the wastebasket. But, I too am concerned about being able to get to a cost-effective - whether this stuff is really cost-effective - we've done a poor - even though we've got an incredible - incredibly large experiment internally, we don't have very good data on how cost-effective it is.

I want to close by one comment by Lord Kelvin, which I fortunately found in a magazine this morning: "When you can measure what you are speaking about and express it in numbers, you know something about it; but when you can not measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind."

Right now we're building some things - I would like to know something about them in numbers - like how many dBs, and how long it takes to get through a page, and this kind of thing. So there's - in order to sell the stuff, we've got to prove that it's really cost-effective for the users. And there are times when I reflect back I'm not sure that it's so cost-effective. So - that's it.

(applause)

Any questions?

Q: Gordon, in the beginning of your talk, you indicated that you would have a strategic shift form what I interpret to be tool-building to applications-building. But later on it appeared that we were building those out of the same components. What is your message about - what do we engineer versus what do we market?

A: What do we engineer...?

Q: Today we engineer and market tools. (unintelligible) applications...

A: No. I think that - I believe that we're getting this next base of tools done. That is, the databases and the languages, and now - the next thing that I think we do is almost the next level of tools. To me the word processing in a sense is more of an in-use(?) rather than a tool. That is, you don't have to pry with it, but it solves a problem by itself. Rather, it doesn't require a program before one can start using it the way it was meant to be used. So I think the shift iis simply moving to a next level of integration, and it's one that where we really (unintelligible) that everybody must have.

November 5, 1979

Dr. Bruno O. Weinschel
IEEE
One Weinschel Lane
Gaithersburg, MD 20760

Dear Dr. Weinschel:

Sorry to have taken so long to answer your letter of 20 September 1979. I have a few comments:

1. Willenbrock seems to have nicely assessed the situation. Your own background material also seems relevant. You're right we need a major change. NEF may be the answer.

2. I hope the Perkins comments were not reflecting the policy or feeling of the NAE membership. I would think that NAE might address the issue of a National Engineering Foundation, because after some thought I can't reject it so easily. The need is clear, if we assume the goals are an effective supply of engineers and encouraging the production and flow of relevant knowledge. Both are missing now.

3. The problems you raise are milignant and deep rooted in the pseudo-science funding agencies (NSF/DOD/NIH) and their university recipients. I see no hope, short of the revolution, which must come

when the current system ultimately collapses (within 10-20 years) because it has no underlying economic support base. We're giving up on balancing our budgets with respect to: energy, manufactured goods, agriculture, and natural resources.

4. In high technology industries, much of the change occurs more rapid than our traditional decision-making processes, hence policies built to address this will really turn out to be irrelevant or completely constraining. For example, in computing we have maintained a positive balance of payments because the government hasn't entered the fray with constraints except to be an intelligent buyer of machines. However, as the purchase of computers for weapons, command and control there is another story. Here, they have a random collection of non-standard, obsolete, vanity junk because they've acted as a bureaucracy used to managing slow changing technologies (e.g. rifles, vehicles).

5. The issue of overfunding (and the wrong funding) can't be too strongly emphasized. The big spenders are health and military. The Japanese fund neither of these. Their life span is longer and health care costs are lower. When you look at R+D as a % of GNP and throw out military R+D because there is no direct output to help the rest of the GNP, we're only spending half of what Japan is! Still, this might not matter, (if we assume we're twice as good) except that the irrelevant needs take the people who could do something useful and relevant in science and engineering out of the supply base.

6. It is important to note that the semiconductor industry has been largely self sustaining in terms of R+D. The technology progress rate has been high. Recently ARPA has entered the fray with money, and I'll wager the rate will slow down because it will be too easy to get funds, versus building useful devices and the supply of people and research will dry up. The added paper work of federal research alone has to be costly in terms of real work that won't get done.

7. Senator Stevenson has proposed to put money into stimulating innovation, small businesses. I hope this will not pass because it too will simply be a dole hole where people have to come to start things. We have to stop having faith that a bigger government, in effect, has an increasingly large brain, and hence can be capable of managing everything. Here again, I assume the funding will take people, who could be useful, out of the work force to be administrators of the program and to be inventors for the bureaucracy rather than manufacturing or use.

8. I think we're losing good engineers to Business Schools because somehow people think this training is needed for industry. In actuality we lose two ways: less engineers; and learning to make a quick buck by believing that the whole world is return-on-investment oriented. The later causes people to go for short term results and to minimize investment and buy

all the products in Japan, transferring our island into a mere warehouse, sales and consumption place.

9. Loss of Manufacturing Engineers also took place when the discipline moved into the business schools. Again, this kind of engineering is a dirty, and presumably less respectable discipline when compared with economics, marketing, etc., which are relatively content free.

10. Japan does a good job of managing the whole science/engineering/manufacturing process. We can learn from them. I've written some comments on what we might learn in the attached paper which I hope might also be relevant.

Good luck.

Sincerely,

Gordon Bell
Vice President,
Engineering

GB:swh
GB0005/53
Attachment - IMPRESSIONS ON HOW WE AND THE JAPANESE ARE CONVERTING U.S. INDUSTRY INTO DISTRIBUTORS

CC: Courtland D. Perkins - President of NAE

January 23, 1980

Bruno O. Weinschel
IEEE
One Weinschel Lane
Gaithersburg, MD 20760

Dear Bruno:

In reply to your letter of January 11. Please find enclosed a copy of my paper, "Innovation in Japan--A Lesson for Us?" This paper was published by Dartmouth in their Engineering Bulletin. Feel free to distribute as you see fit.

Sincerely yours,

Gordon Bell
Vice President,
Engineering

GB:swh
GB1.S1.39
Enclosure
November 1, 1983

Professor Dennis Allison
Stanford University
Computer Systems Laborator, ERL 450
Stanford, California 94305

Dear Dennis:

Thanks for setting up the show for the Museum. Here is an abstract:

The "SEE IT THEN THEATRE": film clips from plug-board programming on the ENIAC in 1947; programming the EDSAC in 1951; Edward R. Murrow interviewing the Whirlwind and J. Forrester in 1951; and the first A.I. written and acted Western in 1961. Gwen Bell, Director of The Computer Museum, brings the Museum to life to you on celluloid.

Gordon Bell, museum-user/collector, then tells "WHAT I'VE LEARNED FROM THE MUSEUM". Gordon's first experience in learning from history came in writing Computer Structures with Allen Newell in 1968. Then, after he returned to Digital Equipment Corporation after teaching Computer Science at Carnegie Mellon, he was Vice-President of Engineering and sponsored the development of The Computer Museum. Presently, Gordon is Chief Technical Officer at Encore Computer Corporation and a Board Member of The Computer Museum.

Sincerely,

Gordon Bell

GB13.4

August 16, 1983

Edward A Torrero
Senior Editor
IEEE Spectrum
345 47th Street
New York, New York 10017

Dear Edward Torrero:

Just a brief comment on the panel discussion, regarding impact.

The nature of the university and the engineering profession should change with the new generation in what is potentially more radical than any of the other impacts on other parts of society. Computer Science as currently formulated deals mostly with parts of software engineering, as such other departments are very slow to adopt and embrace computing. Computing appears to go through several stages:

1. fascination and learning,
2. use for a fixed function (eg. a computer game, word processor, simulation game, spread sheet),
3. understanding by direct, deeper application by user who codes problems by some form of "programming",
4. total embracing by adopting the science and then teaching this science to students so that they may "engineer" using computers as components in the systems they build,
5. encoding of a field's knowledge so that the computer becomes a supplemental partner in human problem solving as with expert systems such as those used in geological surveying or medical diagnostics, and finally
6.elimination of books and papers as we know them to be replaced by full electronic transmission, storage and

constant, active processing of knowledge.

Only a few parts of the electrical engineering discipline operate at level 4. Most engineering disciplines are at levels 1 and 2, and there are a significant number who are totally computer illiterate. It's important to transfer the engineering ability from the computer science department so that it can concentrate on science and the faculties of engineering and management can move to a higher level of use.

I believe that no university has recognized that their charter is this broad discovery and dissemination of knowledge which will enable us to attain a point where our knowledge is completely active (level 6).

Sincerely,

Gordon Bell
Chief Technical Officer

GB8.11

IMPRESSIONS ON HOW THE JAPANESE HAVE CONVERTED WORLD INDUSTRY INTO DISTRIBUTORSHIPS -- CONCERN NOW FOR SEMICONDUCTORS AND COMPUTERS

Gordon Bell
Vice President of Engineering,
Digital Equipment Corporation, Maynard, Mass.
Professor of Computer Science and Electrical Engineering (on leave)
Carnegie-Mellon University; Pittsburgh, Pa.

We must be impressed with the intense drive, technical and manufacturing ability of the Japanese. As an island with few natural resources, and only very bright, hard working people they have set about and accomplished the market domination of virtually all manufactured consumer goods including the components and processes to make these goods. Currently the U.S. has a dominant position in computers and semiconductors. However, there's no fundamental reason why the Japanese won't attain what appears to be a basic goal to dominate these industries, given their history in other industries and helped by our governments.

On a first visit to Japanese computer and semiconductor companies, universities, and a government R&D laboratory, I found them relatively open. This was in contrast to my former experience as a computer science researcher with their one-way scientific interchange and being an information sink. Perhaps their openness is because they are so far along with good products, and their position so secure. Their competence, hospitality, and "apparent openness" made me quite fond of them; but I now fear them more than ever.

Based on a simple system of three islands, two supply energy and manufactured goods and one consumes from the other two, the only apparent stability occurs when the two supplier islands "own" the consumer island.

Furthermore, I question whether we (the U.S.) want to be owned in this fashion, and therefore state we may have a problem. It's not the intent to give solutions here.

Forty-odd reasons are given in the form of "feelings" to support this domination conjecture. These reasons vary from a belief that we are comparatively lazy and greedy to the fact that their government and companies get together and systematically plan to dominate an industry.

The reasons for their ability to dominate are formed from my observations, but like the Japanese, taken freely (generally without credit) from other sources with an attempt to make a better, more complete end product for industrial and government users.

BASIC WORK ENVIRONMENT IS STRONGLY COMPETITIVE WITH PLANS (STRATEGY) AND DRIVE (TACTICS) TO DOMINATE AN INDUSTRY

The new recently announced Fujitsu M200 computer appears to be the highest performance, most reliable, plug compatible 360/370 yet announced. The technology originally from the Amdahl Corporation, was improved and made manufacturable in one computer generation of about 6 years.

The Japanese industry and government team is fundamentally more competitive than any other nation. Competition is built into their culture and reinforced by training. For example in mainframe computers they have carried out a plan to build a successful industry, unlike many companies and countries.

We, the computer and semiconductor industries see a clear pattern based on the Japanese performance in textiles, steel, radios, sewing machines, typewriters, quality cameras/optics, small cars, TV, tape recorders, watches, calculators. (Note the progression from low technology/simple commodities to complex manufactured goods.) Their current position in semiconductors and semiconductor-making equipment indicates they are well on plan to dominate semiconductors as a base for electronics, and computers. This will also be the base of products that are electromechanical and will become more electronic intensive.

There is an amalgamation of the Japanese within an industry which creates something that's often referred to as Japan Inc. The Japan Club is a better name, because there's at least a show of competitiveness at the market level. Not only is MITI supportive, they also have relatively autocratic power. More importantly, they interact with industry in a helping way.

They identified and encouraged DEC's early imports in order to build their own industry as described above. For example, one of our important interactive data base systems, MUMPS* was used in six applications. At this point MITI funded the development of MUMPS on a Japanese Mini. In early July a Japanese researcher asked me for the internal architecture of MUMPS, through an academic channel, in order to study its structure from a so-called computer science viewpoint.

While there isn't direct control an connoted by our phrase "Japan Inc.", there is clear collusion and planning among the government, and companies. Not only is collusion among companies illegal in the U.S., but furthermore the role of government is one of discouraging and being an adversary to industry. In the case of IBM, who developed the mainframe computer market, both the U.S. and Japanese governments are determined to destroy the company and set-up the industry for an American distributorship of Japanese products. IBM is a key resource and should be protected as such.

The Japanese government and companies actually plan to win! Such thinking is totally foreign to us. This includes basic strategy setting among the players to segment and go after various markets (e.g., Fujitsu/Hitachi are IBM System 370 plug compatible -- Hitachi is concentrating on the internal Japanese market against IBM Japan and Fujitsu is concentrating on exports). The companies can plan and talk with one another and do, but certainly compete intensively with one another within a limited domain.

With computers, the Japanese strategy has been to couple individual companies to U.S. Companies for technology acculturation and then to pair companies to build the same compatible machines in a quasi planned, competitive fashion. This is a well-known management technique to make technology gains quickly.

Overall, MITI appears to be very strong and competent! The goal of MITI and the Japanese computer companies is a strong, dominant industry! This is in contrast to our standard regulatory 9-5 bureaucrats, who seem to work for either security or power. However they have no real way to make anything happen. Nor is there any measure of their performance. Who believes that our Department of Commerce has anything to do with our position on imports and exports or any understanding position on standards or technology?

Reicshauer hints at the fact that MITI has high quality people, as opposed to our articulate ones. In addition to the right longevity, power, and process, maybe they segment responsibility and measure results with reward based on performance, as for example "winning" in a trade area. In a few samples, I believe it's simple people quality, and the right process enabling them to accomplish something. Being responsible and measured may be the key variable. Here, this suggests we could probably eliminate the Department of Commerce and have no real change except more output, and less government spending on hand-wringing trips to Japan. For starters, a clear change of management and a clear notion of old fashioned responsibility is in order at the Department of Commerce as we see trade deficits increase with no plan in sight and only a trade trip to Japan by Juanita Kreps as a palative.

As the head of our Osaka sales office, who attended graduate school at the University of Kansas, put it: <u>the Japanese live to work versus the American need to work to live</u>. Thus, there is the basic tactical drive to back up any goal to form a dominant market oriented around a company. This is instilled at birth and trained. Work is a central theme.

A company screens its hires carefully since there is a lifetime commitment. In contrast, a recent Intel ad claimed that no interviews were required for hire. Companies only get graduates from certain universities, more extensive than here.

Housing is provided for the workers and they have what amounts to a lifetime contract. This is bad if a person's incompetent, it also means that it's hard to breathe different life into an organization. On the other hand, turn-over is low to non-existent and a team spirit clearly develops as the various members learn to work with one another.

The <u>pressure to work is fed back producing more work output</u> since everyone is working. Unemployment is non-existent and this creates an environment where non-work is unacceptable. Recall how acceptable unemployment is when the U.S. unemployment starts getting high.

<u>Only half the work is done in large companies</u>; <u>small shops buildsub-assemblies</u>. Since large organizations tend to become inefficient and lithargic, they farm out stable sub-assembly production to small shops on a competitive basis. This limits the organization, provides a buffer, and gets the costs down by a buyer-seller relationship as opposed to operating through a large

bureaucratic organization that typifies governments (invariably large and unbounded), large corporations, and large universities.

Their physical condition certainly reflects working, and they have the longest life span in the world now. On one hand there is much smoking, but an anti-smoking campaign is in progress. However, nearly all Japanese are trim versus being basically overweight. Their diet is conducive to trimness and better health, I'd guess. Although alcoholism is supposedly on the rise, the consumption in business I saw was certainly less than in the U.S.

Invention occurs, though they have large, stable companies. Unlike most large U.S. corporations which lose entrepeneurial drive and operate in a stable non-risk taking fashion, the Japanese structure encourages risk because the entrepeneurs can't escape. That is, the large corporation is the only "game in town". The inventors and entrepeneurs of American business escape large organizations in order to start new small businesses. The effect of mixing the two types in their organizations causes continual reform, rejuvination and risk taking.

I believe their manufacturing output is at least equal to the U.S., even though they have half the population. Numerous factors contribute: investment, equipment, less-overhead at the company and by society, work ethic, more output per person over their lifetime and good management attention to personnel details.

PRODUCT DESIGN IS NOT EGO DRIVEN, BUT IS A PLANNED ACCULTURATION PROCESS

There appears to be less individual egos, although there is a strong group ego! Japan has acculturated customs, technology, etc. from everywhere for centuries and knows how to do it. In the 16th century they apparently set up manufacturing of guns/gunpowder in 18 months once the Portuguese brought them in. Any good idea is fair game, subject to very strict legal patent technicalities. Having adopted an idea they fundamentally understand and improve it.

They seem to be less oriented to technology for its own sake versus what it can do for them in the long run in achieving a particular market domination. For example, they moved more rapidly into semiconductor gate arrays for their computers earlier, quite likely under Gene Amdahl's influence. The computer industry has been unable to get the U.S. semiconductor industry interested in this technology until recently, hence we lag in this basic technology. In Japan since the companies are larger, corrective action can be within a company, or if needed, MITI may force and rearrange priorities.

They clearly think both product and process together in what is a long term view. Again, here they're competitive and they orient the processes to: Quality and Volume (for growth), and finally Flexibility for fast turn-around in order to support and tune the volume. As a quirk, the predominate customer for semiconductors has been their telephone company. Unlike the U.S., where ATT has a fundamentally non-competitive, captive high cost semiconductor supplier. The buyer/seller relationship here has forced a concern for quality that would not be met by simpler consumer use (in calculators, radios and TV).

All of the computer manufacturers have acquired their technology over a one or two technology generation history (approximately 10 years) of dealing with U.S. manufacturers either as a joint venture or under license: Fujitsu (Amdahl/Siemens) and Hitachi (RCA); NEC (Honeywell, GE, Varian) and Toshiba (Honeywell, GE, Interdata); Mitsubishi (Xerox) and Oki (with Univac joint venture); Yokogawa (HP); and Nippon Minicomputer (DG). In all cases, the technology has been improved in terms of quality, performance and manufacturability. The case of Honeywell is ironic. The high performance technology selected for the mainframe is now manufactured more effectively at NEC.

The agreement between Fujitsu and Amdahl Corporation appears to be a good example of the classic Japanese computer acculturation process even though only the first two phases have been carried out. My simple understanding is that in the late 1960's Gene Amdahl explored the basic technology for high performance IBM computers as head of IBM's San Jose advanced development laboratory. As an IBM employee he tried, unsuccessfully, to get IBM interested in building high performance machines. He formed Amdahl Corporation and proceeded to develop the technology. For various reasons, more capital was needed and Fujitsu bought in as an owner. As part of the agreement, Fujitsu got the manufacturing rights to and became the manufacturer for the Amdahl line. In return, Fujitsu was able to use the same technology to design and manufacture computers for their Japanese market. At the beginning of 1978, both Amdahl and Fujitsu have announced their latest computers based on the Fujitsu processes and production facilities.

It appears now that Fujitsu has built a higher performance, incremental performance upgrade, and higher reliability machine than either Amdahl or IBM have so far announced. As an IBM computer it is unorthodox because it is a multiprocessor. Not hampered by the IBM thinking process and appealing to a buyer versus rental market, the Fujitsu machine could have a significant edge, because it also gives users new capabilities that they probably need. Care to bet on the position in 1982?

The current computer manufacturers have a complete line of peripherals, and test and manufacturing equipment, taken from copying and improving counter-part U.S. products. In one very quick casual trip through a computer factory I was able to count a dozen "copied and improved" devices.

For products under license, there is always incremental improvements. Product alternatives range from reverse engineered look-alike, through radical improvements based on key ideas or patents (e.g., video tape recording). Occasionally the Japanese buy U.S. manufactured production machines (e.g., the Gardner

Denver Wire-Wrap machine) where the manufacturer won't grant a license. In general, the emphasis is on making products they can export, versus making manufacturing process equipment that can not be exported.

In one case, (DEC) developed a semi-automatic wiring machine and manufactured a few for internal needs and licensed a U.S. manufacturer. Neither DEC nor the licensee had bothered to patent and protect the design. The Japanese version of this machine appeared in several computer factories. There were Tektronix look-alike scopes and the ideas for a laser printer came from IBM, modified by a Honeywell product and teaching. In the case of disks, they use the reverse engineering techniques (also used by Memorex, STC and Telex) to produce disks identical to those of IBM. They have made improvements in disks technology and will export either head and surface components or complete disks. Geographical separation is not a hinderance, it is a benefit because they are excluded from U.S. laws. With the advent of the IBM 3340 disk organization, NPL (Nippon Peripherals Limited) was put in place to make a comparable product. This "engineering process" required 15 months and produced a disk that was "identical" mechanically. In fact, when comparing the two drives, one might conclude that both drives were made from the same drawings!

A Chronology of Systematic Domination*

"Four phases are involved in the Japanese assault on a market. They include the initial development of a domestic industry, an establishment of an export market base, significant market penetration in the foreign market, and ultimate market exploitation.

I. Development of a domestic Japanese industry. The Japanese industry is developed and grows rapidly. A number of major aspects mark this development. These include:

(a)

Market control. Imports limited essentially to zero. Only a few major manufacturers are permitted. Prices remain significantly higher in Japan than in other competitive markets.

(b)
Borrowed technology. The Japanese borrow heavily from foreign technology, including a large number of purchased licenses and patent rights, and wholesale reverse engineering.

(c)
Vertical integration. During this phase, the Japanese vertically integrate their manufacture almost totally.

(d)
Major investments. This period sees major investments for modern plant, equipment and technology, both for the final product and throughout the vertical chain of manufacturing. Continued heavy research and development and investment expenses keep manufacturing up to date.

II. Establishing an export market base.

(a)
The establishment of widespread sales organizations throughout the United States, and, perhaps, elsewhere.

(b)
A thorough researching and understanding of the foreign markets and their various facets.

(c)
Establishment of a reputation for quality products and reasonable prices.

(d)
A limited focus, especially in those markets less attractive to domestic manufacturers.

III. Major market penetration. Major market penetration occurs usually during an economic downturn in Japan. Previous efforts by the industry have set the stage for them to be successful in this endeavor. It is marked by the following considerations:

(a) Cooperation among the Japanese companies with respect to models, prices, and markets.

(b) Focus at the mainstream of the foreign market.

(c) High inventories because of poor markets in Japan, i.e., an export push at any cost is highly expedient to the Japanese manufacturers.

(d) Extremely low prices to the mass market to gain high percentages of market share rapidly, i.e., a knock-out punch to the domestic manufacturers. Modern plants, reasonable costs, an established export organization, and good reputation set the stage for success.

At this time, marketing muscle is established. Not only was the export market share large, but the domestic market remained closed. It should be pointed out that this major market penetration had been made by a <u>combination</u> of factors, as outlined. The greater marketing muscle allows the Japanese manufacturers to subsequently gain the profits of their long investment.

IV.<u>Market exploitation</u>. This period is marked by higher prices -- often higher than domestic manufactured models. However, the higher prices are often more than offset by perceived higher quality, both real and imagined. There is also continued cooperation on prices and markets, as well as continued limitations on imports to the domestic Japanese market."

JAPANESE DESIGNED PRODUCTS REFLECT A CONCERN FOR QUALITY, PERMANENCY AND NEED

Product design in Japan seems to have a better tradeoff among quality, product cost versus lifecycle and human usability. The lack of structured marketing as we know it provides the opportunity for products to be designed on the basis of user need rather than filling a corporation's revenue gap demanded by its financial growth model.

They're more long versus short term oriented. Their monolithic culture and history reinforces this attitude. They're capable of waiting us out in an area because we're fundamentally impatient and generally "big bang" product/market oriented and because they want long term business domination. NEC, Fujitsu and Hitachi, unlike Xerox, GE, Westinghouse, and RCA, have all persisted with computer manufacturing and now appear to be winning! This timeliness certainly affects their thinking on quality, and lastingness both in markets and products. And they're willing to invest.

Even though they have a concern for long term, they work the short term very hard. This may follow from the competitiveness/growth. They engineer for quick turn around, they have good processes and the engineers at these large companies work very hard. The official work week is 40 hours, but a more accepted pattern is 50-60 hours...particularly to maintain schedule or to win against IBM, Amdahl or Hitachi (if you're at Fujitsu).

They seem to do "bottom-up" product design versus "top-down" market planning as typified by the expensive, heavy, multi-volume market surveys and the classic Edsel. These reports usually report history and extrapolate it in a self-perpetuating fashion gathering data from a variety of sources consisting of discarded product ideas. Using this approach, we continue to build heavy, gas-consuming cars because the market has historically bought them because there is no choice. They look at the needs, and take existing ideas and improve them.

Products are quality/detail oriented versus being the ultra-high volume, low-quality throw-away types. These are characterized by say, Sieko (versus Timex) and anyone of their cameras say, Minolta (versus Kodak or Polaroid which assume an idiot user with no concern for quality picture, but must have it now).

With Japanese Quality Control, although data is kept by the factory reporting and control structure (i.e. management) the analysis, corrective actions and responsibility for improvement is delegated to the workers! Even enlightening American factories go through elaborate analysis to understand and engineer the change of processes that are easily understood and correctable by the workers themselves, given they have the tools to understand how well they're doing. Thus, there is "delegated QC" versus "centrally managed QC".

The <u>long term, quality products makes them built products that are hard to beat on a life-cycle basis</u>. While it isn't clear they really consider all life-cycle costs, their small cars now get very high ratings. In the case of computers, they have begun to design and build multiprocessors because their customers invariably buy and want upgrades. Since IBM rents computers, the multiprocessor approach hasn't been developed. The multiprocessors they sell also permit better Reliability, Availability and maintainability. They seem to do a better job considering life-cycle costs than we do!

<u>Products are designed for people</u> with attention to detail. The styling happens to be also attractive to others, but their technical, gadget-orientation really biasses them to designing technical looking, knob-intensive products as typified in hi-fi sets, complex watches, and cameras). It's probably impossible to have them design a product like the Polaroid One-Step camera. Color TV scopes are used to help operators control the large computing machines. More importantly, less people are involved in operating the Japanese computer centers, giving lower life-cycle costs.

PRODUCTS RESULT FROM UNDERSTANDING AND MANAGING A COMPLETE PROCESS

The basis of competitive performance products in high technology industries depend on understanding a complete process starting with basic research, going through applied research and advanced development, to product development. In addition, a parallel equally complex process, is required in order to design and build the process that manufactures such products. As a new product is introduced, it may be necessary to evolve and enhance it, to adapt it to the real or changing market and finally it must be eliminated when it does not effectively solve a need. There must be astute marketing including forward pricing in order to get on the necessary cost versus volume learning curves.

<u>The Japanese need invest little in basic and applied research because they are effectively coupling the U.S. laboratories into their advanced development</u>. In contrast, aside from hiring, there is very little flow of ideas from our public laboratories into U.S. industry. The university laboratories which have or are receiving significant Advanced Research Projects Agency (ARPA) funding for Computer Science (i.e., 20-30M/year), have post doctoral Japanese visitors. These laboratories include Stanford, MIT, Carnegie-Mellon, the University of Illinois, etc. One finds that the university and industrial laboratories of Japan are headed and staffed by researchers who've spent their research years in the mainstream American laboratories. For example, the head of a major research effort at one company was trained at the MIT Multics Laboratory.

Of the large companies with research laboratories, the Japanese emphasis is on advanced development where the output is a breadboard of a potential product. The quality of these laboratories seemed substantially ahead of comparable U.S. laboratories which often engage in research to ease the corporate conscience by having a research lab. For example, our corporate research laboratories were significant in the development of television. In later years even though labs at GE, Motorola, RCA, Westinghouse and Zenith grew in size and number, there was ineffective coupling and the U.S. TV industry has disappeared.

<u>MITI funds and manages other laboratories and corporations</u> to

carry out research that's oriented to getting experience that will eventually produce products. Funding, as opposed to having a captive laboratory, not only provides a system of checks and balances, but provides an incentive. This minimizes what I call the "dusty-lab syndrome". Many of our government labs were initially set up for a mission, and once the mission has been completed, the lab continues to exist. Since there's no real need, or mission, or review, negligible new work is output. We can all recall visiting these labs in which the dust is blown off the equipment for visitors and the same demo is run year after year. The same equations are on the board, with the same usually vague, unattainable, immeasurable, non-milestone based goal for the research. A buyer-seller relationship can help check this to a great extent whereby an independent organization such as a university manages the lab and takes responsibility for results in a competitive, seller fashion. Government labs set up to provide results to the government are most generally incestuous and ineffective. Also this brings the groups together and technology transfer is more likely to take place.

For example, NBS is setting up a lab to research computing interface standards, with industry being expected to contribute people to them to carrry out the work. This is ridiculous! People capable of this research are clearly going to be employed in developing interfaces. A government group dedicated to this will ultimately be tired and useless, assuming it does become successful in creating a standard. A more fruitful way to bring about the standards is to subcontract several competitive approaches and have industry prototype and report on them to NBS. In this way the expensive, bureaucratic staff is minimized at NBS. Again, such a staff will become obsolete even if it could be acquired. Quality output can be managed by NBS through a buyer role, provided the contact red tape is minimized.

The Japanese <u>orientation is</u> a strongly <u>engineering for trade versus</u> strongly <u>science-based</u> culture! Since the rest of the world does their research, why should they bother? This comes about because of their need for manufacturing novel products and their total dependence on the export of manufactured goods. Since our basic federal research funding for computing comes through the NSF, ARPA, and armed services, the emphasis is on science and research. Their funding comes through MITI and from various corporations, and hence the orientation is on international trade.

The culture supports a strong emphasis on manufacturing, not just product design. In addition to the product engineering process there is a comparable and equally important process responsible for the development and operation of manufacturing. This discipline has been eliminated from U.S. universities. While it isn't clear that the emphasis in Japan universities is stronger, there is more emphasis in the companies on manufacturing processes. People are rotated among the various processes and disciplines, making it equally desirable to be in all functions and phases.

The whole culture appears to understand basic learning and demand curves and they are volume (and growth) oriented, subject to the quality-first constraint. Knowledge of the learning curves is everywhere even the government research labs and universities. Their needs and goals are manufacturing/trade/industry oriented. This also means that, like Texas Instruments*, they're willing to dump and lose money for the short term in order to gain the market. This practice, when carried to certain extremes, was ruled to be illegal for a U.S. company. Although the Japanese put on a good act that their products won't be competitive when the yen is so strong, having gone from 300/1$ to 100/1$, it's a big ruse because of our dependency as a distributor now in many industries. This dependency will be elaborated on the following section.

As a corollary to learning curves and market domination, it's necessary and they are willing to give up profit for growth. For example, RCA is now a rug maker (or distributor), car rentor, publisher, TV distributor etc., instead of an electronics company that pioneered television. Their role is essentially no more than a banker and such a conglomerate is no match for a serious manufacturer. Whereas there is extreme pressure on our business for profit and return on investment, these factors are less in the Japanese companies. Sony is only moderately profitable, Fujitsu does relatively poor financially and I'd bet NEC or Hitachi computer divisions might even lose money. For now, they may still be buying in which is clearly more acceptable than GE, Xerox and RCA could accept. This makes the Japanese doubly hard to beat, since they can lose money on every one and make it up in volume. They can buy the business dumping and why not if there is long term reward?

INDUSTRY DOMINATION BY THE JAPANESE IS SIMPLY PREDICATED ON U.S. NAIVETY, GREED AND VALUES

Whereas as we watched the first few industries of textiles and steel become dominated by the Japanese, we unsympathetically stated that these industries were tired, the workforce was lazy, and the management was incompetent and unagressive about getting capital. Certainly there is no special societal fondness for the automotive and petroleum industries and now it's fitting to import our cars to straighten out the U.S. manufacturers. Now, the domination of all manufacturing is so clear and pervasive that we must look deeper because all society is to blame and is beginning to pay the price.

The domination can only happen with consenting buyers in the U.S. It is these buyers, (nee distributors, including tired old former manufacturers, that are to blame, not the Japanese. Alternatively our values are too short term and too basic as to see and understand the real long term effect.

The (Unstable) Three Island System

Since it's not clear to everyone what the long term, stable situation has to be, let's look at the end point. A system of three inhabited islands, all of which have adequate food, water shelter and land, points out the dilemma:

#1. supplies energy; consumes negligible manufactured goods;

#2.supplies manufactured goods (is supplied raw materials from several small islands it owns, and from discarded goods of island 3); and consumes energy;

#3.consumes energy and manufactured goods; supplies information.

Given that information is generally treated as a waste commodity of zero value there is no stable state for the system until islands 1 and 2 absorb island 3. Or conversely using any monetary system, island 3's paper or tokens will always be worthless. That is, islands 1 and 2 currency values will be out of balance with island 3, until 1 and 2 "own" island 3.

To a first approximation, the Japanese and their counterpart American buyers have systematically transformed American business from inventor-manufacturer-distributor to simply distributorships. This is in complete keeping with the goals of American business as reported and exonerated in business magazines and the teachings of modern business schools. The goal and reward of American industry is clear: return on investment and profit. Secondary measures like market share are occasionally used. Following only the ROI goal, subject to no other constraints, leads U.S. industry directly to being a distributorship for Japanese products. With this strategy, no investment, no planning, and no risk are required . All a company or its potentially enshrined leader has to do to be successful is to buy the right product for resale. Our electronics industry doesn't have to worry where the money comes from to pay the Japanese and Arabs. On the other hand a group who can only run a distributor is probably fairly top heavy and can easily be replaced say, by a hard-working Japanese group.

This merely confirms the classic definition of a capitalist as someone who'll make and sell the rope to hang himself. In this case it's merely reselling someone else's rope as we become too lazy to design and make rope.

The essence of distributorships is competely counter to the principles which made American industry initially great. Now it's simply with no work, no capital, anyone (everyone) can do nothing and succeed. All that's important for us now is to find the right supplier who'll put up the capital, design and manufacture products which we can distribute.

In computing, the trend has already started with Itel buying Japanese manufactured 370-compatible computers. Thus we expect Itel to have high ROI, and a net flow of dollars from the U.S. The solution is obvious:

> No company must be allowed to buy and distribute a foreign product without an offsetting equal export credit which they must arrange! That is, Itel can get agricultural products to sell or it could export its services. This has to be Itel's problem -- not

Carter's, Krep's, or Congress's problem as we now define them.

There's no way a manufacturer can re-enter various lost businesses once he becomes a distributor. The spirit, and capability to catch-up and manufacture are gone. Society and the investment structure are all aimed at continuing a status quo. In the case of TV, radio, hi-fi, and video recorder products all of which were U.S. products and which the first invention or key patents apply, the cause is hopeless.

Again, we can blame the Japanese, but someone in the distributorships acquired by the Japanese had to buy the sets in the first place and had to choose not to design and build competitive products or to insist on bi-lateral flow of goods. In the case of Motorola, the division was purchased by Matsushita and included both manufacturing and distribution. By 1976, the U.S. plant was reduced by 2/3, but the distribution network was left intact.

We (U.S.) have a higher regard to business training versus engineering and technical training. Here the Japanese are in even better shape because they don't yet have many business schools. Therefore instead of getting MBA's their engineering students get engineering master's degrees. In contrast, more engineers, quite erroneously, regard the MBA as necessary or useful to enter industry. This not only makes the Japanese better engineers for the same educational output, but doesn't reinforce the notion that engineering is the route through to the management ladder, or that an MBA is automatically needed if one is to supervise people. The MBA, oriented at every dual-career person being president, and epitomized by the content-free case study methodology, focusses on the quick buck. This is in contrast to the Japanese concern for deep understanding and the long term.

U.S. VALUES ARE CLEARLY DIFFERENT AND AS SUCH WE MAY BE HELPLESS AND SHOULDN'T BOTHER TO MANUFACTURE ANYTHING

At a government/society level they appear to have their act together much more than we do. In societal issues and in their products they seem to have clear, crisp ranking of goals and priorities. For starters, they know them, whereas nearly all our issues that start out simple become entangled as everyone (a new set of referees) enter the fray. These include: human rights vs equal rights; full employment vs inflation and balance of payments; environment vs region vs country; capital vs labor; and consumer protection vs business protection. But worse than a muddy set of design criteria is a muddy set of decision makers and an unclear decision process. The Japanese processes though more complex appear to be clearer. There is less government but it appears to be responsible and accountable!

Because of the need to export, for example there's educational support for engineering and technology, versus lawyers and other semantic accountants. There is a factor of 2 less lawyers per person than in the U.S. while lawyers wouldn't be bad if they only talked to each other. A productive lawyer can consume much productive and creative output of much of society. The Japanese emphasis (priority) is on physical output because they are a manufacturing island with no other visible means of support. With the increased emphasis in legal training, our priorities seem to be on the manufacturing of paper, intergroup contracts, governing and bickering among semantic accountants.

As a simple explanation, more money is available for investment to enable them to manufacture (for their island) because of lower taxes. This clearly affects their ability to invest in industry. They're supposed to be willing to pollute for profit. I didn't observe this. For example, LPG taxis are used instead of gas or diesel. Perhaps they only kill whales outside of Japan and pollute other enivronments. Their environment is just fine, though high density. On the other hand, taxes can be low because their priorities are clearer, more people work and they spend less on government and defense.

Their government spending for military is far less and nearly non-existent. Although there is some fall out of our military spending for a better society, it seems to be small and clearly a by-product. In the case of semiconductors, computers and related research, the benefit is small compared to what it could be compared to more directed goals such as the Japanese have with export domination.

In a similar way the Japanese spend significantly less per capita for health care and medical research. They can capitalize on our research here, but since they have a longer lifespan, its not clear what the extra expenditures we make buy. In effect, the lack of spending in medicine goes to investments which result in full, lifetime employment which is probably the best solution to personal health.

The Japanese don't have the massive federal research over-expenditures, epitomized by NASA and NIH. Here again, in the rare event there are results, the Japanese will capitalize on our research for manufacture and export. These areas seem to have big expenses and contribute little because much of the work has no goal. NASA goals appear to be vague and tenuous now that they've stopped providing the world with exciting space shots and television pictures from the moon, and the immediate needs for this research escapes most of us. National health research is also equally vague. This work only increases health care costs, by a whole series of secondary effects. Here the Japanese have a greater life expectancy with under 1/2 the per capita costs.

They believe computers are fundamental for the long term and they're prepared to invest and wait for return. Machines are used in all products they build for export and they save labor too. Labor is both precious and expensive in Japan as there are only 110M people and 2% unemployment. They're considering raising retirement from 60 to 65 to get the extra productivity. They must have computers to raise productivity! This is vital to their continued domination of manufacturing. As a separate research area, robots are an important component of manufacturing domination. While much of the pioneering work is U.S., the continued work to make robotics practical takes place in Japan! This is the opposite of say the Australian attitude where there is increasing unemployment and a belief that computers must be eliminated. Australia is now almost totally dominated by Japanese products and the small Australian automotive industry of GM- and

Ford-based large cars and is rapidly declining under the stress of small, high volume, quality Japanese cars.

THE JAPANESE SOLUTION TO OUR BALANCE OF PAYMENTS PROBLEM: SELL (IN JAPAN)

This is the answer our industry wants and will willingly, but foolishly looks to. However, the Japanese rhetoric is only for our gullible government and academic communities and the naive business people. For example, trade envoys from Massachusetts and New Hampshire visit Japan with the expectation of selling high technology goods. They'll succeed to sell a few prototypes. The real sales will come in 5-10 years when these products are resold in volume to the U.S!

There has not, nor will there be any serious trading of American products of Japan. The distributor/trading network entirely thwarts such an effort! The results are clear and we must face this.

Japan is a closed society and market. As the most powerful, homogeneous culture in the world there is a long history of being closed. This can be verified by: reading any of the books or articles on Japan; trying to understand the complexity, yet subtly of a formal tea ceremony; looking at any industry manufacturing case; or just visiting and observing.

The language is a code to further segment. It's not clear how difficult the language is to learn, but it's probably relatively useless without the societal understanding. We only teach Japanese minimally on the West Coast of the U.S. On the other hand the technically trained Japanese have several years of English in order to read the literature.

Even though there are major cultural differences among Japan and other far eastern countries (e.g., China, Taiwan, Korea) there is closer proximity among them than with western countries. This closeness is especially advantageous in finding additional sources of especially low cost labor.

The tariffs support the establishment of any industries they target. Now the computer import duty has been reduced to be on a parity with the U.S., but this matters little since their industry is strong enough to withstand imports! As we've seen in other industries, this is a come-on to further strengthen the Japanese manufacturers for export competition by having them compete in a token way with the few imports and thereby gain ideas to sharpen their exports.

For example, in the early seventies the Japanese encouraged U.S. minicomputer imports. These occurred and now there is a significant Japanese minicomputer industry. For example, the basic structure of Fujitsu's minicomputer is identical to the PDP-11 DEC introduced manuals and brochure before the patent application, making the PDP-11 non-patentable in Japan.

By the society and the emphasis on personal relationships it's hard for foreigners to break into or sell, especially on a one shot basis. "Doing business" together appears to be done over a long time period and is almost ritualistic. This means that it's essentially impossible to have an effective international company as we know them. A foreign manager is clearly tabu and sales are limited to one-shot deals with trading companies. There is no trading except as joint ventures! A foreign owned company with ?% of the equity is illegal in Japan.

LABOR COST, LIMITED POPULATION, FULL EMPLOYMENT AND FEW NATURAL RESOURCES, CREATES IMPORTANT BY-PRODUCTS

Transportation and meetings run on time and at full capacity. This is in contrast to U.S. facilities, especially the meetings scheduling and performance. I accomplished roughly twice as much per day as in another western country in terms of customer and plant visits. The cordial, formal protocols help meetings proceed rapidly. By operating in a highly scheduled fashion more work gets done and there is less anxiety as to performance.

There's measurement of and pressure for efficiency. That is, the work-out/work-in ratio is high. In a taxi, there's an automatic back door opener so that the driver can load/unload faster. Of course, the factories graph everything. It feels like the notion of efficiency is taught to all. Concepts like fuel efficiency versus speed, weight and pollution are impossible concepts for Americans to understand. Worse yet, having only briefly lived in a constrained environment during wartime, most of us have no understanding of living with finite resources.

Given a notion of efficiency, there's real concern for saving of physical resources too. At the computation center, printout isn't automatic; it's queued and must be requested by badge reader. Lights, always florescent due to efficiency, are off when not in use. Of course small cars, taxis, a good train/subway are other indicators. The cars have bells that ring when the car is going over 100 Kmh! None of these exist in the U.S.

Contrary to a previous "feeling" they are working the environment issue. There were U.S. environmental people at a conference at the same time I visited; the Japanese were politely ignoring them while taking their basically boondoggle-oriented conference registration fees paid by the U.S. government research establishment.

There is a range of basically human and personal concerns. The result is a longer life span. While the subways and high density trains jostle people pretty badly, and there's no segmented smoker areas (and many smoke), there's great concern for the feelings, privacy and treatment of individuals. Although I had special treatment on the visit, on arrival and departure at every organization, I was given hot cloths and refreshments of tea, juice or coffee to be really considerate to westerners. It was hot and humid in July, but taxis and all buildings had air-conditioning. The hotels, though the most expensive, were also the best in terms of privacy, food and service. This included a large hotel in Tokyo and a 15 room old style, inn in Kyoto. The goal is privacy, and ambiance, with incredible attention to simplicity, design and detail. For example, there was a cloth cover over the telephone because it didn't fit the room decor.

Of course, the food is the ultimate in personal concern. Food served in many courses varied from raw fish to pickled vegetables (e.g., potatoes) and flowers (lotus blossoms) with lots of seaweed, fish and fish eggs. Tempura, teryaki, and hibachi grilled meat and fish are more easily digested by the westerners. The bread crusts were removed when sandwiches were served to westerners. There was much concern that the colors of the food matched; the physical looks were important.

There are Japanese baths, and these are great too!

They are compulsively clean. In an indirect way, this really helps the manufacturing of small, precise goods (including cameras, semiconductors, high-speed computers and disk memories.

There's orderly queueing at each server. The Japanese appear to be the world's best self-queuers. There's probably some protocol for resolving conflict when two persons arrive to the queue at the same time. In general, a system of this type has higher through-put. I also suspect there is lower general hostility arising from competing for a finite resource.

Inventions are to labor-saving devices. I saw countless gadgets of this form. The printers at computation centers had paper cutters on them with conveyors to bring output back to a single station. There are no computer operators and people to serve the users! This direct use of facilities not only costs less, but provides significantly higher through-put.

EPILOGUE

On arriving at Sydney, I was struck with the contrast to dense, intense, humid and hurried Tokyo. I was ecstatic to get back, after 20 years, to a life style, people and place I feel more comfortable with.

Sydney's beaches are the world's finest; the weather's great; people spend lots of time out-of-doors with sports, strolling and simple gardening versus the subtle and very complex Japanese gardens; work starts late, runs slower and ends promptly with twice as many secretaries to do half the work -- but they do make their bosses feel good; and the continental and western food, beer and wine drastically improved having moved away from the early English influence.

Thinking about the Japanese competing with the Arabs to buy American and Australian mines, property and factories is frightening, but remote in my mind. Besides, does it matter who owns us? Will they interfere with our way of life? Maybe we'll change them and make them lawyers rather than manufacturers. If enough of them come to live or vacation with us very long, we'll be back manufacturing and exporting to them if anybody can learn the language. If things don't go our way, we can make it illegal, set up an agency, and then sue them with our incredible bureacracy and legal technology.

IMPRESSIONS ON HOW WE AND THE JAPANESE ARE CONVERTING U.S. INDUSTRY INTO DISTRIBUTORS

Gordon Bell, Vice President of Engineering
Digital Equipment Corporation, Maynard, Mass.; and
Professor of Computer Science and Electrical Engineering (on leave)
Carnegie-Mellon University; Pittsburgh, Pa.

The island of Japan, with few natural resources and over 100 million people, virtually dominates world production of manufactured goods, including the components and processes to make these goods. Every Japanese knows that exports are vital to survival. Also ingrained is the understanding that savings and living within one's own means support the ability to manufacture and export. In contrast, the notion of balanced budgets, savings and manufacturing have gradually disappeared from U. S. culture.

For example, the United States still holds a dominant position in the production of computers and semiconductors, but the Japanese plan to dominate these industries. Unwittingly, U.S. industry, government and society continue to aid the Japanese. Forty odd reasons are given to support this conjecture, each one providing a lesson.

The Japanese have progressed from domination of low-technology simple commodities to complex manufactured goods. The progression has been from textiles, steel, radios, sewing machines, typewriters, quality cameras/optics, watches, small cars, television sets, tape recorders, video tape recorders, calculators and on to state-of-the-art semiconductors and computers. Their current position in semiconductors and semiconductor-making equipment indicates they are well on their plan to dominate this manufacturing as a base for the continued and future market domination of electronics and computers. High-technology industry is increasingly being concentrated in Japan while the Japanese-owned low skill textile and television factories are being located in the U.S.

Dataquest describes how the Japanese go about systematically to dominate a market. Appendix 1 describes the four, detailed phases: initial development of a domestic industry, establishment of an export base, significant market penetration in foreign markets and final market exploitation.

BASIC STRATEGY, AND TACTICS FOR DOMINATION

Japanese industry and government operate as a team reinforcing strategy and tactics with appropriate levels of competition. Unlike many companies and countries that have tried and failed, they successfully planned and built a mainframe computer industry.

The Ministry of International Trade and Industry (MITI), with autocratic power, helps to amalgamate strategies within industry groups creating an organization commonly referred to as "Japan Inc." Because there is no direct control, I prefer not to use the term "Japan Inc." but to name the phenomena "The Japan Club" since there's a structure for the essential competition at the market level. For example, MITI identified and encouraged early importing of minicomputers, including those from Digital Equipment Corporation, as a competitive "straw horse" to build their own industry. One of DEC's interactive data base systems, MUMPS, was sold in Japan for end-user applications. On seeing several lost sales, MITI funded the development of MUMPS on a Japanese minicomputer. In mid 1978, a Japanese researcher asked me, through an academic channel, for the internal architecture of MUMPS in order to study its structure from a so-called computer science viewpoint. We expect to catch MUMPS from Japan soon.

The U.S. has no equivalent of MITI to protect major corporations as national resources. In contrast, U.S. corporations are looked on as adversaries to the national interest. IBM, already under attack from Japanese competition, is also under the gun from most U.S. government departments. Together they seem intent on destroying IBM, leaving it and others as distributors for Japanese products.

The strategy of MITI and the Japanese companies to win dominance of the computer industry is clearly evidenced, but it is not understood by U.S. government and industry. In keeping with the priority, MITI is both very strong and attracts competent people. The Japanese companies, while maintaining competition in limited domains, both plan and talk with one another. For example, Fujitsu and Hitachi have developed IBM plug-compatible machines. Coupling individual, competing companies for technological acculturation in this fashion is an important management technique to assimilate technology quickly.

The U.S. Department of Commerce and the U.S. Labor Department, in contrast to MITI, have neither a plan nor the personnel to help maintain U.S. dominance in high-technology fields important to the future of the country's economy and security. Furthermore, these two adversary departments are adversary to U.S. business. Trade trips to Japan by Secretary Kreps only emphasize our lack of understanding of the Japanese capability to use trade to introduce technology into their society. Our trade deficits cannot be turned around by hand-shaking missions, but demand a strategic and tactical plan based on understanding. Our political system is devoid of planning and accountability of government departments; even if the Secretary of Commerce could plan, her short tenure is inadequate to solve this problem. Once a new administration appears, any policies, plans and commitments are reset to zero!

Japanese tactics focus on the centrality of work and loyality to a company. A company screens each new employee carefully because when it hires an individual it takes on a lifetime commitment. The security promotes risk-taking, a phenomena generally unknown in large U.S. corporations. The team spirit is engendered as the various members learn how to get along with each other.

Quality control is in the hands of the workers. Although data is kept centrally, the analysis, corrective action and responsibility for manufacturing and quality rests with the employees concerned. Quality control is generally centralized and the organization of work often does not lead to self-esteem in the U.S. organization. Such participative management provides a key to the devotion to the workplace and sense of value achieved through work. The incompetent workers become the wards of the organization rather than wards of the state. Pride, family tradition, and because everyone is working, nonwork is socially unacceptable, embedding the importance of work into the fabric of society. A similar effect is observed in the U.S. during periods of high unemployment. At this time non-work is approved since others are unemployed.

In the U.S., the freedom of the individual has superseded work as a goal. The employee mobility is high and as a result companies screen very little as the short tenure is assumed. One recent semiconductor company ad claimed that no interviews were required at all. Turn-over and unemployment here are high with levels of consumption also rising so that some Japanese observers have concluded that the Japanese live to work and the Americans need to work to live. The measurable results are simply that the relative per capita productivity in manufacturing industries of Japan is now almost twice that of the U.S! Also, the sales per employee of a Japanese electronics corporation is about $100K, versus $45K for the U.S.

The Japanese government has been able to nurture both large and small companies while the U.S. government agencies seem to alienate the large and aren't effective at supporting the small ones. Much work in Japan is done in small subassembly operations. Competitive small shops keep the cost down by removing it from the large, hard to manage hierarchical organizations.

USING ACCULTURATED DESIGN AS THE BASIS TO DOMINATE

For centuries Japan has acculturated customs, but mostly it adopts and adapts technology. In the 16th century, for example they began manufacturing gunpowder a scant 18 months after the Portuguese brought it to Japan. Shortly thereafter they were banned. Any idea or product has always been fair game for adoption and improvement. Product and process evolution are merged in a long term view of achieving market domination. They orient the processes competitively considering quality, volume for growth, and flexibility to allow for the fast turn-around needed to maintain full-production capacity in a shifting market.

All the Japanese computer manufactuers have acquired their technology within the past ten years by dealing with U.S. manufacturers either as a joint venture or under license, including: Fujitsu (Amdahl/Siemens) and Hitachi (RCA); NEC (Honeywell, GE, Varian) and Toshiba (Honeywell, GE, Interdata); Mitsubishi (Xerox) and Oki (with Univac joint venture); Yokogawa (HP); and Nippon Minicomputer (DG). In all cases, the Japanese have improved the technology in terms of perceived quality, performance and manufacturability.

The agreement between Fujitsu and Amdahl Corporation, though still at an early stage, provides a good example of the classic Japanese computer acculturation process. In the late 1960's, Gene Amdahl, then head of IBM's San Jose Advanced System Development Laboratory, explored the basic technology for high-performance IBM computers. When he failed to interest IBM in building high performance machines, he formed Amdahl Corporation to develop the technology. When he needed more capital Fujitsu bought an interest and acquired the manufacturing rights to, and became the manufacturer for the Amdahl line. Fujitsu was also able to use the same technology to design and manufacture computers for the Japanese market. In only one computer generation, at the beginning of 1978, both Amdahl and Fujitsu announced their latest computers based on the Fujitsu-Amdahl circuits and packaging. Now, Fujitsu appears to have a machine with higher performance and reliability (the M200) than either Amdahl or IBM have so far announced. Fujitsu has produced a machine based on multiprocessing which provides users with new capabilities; furthermore they can buy more processors rather than trade-in when increased computation is needed.

In addition, Japanese computer manufacturers have a complete line of peripherals and test and manufacturing equipment that is based on counterparts invented in the U.S. The designs range from "reverse engineered", to look-alike copies, to radically improved products based on Japanese inventions. With "reverse engineering" a product is dissected with micrometers, special gauges, etc. and made compatible in nearly every respect. The Japanese make only products for export to the U.S. market that do not violate patents. Tektronix look-alike scopes and reverse engineered IBM disks are common. In 15 months, Nippon Peripherals Limited produced a disk that was mechanically identical to the IBM 3340. From comparing the two drives, one might conclude that they were made from the same drawings.

PRODUCT DESIGN BASED ON NEED, QUALITY AND THE LONG-TERM

Traditional top-down marketing is characterized by expensive, thick market surveys that extrapolate history in a self-perpetuating fashion. Here, the goal is to fill various revenue gaps that develop. Using a market survey approach the U.S. continues to build heavy, gas-consuming cars, because the marketing managers can only think in terms of what has sold in the past. Freed from this approach, the Japanese have been able to look at the real needs, and they have appropriately adapted existing ideas. High-level corporate marketing does not design the products; engineers design according to needs using a bottom-up approach and based on technology.

Japanese companies, with long-term goals and commitments, similarly are not forced to depend on a short-term marketing approach. NEC, Fujitsu and Hitachi, unlike Xerox, GE, Westinghouse, and RCA, have all persisted with computer manufacturing and after years of investment have established successful products. Their long-range thinking from the outset allowed them to invest in long lasting quality.

Japanese companies focus on highly sophisticated quality products rather than ultra-high quantity, low-quality throw-away merchandise. The differences are characterized by comparing Seiko versus Timex watches and comparing Minolta or Nikon versus Kodak or Polaroid cameras. Japanese styling is often technical and gadget oriented, typified by multi-knob hi-fi sets and complex watches. It may be impossible for them to design a product like the Polaroid One-Step Camera because of the differences in picture quality. The emphasis is on an educated consumer who will value his purchase.

Concern for quality and long-term values leads the Japanese to build products that have a long lifecycle. Even their auto industry constrained by Detroit's yearly new model concept is now getting very high ratings for durability and serviceability. Accounting models lead to emphasizing production of long lived versus throw-away goods.

PRODUCTS RESULT FROM UNDERSTANDING AND MANAGING A COMPLETE PROCESS

The successful production of competitive performance products in high technology industries depends on understanding a complete process that includes basic research, going through applied research and advanced development, to product development. In addition, a parallel and equally complex process is required to design and build the process that manufactures such products. After a new product is introduced, it may then be necessary to modify and enhance it to adapt it to the real or changing market, and finally to eliminate it when it is no longer effective.

The Japanese need invest little in basic and applied research because they are effectively coupling the U.S. laboratories into their advanced development. In contrast, aside from the direct hiring of students and researchers, there is very little flow of ideas from our public laboratories into our own industry. As Carver Mead of Cal Tech points out, "I like the Japanese. They listen. Also unlike American industry, they're willing to build from our ideas." The university laboratories at Stanford, MIT, Carnegie-Mellon, the University of Illinois, receiving significant ($20-30M/year) Advanced Research Projects Agency (ARPA) funding for Computer Science, have post-doctoral Japanese visitors. The university and industrial laboratories of Japan are headed and staffed by researchers who've spent their research years in key American laboratories (e.g., MIT Multics). In contrast there is no Japanese training of U.S. engineers and scientists; furthermore, the flow of ideas is minimal.

Most recently, Japan has offered to spend one billion dollars in the U.S. for research, predominately for energy conversion. By accepting these funds, the Japanese can be even more effectively coupled to U.S. research and can "learn" to research, just as they've learned manufacturing, design and advanced development. The scientific community is anxious for more funds, independent of where they come from or what the consequences are. Of the large companies with research laboratories, the Japanese emphasis is on advanced development where the output is a testable prototype, often of a potential product. In contrast, U.S. corporate laboratories hide behind the veil of science where the output is vague and untestable. The quality of these laboratories is high versus many comparable large U.S. companies where research is to ease the corporate conscience instead of providing new development. Although such corporate research laboratories (e.g., GE, Motorola, RCA, Westinghouse and Zenith) were significant in the early development of television, the U.S. television industry has declined with few recent local advances.

MITI funds and manages other laboratories and corporations to carry out research that is oriented toward getting experience that will eventually produce products. Funding specific, as opposed to having a captive laboratory, not only provides a system of checks and balances, but also provides an incentive. Many of our government laboratories were initially set up for specific missions, and although the missions were completed, the laboratories continue to exist. Since they no longer have a real goal, or mission, negligible new work is done. The dust is blown off the equipment for visitors and the same demonstration is run year after year. A buyer-seller relationship, in which an independent organization, such as a university, manages the lab and takes responsibility for results can minimize this "dusty lab" syndrome. Moreover, funding for specific projects can bring together diverse groups and promote technical interchange.

The Japanese orientation is toward engineering for trade rather than being strongly science-based. Since the rest of the world provides research, why should they bother? This comes about because of their need to manufacture products and their total dependence on the export of manufactured goods. Since our basic federal research funding for computing comes through the NSF, ARPA, and the armed services, the emphasis is on science and research. Their funding comes through MITI and from various corporations, and hence the orientation is on international trade.

The trade drive causes a strong emphasis on manufacturing, not just product design. In addition to the product engineering process there is a comparable and equally important process responsible for the development and operation of manufacturing. This discipline has been nearly eliminated from U.S. universities as it has moved from the engineering to the management school. There is a decided emphasis on manufacturing processes in Japan as people are rotated among the various processes and disciplines, making it equally desirable to be in all functions.

Everyone associated with science, engineering and manufacturing understands basic learning and demand curves and they are quantity (and growth) oriented, subject to the quality-first constraint. Knowledge of the learning curves (i.e., increases in the combined number of units produced cause a reduction in manufacturing cost) is everywhere. Fred Bucy comments on Japanese competition in TI's 1978 Annual Report: "...the big difference is that TI is the first major non-Japanese company they have run into that understands and uses the learning curve". The Japanese are willing to sell outside Japan at a lower exported price (dump) and lose money often by selling below cost for the short term (see also Appendix 1) in order to buy market share. This practice is illegal for both U.S. and Japanese companies. Although the Japanese pretend that their products are not competitive because the yen is so strong, they are consciously ignoring our dependency as a distributor now in many industries.

As a corollary to learning curves and market domination, it's necessary and they are willing to give up profit for growth. For example, RCA is now a rug maker (or distributor), car rentor, publisher, television component distributor; it hardly resembles the electronics company that pioneered television. It's difficult to put the whole blame on RCA management because they are constrained by the economic and business temperament of the U.S. environment. Whereas there is extreme pressure on our business for profit and return on investment, these factors are less important to the Japanese companies. Sony is only moderately profitable, Fujitsu does relatively poorly financially and NEC or Hitachi computer divisions may even lose money. None of these companies would compete for capital in the U.S. stock market where return-on-investment is the key criterion. Japanese companies are buying market share and this is clearly more acceptable to the U.S. investors than for GE, Xerox and RCA who left the computer business. They can buy the business through "dumping" and why not if there is long term reward?

JAPANESE DOMINATION IS PREDICATED ON OUR GREED AND VALUES`

As we watched the first few industries of textiles and steel become dominated by the Japanese, we unsympathetically stated that these industries were tired, the workforce was lazy, and the management was incompetent, unimaginative and unaggressive about getting capital. Certainly, there is no fondness for the automotive and petroleum industries and it seems fitting to import our cars as a lesson to our own U.S. manufacturers. Now, however, the domination of all manufacturing is becoming so clear that we must look deeper at the causes.

The domination can only happen with consenting buyers in the U.S. It is these buyers, called distributors, including tired, old, former manufacturers that are to blame, not the Japanese. Our values appear to be too short term and too basic. We really must understand that the following, simple, long-term consequence is complete economic domination.

The (Unstable) Three Island System - Or How and Why We Will Be Dominated

Since it's not clear that continued consumption, with no corresponding export means, let's look at what is the ultimate, singularly stable point simply. A system of three inhabited islands, all of which have adequate food, water, shelter and land, points out the dilemma:

#1. supplies energy; consumes negligible manufactured goods;

#2. supplies manufactured goods (is supplied raw materials from several small islands it owns, and from discarded goods of island 3); and consumes energy;

#3. consumes energy and manufactured goods; supplies information.

Given that information is generally treated as a waste commodity of zero value, there is no stable state for the system until islands 1 and 2 absorb island 3. Or conversely using any monetary system, island 3's paper or tokens will always be worthless. That is, islands 1 and 2 currency values will be out of balance with island 3, until 1 and 2 "own" island 3.

Through greed and short-term values, the Japanese and their counterpart American buyers have systematically transformed American business from inventor-manufacturer-distributor to simply distributorships. This transformation is in complete keeping with the goals of American business as reported in business magazines and the teachings of modern business schools. The goal and reward of American industry are clear: return on investment and profit. Secondary measures, such as market share, are occasionally used. Only a few corporations consider no lay-offs and full-employment to be important; as such, a clear, adversely separation has formed between management and labor. Following only the profit-based goals, subject to no other constraints, leads U.S. industry directly to distributorships for Japanese products. This strategy requires no investment, no planning, and no risk. All a company has to do to be successful is to buy the right product from Japan and then resell it.

This merely confirms the classic definition of a capitalist as someone who'll make and sell the rope to hang himself. However, in this case the capitalist is reselling someone else's rope because he is too lazy to design and make his own rope.

The essence of distributorships is completely counter to the principles that made American industry initially great. The new principle is simply that with no work and no capital, anyone (everyone) can do nothing and succeed. All that's important is to find a supplier who'll put up the capital, design, and manufacture products that we can distribute. In computing, the trend has also started: Itel is buying Japanese-manufactured IBM 370-compatible computers. Thus we expect Itel to have good financial metrics and be a good investment. It will also cause a high net flow of dollars from the U.S. as it becomes more successful.

American business, of course, is only slightly at fault because the U.S. non-business communities (politicians in government, consumers, and academics) have introduced and strongly support heavy borrowing, beyond income. These thwart an environment conducive to manufacturing. Both the per capita rate and amount of savings for both individuals and corporations in Japan is twice that of their U. S. counterparts! For example, the retirement system in Japan is actuarially sound. Of course, the Japanese government operates a balanced budget and taxation supports savings. Furthermore, as a society, they understand themselves simply as an island that must have a favorable balance of trade.

There's no way a manufacturer can re-enter a lost business once he has becomes a distributor. The spirit, and capability to catch-up and manufacture are gone. Society and the investment structure are all aimed at continuing a status quo. Radio, television, hi-fi, and video recorder products are all built using key U.S. developed ideas and patents, yet are no longer built by U.S. manufacturers. Again, we can blame the Japanese, but someone in the distributors had to choose to buy the products rather than design and build competitive products. In the case of Motorola, the television division was purchased by Matsushita in 1974 and included both manufacturing and distribution. By 1976, the U.S. plant was reduced by 2/3, but the distribution network was left intact.

We (U.S.) have a higher regard for business training versus engineering and technical training. In the U.S. many engineers regard the MBA degree as necessary for a career in industry. The Japanese do not yet have many business schools; therefore, instead of MBAs, engineering master's degrees are sought. This makes the Japanese better engineers for the same educational investment. Also, the management of manufacturing organizations are the better equipped to understand technology and products.

By having more people just concerned with distribution, we are becoming a nation of shopkeepers. The emphasis is simply to keep stores open longer and to find new ways to distribute Japanese manufactured goods. Not only does this further stimulate consumption, but it takes people from the primary production work force and makes us merely an island of consumers with no material means of support.

THE JAPANESE HAVE PRIORITIES AND SUPPORT FOR TRADE

At a government/society level the Japanese appear to have their act together. The Japanese seem to have a clear, crisp ranking of goals and priorities. For starters, the Japanese know their goals and priorities, whereas nearly all our goals that begin simple become entangled as special interest groups enter the fray. Some issues that compete for priority include: human rights versus equal rights; full employment versus inflation and balance of payments; environment versus region versus country; capital versus labor; and consumer protection versus business protection.

Because of the need to manufacture and export, the Japanese educational system supports engineering and technology, while we support lawyers and other semantic accountants. There are fewer lawyers per person by a factor of two than in the U.S. The Japanese emphasis (priority) is on physical output. The increasingly large number of U.S. lawyers: consumes productive and creative output of workers; creates a self-perpetuating, non-productive body; detracts from persons who would otherwise enter productive occupations; and tends to build an even larger governing body. With an increased emphasis on legal training, our output is measured by intergroup contracts, policies, laws, rules, regulations and other forms of bickering among semantic accountants.

As a simple explanation, more money is available in Japan for investment to enable them to manufacture (for their island) because of lower taxes. This clearly affects their ability to invest in industry.

Their government spending for military is nearly nonexistent. Although there are prototypes from our military spending, they seem small and are by-products. In the case of research for semiconductors and computers the benefit though impressive might have been as great, given a different goal (e.g., energy self-sufficiency).

The Japanese don't have the federal research over-expenditures, epitomized by NASA and NIH. In the event of results, the Japanese will capitalize on our research for their manufacture and export. The NASA goals, for example, appear to be vague now that they've stopped providing the world with exciting space shots and television pictures from the moon, and the immediate needs for this research is unclear to most of us.

National health research seems equally vague. This research appears to increase health care costs, through a number of secondary effects. By contrast the Japanese spend one-half of what we do per capita for health care and medical research. They can capitalize on our research, but since they have a longer lifespan, it is not clear what we gain with the extra expenditures. In effect, Japan's lack of spending in medicine goes to investments which result in full, lifetime employment which is probably the best solution to personal health.

The Japanese believe computers are fundamental for the long term and they are prepared to invest in them and wait for return. Non only are machines used in all products they build for export, but they save labor too. Labor is both precious and expensive in Japan: there are only about one hundred million people and two percent unemployment. They're considering raising retirement from 60 to 65 to get the extra productivity. They must have computers to raise productivity; computers are vital to their continued domination of manufacturing. As a separate research area, robots are an important component of manufacturing domination. While much of the pioneering work was done in the U.S., the continued work to make robotics practical takes place in Japan. By contrast, in Australia where there is increasing unemployment, there's a belief that computers must be eliminated. Australia buys nearly all Japanese products, produces less and less, and the small Australian automotive industry of GM- and Ford-based large cars is rapidly declining under the stress of small, mass-produced Japanese cars.

THE JAPANESE SOLUTION TO OUR BALANCE OF PAYMENTS PROBLEM: SELL (IN JAPAN)

Can we solve our balance of payments problem by selling to Japan? Selling to Japan is the answer our government and industry want and willingly, but foolishly, look to. However, the Japanese rhetoric is only for our gullible government and academic communities and the naive business people. Furthermore the trade missions are only stocked with powerless, non-responsible, short-lived politicians whose main purposes include visiting Japan and being able to say something to the folks back home. For example, when state trade envoys visit Japan with the expectation of selling high technology goods, they succeed in selling only a few prototypes. The real sales will come in 5-10 years when these products are resold in volume to the U.S!

There has not been, nor will there be any serious trading of American products with Japan. The distributor/trading network entirely thwarts such an effort! The results are clear and we must face them.

Japan is a closed society and market. As the most powerful, homogeneous culture in the world it has a long history of being closed. There is no counter-evidence that an open market exists. The language is a code to further segment. Although business people do learn the language in crash courses, the language is relatively useless without the societal understanding. We only teach Japanese minimally on the West Coast of the U.S. On the other hand the technically trained Japanese have several years of English language training.

Even though there are major cultural differences among Japan and other far eastern countries (e.g., China, Taiwan, Korea) there is closer proximity among them than with western countries. This closeness is especially advantageous in finding additional sources of especially low cost labor.

The tariffs support the establishment of any industries they target. Although the semiconductor and computer import duties have been "advertised" to be on a parity with the U.S. they aren't there yet, but this matters little since their industry is strong enough to withstand imports. Still prices of U.S. produced computing machines are cheaper. In semiconductors the rationale for high tariffs has been protection of infant industries, yet outside of Texas Instruments and Western Electric, Japanese companies have been manufacturing longer than all other U. S. corporations. As evidenced in other industries, this is a come-on to further strengthen the Japanese manufacturers for export competition by having them compete in a token way with the few imports and thereby gain ideas to sharpen their exports.

For example, in the early seventies the Japanese encouraged U.S. minicomputer imports, although there were high tariffs. These occurred and now there is a significant Japanese minicomputer industry. For example, the basic structure of Fujitsu's minicomputer is quite similar to the DEC PDP-11.

Because of the closed nature of society and the emphasis on personal relationships, it is difficult, perhaps impossible to have significant Japanese sales. There are no significant examples to the contrary. "Doing business" together appears to be done over a long time period and is almost ritualistic. This means that it's essentially impossible to have an effective international company as we know it. A foreign manager is clearly tabu and sales are limited to one-shot deals with trading companies. There is no trading except as joint ventures. A foreign-owned company with controlling equity is so rare that it is an effective unwritten law.

JAPANESE HIGH LABOR COST, LIMITED POPULATION, FULL EMPLOYMENT AND FEW NATURAL RESOURCES, CREATES IMPORTANT BY-PRODUCTS TO FURTHER HELP TRADE

Japanese transportation and meetings run on time and at full capacity. Roughly twice as much as in the U.S. can be accomplished per day in Japan, especially those requiring meetings. The cordial, formal protocols help meetings proceed rapidly.

There's measurement of and pressure for efficiency. That is, the work-out/work-in ratio is high. For example, taxis have a driver-operated back door opener so that passengers can load/unload faster. The notion of efficiency seems to be taught to all and factories measure, graph and display key results. Concepts like fuel efficiency versus speed, weight and pollution are difficult concepts for Americans to understand, yet the Japanese "feel" them.

Given a notion of efficiency, there's real concern for saving physical resources too. At the computation center, printing isn't automatic; it's queued and must be requested separately. Lights, always florescent for high efficiency, are off when not in use. Of course small cars, taxis, a good train/subway are other indicators. The cars have mandatory bells that ring when the car is going over 100 Kmh! None of these artifacts for efficiency exist in the U.S.

Contrary to our "feelings", they are working the environment issue by less consumption, for example. This will indirectly make more money and resources available for production at lower costs. For example, cars don't pollute. U.S. environmental people at conferences in Japan are politely ignored while taking their basically boondoggle-oriented conference registration fees paid for by the U.S. government research establishment.

There is a range of basically human and personal concerns which encourage and support productivity. The result is a longer life span in the face of stress on productivity. While the subways and high density trains jostle people pretty badly, and there's no segmented smoker areas (and many smoke), there's great concern for the feelings, privacy and treatment of individuals. On arrival and departure at every organization, one is given moist cloths and refreshments. Taxis and buildings are air-conditioned. The hotels, though very expensive, provide privacy, ambiance and excellent food and service. For example, one expects a cloth cover over the telephone to enable it to fit the room decor. There are Japanese baths, and these are great too!

They are compulsively clean. In an indirect way, this really helps the manufacturing of small, precise goods including cameras, semiconductors, high-speed computers and disk memories.

There's orderly queueing at each server. The Japanese appear to be the world's best self-queuers. Queued systems of this type have higher through-put and make the best use of resources. One might suspect there is lower general hostility arising from competing for a finite resource when queueing.

Inventions are to labor-saving devices. There are countless gadgets to save scarce labor. Computation center line printers have paper cutters and conveyors in order to bring printing back to a single station. There are no computer operators and people to serve the users! This direct use of facilities not only costs less, but provides better service and through-put.

Conclusions

We must be impressed with the intense drive coupled with the technical, manufacturing and marketing acumen of the Japanese. This drive and ability, coupled with many factors of our society, has enabled the Japanese to systematically plan and dominate every U.S. market that they've attempted. Although there's been a "feeling" that the market domination is limited to low technology, there is evidence that nothing is immune.

However, despite a desire to blame the Japanese for dominating our manufacturing, it comes about because there are U.S. buyers and distributors for their goods. Distributors come about because of the intense emphasis we have on profit and return-on-investment. By only distributing and not designing and manufacturing the investment is negligible, giving a high return-on-investment.

The intent of the paper is to describe variously "how" this market/product domination is carried out. Like any good Japanese product, the ideas within the paper have been taken liberally from many sources -- mostly without credit. It should be self evident that, we (the U.S.) have a problem. Each of us, whether we be part of industry, government, or academia, can now address the issues we're responsible for. There's no real need for another fact-finding trip to Japan to further define the problem. Japan is clearly not a place to search for the solution.

Many solutions are required. Freezing the current level of government size spending and non-productive people (e.g., lawyers) would be fine first starts. Living within our collective energy budget is also needed. Rather than engaging in a trade war the following mechanism could simply address the trade deficit:

> No company can import and distribute a foreign product without arranging an equal export credit. That is, a company; such as Itel who buys and resells Japanese computers can get agricultural products to sell or it could export its own services in an equal amount. The trade balance has to be the distributor's problem -- not that of the President, or the Secretary of Commerce or Congress.

<u>Appendix 1</u>. <u>A Chronology of Systematic Domination</u>*

I. "<u>Development of a domestic Japanese industry</u>. The Japanese industry is developed and grows rapidly. The major aspects that mark this development include:

(a)Market control. Imports limited essentially to zero. Only a few major manufacturers are permitted. Prices remain significantly higher in Japan than in other competitive markets.

(b)Borrowed technology. The Japanese borrow heavily from foreign technology, including a large number of purchased licenses and patent rights, and wholesale reverse engineering.

(c)Vertical integration of most manufacturing.

(d)Major investments. Major investments are made in modern plant, equipment and technology, both for the final product and throughout the vertical chain of manufacturing. Continued research, development and plant investment expenses are made.

II.<u>Establishing an export market base</u>.

(a) The establishment of world-wide sales organizations.

(b) Researching and understanding of the foreign markets.

(c) Establishment of a reputation for quality and reasonable prices.

(d) A limited focus, especially in those markets less attractive to domestic manufacturers.

III. Major market penetration. Major market penetration occurs usually during an economic downturn in Japan. Previous efforts by the industry have set the stage for them to be successful in this endeavor. It is marked by the following considerations:

(a) Cooperation among the Japanese companies with respect to models, prices, and markets.

(b) Focus at the mainstream of the foreign market.

(c) High inventories because of poor markets in Japan, i.e., an export push at any cost is necessary and expedient.

(d) Extremely low prices to the mass market to gain market share rapidly, i.e., a knock-out punch to the domestic manufacturers. Modern plants, reasonable costs, an established export organization, and good reputation set the stage for success.

At this time, marketing muscle is established. Not only was the export market share large, but the domestic market remained closed. It should be pointed out that this major market penetration had been made by a combination of factors, as outlined. The greater marketing muscle allows the Japanese manufacturers to profit from their long investment.

IV. Market exploitation. This period is marked by higher prices -- often higher than domestic manufactured models. However, the higher prices are often more than offset by perceived higher quality, both real and imagined. There is also continued cooperation on prices and markets, as well as continued limitations on imports to the Japanese market."

paper 1, printed 10/19/86. Original paper 10/78.

JAPAN IMPRESSIONS (Part I)

Reaffirmed to be #1 in Sales and Technology (see slide on IBM hi-end)

Now claim to be #1 in supercomputers

Technologies that are dominated:

- Base materials and production (esp. Quality)
- CRT, LCD, EL
- Printing, fax, thermal, xerography
- Magnetic recording and video disk
- Video, video b/w compression and image processing
- Voice i/o
- Communications (installing systems)
- Fiber optics (installed LANs)
- Packaging and PWB's
- Semis and Semi CAD
- ECL, Bipolar, MOS, CMOS
- Research in J^2, (GaAs and HEMT in factory?)
- Robotics?

The Japanese Computer Industry

- (MITI and Fujitsu, Hitachi, NEC, Mitsubishi, Toshiba, Matshusta, Oki)
- have and are implementing a vision of 5G computing based on AI and high performance processing.

Program includes:

- Supercomputer technology, 1981 + 8 years (VLSI, J^2, GaSa, HEMT)
  - new architecture and technology

- Dist. Proc. and LAN's ($40M over 3 years)
- ICOT (the main push)

. Next generation (farther out)

. NTT Si Compiler - a real compiler that's so far produced a 13k t chip in 2 months <u>without</u> using a CRT. Totally language driven with separate backends for CMOs, HMOs, bipolar or ECL. 2,000 people are working on VLSI and this will be used for smaller companies. Also, several new architectures, including 2 data flow.

<u>ICOT</u> - <u>INSTITUTE OF NEXT GENERATION COMPUTERS</u>

. Headed by Dr. Fuchi

. Coupled into universities - 5 people + 5 x 7 company research

. Use a 2060 for Prolog and LISP

. Two machines are to be built by companies (in 2 years)

- RDMS
- Prolog processor

(Data flow for Resolution desired)

. Prof. H. Goto, TU, believes Prolog is wrong and has Mitsui building a 10 mips LISP machine for him.

. Government funded, Company's fund space,...

. <u>They</u> <u>are</u> <u>driven.</u>

. Three Groups: Architecture, Application, Human Interface

JAPAN IMPRESSIONS (Part I)

Reaffirmed to be #1 in Sales and Technology (see slide on IBM hi-end)

Now claim to be #1 in supercomputers

Technologies that are dominated:

- Base materials and production (esp. Quality)
- CRT, LCD, EL
- Printing, fax, thermal, xerography
- Magnetic recording and video disk
- Video, video b/w compression and image processing
- Voice i/o
- Communications (installing systems)
- Fiber optics (installed LANs)
- Packaging and PWB's
- Semis and Semi CAD
- ECL, Bipolar, MOS, CMOS
- Research in J^2, (GaAs and HEMT in factory?)
- Robotics?

The Japanese Computer Industry

- (MITI and Fujitsu, Hitachi, NEC, Mitsubishi, Toshiba, Matshusta, Oki)
- <u>have</u> <u>and</u> <u>are implementing</u> a vision of 5G computing based on AI and high performance processing.

Program includes:

- Supercomputer technology, 1981 + 8 years (VLSI, J^2, GaSa, HEMT)
  - new architecture and technology
- Dist. Proc. and LAN's ($40M over 3 years)
- ICOT (the main push)
- Next generation (farther out)
- NTT Si Compiler - a real compiler that's so far produced a 13k t chip in 2 months <u>without</u> using a CRT. Totally language driven with separate backends for CMOs, HMOs, bipolar or ECL. 2,000 people are working on VLSI and this will be used for smaller companies. Also, several new architectures, including 2 data flow.

ICOT - INSTITUTE OF NEXT GENERATION COMPUTERS

- Headed by Dr. Fuchi
- Coupled into universities - 5 people + 5 x 7 company research
- Use a 2060 for Prolog and LISP
- Two machines are to be built by companies (in 2 years)
  - RDMS
  - Prolog processor

  (Data flow for Resolution desired)

- Prof. H. Goto, TU, believes Prolog is wrong and has Mitsui building a 10 mips LISP machine for him.
- Government funded, Company's fund space,...
- They are driven.
- Three Groups: Architecture, Application, Human Interface

INDUSTRY, GOV'T, ACADEMIA,...SOCIETY CONSPIRE TO AID THE JAPANESE.

THEY HAVE SYSTEMATICALLY DOMINATED TRADE BY:

1. DEVELOPMENT OF DOMESTIC INDUSTRY
2. ESTIMATED EXPORT BASE
3. MARKET PENETRATION
4. MARKET EXPLOITATION

STRATEGY & TACTICS OF THE JAPANESE

. INDUSTRY, GOV'T, ACADEMIA OPERATE AS TEAM.

. MITI IS AUTOCRATIC - CREATES JAPAN CLUB.

. WE HAVE NO MITI TO PROTECT AND BUILD TRADE RESOURCES.

. THE INTENT IS TO DOMINATE SEMIS + COMPUTERS.

. U.S. DEPT'S OF LABOR & COMMERCE AREN'T SKILLED & AREN'T TOGETHER.

THE JAPANESE "LIVE TO WORK VS. WORK TO LIVE"

. FOCUS IS ON WORK + LOYALTY.

. RISK TAKING IS POSSIBLE (WITH SECURITY).

. QUALITY CONTROL IS AT WORKER LEVEL.

. KNOW-HOW FOR TEAM (INTER-DISCIPLINARY) WORK.

. WORK IS THE GOALS VS. FREEDOM (NON-WORK).

J - "KNOW HOW" FOR DESIGN/TECH. ACCULTURATION

. PROCESSES ARE ORIENTED FOR COMPETITION, QUALITY, GROWTH, FLEXIBILITY.

. ACQUIRED COMPUTER TECH. FROM WORLD, (U.S.) - BUT IMPROVED ON IT.

. DESIGN INCLUDE: LOOK-ALIKE, LICENSE, REVERSE - ENGINEERING.

ENGINEER/DESIGN FOR LONG-TERM/NEEDS.

. DON'T DO MARKETING.

. GO FOR QUALITY VS. THROW-AWAY
        LONG-LIFE CYCLE

J - UNDERSTAND AND MANAGE A COMPLETE PROCESS

- INVEST LITTLE IN RESEARCH - THE U.S. DOES IT FOR 'EM.   COUPLING TO ARPA

- MITI HAS FEW LABS, BUT FUNDS (CONTROLS) WORK.

- ENGINEERING VS. SCIENCE

- UNDERSTAND GROWTH, VOLUME, DEMAND, ETC.

- THEY GIVE UP PROFIT FOR GROWTH (IN SHORT TERM)

BAD? PICTURE 3 ISLANDS:

```
             ________         _______________     _______
             _______
        !        !          !               !     !       !
!       !
        ! ENERGY ! ---->    !   CONSUMER    !<--- ! MFG.  !<--
>! RAW   !
        !        !          !               !     ! GOODS !
! MAT   !
        !________!          !_______________!---> !_______! --
>!_______!
                                               !
                                               !
                                               !INFORMATION
```

J AND AMERICAN BUYERS HAVE CHANGED FROM INVENTOR - MFG. - DISTRIBUTOR

TO DISTRIBUTOR

NO WAY FOR US TO RE-ENTER LOST BUSINESS.

(TEXTILES, STEEL, RADIOS, SEWING MACHINES, TYPEWRITERS, CAMERA/OPTICS, SMALL CARS, TV, TAPE RECORDERS, WATCHES, CALCULATORS, VIDEOTAPE, SEMICONDUCTORS, COMPUTERS.)

AMERICAN REGARD (WORSHIP) OF MBA.

AMERICAN BUSINESS FOCUS ON

AT A SOCIETY LEVEL THEY'RE TOGETHER

ENGINEERING & SCIENCE VS. LAW AND BUSINESS TRAINING

LOWER TAXES.

LESS MILITARY, NIH, NASA EXPENSES.

COMPUTING IS SUPPORTED.

```
+---------------------------+                                         GB0001/52
|   |   |   |   |   |   |   |
| d | i | g | i | t | a | l |  i n t e r o f f i c e   m e m o r a n d u m
|   |   |   |   |   |   |   |
+---------------------------+
```

Subject: **Make vs Buy Guidelines Update (from 3/5/76)**

To: File

Date: 3/28/79

From: Gordon Bell

Dept: OOD

Loc: ML12-1/A51 Ext: 223-2236

What DEC SELLS not what it BUILDS is the more important issue for continuing success. In a rapidly changing industry where technologies can quickly become obsolete, it is essential that DEC maintain flexibility and not become over committed to any particular technology or process. As we make more and more of the items we sell, we become more rigid. Opportunities in the marketplace can be delayed or lost forever. Opportunities to cost reduce by building more inside will stay with us. The following guideline is intended to help us focus on these issues.

1.DEC wants to build unique products that offer specific advantages to its customers. Profitability alone is not sufficient.

2.High ROI by itself is no reason to build anything (e.g., it robs resources from other, more essential projects).

3.The general rule should be, if we don't make it now, buy it.

4.Proposals to build must explicitly demonstrate that:

a.project will result in a quantum jump in technology or

b.needed to introduce (or confine) a vital technology to DEC or

c.present or developing vendors are unable to supply demands of ON-GOING high production item.

5.All proposals to build should address and be screened by at least the following criteria:

a.DEC's forecasted needs exceed the volume of at least the smallest economically viable vendor.

b.DEC's engineering resources to accomplish task is at least comparable to vendor.

c.Incremental NOR/employee will be above the corporate average for the effort. [We should strive to increase "PRODUCTIVITY".]

d.Hardware products can be sold through the Components Group. [The product is inherently good enough to stand on its own.]

e.ROI analysis of not only the results of pursuing the project but the corresponding results when using the vendors part.

f.Level-of-integration of the project. [We should tend to increase level-of-integration-focus on MAKING what we sell--NOT what we BUY.]

g.The resulting incremental NOR to development cost ratio compare with Corporate NOR to total engineering ratio budgets. [Won't become an engineering sink.]

6.We must have a "buy out" advocate to test analysis (in Manufacturing, Purchasing, and Engineering?).

7.Proposals to "make" must be explicit with respect to the level-of-integration covered (i.e., which parts). "Making" is not a carte blanche licensing to make everything.

00 CORE DECGRAM ACCEPTED S 005092 O 429 04-AUG-82 16:44:19

```
*****************
* d i g i t a l *
*****************
```

TO: see "TO" DISTRIBUTION

DATE: WED 4 AUG 1982 3:58 PM EDT

FROM: GORDON BELL
DEPT: ENG STAFF
EXT: 223-2236
LOC/MAIL STOP: ML12-1/A51

MESSAGE ID: 5171520001

SUBJECT: ALPHA OMEGA...POST VN COMPUTING: PERSONS & COMMENTS?

Bruce Delagi and I led a group from CDC, Univac, and Harris to define a research program on parallel computing for use across a large array of problems including AI. This is a program which we believe must be executed in order to do the basic work necessary to produce machines to compete with those produced by the Fifth Generation Computer Program.

CALL FOR COMMENTS ON THE DRAFT PROPOSAL

Please call or EMS me for a copy of the research proposal. Bruce is going to give a seminar at Hudson on it and we'd like to get your comments on how the proposal can be improved either by narrowing or widening the scope.

CALL FOR PEOPLE WHO WOULD LIKE TO WORK ON THE PROJECT

Now, we would like to carry the proposal into the next phase by having the group who are going to carry out the work write the detailed research proposal/plan.

If you would like to work on this, please let me know now.
Individuals are needed.

We are looking for someone to head the program. Any candidates?
(Please forward mesage as appropriate).

"TO" DISTRIBUTION:

```
BRUCE DELAGI               ARNOLD KRAFT              PEG:
RAD:                       BARRY RUBINSON            TMC:
```

WPS USERS - Leave HP mode and type <CR>

THE ENCORE CONTINUUM:
A MULTIPROCESSOR AND DISTRIBUTED WORKSTATION COMPUTING ENVIRONMENT

C. Gordon Bell[1], Henry B. Burkhardt III[1], Steve Chapin[2], Steve Emmerich[1], Russell Moore[2], Isaac Nassi[3], Charle' Rupp[3], David Schanin[2]

1Encore Computer Corporation, Wellesley Hills, Mass. 02181

2Hydra Computer, an Encore Company responsible for Multimax, Natick Mass.

3Resolution Systems Inc., an Encore Company, Marlboro, Mass.

SUMMARY

The Encore Continuum is a UNIX compatible, computing environment designed to provide a range of computing styles from distributed workstations to multiprocessor, superminicomputers. The Appendix provides a taxonomy for multiprogramming, distributed processing, and parallel processing, and defines the "Multi," or multiple microprocessor, a new computer class which we believe will lead to parallel processing.

INTRODUCTION

Traditionally, companies have used multiple bit-slice technologies to produce a "computer family" to cover a price-performance range. Powerful MOS and CMOS microprocessors are likely to change this strategy. Some companies have built proprietary microprocessors to implement new computers in their range (e.g. IBM's PC/370, DEC's MicroVAX, and Data General's Micro-Eclipse). Others have built a new computer family that leverages off of the performance range of these standard microprocessor families (e.g. IBM PC and AT).

A multiprocessing approach with up to four processors to achieve a performance range is used by a few mainframe vendors. The high interconnection costs between processor, memory, and input/output components limits the success of multiprocessing.

Encore has constructed an entire computer family, of both workstation and shared-resource styles of computing, in which the performance range stems from a microprocessor-based multiprocessor computer - the "Multi." Encore's Continuum consists of multiprocessors, distributed processing servers, high-resolution terminals, and workstations, all interconnected by local area networks. Microprocessors offer sufficient performance to support today's general-purpose computing, at all levels of performance. The Encore Continuum architecture is designed to exploit rapidly improving micros. The Multi scales linearly in price and performance over an order of magnitude. Networked workstations scale over a factor of several hundred.

GOALS OF THE ARCHITECTURE

Major objectives of the hardware architecture include:

- Cost-effective multi-user computing, with incremental scalability of an order of magnitude for processor, memory, input/output computing resources.
- Hardware-independence of microprocessor architecture, data formats (e.g. floating point formats) and communications technologies, in order to "track" the transitions to new technologies without obsoleting the architecture.
- Arbitrarily large virtual memories to support memory-intensive applications; and accommodation of very large physical memories for high-performance memory utilization and input/output buffering.
- Large numbers of high-bandwidth Input/Output channels with dedicated intelligent controllers.
- Ability to expand or rearrange systems easily to respond to changing requirements, new applications and new computer and communications technology.
- Access to the Continuum both within a local area, and from long distances, using industry-standard Local and Wide Area Network protocols from a variety of industry-standard user-interface devices, including terminals and Personal Computers.
- High-quality human interface devices, with full-page display formats, integral graphics, distributed windowing, and industry- standard connections, protocols, and application software environments.

The goals of the Encore Continuum include a complete compatible software environment:

- To support multiprogramming, distributed computing, and revolutionary parallel processing applications, with minimal degradation of throughput and response-time due to systems software overhead.

.With UNIX-compatibility for multiprogramming -- to achieve longevity of stable, system interfaces; and to acquire a large flow of compatible, competitive software from many sources. By having a single, standard interface, applications software can be written that runs on all hardware rather than just a single vendor's. Thus, the software industry is in competition to provide better products.

.To provide identical programmers' and users' interfaces across the Continuum -- in order to protect investments in applications programs and personnel training, and to enforce security restrictions uniformly, where needed.

. To store data in the appropriate locations in order to minimize communications cost, to enhance system responsiveness, to enable data and equipment to be secure, and to give users access to data when and where it is needed.

.To allow sharing of resources, both physical and logical, between nodes in the Continuum, and between the Continuum and non-Encore computer equipment.

ENCORE PRODUCT OVERVIEW

Figure 1 shows the hardware products of the Encore Continuum -- the Multimax, Resolution HostStation and WorkStation, and the Annex computer for network input/output:

.Multimax: A multiprocessor that spans a processing performance range of 1.5 to 15 Mips in expandable increments of 1.5 million instructions per second (Mips). Input/Output throughput can be expanded in increments of 1.5 Mbytes/sec. to 15 Mbytes/sec. Memory can be expanded in increments of 4 Megabytes to 32 Megabytes.

.Annex (Ancillary Network EXchange computers): Intelligent, low- cost terminal and PC concentrator and gateway computers for Multimax systems and Resolution Stations. Functions such as terminal access to the Continuum, and gatewaying to personal computers, public data networks, and to external computing environments such as SNA, take place via Encore's Annex computers.

.Resolution R100 Host Stations and R500 Workstations: The R100 is a high resolution, large screen, multiple-window, multiple host access station designed for host-based computer access. The R500 is a compatible Workstation with local processing and mass storage.

.Interconnection with other computer vendors which support the TCP/IP protocols.

.Local Area Network (LAN). Ethernet (IEEE 802.3) LAN is used to interconnect all computing elements and provide computer to computer intercommunication, common gateways to other computers and public communications networks, and common access to terminals and Personal Computers.

.UMax 4.2 and UMax V. Two, software environments are provided which are derived from the University of California/ Berkeley BSD 4.2 and AT&T Unix. These are primarily for technical and commercial applications use, respectively. Software includes the 200 general-purpose tools provided with UNIX, a full complement of user productivity tools, editors, languages and debuggers, and a relational database with Ally, our 4th- generation language.

.A communications architecture for supporting cooperative, intelligent processing between networked Multimax systems, Resolution Stations, and Annex computers.

The Multimax

Encore's Multimax computer uses multiple microprocessors sharing a common memory to achieve a scalable, large performance range, lower price/performance over range, and higher reliability and availability than a uniprocessor.

Multimax's power is derived from the Nanobus^tm, used to interconnect the Multimax options within a backplane card holder for 20 cards which is about 12" long, and corresponds to a transmission time of approximately one nanosecond (see Figure 2). Every 80 nanoseconds a 32-bit address (corresponding to an ability to access up to 4 Billion bytes of memory) and a 64-bit data word can be transmitted from card to card plugged in along the bus; thus Nanobus has a data carrying capacity of 100 Million 8-bit bytes per second. (By comparison, standard and emerging busses for Multi's are usually one-tenth to one- fourth as fast.) All cards can operate either offline or online in completely standalone basis. Each can carry out a complete self- diagnosis.

The Nanobus provides the key to product longevity by being able to accept new, higher speed processors that will evolve with CMOS VLSI. Real-time data can be processed at up to full bus bandwidth (100 Mbytes per second) using direct memory access (DMA), and via priority, programmed interrupts (40,000 events per processor/second maximum) and via direct program control.

At least one of each type of the following four card-type options is required:

.Dual Processor Card (DPC) - two, National 32032 processors share a common 32 Kilobyte cache. A high performance floating point option utilizing Weitek chips is provided for arithmetic intensive applications. Encore rates this processor at 0.75 Million instructions per second. With 10 DPC's, a single Multimax can process up to 15 Million instructions per second.

.Ethernet/Mass Storage Card (EMC) - interfaces to Ethernet and to disks. This card contains a 32032 for managing input/output transfers and diagnostics. It has sufficient capability to operate as a LAN-based file service computer. Up to 11 DPC or EMC cards can be placed in one system.

.Shared Memory Card (SMC) - 4 Megabytes of memory, organized in two independent banks with error detection and correction codes and utilizing 256Kbit memory chips. Eight SMC's cards can be placed in a single system, providing up to 32 Megabytes of memory. An on-card computer can completely check and diagnose the memory in an offline basis.

.System Control Card (SCC) - performs bus arbitration, logs errors, provides system diagnosis, and communicates with an operator and a remote console.

Peripheral Options. In addition, Multimax offers: battery backup;

fixed and removable disks of 520 and 300 Megabytes respectively; and 1600/6250 bpi magnetic tape options.

The Local Area Network

Encore currently uses the most accepted Local Area Network standard, IEEE Standard 802.3 (Ethernet), to interconnect its computing nodes at a rate of 10 million bits per second. Other standards will be adopted in response to market requirements. The function of the LAN is for:

.computer to computer communication for distributed processing, file transmission, and virtual terminal access among computers.

.common access to other networks via Annex Gateway Computers.

.common access from terminals and PC's via Annex Concentrator Computers.

.formation of a fully distributed computing environment using Encore's powerful, single user workstations.

.connection to existing personal computers, minicomputers and mainframes

A LAN is not required for system operation.

Annex **Terminal and Personal Computer Access (Concentrator) Computer**

Each Annex Concentrator Computer attaches up to 16 terminals and printers along the LAN in a fully distributed fashion, permitting up to several thousand terminals to access all computers within a single LAN. Five Annex's can be connected to a single LAN port, or it can be directly connected to a Multimax if there is no LAN. Annex roughly doubles the processing power in the Ensemble since roughly one is used with each Multimax processor. Wiring from terminals to computer is simplified by distributing the physical connections to the Annex concentrators, unlike most terminal architectures which require all RS-232C terminal lines to be connected to a particular computer. Any serial port on an Annex can communicate with any Multimax or Resolution on the LAN since the LAN is basically a distributed switch. Annex incorporates a general-purpose remote procedure call facility to communicate with Multimax systems and Resolution workstations.

Annex is programmed to perform time-consuming functions such as character processing on input from terminals, and screen updating on output to terminals, which require no host or central database interaction.

On terminal initialization, a switch program asks the user which of the available hosts he wishes to log in to. The Annex notifies the selected host, the terminal becomes connected to it, and the host runs the standard log-in process. The software also supports connection switching and connection binding. Connection switching allows users, once a connection to a host is established, to connect as well to other hosts, and then to switch between hosts. Connection binding allows site managers to make certain ports on certain Annexes "bound" to a given host, for users who are bound to an application, or for dedicated peripherals such as line-printers.

Annex is programmed to perform time-consuming functions such as character processing on input from terminals, and screen updating on output to terminals, which require no host or central database interaction.

Annex contains a National 32016 microprocessor with 128 Kilobytes of memory. Annex has options for both asynchronous and synchronous communications and direct and modem connections. Standard terminals and Encore Host Stations communicate with hosts at up to 38.4 Kilobaud transmission rate per terminal.

Hard copy options include a 200 character per second matrix printer and 800, 1200 and 1800 lines-per-minute printers.

Annex **Gateway Computer**

The Annex Gateway to the LAN and provides gateway access to various communications and industry networks using protocol conversion hardware and software. The protocols include: IBM SNA, IBM Block Mode Terminals (3270), IBM PC, and X.25. Presentation-level services associated with gatewaying generally run in the host, which communicates via the remote-procedure call communications architecture with protocol-conversion software running in the gateway.

RESOLUTION COMPUTING STATIONS

Resolution Stations are Encore's host station and workstation product line. The host stations provide the workstation characteristics including large-screen, high-resolution bitmap, support of terminal and graphics protocols, high interactivity, multiple windows and access to multiple hosts.

The Resolution Stations use a 19" screen size to give an unscaled, ledger sized 11" x 14" page at high resolution using 1056 x 864 pixels. A ledger sheet of 176 columns and 86 rows can be displayed. Keyboard and pointing device (e.g. mouse) input are provided. The Stations (without keyboard) occupy a desk space of 16-1/2" square. Text and graphics protocols are provided which allow existing and future software to be run without modification, including: VT100, ANSI 3.64, Tektronix 4010/4014, Regis, and VDI for GKS.

The stations are designed to address a variety of applications including: the station of choice for the professional programmer; text and typographic input; engineering; business and accounting where computational power and large screens are required; and special functions such as translation where side-by-side text is required.

Resolution Host Station - R100

The R100 is a single, but universal Host Station because it can communicate with as many as three computers through separate windows. For example, the R100 can simultaneously access Hydra, a traditional host (eg. IBM 370 or VAX-11), and a PC AT for personal computer software. All the functions of the R100 are carried out under the program control of a National 32016 microprocessor. The R100 is also designed to be used as a remote, slave station to conventional workstations (i.e. a user can have a workstation at home or a second office).

The R100 can be upgraded to become an R500.

Resolution Workstation - R500

The R500 is self-contained computer system with a primary memory of two megabytes and disk memory of 20 Megabytes. The processor, a National 32016, is completely compatible with other computers in the Encore Continuum. Thus, software can be run either within the Workstation, among Workstations, or among Host Stations and Multimax systems in a completely flexible and transparent fashion.

SYSTEM SOFTWARE IN THE ENCORE CONTINUUM

The software environment in the Encore Continuum is UNIX-compatible, enhanced to support both distributed and parallel processing. Distributed processing support is provided by a communications architecture that provides for cooperative, efficient inter-processing between networked Multimax systems, Resolution Stations, and Annex computers. Language constructs for assigning task forces of processors to a single process for support of parallel processing are also provided.

THE UMAX 4.2 AND V DISTRIBUTED SOFTWARE ENVIRONMENT

UMax 4.2 and UMax V constitute Encore's standard operating systems. Programs that run under either UNIX System V or UNIX BSD 4.2 are compatible and portable to the corresponding Multimax and Resolution systems.

In addition to UNIX standards, the Encore Continuum extends UNIX:

. to take full advantage of demand-paged virtual memory, multiprocessor performance, and distributed terminal architecture,

.to provide data sharing and synchronization mechanisms between user processes in UMax 4.2. Additional system calls and library subroutines support these new multiprocessor functions.

. by unifying language standards and language-related data formats across both versions of the operating system, to simplify portability of applications between environments.

UMAX PERFORMANCE ON THE MULTIMAX

UMax 4.2 and UMax V incorporate three strategies for high-performance that are inherent in the Encore Continuum-- symmetrical multiprocessing, scalability to a large number of processors, and distributed intelligent peripheral control.

Symmetrical multiprocessing, or multithreading achieves maximal performance in the Multimax by assuring that any processor can execute any user process or part of the operating system. This assures no inherent bottlenecks. One copy of the operating system supports all the processors, memory, and Input/Output computers. In order to allow multiple processors to gain simultaneous access to operating system services concurrent access must be controlled to each process and operating system routine. Processors must be locked while shared data structures are being read and written, to minimize performance degradation due to processor idling and context switching.

Controlled, concurrent access to internal UMAX resources is achieved with the following three mechanisms:

.Spin locks -- accomplish synchronization by executing tight instruction loops until the expected condition occurs (used only for critical, short-duration events).

.Semaphores (Dijkstra style) -- accomplish synchronization by putting requesting processes to sleep until the requested resource is available.

. Read/write locks -- specialized forms of semaphores that provide access to data structures for a single writer or multiple readers.

Scaling performance to many processors and very large memories is a major issue. Multithreaded operation alone will not realize the performance potential inherent in the Multimax since there are more resources in Multimax. Two additional performance enhancements have been added to accommodate a large processing load: shared data in the operating system is minimized; and the UNIX terminal driver has been redistributed to the Annex concentrator computers.

The first method caches frequently used in-memory resources, such as file and directory entries. For resources that are tables, it is generally appropriate to lock individual entries rather than the whole table. In other cases, kernel tables have been divided into subpools of entries, linked together and located by hashing. This minimizes search times, and allows for locking of subpools rather than whole tables.

Annex computers have their own processors and memory, modified UNIX terminal and printer drivers are down-line loaded to Annexes to minimize Multimax loading. Thus Annexes handle the major terminal processing. In large configurations where the interrupt- and computational overhead for character processing is high, Multimax performance is increased by Annex pre-processing.

High-Level Languages and Debuggers

.C. This language is supported by an optimizing compiler. Traditional assembly languages for system-level and time- critical applications programs ARE minimized.

.FORTRAN-77. Fully conformant to the ANSI standard, and also a highly optimizing compiler.

. Pascal. ISO standard.

. COBOL 74. FIPS intermediate-level 2.

UMax requires a remote debugger of supervisor mode programs, with a "host" portion running on a remote system which communicates to the "target" Multimax through serial ports. Although not intended for Multimax users, this debugger facilitates remote Encore diagnosis of system problems. The debugger supplied to users is for local debugging of user mode code. It is intended for user-developed C and FORTRAN-77 applications.

CONCLUSIONS

The indefinite expandability goals of the architecture are satisfied by allowing almost unlimited numbers of each product to be added to a Local Area Network. Multimax and Resolution are both incrementally upgradeable to higher levels of performance, by additional processors, memory, and mass storage over an order-of-magnitude range. The cost-effective hardware, and the software environment, UMax, makes efficient use of incremental resources, and also providing standardization, portability, and ease-of-use inherent in UNIX.

THE MULTI - A NEW COMPUTER CLASS

The Multi (for multiple, microprocessor) is an emerging computer class made possible by recent, powerful micros that have the speed and functionality of mid-range super minicomputers. A Multi is scalable, permitting a single computer to be built which spans a performance range, in contrast to computer families implemented from a range of technologies. The Multi is a significant alternative to conventional micros, minis, and mainframes.

Multis can be used today - without redesigning or reprogramming of applications - because computer systems operate on many independent processes. With Multis, it is possible to operate on many of these processes in a parallel fashion, each on an independent processor, transparent to the user. Most importantly, the Multi is likely to be the path to the Fifth Generation based on parallel processing.

This Preface briefly summarizes the generic Multi - what it is, why it has come to be, and how it is applied - to better prepare those unfamiliar with this new concept for the Multimax design discussions which follow.

THE MULTI - ITS HISTORICAL AND TECHNOLOGICAL BASIS

Computer systems with multiple processors have existed since the second generation (the Burroughs B5000, a dual symmetrical processor, was introduced in 1961). Most mainframe vendors and some minicomputer suppliers currently offer systems with two to four processors. However, these structures have been expensive to build - due to the high cost of typical processors - and hence have found application mostly for high-availability computing (e.g., communications, banking, airline reservations).

The modern 32-bit microprocessor's function, performance, size, and negligible cost are creating a new potential for multiprocessors. With 32-bit addressing, hardware support for paged, virtual memory, and complete instruction sets with integer, floating, decimal, and character operations, these chips offer performance levels comparable to mid-range superminis such as the VAX™-11/750.

The Multi is a multiprocessor structure designed to use these new micros to advantage. It employs an extended UNIBUS™-type interconnect, whereby all arithmetic and input/output processor modules can access common memory modules. Cache memories attached to each processor handle approximately 95% of its requests, limiting traffic on the common bus. With these local caches, an order of magnitude more processors can be attached before saturating the common bus.

With proper attention to design of critical elements (e.g., the common bus), large multis using current-technology micros can outstrip high- end superminis, and even some mainframes, in total performance. This advantage should continue to grow. The performance of MOS and CMOS microprocessors has improved (and is expected to continue to improve) at a 40% per year rate, while the TTL and ECL bipolar technologies (on which most traditional minis are based) have shown roughly a 15% per annum improvement.

When compared to traditional uniprocessor designs, the Multi delivers improved performance, price, and price/performance.

.**Configurability Range** - through modular design, the Multi allows the user to "construct" the correct level of performance or price, without having to choose among a limited number of computer family members.

.**Availability** - the Multi has inherent reliability through redundancy because it is built from a small number of module types (typically, four). With appropriate software support, faulty modules which are replicated can be taken out of service - allowing continued operation with minimum downtime.

. **DesignabilityandManufacturability** - because the Multi is comprised of multiple copies of a small number of modules, instead of the large number of unique boards in a typical minicomputer, it is faster and less expensive to design. Individual module types are produced in larger volumes, producing improvements of 30% in manufacturing costs due to a learning curve over conventional uniprocessors.

.Evolutionary Technology Upscaling with appropriate design, Multis allow long-term performance upscaling through evolution. As key components of the processor and memory cards improve over time, the computer can be upgraded without replacement in an evolutionary fashion. In addition, increased cache sizes through denser parts and improved cache management disciplines will permit substantially greater numbers of processors to be installed without saturating the common bus (provided the bus design has allowed for this performance growth). All of this will permit graceful and cost-effective evolution in processor performance, input/output throughout and memory size over a range of one to two orders of magnitude over a ten-year period.

APPLYING THE MULTI

Multis will be widely used for many applications because they can provide the most cost-effective computation unless the power of a single, large processor is required today on a single, sequential program. Because of the rapid rate of microprocessor evolution, the percentage of applications requiring single-stream performance in excess of that delivered by each of the Multi's processors is already quite small and will continue to shrink. On the other hand, we believe the emergence of the Multi will lead to parallel processing.

We can better understand where Multis may be applied by classifying the degrees of parallelism achievable. Grain size is the period between synchronization events for multiple processors or processing elements. Synchronization is necessary in parallel processing to initialize a task, parcel out work, and merge results. The Multi exploits the Coarse- and Medium- grain parallelism within an application, not the Fine-Grain, which is the focus of vector, pipelined computers (e.g. Cray 1) on wide word microprogrammed array processors. Groups of Multis and conventional Workstations can interact over networks to implement Very Coarse granularity.

		Synchronization	Encore Computer
	Construct for	Interval	Structures to
Grain Size	Parallelism	(instructions)	SupportGrain
Fine Array)	Parallelism inherent in single instruction or data stream	1	Specialized Processors (e.g., Systolic or added to Multimax
Medium	Parallel processing or multi-tasking within a single process	20-200	Multimax
Coarse	Multiprocessing of concurrent processes in a multi-programming environment	200-2000	Multimax
Very Coarse	Distributed processing across network nodes to form single computing environment	2000-1M	Multiple Multimaxes, Encore workstations, and other machines, on Ethernet

As all modern operating systems are multiprogrammed, whereby each job in the system is at least a single process, and many support multi- tasking or sub-processes, most current applications are already designed to take advantage of the Multi at the Coarse-Grain level. Also, when used in a timesharing or batch environment, each processor of a Multi can be assigned to a separate job to exploit the parallelism inherent in the work load. The UNIX™ pipe mechanism allows multiple processes to be used concurrently on behalf of a single user or job to achieve parallelism in reading a file, computing and output to one or more files. Transaction Processing is inherently a pipeline of independent processes.

The Multi can be a more efficient multiprogramming computer than the traditional uniprocessor, because the number of context switches (and hence lost time) is dramatically reduced. Additional parallelism is in the operating system itself. Execution of operating system code often accounts for 25% or more of available processing time, when file, database, and communications subsystems are included. By restructuring the operating system internals, multiple, independent system functions can be executed on independent processors.

When reprogramming of subsections of the application is possible, Multis permit additional parallelism to be realized, at the Medium- Grain level (i.e., parallel processing), by segmenting a problem's data for parallel manipulation by independent processors. This has been shown to be quite effective on simulation, scientific modeling, and analysis problems (such as matrix operations, linear programming, partial differential equation solution, etc.) which permit data elements to be processed in segments.

Finer granularity of parallelism is achievable in the framework of the Multi through specialized processors installed into its common bus. This is most effective when the algorithms are known a priori, such as in certain signal processing applications.

We believe multiprocessors, augmented by both programmable pipeline, i.e., systolic, and specialized processors for Fine-Grain parallelism, will cover the widest range of problems of any computing structure.

Notes

tm Digital Equipment Corporation

tm AT&T

tm Encore Computer Corporation

FIGURES

Figure 1. The Encore Continuum, a distributed processing and multiprocessor computing environment.

Figure 2. The Encore Multimax, multi(ple) microprocessor computer.

Figure Configurability ... either a space or the planes from my multi paper

Figure Price/Performance ... could be a dimensionless graph which shows Hydra against VAX, but doesn't name either or put price or performance dimensions.

GB.PA.NC.AD1

THE DIGITAL COMPUTING ENVIRONMENT

Gordon Bell
Encore Computer Corporation
(formerly Digital Equipment Corporation)

Bill Strecker
Digital Equipment Corporation

ABSTRACT

The Digital "E" Enviroment for computing is the aggregate of a wide range of compatible computers and ways of indefinitely interconnecting them to provide its users with both generic (eg. word processing, electronic mail, database access, spread sheet, payroll) and basic (eg. languages, file sytems) computing facilities that he may use directly, or "program". The intent is to provide the widest range of choices, by

having complete compatibility, for where and how computing is to be performed without having to make a priori commitments to a particular computer system class (i.e. mainframe, minicomputer, team computer, personal computer). The design of the Environment is substantially more than a single range of compatible computers because different styles of use are required depending on the machine class and all the computers must be interconnected and work together. The Environment is the first of what we define as a multi-level, homogeneous computing architecture.

BACKGROUND

BATCH MAINFRAMES FOR CENTRAL SERVICES

In the first two computer generations, 1950-1970, computers were used in batch processing under the name of mainframe computing. During the 70's the mainframe began to be used almost interactively from remote job entry terminals at "glass key punches". The general direction is to have larger mainframes and larger terminal networks that interconnect to a single computer by an array of front end computers. When more power is required, more switching computers are connected to several mainframes each of which perform a particular function. Attached, dual processors are used to provide increased power for what is fundamentally a single system. Over time, the evolution will be to small scale, multiprocessors for incremental performance and higher availability.

MINICOMPUTERS AND TIMESHARING FOR A GROUP

In the mid-60's both minicomputers and timesharing were developed at Digital around the PDP-8 and PDP-10 computers, respectively. Minicomputers were initially used as components of real time systems and for personal computing. The LINC minicomputer, developed at M.I.T.'s Lincoln Laboratory was the first personal computer, providing a personal filing system and the ability to write and run programs completely on line.

Timesharing started out as a centralized mainframe facility for a large group. Access was via individual Teletypes which were eventually replaced by cathode ray tube terminals, or "glass Teletypes". By the mid-70's low cost PDP-11 timeshared computers began to be used by seperate groups and departments to provide "personal computing". In the early 80's, low cost disks and large memories permitted two evolved computer structures: the 32-bit supermini, and the microprocessor based "team computer". The supermini had all the power of its mainframe ancestors, especially the critical 32 bits to access memory. The "team computer" based on modern, powerful microprocessors is simply much lower priced, (eg. $15,000) providing "personal computing" at a price below personal computers.

MICROPROCESSORS, PERSONAL COMPUTERS AND POWERFUL WORKSTATIONS

The fourth generation appeared in 1972 with the microprocessor. With the second 8-bit generation microprocessor, floppy disks and 16 Kilobit semiconductor memories, circa 1976, Personal Computers were practical and began to be manufactured by Apple, Commodore, Radio Shack, etc. With 16-bit microprocesor and 64 Kilobit rams, the second generation of PCs appeared in the early '80's.

In 1979, Carnegie-Mellon University wrote a proposal for personal computer research, stating:

> "The era of time-sharing is ending. Time-sharing evolved as a way to provide users with the power of a large interactive computer system at a time when such systems were too expensive to dedicate to a single individual...Recent advances in hardware open up new possibilites...high resolution color graphics, 1 mip, 16 Kword microprogrammed memory, 1 Mbyte primary memory, 100 Mbyte secondary memory, special transducers,...We would expect that by the mid-1980's such systems could be priced around $10,000."

Today's powerful Workstation such as the Apollo or SUN Workstation connected with shared facilities on a Local Area Network characterize this type of machine.

Numerous information processing products are possible using the modern, high performance microprocessor. These include:

- terminals and smart terminals
- personal computers and special word processors
- high performance workstations
- PABXes for voice and data
- Smart telephones and telephone-based terminals
- Conventional, shared supermicros
- High availability supermicros using redundancy to form seperate computers or seperate processors within a single computer

PERSONAL COMPUTERS CLUSTERS AS AN ALTERNATIVE TO SHARED COMPUTERS

In the early 70's Xerox Research Park researchers developed and provided itself with a personal computing environment consisting of a powerful personal computers all linked together via the first Ethernet cable (3 Mbits), and created the notion of the Local Area Network. Their network had various specialized function servers, including a shared central computer that was compatible with the DECsystem 10, for archival memory and large scale computation.

Figure Evolution shows the hardware and software of a multiprogrammed computer used for timesharing, and the corresponding structure of a Personal Computer Cluster consisting of functional services and interconnected by a common interconnect which provides basically the same capability. The timeshared system has a central memory containing various jobs connected to terminals and operating system which attends to the users and handles the particular functions (eg. real time, files, printing, communication). Personal Computers are connected to timesharing systems as terminals. By comparing the shared system with the systems formed from functionally independent modules, one would expect two design approaches:

1. decomposing systems to provide shared LAN services; and
2. aggregating Personal Computer to Form PC Networks and Clusters.

Decomposing Systems to Provide Shared LAN Services

As shared computers become more complex and more centralized, it's desireable to decompose the functions for execution on smaller computers that can be distributed to be nearer the use. Thus, the decomposition of a shared system into various boxes, each of which perform a unique function permits the evolution of the parts independent of the whole, the physical distribution of a function and the ability of several computers to share a function. While we have described the evolution of LANs as a decomposition of a single system, LANs are generally an aggregate of heterogenous systems which access a shared service of some kind as described below.

LANs differ from Wide Area Networks (WANs) in that they assume a low latency, high bandwidth interconnect. This permits file access as well as file transfer applications. With file access, it is possible to remotely locate part or all of a system's mass storage to a file serving computer. File access requires bandwidth and latency which are roughly equal to that of a disk (i.e. 10 Mhz rates) file transfer can be done at substantially slower rates (56Khz to 1 Mhz).

Using the reasoning which allowed the formation of the file server, we continue the decomposition of a large central system into servers or stations and then combine these servers into a LAN. The major servers are:

1. Person Server (personal computer or workstation) - local computation and human interface, possibly private storage of files
2. File Server - mass storage
3. Compute Server - batch computation or existance of particular programs
4. Print Server - printing
5. Communication Server - terminal, telephone and PABX, Wide Area Network access including international standards, other companies (eg. SNA)
6. Name/Authentication/Directory Server - naming the networks resources and controlling access to them.

A LAN formed as a complete decomposition of a single system and containing no other incompatible servers would be defined as a homogeneous cluster of Personal Computers or Workstations.

Aggregating Personal Computers to Form PC Networks and Clusters

As personal computers require more facilities (e.g. printing, communication and files), and the number and type of PCs grow, the need to directly communicate for sending messages and sharing files. Furthermore, as a collection of computers in one place forms, economy is gained by sharing common facilities such as printers, phone lines, and disks. Applenet and Corvus Omninet are relatively short and low data rate Local Area Networks used to permit the construction of what might best be called a network of Personal Computers because of the heterogenity of type. The 3 Com system for interconnecting IBM PCs is more characteristic of the homogeneous network, or cluster.

For a PC Cluster, one would expect to have a single File Server which can supply records at random to any of its constituency. Table (of what timesharing, PC's and PC clusters provide) shows what timesharing, PC's and PC Clusters provide.

DISTRIBUTED PROCESSING USING CAMPUS AND WIDE AREA NETWORKS

The proliferation of timeshared computers required the development of networking in order for various systems to communicate with one another and to mainframes. Thus, dispersed computing became distributed computing. Store and forward wide area networks evolved from the ARPA-net, which was used to interconnect timeshared mainframe computers (mostly PDP-10's).

Campus Area Networks

When a collection of Local Area Networks are connected together in a single area which extends beyond a typical LAN, we call this a Campus. Universities clearly typify the campus as does a collection of buildings. Gateways are used to interconnect LANs of different type (eg. Omninet, Ethernet, 802 Rings, Applenet, Arcnet, PCnet), whereas bridges or repeaters are used to interconnect networks of the same type to form one larger network.

Wide Area Networks

WANs are characterized by low bandwidth, high latency, and autonomous operation of the nodes. The applications typically include: mail, file transfer, database query, and low interaction remote terminal access. Wide Area Networks can be constructed in several ways: direct dial up using conventional circuit switching using voice grade circuits, an intermediate store and forward network such as Telenet, or a hybrid approach where various worker computers do store and forward switching.

THE E

Although the specific design of the E began in December 1978 with the approval of the Board of Directors, it's origins include:

- the original VAX-11 goals for a 1000:1 range of computers,
- evolution of distributed processing minicomputer networks, in Wide Areas, "Campuses", and Local Areas,
- the appearance of powerful Personal Computers and Local Area Networks, permitting the aggregation of tightly coupled "PC networs and clusters" that provide some of the benefits of timeshared minicompters and mainframes,
- the ability to aggregate minicomputers and mainframes into multiprocessors and multicomputer clusters that appear to be a "single" system in order to provide higher reliability, higher performance and incremental performance.

The December '78 statement of the Distributed Computing Environment, Fig. DCE 12/78, and subsequent evolution [shown in brackets] was:

> "Provide a set of homogeneous distributed computing system products based on VAX-11 so a user can interface, store information and compute without re-programming or extra work from the following computers system sizes and styles:
>
> - via [a cluster of] large, central (mainframe) computers or network;
> - at a local, shared departmental/group/team (mini) computer, [and evolving to a minicomputer with shared network servers];
> - as a single user personal (micro) computer within a terminal [and evolving to PC Clusters];
> - with interfacing to other manufacturer and industry standard information processing systems; and
> - all interconnected via the local area Network Interconnect, NI (i.e. Ethernet) in a single area, and the ability of interconnecting the Local Area Networks (LANs) to form Campus Area and Wide Area Networks."

Fig. DCE 12/7 shows the origin of the "E" shape that characterizes the present Environment of Fig. E. The three horizontal segments of the E provide the different computing classes which roughly correspond to different priced computers; the functions are described in the Table of Computing Styles. In order to implement the environment, many requirements were initially posited, and several developments evolved from necessity:

- a range of VAX-11 and 11 compatible computers to meet the requirements of the various computing styles based on different classes of computers;
- interconnection schemes and the corresponding protocols for building multiprocessors, tightly coupled centralized VAX Clusters, LAN-based PC Clusters, LANs, Campus Area Networks and Wide Area Networks;

GOALS OF THE ENVIRONMENT

RANGE GOALS

The important goals and constraints of the Environment are contained in the original statement about what the Environment, which is simply "to provide a very wide range of interconnectable VAX-11 computers". The original goal of VAX was to be able to implement the range [for what appears to be a single system] of a factor of 1000 price range... with no time limit given. Since a given implementation tends to provide at a maximum, a range of 2-4 in price and 10 in performance if performance is measured as the product of processor speed times memory size, then many models and ways of interconnection were required.

At the time the 780 was introduced, the total range of products for both the VAX-11 and 11 family was almost 500, beginning with $1,000, LSI-11 boards and going to a $500,000 VAX-11/780. If the LSI-11 is used as a Personal Computer, the price range is reduced to only a factor of 50! While the two ends of the system were "compatible" and could be interconnected via DECnet, they lacked the coherency necessary for a fully homogenous computing environment.

By introducing "VAX Clusters", the range can be extended by a factor of up to the number in the cluster. For VAX, we now have a price range of from about $50,000 for a 730 to about $7.5 million for a cluster of 12, 782 dual processors and a corresponding performance range of over 100. In the following section, we will show how the cluster provides what appears to the user as a single system.

Obviously both higher performance machines, and VAX on a chip microprocessors are necessary in order to attain the range and computing style goals.

STATIC AND DYNAMIC ASSIGNMENT OF PROGRAMS TO NODES

Ideally, a user can decide on how to compute on a completely variable basis at the following times:

. at sytem purchase or rent time ranging from outside facilities reached via gateways, to a central facility, to a shared department or team computer, to a users own personal computer

. at system use time, ranging from access via a terminal, or personal computer interconnected to the system LAN or a particular, shared computer. Here, work is bound statically to a particular set of system resources. Most likely, particular nodes would execute special programs on data located at the node.

. at task time on the basis of reliability. VAX clusters provide the complete dynamic

. at task use time on a completely dynamic basis, ranging from computing on his own local system to being able to collect any resources and move work dynamically while programs are in execution. With this ability, as a program goes though its various stages of development, it might be moved from small system to large system to take advantage of increased computational power at higher level nodes.

. at task time on a dynmic basis with the ability to acquire arbitrary resources to engage in parallel computation.

FIGURES

EVOLUTION FROM TIMESHARING TO PC CLUSTERS

DCE 12/78

E

TABLES
Table of Computing Styles
Table of What Timesharing, PC's and PC Clusters Provide

THE MICROCOMPUTER-BASED INFORMATION PROCESSING INDUSTRY

Gordon Bell

Encore Computer Corporation

LEARNING FROM MINICOMPUTERS (1969-1972)

With the advent of the third generation characterized by small and medium scale integrated circuits, a relatively small group could easily design and build a minicomputer. Gestation time was in the order of 2 years, including the time to design the computer and write an assembler, mini operating system and utility routines for the sophisticated users. A relatively large number of skills were required to design logic, core memories, and power supplies; to interface peripherals, and do packaging; and to do system software (eg. operating systems, compilers, assemblers) and all types of applications software (eg. message switching).

The early minicomputer was characterized by a 16 bit word length and four kiloword memory. Applications of the mini varied from factory control, laboratory collection and data analysis, and communications, to computing in the office and small business. The OEM relationship was established to design and market hardware and software and software-only applications in a two-tier fashion, increasing the market for what was basically a general purpose computer. In effect, many more markets were created than could be reached by a single organization with a limited view of applications.

OBSERVATIONS ABOUT THE ORGANIZATIONS THAT BUILT MINIS

From 1967 to 1972 about 100 minicomputers efforts were started by four different kinds of organizations. The following Table of Minicomputer Companies describes the situation. At least 50 new companies were formed by individuals who came from established companies or research laboratories. Of these, some merged with existing computer or other companies. Established small (e.g. Scientific Data Systems) and mainframe computer companies attempted to develop a line of minis and other electronic related companies looked at the opportunity to enter the computer business.

There were no significant minicomputer entries after this period, except the IBM Series 1 (1980) when IBM decided that distributed departmental computing, using multi-channel distribution (OEM/end user), was not a fad. Tandem started in 1974 to supply high availability and cluster expandable systems. Several companies started up to build special signal and image processing niche products.

The following conclusions can be drawn from looking at the companies (see Table of Minicomputer Companies) that built minicomputers:

0. At most, only 1/4 of the organizations who tried to build minicomputers were successful to any degree! While virtually all companies built working computers, they did not build organizations with any longevity for a variety of reasons (eg. failure in engineering, marketing, manufacturing, product depth

or breadth).

1. Only six, real winning companies, or 7%, survived long enough (10 years) to enter and defend themselves in the current microprocessor based sweepstakes.
2. Only 14 groups succeeded marginally, adding 16%.
3. Only 2 of 50 4%, startups lasted, 4%; although 10/50 (20%) continue in some fashion.
4. For startups, merging increases the chance of survival; 4 of 59 (7%) could be considered in the winning column.
5. But, the probability of a successful merger is only 33%.
6. Being part of a larger organization that is part of some other business is pretty likely to fail, note only 1/22 (4.5%) really made it. A startup within a large, existing company is about like a standalone startup.
7. Being in the computer business and trying to make minis wasn't a way to win. Companies selling in a different market or price band were unable to make the transition. Only DEC made the transition; but one might argue that DEC was already in the mini business and maintained its market when everyone else started making minis.
8. IBM eventually started making credible minis in the 80's with the Series 1. Alternatively, we might argue that the System 3, circa 1972 was the most successful business minicomputer and as such took most of the business market.

Companies that differentiated their products by attempting to carve out particular niches with specialized hardware and software were prone to failure. Vendors who made special hardware for an application such as communications or testing (real time control) ALWAYS failed to make successful minis, and often failed or fell behind in their main business. Specialized hardware limited the market instead of broadening it and specialized software might leverage sales, but it was typically inadequate when used with limited hardware for a single market.

In the mini generation the truism held: having a high performance, low cost, general purpose minicomputer capable of being applied broadly, insured the largest market. DEC excelled by having a variety of operating systems aimed at the real time single user (which served as the model of the CP/M operating system for personal computers), and by providing communications, real time control and timesharing. The real time system ultimately was extended for transaction processing. Minis became especially useful for business applications because they were designed for high throughput. Note that although business computers weren't useful for real time, minis designed for real time were very good for business and timesharing use.

Table of Minicomputer Companies

```
88 minis*-50 starts-----02 win:  DG, Prime
         |             |
         |             -08 ?:    Adage, Basic 4, CA, Datapoint, Four
Phase,
         |             |         GA, Macrodata, Microdata, Modcomp
         |             |
         |             -40 fail: see below
         |             -other:   lots, but unknown
         |
         -09 mergers----02 win:  Interdata/PE, SEL/Gould
         |             -01 ?:    Datacraft/Harris
         |             |
         |             -06 fail: ASI/EMR (Schlumberger), CCC/Honeywell,
         |                       DMI/Varian/Univac, PDS/EAI,
         |                       SDS/Xerox/Honeywell, Tempo/GTE
         |
         -07 continue---01 win:  DEC
         |             |
         |             -02 ?:    Bunker-Ramo, CDC, IBM
         |             |
         |             -04 fail: GE, Litton, Packard-Bell, Philco, Recomp
         |              Victor
         |             |
         |             -03 try?: Burroughs, NCR, RCA
         |
         -22 other trys-01 win:  HP
                       -03 ?:    Hughes, TI, Raytheon
                       -18 fail: see below
```

-other: probably lots

*All general and special purpose Minicomputers for real time, communications, business, etc. sold through OEM, end user and bundled for process control, testing, etc. but not including scores of military, AT&T, European and Japanese computers. At a later time Tandem formed and array processing systems were developed for niche markets.

startup failures: American Computer Tech., Atron, BIT, Cascade, Compiler Systems, Computer Devel. Corp., Computer Logic Systems, Computer Property, Comten (purchased by NCR), Datamate, Data Technology Corp., Datac, Decade, Digital Electronics, Digital Computer Corp. (ultimately merged with DG), Digital Scientific, Dresser, Electronic Engineering, Foto-Mem, GRI, Hetra, Information Tech. Inc., Infotronics, Linolex, Minicomp, Monitor Data, Multidata, Nanodata, Northeast Data, Nuclear Data, Omnicomp Computer, Omnus, Redcor, Scientific Control Corp., Standard Computer Corp., Spiras Systems, TEC, UniComp Inc., Unicom Inc., Viatron

other tries (e.g. new businesses, special niches, vertical integration) failures: AC Electronics, Bailey Meter, Beckman Instruments, Cincinnati Milling, Clary, Collins, EAI, Fabritek, Fairchild, Foxboro, GTE, Interstate Electronics, Lockheed, International Telephone and Telegraph, Motorola, Philco-Ford, Singer, Teradyne, Westinghouse.

WHY DID DG AND PRIME WIN AS STARTUPS?

In both cases, the initial products were unique and had a relatively long time to become established before the established leaders (e.g. DEC) reacted to the threat. The company was established by engineers who had built successful products, in contrast to many startups who had little or no experience in designing real products. DG had the first, simple-to-build, yet modern 16-bit minicomputer based on integrated circuits which enabled it to be priced below all the existing products even though it was relatively late to enter the market. Being later allowed using more modern parts and learning from experience. The simplicity allowed rapid understanding, production and distribution, especially to OEMs. The OEM form of distribution is particularly suited to startup companies since the volume use of a computer is delayed 1 to 2 years after the first shipment of a product.

In the case of PRIME, the design for a large, virtual memory had been done prior to the formation of the company in a laboratory. The product was unique, and the first of the "32-bit (address) mini's", enabling large system programs such as CAD to be run when it was introduced in the mid 70's. (DEC didn't provide this capability until 1978 with VAX.)

In both cases, the marketing was superb, followed by the building of a large organization to build and service in accordance with the demand.

WHY DID DEC CONTINUE TO DOMINATE MINIS?

After several false starts, DEC was able to compete with DG and other startups based on its momentum in three other product lines which were all minis. Thus, its fundamental business from the beginning (1957) was small computers even though it produced the first, large scale timesharing computer. It produced the first mini, the PDP-8 in 1965.

With the onslaught of the wave of minicomputer startups, including DG which was formed by former DEC engineers in 1968, DEC finally responding with a competitive 16-bit minicomputer, the PDP-11, in

1970. The 11 was comparatively complex and sold as a premium product, quickly regaining the market. The Unibus made interconnection with OEM products easy, and the extensive hardware facilitated the construction of complex software. By 1975 several different operating systems were available for the various market segments.

DEC was relatively early in converting the PDP-11 to a multi-chip set and entering the board market to compete with microprocessors to some degree. It lead the 16-bit micro market until recently when chip based micros became commodity parts enabling the trivial assembly of Personal Computers. It failed to license the PDP-11 chips or make them available for broad use, including Personal Computers. The result is simply that the PDP-11 is merely another interesting machine that failed to realize its full potential.

DEC introduced VAX-11, a 32-bit mini, about 6 years after Prime but at a time when physical memories were large enough to support virtual memories and provide optimum cost performance. Because of its much larger manufacturing and marketing organization, it quickly regained the market lost to the smaller manufacturers such as Prime.

WHY DID IBM CONTINUE TO WIN?

IBM continues to read computing use correctly even though it was late in realizing the minicomputer was not a niche market. It did have low cost computers for technical computing (the 1130), real time (the 1800) and business (the System 3 was introduced in 1971). During the time when the minicomputer was forming, IBM was preoccupied with the 360 introduction. It should also be noted that the antitrust suit against IBM was started in January 1969.

Recently, IBM waited until PC's were established before it came in and established the standard where it now has the largest market share only two years after entering the market. Thus, because of its size, IBM can dominate ANY (and perhaps ALL) market segments of information processing in just a few years.

The conservative way to look at computing is in terms of many alternative price and performance levels that can be supplied to users on a substitutable basis. The price is of interest on both a cost per user and entry cost basis. A low entry cost means more people can decide to buy a product whether they be small company presidents or department heads in a large company. The cost per user determines the attractiveness against other alternatives (substitutable forms of computation). By both measures, IBM missed the minicomputer market until it introduced the Series 1.

Fundamentally, IBM views all forms of computing (and possibly communication) as part of its market to be aggressively pursued!

WHY DID NO ESTABLISHED COMPANY, BESIDES HP, SUCCEED?

For most non-computer companies, computers were too much of a diversion from what they understood and could manage. In the third generation, many new skills were required that weren't present in an existing organization. Clearly the gestation time for developing computers was shorter than for conventional electronics. For conventional computer companies, the status quo prevailed: costs, limited engineering design capability, a customer base, and perceived high growth in the existing business all argued to not invest in smaller computers.

HP was used to shorter product cycles. Even HP purchased a small startup, DYMEC, to enter the minicomputer business, and thus could be considered as a merger even though it integrated the product right from the start. HP's fundamental business was to produce information from instrumentation equipment. Since all of their instruments were being interfaced to computers, they regarded computing as fundamental.

The moral: leaders in an industry usually remain leaders, unless too much change is required. Technology transition, that typifies the generations, acts to limit the number of organizations that attempt the change. Only a few successful new companies such as Apple, Commodore, and Radio Shack define and typify a new generation (such as personal computers) even though many companies attempt to become part of the new segment of the broad, information processing industry. Being the first, such as Osborne with their portable PC, is no assurance of longevity.

The Future

Since most mainframe and minicomputer companies regard selling to the installed base as their market, they will have a hard time reacting to the microcomputer.

Since microprocessors evolve at a more rapid rate (50% vs 20%) than discrete and gate array bipolar-based designs, the conflict is inevitable. Note that microprocessor performance is measured in terms of the VAX-11/780 super-minicomputer introduced in 1978. In 1984 the two are approximately equal in performance but the processor cost difference is almost a factor of 100! If a method can be found to utilize a large number of essentially zero cost microprocessors in some sort of parallel structure, then micros can compete with all forms of computers including mainframes.

MICROPROCESSORS, THE BASIS OF THE NEXT GENERATION

Like the minicomputer, hundreds of information processing products can be built using microprocessors and their associated semiconductor and electromechanical peripherals because they are so inherently universal (i.e. programmable). In fact, many times more products can be built than with the mini. Let's totally ignore the products such as cars, instruments, testers, etc. that embed a computer in a larger product and whose primary function is NOT information processing. When PABXes and telephones compute and store information, they too, must be considered as part of the computer industry. That is, there can be substitution among personal computers, games, terminals, typewriters, computing telephones etc. when they perform the same function as general purpose computers.

IS THE INDUSTRY HI TECH?

The rapidly evolving high density semiconductor and magnetic recording products are clearly high technology, representing significant investment, high risk and high entry cost for a company. However, the systems assembled from these components are clearly NOT high tech and the barriers for entering an end user OEM or system level business with a generic product are negligible, especially when compared with previous computer generations such as the minicomputer which demanded a comparatively large number of disciplines. A company can be formed by a part-time president, someone who can assemble the various circuit boards, a programmer to do a version of UNIX and one or two helpers.

The Workstation Case

There are approximately 100 groups of one to a dozen or so engineers who are building workstations for various engineering and business professionals. Design consists of "assembling" the following:

- boards with microprocessors, disk, CRT and communication controllers that use one of several standard busses eg. Multibus, Qbus, or VME/Versabus
- appropriate disks and CRTs
- standard or custom enclosures

. a licensed version of UNIX available from a myriad of suppliers

. generic software including word processing, spread sheet, etc.

Each new startup company believes their product and business plan will beat Apollo, one of the first entrants into the high performance workstation market. Apollo is valued at about 700 million dollars in the fall of 1983 on annualized sales of less than 100 million with less than 1000 employees. By contrast, Digital has a valuation of about 4 billion with sales of 4 billion and a workforce of over 70,000.

A typical workstation startup company compares themselves with Apollo on two data points: their startup date which is usually 1-2 years after Apollo (when systems were easier to build), and the current month's annualized shipments. In this fashion, within 2 years each of 100 companies will be valued at 1 Billion dollars, giving a valuation of Workstation companies of 10 to 100 Billion... at least one order of magnitude greater than any optimistic projection of the market. This valuation doesn't include the established companies, such as IBM, whose values are approximately equal to sales and who may believe workstations are mainstream products.

Thus today's assembly operations that permit this great overstatement about hi tech, is at the root of what is surely to be one of the largest corporate re-evaluations in the history of American business. The only barrier to entering the industry as a board, software or system supplier is having a Personal Computer capable of generating a business plan.

A PRODUCT segmentED INDUSTRY, STRATIFIED BY LEVEL OF INTEGRATION AND FUELED BY ENTREPRENEURIAL ENERGY

We see an industry composed of thousands of independent, entrepreneurial-oriented companies that is fundamentally both:

stratified by level of integration; and

segmented by product function (eg. microprocessor, memory, floppy, monitor, keyboard) within a level of integration.

In contrast to the first computer generation when a company built the whole system from circuits to tape drives through end user applications in a totally vertically integrated fashion, a stratified industry consists of a set of industries within an industry each building on successive layers in a hierarchical fashion. Today, each company only designs and builds a single product within each level. Only the semiconductor part is hi tech, and that business is determined by the established semicomputer suppliers: Intel, Motorola, National, and the Japanese suppliers.

Entrepreneurial Energy

Companies form in an entrepreneurial fashion and are able to participate in every level of integration on a product by product basis. The amount of energy released to build products through entrepreneurial self determinism is truly incredible. Improvements in productivity of several hundred over a single, large monolithic organization have been observed!

The levels, and corresponding commodity standards, are:

- hi tech semiconductor microprocessors (Z80 -> 8086 -> 68,000 or National 32032) and memory chips (16K -> 64K -> 256K bits/chip)
- boards based on a standard bus (S100 -> Multibus I -> Multibus II or Versabus/VME; IBM PC cards)
- electromechanical disks (8" floppy -> 5" floppy -> 5" winchester disk)
- hardware systems are particular to the structure, but usually use Multibus form factors
- operating systems (BASIC -> CP/M -> MS/DOS -> UNIX)
- languages (BASIC -> PASCAL -> C)
- generic applications (eg. word processing, spread sheets)
- vertical applications within various professional and intellectual domains (eg. accounting, structured design, project scheduling)

GENERIC COMPUTER PRODUCTS THAT CAN BE FORMED FROM MICROS

Given the basically commodity structure of the products within a strata, what are the generic products that can be formed? Unlike the mini, many more structures are possible over a wider price range, creating what amount to many markets that are differentiated directly by function and indirectly by price. Like the mini, the information processing products are quite general, and are fundamentally only differentiated by price classes AND the software they run. That is, a programmer's workbench, a $10,000 personal computer and a CAD workstation all use the same basic hardware! As during the era of the mini, a niche strategy by applications software companies is likely to be fatal because software publishing is a cottage industry and nearly any program that's used to lock in a hardware base can be rapidly eroded by a "functional look alike" when the product is demonstrated. A list of the computer classes formed by Micros is given in the following table.

In these clearly established machine structures it's highly likely that the winners and losers are already established when one considers the product, organization and marketing approach. There are still many, many micro-based products to be invented providing they aren't in the time-worn, computer classes.

TABLE OF MICROPROCESSOR-BASED COMPUTER PRODUCT CLASSES

Terminals	What will happen when all terminals built by the already shaken out terminal companies become Personal Computers?
COMPUTERS ON A DESK	
Home (and game) computers	note failures at Atari, Matel, TI
Portable PC's	expect more losses (e.g. Computing Devices, Osborne) as the market anticipates the IBM Portable PC. Finally, there's a standard--the IBM PC.
Word processing	mature; first shakeouts and mergers occurring. Will it be completely eroded by PC's, smart typewriters, smart telephones, Workstations, etc.?
PC's	a general purpose structure that will prevail!
Workstations	vastly oversubscribed with 100 68,000 based UNIX companies, finale will come with competition from evolving high performance IBM compatible PC's
DEPARTMENTAL AND GROUP-LEVEL COMPUTERS	
[illegible]	basically a mini replacement at lower cost
[illegible]	a supermini replacement... but what about the minicomputer suppliers?
SPECIAL PRODUCTS	
[illegible]	a niche, but already oversubscribed by Tandem, current and emerging companies

Special processors an infinity of types??

HIGH PERFORMANCE, MAINFRAME COMPUTING

Clustered micros essentially zero cost micros deliver the best performance/cost. We expect gains through our ability to utilize these clusters through parallelism which will in fact mark the next generation.

COMMUNICATION NETWORKING

PABXes for voice/data already oversubscribed

Local Area Networks can exist if the standards firm up

Smart telephones a niche that add-ons to PC's will most likely fill

OTHER PRODUCTS

Robots, voice i/o, etc. an infinity of products and industries are possible based on the microprocessor sdand specialized hardware and software

WINNING IN THIS COMPUTER GENERATION THROUGH APPLICATIONS

We have shown that large vendors such as DEC and IBM treat computing as a substitutable commodity in a complete marketplace. Computing can be traded off among the personal, shared departmental mini, and mainframe levels over a price range of 1,000 to 1,000,000. (There is some evidence that increased use of personal computers increases the need for mainframes by giving more users access to tightly connected databases , instead of decreasing the need for mainframe power.)

Computing done in any fashion should be treated as part of a single, available market!

APPLICATIONS NICHES FOR BASICALLY GENERAL PURPOSE HARDWARE

Many alternatives are possible for supplying a range of products from the purely general purpose base system, to the product that has been highly customized by hardware and software.

The critical mass (or economy of scale) is in the widescale sales, distribution, installation and service of hardware products. In addition to having very good products, winning requires BOTH the OEM and end user channels to get critical mass of product distribution and amortize costs.

An OEM approach usually requires a product range, not just a point product. An OEM customer often requires service, and always requires high level applications and field support assistance.

An end user approach requires both a wide product range and complete sales/service.

The applications software company (e.g. CAD) that has to invent its own hardware system is likely to become either obsolete with hardware, especially when reviewing what happened in the case of minis, or fall behind in its software development. Furthermore, the company is limited by growth in their own market because investment is required in BOTH vanity hardware and its specialty added value software. The basically hardware vendors will surpass the combination supplier, and the software only CAD companies are likely to provide better software as described in more detail.

The Base, General Purpose Computer System

The simplest form of distribution is the base system with various generic software such as languages, utilities, editors, communications interfaces and database programs. The system is provided by a manufacturer, sold and distributed via the manufacturer or a third party channel of some sort. Eventually the system is Installed and the user is Trained (I/T) and finally Serviced (S) as follows:

```
Base system------------------------------------------S/I/T/S->USER
```

General Purpose Systems Enhanced Using Vertical Application Programs

As users require more specialized applications for particular environments such as Computer Aided Design of electrical circuits. Various industries supply these programs. The manufacture of the base system in concert with an independent software industry takes the particular programs to the distribution network:

```
Base system (eg. IBM) -----Integrate (eg. IBM)--------S/I/T/S->USER

         CADswco-----------|

         WPSswco-----------|

         ...---------------'
```

Note that a manufacturer can acquire a variety of packages and transform what is a general purpose system into a variety of special purpose systems. The suppliers of the software are likely to be the "best", because they have only focussed on the particular, vertical application. The software suppliers have the largest market, because a program can be transformed to run on many different base systems.

OEMing Hardware By Vertical Application Program Companies

Since the perceived (and often the actual) price of software is low, a company that has a software product and wishes to enhance its sales volume does so by buying hardware and then reselling the hardware as a complete system. In effect, a company competes with the mainline manufacturer supplying a similar, but greatly expanded product. While the gross sales are up, the costs can easily outrun the sales since the company must support a hardware too. In addition, the software company doesn't usually market the range of products that a mainline hardware supplier has. Such a system is likely to be less profitable than a pure, software supplier. Furthermore, the supplier is cut off from a large number of channels of distribution made possible when a basic software package is made to operate on many different base systems. Note the case of Computer Vision (CV) who buys products on an OEM basis from Apple and IBM:

```
Base system (eg. IBM)------Integrate (eg. CV)---------S/I/T/S->USER

         CAD sw (eg. CV)---'
```

Base System Hardware Enhanced By Unique, Vertical Application Programs

While selling a complete system enables a company to have higher sales, the margins may suffer. A way to decrease the cost of goods is to backward integrate and manufacture the base system. This is clearly the most difficult, with rapidly changing products because a company now has three areas of management attention: its basic applications business, the development of a vanity base system and the maintenance of a field organization which has to support complex hardware and software. Observe companies such as Valid Logic or Daisy Systems:

```
Base System (eg. V'd)------Integrate------------------S/I/T/S->USER

         CAD sw (eg. V'd)--'
```

Traditional OEM: Unique Hardware, Base System and Application Programs

The traditional nature of OEM which DEC pioneered is still relevant. A company skilled in a particular area such as Computed Axial Tomography or testing builds a highly complex instrument. A computer may constitute up to 1/2 of the cost of the system. Products of this nature are NOT basic, general purpose computers and as such, the customer will not require other software beyond the control of the device. Such a device requires a specialized field organization to sell, install, train and service the system... something that can't be done by a conventional computer company.

```
Basic product--------------Integrate------------------S/I/T/S->USER

         Special SW--------|

Base System (eg. IBM)------'
```

Thus, the range of product approach appears to be strongly supported: basic general purpose computer systems support many applications that form a variety of systems using unique hardware and/or software.

<u>Summary</u>

Look at the case of minis. The system winners did so by having fundamentally good general purpose hardware and then distributing it:

- as a generic product for a variety of uses. In the case of minis, DEC opened up many markets with the PDP-11 as the range included boards for embedded computing, a range of systems for personal computers, business, communications, general purpose timesharing

- by facilitating the formation of a software industry to write both generic and vertical professional applications which the manufacturer integrated and distributed

- through traditional OEM arrangements by encouraging others to embed a computer in other systems as characterized by the early Technical OEM marketing of DEC

- through other channels such as distributors, distributors to sub-distributors, retail outlets, installers, etc.

WINNING IN THIS GENERATION THROUGH OTHER AVENUES

UNIX AND THE MARKET

Highly interactive computing with UNIX is no longer a niche, but rather something that a user should be able to specify. IBM has shown its flexibility in adopting industry standards rather than forcing its own in a defacto fashion. If customers want it, IBM will likely supply it. IBM appears to be near announcement of UNIX across the range from PC's to mainframes. In a similar fashion, every minicomputer and microcomputer supplier will supply this standard in a commodity like fashion. While the combined market is large, the fundamental market has NOT been expanded, but merely made more accessible by every manufacturer. The result will be a much greater fall out of the smallest manufacturers who have inadequate marketing and manufacturing organizations.

UNIX is almost a standard. There is not, however, a single version that is available in "bubble packs" that operates on a variety of systems. Each version requires the overhead of support, making the distrbution again a matter of critical mass. Furthermore, each version has to be supported independently and on a sub-critical mass basis because there are idiosyncrasies for every hardware and software version. Having UNIX as a standard can let everyone enter the market on a commodity basis: yes and no. It's easy to develop the product, but how does one compete with the large organization distributing products on a wide scale?

The bottom line: UNIX is the opiate that lets 100 companies form and assemble a product in a trivial fashion. The final result in this case will be far more brutal than in the case of minis where at least some technical skills were required and acted as another filter to limit entrants!

WHY THE OFFICE/WPS MARKET IS ON ITS WAY TO SHAKEOUT

The past has been marked by what is a special purpose, albeit large application. The history of computing goes to the general purpose solution. Look at the competitors:

- general purpose PC's (i.e. the IBM PC)... clearly the structure to watch as all the conventional WPS software becomes available and replaces simple editors
- typewriters with built in modems
- terminals connected to large systems for the casual users
- workstations for the professional
- computing telephones

There are over 100 vendors in what is a commodity-like product valuing themselves at 10 to 100 Billion for a limited market to engineers, scientists and business analysts. All have the organizational overhead to start, but none have the critical mass to succeed except those who are currently well established such as Apollo, Apple, Convergent Technology and SUN.

This is a MAINLINE product when viewed by all established companies. Then best bets here are: perhaps AT&T (via new Teletype computing terminals), DEC, HP, and IBM.

Finally, the 32-bit Personal Computers (circa 1984-85), led by IBM using 256 Kbit chips and the Intel XX86 evolution, will provide the power of the emerging 68,000 based UNIX workstations at a fraction of the cost.

WHY THE SUPERMICRO AND CLUSTERED SUPERMICRO WON'T STAND NEW SUPPLIERS

Basically this structure competes with old line mini AND mainframe makers both of which are beginning to be distributors for supermicros as in the Convergent Technology model. Neither group will let their base erode without resistance. Both groups are ultimately capable of backward integration of OEM'd hardware.

HIGH AVAILABILITY COMPUTING

Reliable computing ala Tandem should no longer be treated as a niche, but rather something a user should be able to tradeoff. Tandem's aging product line is due for upgrade. DEC has fielded a cluster system in the high end market, but VLSI will reduce the cost. A product called HYDRA from IBM has been rumoured. There are a dozen new companies focusing on this market area now with microprocessors and UNIX.

Because there is a somewhat different structure involved in building reliable computers, especially with respect to software, there is a possible niche market as evidenced by Tandem. As the overall reliability of computers increases, however, it's unlikely that anyone will pay even a 25% premium, let alone a 100% premium for reliability unless it drastically reduces operating cost.

There is still interest in making a self diagnosable, self repairing computer that NEVER fails, however. While this is possible for the CPU portion of a system, the software and peripherals don't permit this ultimate machine to be built for sometime.

The most important aspect of high availability computers is that they can be designed for incremental upgrade using both the multiprocessor and multicomputer structures. This incremental upgrade capability is why many of the computers are sold, independent of their availability. With much lower priced machines, a broader range, and the introduction of fully distributed computing in Local Area Network (LAN) clusters, the need for high availability computers for incremental expansion will decline.

In short, high availability appears to be a fully filled niche.

MAINFRAME (BUNCH) AND THE MINICOMPUTER COMPANIES

Several of BUNCH have been relegated to decline through a declining base as all customers standardize on IBM compatible hardware. The microprocessor based systems are a convenient product to market. Companies have signed agreements with various microprocessor suppliers such as Convergent Technology.

With the various microprocessor structures taking over the tasks that were done by the traditional minicomputers, these vendors find themselves in situation similiar to BUNCH. In some cases, eg. SEL and Prime, marketing/distribution agreements have been signed with Convergent Technology. Companies in decline because of poor product competitiveness will witness rapid decline as high performance, commodity oriented, 32-bit microprocessors provide the same function as the traditional TTL-based minicomputer at a fraction of the cost!

Even though the trend is clear, the installed base, proprietory standards and unwieldy organizations all mean that the existing companies will have difficulty moving to meet the challenge.

SUMMARY

Beware of treating any niche as sacred or large enough. Generality beats niches every time provided the cost is adequate. Virtually all of the microprocessor based structures supply what is basically a single information processing market. At most, these structures attack what is the traditional minicomputer market. In summary;

- there are no barriers to entering what is decidedly not a high tech industry,
- economy of scale is most important in distribution and service,
- economy of scale of manufacturing may hold for a single product and single company such as for the Personal Computer and IBM, but not in general
- time to market is far more important than economy of scale in engineering and manufacturing--which decidedly favors the Entrepreneurial Energy of startups who provide a single product,
- large vendors such as IBM and DEC believe it's important to supply computing on a full service basis--virtually no organization provides a full line of networked, compatible, multi-vendor products,
- old line minicomputer and mainframe supplier markets will not be easily supplanted by new supermicro suppliers because system pricing makes distributing one product, low priced, complex systems difficult,
- generic and unique (e.g. CAD) software applications which run on a few generic structures (PC's, Workstations and Supermicros) will fuel this generation, and
- truly unique structures (eg. home robots) are rarely revolutionary or protected by patents long enough to become

established before the large supplier enters the market and takes over (eg. IBM now dominates the PC market via its late entry).

The Encore business and product plan is based on an understanding of these lessons and experience in dealing with these issues. We believe that this understanding and experience provide Encore with a strong foundation on which to fuild a large and successful computer company.

Def I mark generations of computing devices by these four factors:

one - an identifiable new machine structure,
two - the physical technology,
three - the basic needs for computation, and
four - the actual uses of the machine.

Technology and need are constraints to satisfy. Often the use is quite different than the specified or perceived need. Generations are the major concurrent breaking points of all these factors.

Op rate Generations result from pipelined, asynchronous, and parallel processing on numerous technologies. The pipeline starts with discovery, goes on to prototype construction of the principle, construction of some sort of working system, manufacturing, evolution of manufacturing processes, enhancement, possibly hybridization, and finally, most likely replacement.

During the 400 year, 10 generation period from 1600 to 2000, technology has evolved roughly a factor of 10**12. Using the product of processing rate and the memory size to measure computing power, then the computer has evolved almost 20 orders of magnitude since stone-based manual, single register devices supplemented fingers and toes for counting and arithmetic.

Genera Technology and physical computer structures are used to mark generations. Emphasis on software engineering might produce a slightly different segmentation.

Technology is the way that groups provide themselves with the material objects of their civilization. This conventional definition separates the physical technology that we arrange to form a physical machine from the specific memory patterns, or software, that has come to operate or control the machine.

Tab I For each of the four pre-computer and eight computer generations, a high level need, the specific use, and appropriate machines are identified. Generations are named for the predominant technology, not according to the date of the first invention. 1600 to 1800 marks the earliest pre-computer generation, with manual technology; 1800-1890, mechanical; and 1890-1930, electric-motor driven and electro-mechanical.

bones In the early 1600's Napier developed the first pocket calculator, based on a table look-up method.

.Lesson One: Throughout history man has sought to develop hand-held, light, personal, general purpose computing aids. The idea of the personal computer is hardly frightening or revolutionary.

Liebniz Shickard, Pascal and Leibniz designed and built mechanical calculators in the early sixteen hundreds -- the manual generation. Their machines worked in principle but not fact because the mechnical technology for constructing them didn't exist.

Thomas Then two centuries later, in the 1850's, Thomas used Leibniz's principle of a stepped wheel to build a mechanical calculator that worked.

Tates In fact, the machine worked so well that Tates and other companies copied them, worked out some bugs, and went into profitable businesses.

.Lesson Two: Ideas a generation ahead of their practical application and use. Few inventors become millionaire manufacturers.

Abacus The abacus is simple calculator that started well before my categories of pre-computer generations. I've had trouble actually identifying its place of invention. It has been claimed to be invented in Egypt, the Roman Empire and China.

.Lesson Three: If it is a good idea, then everyone will take credit for it. The Chinese abacus could represent up to 15 in a digit with 5 + 2 beads similar to what we invented several times and call the bi-quinary system.

Soroban Ultimately the Japanese refined it, first using 5 + 1, and then 4 + 1 beads for lower cost and faster operation.

.Lesson Four : Any basically good idea can be evolved.

Sor/cal This 1979 Casio calculator/Soroban is ideal in several ways: low cost storage of a second number is provided; simple operations can be done traditionally and more rapidly on the soroban; users can be gradually trained on the new machine without losing any traditional computational capability; the market is larger; and a culture is preserved.

.Lesson Five: Compatibility is important for a transition machine.

Babbage The computer itself can be directly traced back to Charles Babbage who worked within the mechanical era trying to build machines to calculate tide and navigation tables for the navy. His early struggles provide considerable insight.

Dif eng Babbage's ideas were always racing ahead -- funds lagged, technology lagged, and even his patience wore out. His first machine, the difference engine, was barely half-finished when he left it in pursuit of building an analyytic engine.

Sch eng Scheutz then built a working difference engine for which he received some acclaim that slightly bothered Babbage.

.Lesson Six: Don't be concerned if someone else takes your idea and perserveres to make it work.

Card in Babbage himself freely used ideas of others. The Jacquard card-driven loom gave him inspiration for program storage sequencing machine control.

.Lesson Seven: Freely borrow ideas and technology from other mechanisms or disciplines.

Babcard Although Babbage appeared to be frustrated as to his lack of recognition, his work on the analytic engine provided society with a significant goal for over a century. The goal was well known for various periods.

Anal e .Lesson Eight: If you set ambitious goals and fail, don't necessarily expect to be recognized for this work, at least until after your death.

Mark I It provided Aiken and IBM with the template for the Harvard Mark I, or to IBM, the IBM Automatic Sequence Controlled Calculator.

Engine If the IBM engineers had known about it, the machine certainly could have been built instead of some of the accounting tabulators. Although people think they have built the analytic engine, the notations about the design are still being unraveled.

.Lesson Nine: Document design notations so that it and the clues for its replication are clear.

Babbage A substantial portion of Babbage's efforts were devoted to pushing back the limits of technology and generating funds from the government. While aiding the industrial revolution and the birth of computers, he did not live to see the analytic engine implemented.

.Lesson Ten: I_f_ building an operational machine is important, then it takes a steady supply of funds, workable technology, and the machine design. One or two out of three isn't enough.

Tab II   In the first generation of pre-computers, 1930-45, the groundwork was laid for computer evolution. From an historical worldview, the whole period could be thought of as dominated by war. Turing was involved in electronic cryptography that helped form the British computers, especially the NPL ACE, and training of people to build machines at other instituions, including Manchester. The four U.S. efforts that I'll describe were funded by the war effort: Aiken at Harvard with Charles Lake of IBM doing the engineering; Stibitz and Andrews at Bell Labs; Eckert-Mauchly at the University of Pennsylvania, later aided by von Neumann and J. Forrester at MIT. Many who built later computers were trained on these first machines.

.Lesson Eleven: Although early machine evolution was driven largely by individuals; larger machines need larger teams, and organizations to fund and build them.

Mark I   Even though Harvard's Mark I was built by engineers from IBM who had a very good relay background, it was a copy of Babbage's mechanical machine, with some electromechanical control. It had 23 digits, 72 numbers for primary memory, other storage, tape control with operations varying from 1/3 s +, 6 s *, 12 s /.

Mark I   The machine took about 8 years to develop and was running between 1943-45 and ran until 1959. It was the last machine that one could hear. There was some controversy as to whether it was worthwhile. Comrie stated, "It is disappointing to have to record that the only output of the machine ... consisted of tables of Bessel functions. ... If the machine is to justify its existence, it must be used to explore fields in which the numerical labour has so far been prohibitive".

Aiken Aiken estimated that the Mark I was equivalent to 100 desk calculators. He later predicted: "If all 3-4 machines currently under construction worked, it would saturate all conceivable need for computing."

.Lesson Twelve is clear: be careful about predicting the ultimate computer. With every computer, new applications emerge commensurate with exponential machine population and capability growth.

MK I Pr Aiken went on to build advanced versions of basically the same machine: the Mark II, a relay computer in 1947, and an electronic machine in '50. The ballistic benchmark took 12 hours on hand operated calculators; on Mark I, 2 hours; and on the 1950 machine, 15 minutes, the same as the differential analyzer. The grand ideal of Babbage had climaxed and by 1950, general purpose, stored program computers were starting to operate to replace the principles on which the analytic calculator was built.

BTL II In contrast to Mark I, the first Bell Labs computer, operated in 1939, made excellent use of the available relay technology in use then. Although similar to the 1920 Torres calculator, George Stibitz produced the prototype design independently.

Stibitz It was the first calculator that could do complex arithmetic and was operated via teletypes in an interactive fashion. It could also be operated remotely and in a shared fashion, albeit on a one-at-a-time basis. The concern was reliability and like subsequent designs, it contained exhaustive checking and diagnostics.

BTL 5 The final 1944 machine had these specs: + .3 s, * 1 s, for 7 digits. The machine ran 20 minutes on the ballistic benchmark.

Bush Computing in the thirties and forties often confused the notion of analog machines, in particular Bush's Differential Analyzer, which provided one source of people for Whirlwind. A DA at the University of Pennsylvania acted as a model of computing for building ENIAC.

E & M Eckert and Mauchly worked on mechanical control for improved function generators, but clearly understood limits of mechanisms and need for speed that caused the shift to vacuum tubes. Although Eckert claims to have invented the digital differential analyzer, they abandoned the principle because each order of magnitude required an order of magnitude increase in speed.

At the beginning of the war, John Mauchly proposed the construction of ENIAC. Ultimately, Herman Goldstine funded the project to compute firing tables but the machine did not run until 1946.

ENIAC ENIAC was roughly 500-1000 times faster than the relay machines, with a 200Khz clock. Times were + .2 ms, * 3 ms, / 30 ms. for 10 digits. It had 20 ac's, 3 function tables of 104 values. It held temporaries in relays with card i/o. Because ENIAC contained 18K vacuum tubes each with a predicted 500 hour life, reliability was an issue. But Mauchly was unconcerned. He reasoned that -- compared with the contemporary machines -- the machine would get a lot of calculating done in the few minutes it ran. Fortunately the 500 hour number was wrong, otherwise the exponentially increasing repair time for multiple tube failure would have bootstraped the machine to its death, that is, if it ever lived.

ENIAC The results are mixed as to its reliability: For example, all problems were run twice to insure accuracy. Franz Alt commented that its 40 plugboards and cables caused a significant reliablity problem and he estimated that the overall effective rate was 5% utilization. Goldstine used a different metric, observing that there were only 3 tube failures per week, or giving a tube failure rate of about 1 million hours, being achieved by derating the filament and plate. Thus, even if we reduce the factor of 500-1000 to 25-50 times the relay machines, it was still very worthwhile. The fact that such a large system ran was a tribute to significant engineering, mostly on the part of Prespert Eckert.

Met Tre Eckert also described the situation from which the stored program computer came about. Various priced memories were a given, and in fact, he stated that von Neumann coined the phrase, "memory hierarchy." They speculated that to generalize ENIAC to solve other kinds of problems it would be very difficult to determine how much memory should be available for various kinds of data, functions and programs. This immediately led to the notion of a common memory pool. It couldn't be implemented because there was no adequate primary memory. Mercury delay lines, magnetic drum and storage tubes were subsequently developed.

The effort surrounding ENIAC led to the stored program concept as embodied in the EDVAC draft report. EDVAC was, of course, to be the successor to ENIAC.

EDSAC Maurice Wilkes, visiting at the University of Pennsylvania from Cambrdige University, returned home to build EDSAC, the first stored program computer that continued to be in service.

Manch Meanwhile Williams, the inventor of the electrostatic storage tube, and Kilburn, had embarked on a much more ambitious computer, MADM at Manchester. Their effort eventually produced five innovative, influential machine designs. En route to their first machine, they produced the Mark I prototype to test the electrostatic memory -- in fact, I believe that the Mark I was the first, operational stored program computer -- but the machine was not put in service.

ENIAC .Lesson Thirteen: An operating system can be built by a small number of changes: Aiken contributed to adopting Babbage's program-controlled concept, using very conservative, even reactionary technology. Stibitz, concerned solely with reliability in terms of operations per month, built the first machine providing computation. The designers of ENIAC stressed speed in operations per second, and for various reasons really originated modern stored program computers.

As a corollary, machines that have changed too much have had problems. The Babbage machines suffered in this way. ILLIAC IV changed technology and organization of hardware and software, perhaps it also was without a clear need or problem. Perhaps Stretch and STAR, may have changed too many variables.

DEC6205 In my own case, I remember a parallel which this bit slice module from the PDP 6 may recall to some of you. We did have a clear need and use in mind, but we only made 20 PDP 6's. The Six can be best viewed as an advanced development effort for much of our own and others interactive computing and for the PDP 10. This lesson now might have been useful to me 16 years ago. We thought that there would be little risk to doubling the circuit speed; using a new mechanical packaging technique placing connectors on both the front and back of the modules in order to get the requisite number of pins; specifying a new architecture with a megabyte address when everyone else was at 256K; organizing a flexible structure that would permit building a large multi processor in an evolutionary fashion so that we could build subsequent machines on the same base; presenting a straight forward interface which as a side-effect probably started the whole idea of third party vendors at Stanford; and predicating the design on timesharing -- a concept that was just being breadboarded at BBN, MIT, Stanford and SDC.

PDP 6 This is exactly counter to the lesson of minimizing change, but the only mistake was not changing the packaging more to avoid the mechanical problems ultimately solved in the six's successor, the PDP-10. The ten's only change was the floating point format which was changed back to six format in the next model. On the 10, by changing the package to one permitting machine wirewrap we got a side benefit, as well, the ability to really produce computers, the key to the formation of the minicomputers.

.Lesson Fourteen: In making a revolutionary change make sure that every discipline of engineering is covered. In this case understanding the seemingly trivial aspects of sound mechanical connector mounting was critical.

.Lesson Fifteen: When making a revolutionary change look for all ways to reduce risk. Building enough prototypes is now a standard practice.

Mark I .Lesson Sixteen: The justifaction for a machine may be independent from its contributions. Retrospective looks are sometimes flattering and sometimes not.. but often exaggerated, at least the ones I've been involved with. Mark I provided a training ground for people who had concerns about programming. Some of the leaders in computing that came from it include Bob Ashenhurst, Gerrit Blauuw, Fred Brooks, Grace Hopper, and Jerry Salton. Stibitz' machines provided useful computations, first for engineers and then the military. ENIAC proved a point with electronics and speed that gave credibility to computing as an endeavor.

ENIAC .Lesson Seventeen: Greater than an order of magnitude change is needed in order to change future generation directions: ENIAC provided this, but it was close. Although evolutionary changes in relay technology may have resulted in the same performance as ENIAC in terms of operation per month, ENIAC's high speed in terms of operations per second permitted revolutionary use.

ENIAC The plugboard programming model of the ENIAC based on its differential analyzer predecessors constrained its use. The unreliability, in the way programs were plugged, provided one of the drives for the stored program concept.

.Lesson Eighteen: Even a poor technology or adversary design can provide a constraint or need to be useful in future computing.

DEUCE I recall writing an optimizing macro-assembler for DEUCE, a machine that Turing and Wilkinson had worked on at NPL, and have this module as a memento. Because it was so intricate to program, it established alot of strong goals within me that computer architecture had to satisfy.

MK & EN .Lesson Nineteen: When working in a field, keep abreast of your contemporaries. It should be noted that all these efforts would have moved faster had there been earlier communication between them. Mark I could have used relay technology and some of the design techniques developed for the Bell Labs machines; Bell Labs and ENIAC could have used control mechanisms of Mark I avoiding the large tube count through better organization.

WW I believe Whirlwind was the most significant computer of this period because it did build on all these efforts. It attended to technology and was designed to solve a significant real time interactive and control problem. Every other computer built in the forties was either oriented to arithmetic computation or data processing.

The original task of project Whirlwind was to build a Aircraft Stability Control Analyzer, requiring real time simulation of an aircraft. This need constrained the problem in three ways: reliability, accuracy, and speed. Over 100 simultaneous equations, with an accuracy of 0.1%, had to be solved at a 10-20 herz rate, forcing a parallel organization.

Bush The program was conceived as an extension of Bush's work at MIT on analog and differential analyzers -- with the project starting in the servo-mechanisms lab. As the work progressed, the transition from analog to digital was based on a suggestion by Perry Crawford. His 1942 thesis, also read by the ENIAC design team, was on digital computation.

WW dia The MIT team, led by Jay Forrester, investigated computing efforts going on elsewhere including those at Penn and Princeton, and chose two unusual designs for the period. The serial approach was ruled out in favor of going to a parallel computer. They also moved from the 40 bit word length to a 16 bit word, another difference. To a large extent the word length was chosen as a factor of the speed and accuracy needs, providing the precision (for what had been an analog problem), and also satisfying the size and cost constraints.

ww On new projects people are concerned about both failure and overwhelming success. In the case of MIT the question was: Should MIT be in the business of building computers? Forrester commented; "experimental equipment, merely for demonstration of principle and without inherent possibility of transformation to designs of value to others, does not meet the principle of systems engineering".

AC mod .Lesson Twenty: Build real things, not toys. MIT never got into the computer business. The modules were taken verbatim by Burroughs and by ERA for the 1101. The University of Illinois designed and built the IAS machine and other universities replicated their design. At Harvard, the subsequent Marks were less significant, possibly because IBM wasn't there to do engineering.

ww tubes Jay Forrester focussed on real time computing with high reliability. He knew that the estimated tube reliability of 500 hours had to be increased several orders of magnitude. An outside review also prodded atthe gradual failure mechanism of the tubes that led to marginal checking. By understanding the tube failure mechanism, the manufacturing process, and introducing marginal checking, reliability was raised to 5 million hours. In fact, the vacuum tube IBM AN/FSQ7 SAGE computers, originally called Whirlwind II are still in service.

.Lesson Twenty-one: Question the technology suppliers and pay attention to all details, using available resources and outside critics and consultants.

electro In the late forties, everyone building machines was searching for a reliable primary memory matched to the machine speed. The 2 Mhz clock and 50Kips

speed using MIT designed Williams Storage tubes costing $1K/1Kbit/month, was quite impressive but expensive. Jay Forrester searched for a better solution and eventually came to the core memory. In making the first cores they used wound magnetic tape (Deltamax) cores.
cer core   Then they discovered beautifully made, but little understood, ceramic cores. The only theory put forward by the producers at Philips, according to Forrester, was that they could not be used for storage.

.Lesson Twenty-two: Don't be undone by theory, especially if the art is much ahead of it.   Forrester commented: "This is an example of where the art was substantially ahead of the theory. Cores worked and could be made by trained ceramicists. Years later scientists understood how and why, but for many years production of ceramic cores was a materials art."

core   The University Research Corporation did not see fit to patent it because they considered that its commercial applicability would be neglible. Forrester got MIT to patent it, and to his chagrin (and probably many others) kept many patent lawyer in business for many years. He also commented, "The patent effort and litigation took about 1000 times the effort of the design. It took six years to convince industry to use the core and then six years to convince them they hadn't invented it." It is unclear whether Forrester or the other computer pioneers gained the personal wealth accumulated by those in industry -- who exploited -- or attended to the details of learning to manufacture computers like any other mass produced product.

.Lesson Twenty-three: The pioneers and the role of the universities was critical then and continues to be critical for generating new computer generations. Openness for ideas across disciplines and cultures are much more likely to occur in university environments where immortality is bought on the intellectual and not monetary marketplace.

ww   In 1948, Forrester was very much aware that the best, and only way, to learn computer programming was to program a computer, not learn about it theoretically. He made the following statement in one of his reports: "if a high speed computer capable of 1K to 20K ops were sitting here today, it would be nearly 2 years before the machine were in effective and efficient operation. One would be caught totally unprepared for feeding to this equipment problems...this represents one-half of the vicious circle in which an adequate national interest in computer training cannot be developed until the equipment is actually available." Whirlwind had one of the first operating systems and certainly the first one with real time processing.

.Lesson Twenty-four: Understanding and training about computers requires computers. (This was recently reinforced by the Feldman report last year arguing for equipment for experimental computer science. I would hope Forrester might even support it.)

Sage c Eventually Whirlwind was used in demonstrating the SAGE air defense system. It had a real time input from radars whose information were transmitted via phone lines, its real time operating system and the first CRT's and light pens. In addition, it was used for at least two purposes not concived in its design: the first computer speech research and Linvill's work on sampled data.

.Lesson Twenty-five: Build in generality, because the system may be used for something entirely different from what it was intended.

A significant transfer of technology occurred from Whirlwind to the machine contractor,IBM, via the establishment of Lincoln Laboratory and then MITRE and Systems Development Corporation, to work on the Air Defense project. Today, at IBM the corporate memory has so embodied the project that they may tell you, as they told me, that IBM invented the core memory.

Barta B In the early fifties, when Whirlwind was in full operation in the Barta Building in Cambrige, the engineers were anxious to move from the first generation computer in a building to the next generation where a computer could be built from transistors and kept in a room.

TX-0 Whirlwind with all its investment in operating programs was essentially discarded for the TX-0, the first transistor machine, designed to test transistor circuitry and large memories.

TX-0 It was 18 bits, and quite impressive--so much so that you can see the Japanese had already taken cognizance of it for speech research. Note I was, and still am, surrounded by the Japanese. (I'm leaning over the machine with my hand over my mouth -- holding in our secrets.)

PDP-1 The people who designed the circuits started DEC, first building logic modules using the basic circuits and then the PDP-1. Much of software investment in both Whirlwind and then the TX-0 was lost.

TX-0 .Lesson Twenty-six: When building a machine, don't be too hasty at not adopting a previous one, especially if there is a significant investment in software.

TX-0 In the case of Whirlwind and TX-0, the significant change needed was a longer word length because there weren't enough address bits to access the 65k memory that machine was design to test. Given the technological progress in memory cost reduction of about 30 percent per year, meaning that, the cost of a given size memory declines by a factor of two every six years. If a given user spends a constant amount for a system, and a certain fraction is memory cost, then an extra bit is required somewhere to access the memory each two years. Nearly, all machines have been designed with inadequate address space expansion to get to the next generation.

minis .Lesson Twenty-seven: A general purpose machine (including a language) should be designed for orderly extensibility, especially in address-size, or in the case of language machines, datatypes, otherwise the past machine will have to be emulated in succesive generations because of perceived software investments.

Given that I've opened the issue of building successor machines that are compatible with or build on the past, I feel duty bound to state a less that RCA ignored and the Japanese eventually learned:

.Lesson Twenty-eight: If you copy a machine, do it exactly--not just closely. The test has to be that the software, including all user data and files can't know the difference between the original and the copy. Furthermore, if there is a desire to attract and then entrap a given set of users to your machine (or language), then build it compatible with extensions that other machines don't have which your users will feel duty-bound to use.

.Lesson Twenty-nine: Getting the right standards at the right time is essential. If a defacto standard exists, such as the IBM channel and Unibus, let it be. If a standard is needed, then go all out to create it so that others can avoid the hassle of having to invent in an area that will genrally make work. Alternatively we can let anarchy reign until IBM makes an ad hoc decision, and then we can accept it in a de facto fashion. I hope the forthcoming standard based on Ethernet will permit us to build communicating systems.

TAB III Whereas early computing technology was marked by a change in the basic phenomena, now, it is a refinement of the semiconducting phenomenon. The end of the fourth computer generation is marked by the number of semiconductors on a single silicon chip. The fifth generation microprocessor, where a single computer is placed on a silicon substrate, has emerged. The sixth generation will be limited by the time to make and refine a design and to find the next collection of ideas that generates the next structure. The estimate is 7 years, which is also the time taken to get a factor of one-hundred times increase in the bit density on semiconductor memories.

With the fifth generation or perhaps near the end of it, we may see the beginning of the end of the computer as it becomes part of more of our goods. Soon, cams and levers in typewriters will disappear as we form all electronic typewriters and make the transition to all electronic transmission, storage and transduction of information...this later step is just a matter of time unless we find out that there really is an infinite supply of energy for transmitting us and our paper.

$ vs. g. Another positive feedback cycle exists for continuing to supply machines at a constant cost with increasing performance because the existing user-base metric is cost/performance or productivity. Given a substantial investment in operations costs, increasing performance at the same costs gives the highest overall increase in productivity.

Lesson Thirty: There is a natural economic-based mechanism that favors evolution of current priced machines to aid productivity.

The figure illustrates that a given technology permits 3 different paths for forming new computers:
1. constant cost and increasing performance while evolving use;
2. the new structure based on decreasing cost and constant performance, use is likely to be taken from the previous structure and simply widely applied; and
3. a newer, larger structure where new uses come from free resources. Technology permits the structure, based on increased component reliability, speed, and density. Price is constrained to about $10M and to the overall system reliability.

.Lesson Thirty-one: New uses come out of free and available resources not out of a computer system that has high throughput. Is this another reason why batch processing disappeared?

.Lesson Thirty-two: Machines evolving along a decreasing price line are based on the previous generation. As such, they are likely to make the same errors and go through the same evolution as their predecessors. This is especially true with such a short generation life.

TAB IV As I look forward, I am concerned whether government can provide any useful help since national support has not been rallied around any fundamental goal or need. There are two goals that could significantly force the evolution of computing: energy self-sufficiency and economic self-sufficiency through production. Regaining a number one position in overall science and technology might then be a fallout.

.Lesson Thirty-three: Change when the technology is obsolete: It's all in the timing. Looking back over the generations, the same mistake is made over and over again. Deviate when one can provide a significant gain, but don't necessarily throw out or ignore the old.

TABLE OF COMPUTING GENERATIONS, WITH NEED, USE AND STRUCTURES

GENERATION	HIGH LEVEL NEED	SPECIFIC USE	COMPUTER STRUCTURE
taxes, land	counting	abacus, counting tab.	
Manual 4.p.c. 1600	Trade & exploration	Arithmetic	Pascal, Napier's Tables, & Gunter's Scale
Mechanical 3 p.c. 1800	Industrial	Survey, navigation, loom control	Arithmometer, Difference engine, Planimeter, Slide rule, & Tables.
Electro-mechanical 2 p.c. 1890	Mass production & census	Census & modern accounting	Comptometer, Electric calculator, Hollerith & accounting machines
Electronic (thermonic) 1 p.c. 1930	Power, highway & communication grids	Engineering calculations & cryptography	Network analyzer, Mark I, Bell Labs calculators, ENIAC, Collosus.
Electronic (magnetic) 1 c. 1945	Defense	War-machine control via tables & real time	EDVAC, EDSAC, IAS, Whirlwind, LGP30, IBM 650, 701, 709, UNIVAC.

Transistors 2 c. 1958	Space & science	Air defense & traffic control; Engineering & science education	TX-0, IBM 7090, Atlas, Stretch
Integrated Circuits 3 c. 1966	Transport flow control & welfare	Process control & social accounting	PDP-8, B5000, PDP-6, IBM 360, 6600
LSI 4 c. 1972	Economic models & r.t. control	Interactive computing	Intel 4004, 8008, VAX-11, Cray 1
VLSI 5 c. 1980	Energy & productivity	Office & home computing	
ULSI 6 c. ~1985	Information & program overload	Knowledge-based systems	
Electro-optical 7 c. ~1990	Arts, leisure, food & energy crisis.	Travel substitute & environmental management.	
Particle 8 c.	Computer-assisted micro-environmental		

PAPER2/4
10/13/80

GB2.S3.22

GENERATING COMPUTER GENERATIONS

DEF I mark generations of computing devices by four factors:

one - an identifiable new machine structure

two - a new physical technology,

three - meeting new needs, and

four - a new level of use.

Generations are evolutionary, with family trees that can be followed. A revolutionary change -- such as the computer itself -- marks a discontinuity and the start of a new set of family trees.

A simple model of the process includes all four factors. The "perceived needs" or aspirations of a society generally are at some level above that of the existent technology. Technology is defined as the way that groups provide themselves with the material objects of their civilization. The light bulbs floating between the levels of technology and perceived need represent the ideas of inventors and knowledge from science. In each generation, a number of isolated ideas precede the actual project -- the identiable new machine structure. All levels then adjust to anew status quo.

The project pipeline starts with discover, goes on to prototype construction of the principle, construction of some sort of working system, manufacturing, evolution of manufacturing processes, enhancement, possibly hybridization, and finally, most likely, replacement. This pipeline may extend through several generations -- and replacement may not take place until several orders of magnitude improvement can be realized. For example, many second generation computers operated through the fourth generation. Replacement often takes several orders of magnitude improvement, especially in periods of rapid change.

During the 400 year, 10 generation period from 1600 to 2000, the technological change is roughly a factor of 10^{**}12. Using the product of processing rate and the memory size to measure computing power, then the computer has evolved almost 20 orders of magnitude since stone-based, manual, single register devices supplemented fingers and toes for counting and arithmetic.

The four factors marking generations are listed for each of the pre-computer and computer generations. The generations are named for the predominant technology of the time. The key invention for each generation, thus precedes that generation in time.

The abacus, a simple calculator, started well before my categories of pre-computer generations. It is such a good idea and simple device, that it has been claimed to be invented in Egypt, the Roman Empire, and China.

.Lesson One: If it is a good idea, then everyone will take credit for it. The original Chinese abacus represents up to 15 in a digit with a combination of 5 and 2 beads; It is similar to what computer engineers invented several times and call the bi-quinary system.

Ultimately the Japanese refined the abacus, first using 5 and 1, and then 4 and 1 beads for lower cost and faster operation.

.Lesson Two: Any basically good idea can be evolved.

This 1979 calculator/soroban is ideal in several ways: low cost storage of a second number is provided; simple operations can be done traditionally and more rapidly on the soroban; users can be gradually trained on the new machine without losing any traditional computational capability; the market is larger; and a culture is preserved.

.Lesson Three: Compatability is important for a transition machine.

The beginning of the pre-computer generations

i

need for a reliable aid to the calculation of multiplication so that, in his words, it would be "free of slippery errors." Napier's rods are inscribed with number series on all sides so that they can be manipulated to the needed set of digits. The set of bones fit nicely into a box and might be thought of as the first pocket calculator.

.Lesson Four: Over the years, a hand-held, personal, general purpose computing aid has been a perceived need. People always want more computational power in their pocket.

About 50 years later, Leibniz designed and built one four-function mechnical calculator. He had a vision of many machines, calling them his "living bank clerks". The one he built worked in

principle, but not in fact.

EL The mechanical technology for accurately milling the stepped wheel mechanism did not exist.

Two hundred years later, Thomas manufactured a calculator using the stepped drum principle of Leibniz.

ER Then when the machine was technically feasible to manufacture, the scientific establishment viewed it with skepticm. In 1849, SCIENTIFIC AMERICAN wrote, the Thomas machine "is said to be one of the most astonishing pieces of mechanism that has ever been invented, but to our view, its complexity shows its defectability."

Subsequent manufacturers streamlined the machine and continued to make and sell them into the twentieth century.

.Lesson Five: Ideas come at least a generation ahead of their practical application and use.

Babbage's ideas for computing formed between 1820 and 1850, a century ahead of their time. He was driven by the need to build machines to calculate tide and navigation tables for the navy.

His own ideas were racing far ahead of the technology. He left his first machine, a difference engine, barely half-finished and certainly not working, in order to build a better, more powerful machine -- an analytical engine.

Later Scheutz took Babbage's ideas and built a working difference engine. He received acclaim for it, to the annoyance of Babbage.

.Lesson Six: Don't be concerned if someone else takes your half-finished idea and perserveres to make it work. ... or even gets rich manufacturing it.

Babbage himself freely used ideas of others. The Jacquard card-driven loom gave him inspiration for program storage sequencing machine control.

.Lesson Seven: Freely borrow ideas and technology from other mechanisms and disciplines. The converse is also true: Don't keep industrial cliches around.

The card which was the savior of the 1890 census became so tied to some corporations approach to computing that they could see no alternative methods for input or output.

When the 80 column card was on the way out, true believers in card computing invented a 132 column card.

.Lesson Eight: Beware of the dinosaurs that are

created as a last gasp to extend a dying species. Somehow I continue to see larger and larger beasts created on a small bone structure (or architecture) just when a technology should be let go.

In Digital's case, we engineered this memory plane to get two bits from each core. While it's a beautiful show piece in the museum, the engineers called it everything but beautiful. Of course, it was never used. By the time it was completed semi-conductor memories were more practical.

Babbage never finished the analytic engine nor did he clearly and cleanly annotate it. Scholars are still unravelling and learning from Babbage's notations. At least Babbage was a prolific writer and did speak and write about his machine designs. The primary effect was not for people to steal his ideas -- but to applaud Babbage as an interesting thinker.

When Babbage was not trying to push back the limits of technology, he was trying to generate funds from the government, friends, and various agencies. He tried everyone's patience by not completing any projects or producing any results but promising the "fantastic" if only monies were available for the next machine.

.Lesson Nine: If building an operational machine is important, then it takes three ingredients:

a steady supply of funds,

useable technology, and

the machine design.

Two of the three is not enough. And having only one of the three -- only the machine design as Babbage had -- dooms a project to failure.

A more recent example is the ILLIAC IV, built at the beginning of the 3rd computer generation for scientific use, based on the yet undeveloped technology of semi-conductor memories. ILLIAC had a steady, seemingly, inexhaustible supply of funds that were thought as the universal solvent making up for under-developed technology and a trival but complex architecture.

Between 1833, when Babbage was working on the analytic engine and 1945, all component technology for the computer had been developed: teletype equipment for i/o; magnetic recording in the form of drums; diodes and triodes;

the Eccles Jordan flip-flop; and switching algebra. Babbage's ideas for the analytic engine were a century ahead of their realization in the Harvard Mark I.

Ironically, the Mark I built by Howard Aiken was given more notice at the time than the other pioneer computers. John Vincent Atanasoff at Iowa State and

Konrad Zuse in Germany who built highly original complex calculators were barely known of until the sixties, and only recently widely hailed as pioneers.

. Lesson Ten: If you have the foresight to be ahead of your generation with an idea, don't expect acclaim unless you plan to have a long, long life. A number of concurrent machines, each built incorporating a small number of changes laid the groundwork for the computer era.

Atanasoff built the first electronic digital calculator introducing the notion of direct, serial, binary computation using a regenerative memory.

Stibitz, attending mainly to obtaining a reliable number of operations per month, built the first machine providing computation.

The ENIAC group, stressing speed in operations per second was at the fulcrum of the revolution.
Later Williams and Kilburn added a new storage device; and Wilkes incorperated micro-programming.

.Lesson Eleven: The computer revolution, LIKE THE INDUSTRIAL REVOLUTION, cannot be marked by one machine, one person, or one idea.

Turing's Report of the Pilot ACE and Von Neumann's Edvac Report mobilized projects for computing on each side of the Atlantic, and led to some distinction between American or Von Neumann and the British or Turing type machines. The difference is distilled in this pair of quotations: In December 1946, on being shown an outline proposal for Wilkes' EDSAC, Turing commented, "The code which he (Wilkes) suggests is however very contrary to the line of development here and much more in the tradition of solving one's difficulties by means of equipment rather than thought."

Simon Lavington from Manchester notes, "Turing's design may have been economical of equipment but it certaily made the programmer work hard." And now, for a commercial. These quotes come from Simon Lavington's book, EARLY BRITISH COMPUTERS published by Digital Press.

The controversy over the roles of Eckert, Mauchly, and Von Neumann on the idea for the stored program computer concept continues. When Von Neumann joined the project group the EDVAC machine design had apparently been set but virtually nothing had been written down, although meetings were recorded on a wire type machines. Von Neumann started to take minutes and wrote these up in consolidated form as THE EDVAC report carried his name. Some people think that this report gave Von Neumann an exalted role as a computer pioneer. But the EDVAC Report, Turing's Report and Babbage's papers and books were more critical to the development of computers than built machines. They effected technology transfer.

.Lesson Twelve: If you want your ideas to be used and understood, then clearly document the design intent and the details and put your name on them.

Von Neumann and Turing, both brilliant theoreticians, provide an interesting contrast. After Von Neumann wrote the EDVAC Report, he felt that he also had to build a computer. It is clear that the architecture of the IAS machine was important but its implementation at Princeton

seemed to me totally irrelevant.

Turing, in contrast, didn't oppose the idea of building the Pilot ACE, but never fully associated himself with it. Harry Huskey, on Fulbright from the U.S. gave the impetus for implementation of the Pilot ACE while Turing went off on Sabbatical at King's College.

Lesson Thirteen: Four kinds of people can be identified in the process: IDEA GENERATORS, ENGINEERS, ENTERPRENEURS, AND EXPLAINERS.

On project Whirlwind, Jay Forrester who is credited with most of the ideas, led a strong engineering group that included Bob Everett and Ken Olsen. Ken, my boss, then went 25 miles outside Cambridge to develop his entrepreurial skills, and MIT's Electrical Engineering Department set to work explaining the machine to students.

S

Kilburn and David Edwards as engineers. Ferranti Corp. headquarters in the same town fed on these machine ideas, while the university program explaining them, also grew.

The idea generators can usually tell exactly when and where the components fell into place for them. George Stibitz who in working in his garage one November weekend; J. Forrester was standing on the

Mass Avenue steps of MIT talking to Perry Crawford,

and John Atanasoff made a long drive across Iowa to a roadhouse in Illinois on a wintry nite in January when the ideas came with the bourbon. In listening to these descriptions, the feeling of true peak experiences can be detected. A. Maslow, the psychologists believes these offer important lessons for the scientist:

"...the creative scientist lives by peak experiences. He lives for the moments of glory when a problem solves itself, when suddenly through a microscope he sees things in a very different way, the moments of revelation, of illumination, insight, understanding, ecstasy. These are vital for him. Scientists are very, very shy and embarrassed about this. They refuse to talk about this in public. ... if one can manage to convince a creative scientist that he is not going to be laughed at for these things, then he will blushingly admit the fact of having a high emotional experience from, for example, the moment in which the crucial correlation turns out right. They just don't talk about it, and as for the usual textbook on how you do science, it is total nonsense. My point here is that it is possible; that if we are conscious enough of what we are doing,... we may be able to use those experiences that produce ... revelations, illumination, bliss ... as a model." (Maslow, 1968)

Lesson Fourteen: Putting together a new machine or language has provided and will continue to provide peak experiences. In my own case, the mental picture of afternoon when I generalized the idea for the flip flop to handle multiple stables states is still very clear. I also have clear memories of inventing the unibus and the general registers used in the PDP-11.

These breakthroughs are distinct from futuristic dreaming or copyists. In the 40s, criticizing the ENIAC because it took a room and consumed half a city's electricity instead of being able to sit on each Caltech students desk, was only a pipedream. It had only become possible to construct Babbage's machine.

The design of Harvard's Mark I was fundamentally a copy of Babbage's mechanical engine with some electromechanical control, despite the fact that the IBM engineers had good backgrounds in relay technology. It took about 8 years to develop and ran from 1943 until 1959.

It was the last machine that you could really hear. The Mark I had 23 digits, 72 numbers for primary memory, other storage and tape for program control. In evaluating the Mark I, Comrie stated, "It is disappointing to have to record that the only output of the machine...consisted of tables of Bessel functions...If the machine is to justify its existence, it must be used to explore fields in which the numerical labor has so far been prohibitive."

Aiken estimated that the Mark I was equivalent to 100 desk calculators. He predicted, "If all 3 to 4 machines currently under construction worked, it would saturate all conceivable need for computing."

.Lesson Fourteen is clear: be careful about predicting the ultimate computer. With every computer, new applications emerge commensurate with exponential machine population and capability growth.

George Stibitz designed and built the Bell Lab Machines within the constraints of the telephone company, that is with a lot of telephone relays. The 1939 machine was the first calculator that

could do complex arithmetic and operated via

T

Mathematical Society at Dartmouth College. S.B. Williams designed an interface so that the number signals could be transmitted over a standard telegraph line. Attending mathmaticians were invited to transmit problems from a teletype at Dartmouth to the computer in New York. Answers returned over the same telegraph line and were printed out on a teletype.

Bell Labs went on to produce four advanced versions of the same machine, although the base technology was rapidly changing.

.Lesson Sixteen: Users and manufacturers are highly conservative. They want to preserve their economic and emotional investments as long as they can. Almost every company that listens to its user base, produces one too many of a given machine design. The key is to know which machine is one too many, and then not build it.

The converse also holds:

.Lesson Seventeen: A high growth organization can only be built on a new architecture. In buying up a bunch of old computer designs, Honeywell had no chance of high growth. They were self-limited by existing conservative users of these machines.

High growth has come from the new architecture of ,organizations like Apple, converting to the VAX base at Digital. But, believe me, when you're dealing with an existing organization with a set of happy and content users, suggesting change, and implementing it, is difficult but critical for success.

Harvard's Mark I and the early Bell Lab's machine, worked within existing technology to build large scale calculators. While they were an order of magnitude faster than the desk calculators, ENIAC w

the computing revolution came from funding for the war effort, particularly running ballistics analysis. These were being carried out by large crews of women on hand calculators and by differential analyzers.

It took a month to calculate a set of firing tables on a D.A. -- a big and expensive machine. Lt. Herman Goldstine, in charge of the substation of the Ballistic Research Laboratory at the University of Pennsylvania's Moore School of Electrical Engineering, wanted a device that would be much faster than anything that existed. He became an enthusiastic backer and fundraiser for the ENIAC. Vannevar Bush and George Stibitz who reveiwed the project as members of the National Defense Research Committee, were highly skepical of the feasibility of ENIAC fulfilling its promise. But Goldstine -- driven by his need -- and his knowledge as a mathematician that the project looked feasible persisted to play midwife.

The design of the ENIAC was influenced by the machine of John Vincent Atanasoff, a physicist at Iowa State. He described the difference between analog and impulse, i.e., digital, computation...and probably invented the phrases analog computer and impulse computer. He invented the notion of direct digital computation with electronics in the 30s, and by the time he stated carrying on discussion with Mauchly, considered that differential computation was a blind alley. He certainly reinforced Mauchly to take the digital route.

Teaching physics, Atanasoff was driven by the needs of his students who used electric Monroe calculators to solve simultaneous linear equations. He had reduced the solution of partial differential equations to an interactive set of linear equations.

Atanasoff worried the problem of building a

calculating machine through the early thirties, and in 1937 specified a serial computer with a serial regenerative memory using an electro-static drum.

This electrostatic drum is the only part of the machine that is left. With Clifford Berry doing the engineering, Atanasoff built a working machine that operated until 1942. During that time he gave a paper on it at the AAAS meeting. Mauchly visited Atanasoff for three days, saw his machine and looked at his circuit diagrams.

Prior to coming to Penn, John Mauchly taught physics at Ursinius College and experimented with gas tube and vacuum tube counting which would have naturally led to a digital differential analyzer approach. Presper Eckert claims to have invented the digital differential analyzer, but abandoned the idea because each order of magnitude required an order of magnitude increase in speed. The ENIAC reflected both Eckert and Mauchly's interest in differential analyzers. The plugboard programming was modelled on an analog machine. Eckert and Mauchly persisted on funding for the full-blown computer, spending considerable time on these efforts.

The first operational stored program computer was not the ENIAC, but the prototype the Manchester University Mark I. It was built by Tom Kilburn and Sir Frederic Williams, to test electrostatic memory and as the prototype for their large-scale machine, MADM. The efforts at Manchester produced five innovative, influential machine designs.

Similarly, Wilkinson, building the Pilot Ace, an almost portable machine, kept it simple and prototypical. Wilkinson stated "In deciding whether or not a feature should be included, the questions we asked ourselves was, could we do without it?"

.Lesson Eighteen: Prototype development provides a way to reduce risk. Many computer pioneers assumed that they had to build the computer itself. Unfortunately, I feel that the computer industry may have learned this lesson too well. Too much time is now spent on testing and prototypes.

By not building a prototyae, ENIAC did not run until the war was over. ENIAC with a 200K hz clock was roughly 500 to 1,000 times faster than relay machines. It had 20 accumulators and three function tables of 104 values. It held temporaries in relays with card i/o.

.Lesson Nineteen: Totally new ideas, often coming under highly skeptical criticism of the establishment, are needed in order to change the direction of future generations. ENIAC provided this. Although evolutionary changes in relay technology may have resulted in the same performance as ENIAC in terms of operation per month, ENIAC's high speed in terms of operations per second permitted revolutionary use.

Reliability was an issue on ENIAC. Both vacuum tubes and the plugboard programming gave cause for concern. ENIAC contained 18,000 vacuum tubes each with a predicted 500 hour life. Nevertheless, Mauchly was unconcerned. He reasoned that even if ENIAC only ran a few minutes it would accomplish more than they slow relay machines. Goldstine observed that due to derating the filament and plate only three tube failures occurred per week. The actual tube failure rate of about one million hours was achieved by very conservative engineering and not tackling the problem at its source. If the machine had been designed using the tubes at capacity, the exponentially increasing repair time for multiple tube failure of 18,000 tubes each with a 500 hour life would have bootstraped the machine to its death, that is, if it ever lived.

Franz Alt, commenting on the 40 plugboards and extensive cabling, estimated that the overall effective rate was five percent utilization. Because of the potentially compound problems of tube failure and plugboard connections all problems were run twice to insure accuracy. Taking into account the amount of time the machine ran, it was still 25 to 50 times faster than the relay machines. The fact that such a large system ran is a tribute to significant engineering, mostly on the part of Prespert Eckert.

Eckert in the historic tapes produced by the Science Museum, London, describes how the stored program computer came about. Various priced memories were available and Von Neumann coined the phrase "memory hierarchy." The ENIAC team speculated that it would be very difficult to determine how much memory should be available for various kinds of data, functions and programs. This led to the notion of a common memory pool. But it couldn't be implemented because primary memory was not adequate. Mercury delay lines, magnetic drums and storage tubes were subsequently developed. The mercury delay line holding regenerated shock waves is the exact dual of Atanasoff's electrostatic drum holding regenerated electronic charge.

The effort surrounding ENIAC led to the stored program concept as embodied in the EDVAC draft report written up by VonNeumann. The EDVAC was, of course, to be the successor to ENIAC. Eckert and Mauchly, like Babbage, were often thinking about the next machine before they realized the full potential and had all the bugs out of the one at hand. Eckert and Mauchly left the Moore School and the ENIAC project during the infancy of the machine; shipped the BINAC without really making it work and following through on the idea; and never lived with the UNIVAC systems long enough to make them great.

.Lesson Twenty: Don't only design the machines, document them, build them and then use and understand them.

Maurice Wilkes, who took the 1946 summer course on the ENIAC at the Moore School, returned to Cambridge University and built and programmed the EDSAC. He kept on with this successful venture which included the invention of microprogramming, but that's another story.

.Lesson Twenty-one: Don't be too hasty at throwing out a previous set of technology especially if there is significant investment in software.

In 1949, only one month after EDSAC was operational, Maurice Wilkes perceived the value of a series of computers sharing the same instruction set. He stated, "When a machine was finished, and a number of subroutines were in use, the order code could not be altered without causing a good deal of trouble. There would be almost as much capital sunk in the library of subroutines as the machine itself, and builders of new machines in the future might wish to make use of the same order code as an existing machine in order that the subroutines could be taken over without modification."

While ENIAC gave proof to the value of computing, the plugboard programming was so unwieldy and unreliable, that it provided a drive for the stored program concept.

.Lesson Twenty-two: Adversary designs and the poor use of technology can provide the definition of a need that is useful in determining evolutionary designs.

In my own experience, I've found that an adversary design has often created an extraordinarily strong driving factor for change. This module from DEUCE, a machine that was derived from Alan Turing's Pilot Ace of Britain's National Physics Laboratory, is a memento of a year I spent programming the machine. In 1958, when I started to work on the Deuce it was programmed in punching row binary and I was driven to write an assmbler that provided symbolic programming using three addresses. The program allocated instructions to positions in the delay line in an optimum fashion. This assembler may be the first one-level store machine using the 8 K word secondary memory and 320 primary memory as one. I also helped on a program George, which was perhaps, the first polish postfix and stack software machine implementation. It was later built as the English Electric KDF9. Since the first computer generation, simple stack structures

have been fascinating.

HP and Burroughs still can't get away from them. These personal experiences, fixed strong goals for architecture in my mind. I hope you don't have to relive them!

.Lesson Twenty-three: Nearly all mechanisms that appear in computer hardware structures start with software implementations. John Backus of IBM tells the following story on the introduction of floating point. He observed that many customers were running their 701s with a floating-point interpreter, slowing the machine to 50 multiplications were second. He tried to get the engineers to include floating point hardware, but they were more interested in speeding up the drum. He then created, "The most incredible design for building floating point inot the 704. It involved adding four or five new registers, which was unheard of in those days. At the next meeting of the engineering design committee, he remembers, "I stood up and spent an hour describing my insane design and people listened. At the next meeting Gene Amdahl got up and said, 'Backus, you're an absolute idiot; you can build in floating point without adding any registers at all to the computer, and it will cost almost nothing, and here's how to do it.' And that's how it happened."

I've been able to draw a number of these lessons from comparing the properities of pioneer computers. If there had been learning between the different efforts, then the computer revolution might have happened faster, . Mark I could have used relay technology and some of the design techniques developed for the Bell Labs machines; Bell Labs and ENIAC could have used some control mechanisms of MARK I avoiding the large tube counts.

.Lesson Twenty-four: When working in a new area, determine other pioneers and keep abreast of what they are doing.

I

getting overnight funding to build the IAS machine in Princeton. Part of the agreement was to send working drawings to Los Alamos Laboratory, the University of Illinois, Oak Ridge National Laboratory, Argonne National Laboratory, and the Rand Corporation. According to Julian Bigelow, who led the engineering team on the IAS, at the outset "we anticipated that any mistake we might make in sending out piecewise the fruits of our efforts would be exposed to possibly hostile or competitive criticism leaving us no place to hide, but in fact problems of this sort never arose, and communication with people at these laboratories was

entirely friendly and stimulating."

Today, the ARPAnet is a prime example of continued cross-communication - one of the few good things the defense department supports.

I think Whirlwind was the most significant first generation computer. The design team investigated other machines and a varieties of technology and they designed a machine to solve a significant real time, interactive, and control problem. Every other computer built in the forties was either oriented to arithmetic computation or data processing. The original task of project Whirlwind was to build an Aircraft Stability Control Analyzer, requiring real time simulation of an aircraft. This need constrained the problem in three ways: reliability, accuracy, and speed. Over 100 simultaneous equations, with an accuracy of .1 percent, had to be solved at a 10-20 herz rate, forcing a parallel organization.

Like ENIAC, the program was conceived as an extension of the work on analog and differental analyzers in MIT's servo-mechanisms lab. As the work progressed, the transition from analog to digital was based on a suggestion by Perry Crawford who worked for Vannevar Bush. Crawford's ideas, based on his 1942 thesis on digital computation, were critical to the decision of both ENIAC and Whirlwind to become digital computing projects.

The MIT team, led by Jay Forrester, investigated the efforts of ENIAC and EDVAC. They made two unusual design decisions for the period. The serial approach was ruled out in favor of going to a parallel computer. They also moved from the 40 bit word length convention to a 16, bit word, 32 feet long. To a large extent the word length was chosen to gain speed and accuracy within the size and cost constraints.

At both the University of Pennsylvania and at MIT, the administration and the design teams tangled over the "value" of building a computer. Eckert and Mauchly put a high value on the economic potential of computers and insisted on holding all the patents themselves. This ultimately led to their leaving the University. Forrester at MIT was not interested in building his own company. He was interested in sound engineering practices, stating, "Experimental equipment, merely for demonstration of principle and without inherent possibility of transformation to designs of value to others, does not meet the principle of systems engineering." MIT never got into the computer business -- but the Whirlwind did provide many businesses with proven designs and trained engineers.

.Lesson Twenty-five: Build real things, not toys. The Whirlwind modules were taken verbatim by Burroughs and by ERA for the 1101, and the machine itself was built by IBM to serve the SAGE system. ENIAC was the breadboard for the UNIVAC machines. These real, engineered efforts at universities were significant spurs to American industry, the economy, and computing. In contrast, the Harvard Marks and Atanasoff's machine were toys for training graduate students. Jay Forrester, concerned with highly reliable, real time computing, knew that the estimated tube reliability of 500 hours had to be increased by several orders of magnitude. An outside review prodded at the gradual failure mechanism of the tubes and led to marginal checking. By understanding the tube failure mechanism, the manufacturing process, and introducing marginal checking, reliability was raised to five million hours.

In fact, the Vacuum tube IBM AN/FSQ7 sage computers, that should be known as Whirlwind II except for the stuffiness of IBM, are still in service.

.Lesson Twenty-six: Question the technology suppliers, solicit outside reviews, and pay attention to all the details. As Mies VanderRohe--the most pristine architect/engineer said, "God is in the details."

As a corollary, don't try to change too many things at once because you just can't keep track of all the interacting variables.

I learned this the hard way in designing the PDP 6 about 1964. This bit slice module is my memento. We thought we could change everything, that there would be little risk in doubling the circuit speed using a new mechanical packaging technique placing connectors on both the front and back of the modules in order to get the requisite numbers of pins; specifying a new architecture with a megabyte adress when everyone else was at most 256K; organizing a flexible structure that would permit building a large multi-processor in an evolutionary fashion so that we could build subsequent machines on the same base; presenting a straight forward interface which as a side effect probably started the whole idea of third pary vendors at Stanford, and predicating the design on timesharing -- a concept that was just being breadboarded at BBN, MIT, Stanford, and SDC.

Only 20 PDP 6s were made and several are still in service. The team stayed together and gained experience for the PDP 10. I would have hated to say to customers at the time that we were selling them an advanced development effort for our own, and others, interactive computing. Thinking of the 6 as a breadboard, probably the main mistake was not changing the packaging more to allow wirewrapping. Then the mechanical problems of building the PDP 10 would have been to allow wirewrapping. Wirewrapping was the second generation a technology that allowed computers to be mass-produced and not handcrafted. This was the key to the formation of minicomputers and the explosion of the computer population.

But back to the first generation. In the late

forties, everyone building machines was searching for a reliable primary memory matched to the machine speed. The two Mhz clock and 50 K ips speed using MIT adapted Williams Storage tubes cost $1 per bit, or $16,000 per month. Impressive, but expensive. Searching for a better solution Jay Forrester started to investigate using magnetic cores. At first they used wound magnetic tape Deltamax cores.

Then beautifully made, but little understood, ceramic cores were found at Philips. According to Forrester, the manufactuers claimed that they could not be used for storage. Theoretically this was true, but it didn't stop Jay Forrester from trying ceramic cores and succeeding.

.Lesson Twenty-seven: Don't be undone by theory, especially if the art is much ahead of it. Forrester commented, "This is an example of where the art was substantially ahead of the theory. Cores worked and could be made by trained ceramicists. Years later scientists understood how and why, but for many years production of ceramic cores was a materials art."

At Princeton, about the same time, Von Neumann was increasingly concerned about being stuck without a fast parallel memory. The team had been relying on an RCA team to produce the selectron tube. After two years of work, with continually optimistic quarterly reports, not one had worked. Julian Bigelow reports, "No one in the IAS team was sufficiently expert in electron tube design and manufacture to be able to assist it, but in conference with Von Neumann I made an attempt to list the variables which would have to be kept under control to produce a 50% yield of successful Selectron tubes, covering a range of digital capacities from the original goal of 40% digits per tube, down through 2048, 1024, 512, etc. It appeared that...the goal of 4096 per Selectron was far too ambitious, and that acceptable production yields might be far sooner attained if the goal were reduced to 128 digits per tube." The IAS machine worked in 1950 using Williams tubes. The Selectron actually became available in capacities of 256 digits per tube and were used on the Johnniac, built in the early 50s at Rand.

.Lesson Twenty-eight: Don't depend on a critical component from a supplier who is developing an unknown technology. Examples of making this mistake are manifold. For example, the original line sharing system was predicted on the swapping drum that was always just around the corner.

At MIT, Forrester did not depend on outside

suppliers but did his own experimental development on the core memory. MIT's University Research Corporation did not see fit to patent the core because they considered its commercial applicability would be neglible. Forrester got MIT to patent it, and to his chagrin (and probably many others) kept many patent lawyers in business for years. He stated, "The Patent effort and litigation took about 1000 times the effort of the design. It took six years to convince industry to use the core and then six years to convince them they hadn't invented it." In this case, IBM lost the suit against Forrester and MIT, but they still will not readily admit it. I was recently told that IBM invented the co-incident current core memory; An Wang and Jan Rachjman of RCA also claim invention of the core. It was such a good idea at the time, everyone wanted the credit, just like the abacus. The idea did come from the university environment where openness across disciplines and cultures are much more likely to occur than in industry.

.Lesson Twenty-nine: The role of the universities continues to be critical for generating new computer generations. President Killian who was at MIT during the early electronics boom stated that it was the mix of three things -- teaching the bright young undergraduates who were free of preconceived ideas, the drudgery of the graduate students plodding on theses, and faculty consulting to industry that made the daisies -- the new ideas -- bloom.

Can you conceive Carnegie, Stanford, MIT, or even Pitt without computers? In 1948 that was the case. One of Forrester's reports gives some feeling for the frustration that he felt. It stated, "If a high speed computer capable of 1 K to 20 K operations per second were sitting here today, it would be nearly two years before the machine were in effective and efficient operation. One would be caught totally unprepared for feeding to this equipment problems ... this represents one-half of the vicious circle in which an adequate national interest in computer training cannot be developed until the equipment is actually available."

.Lesson Thirty: Understanding and training about a revolutionary new device requires the device. The problem is still here. The 1979 Feldman report argued for funding for equipment for experimental computer science. Carnegie, Stanford, and MIT, still need more computing power for training tomorrow's pioneers.

It doesn't always mean that good machines are needed, since we know that adversary designs can be useful. Aiken's calculating machines -- in use before Whirlwind -- are not remembered for solving Bessel functions, but for training a number of leaders in computing were trained: Bob Ashenhurst, Gerrit Blauuw, Fred Brooks, Grace Hopper, and Jerry Salton.

.Lesson Fifteen: The original justification (that we call need) for funding and developing a machine is often different than its use and contribution.

Whirlwind also had significant spinoffs. It was built to demonstrate the SAGE air defense system using real time input from radars whose information was transmitted via phone lines. It also had the first crt's and light pens. In addition, Whirlwind was used for at least two purposes not conceived in its design but that fell out of it: the first computer speech research and Linvill's work on sample data.

.Lesson Thirty-one: Build in generality, because the system may be used for something entirely different from what it was intended. Another example of this lesson comes from industry.

Texas Instruments Advanced Scientific Computer -- the ASC - a super large pipeline machine built in

1

company, and division involved in oil exploration. Several of the machines are used for this purpose but the generic aspects of large scale number crunching also made the machine desirable for others - hence the change of the name from Seismic to Scientific. The extra sales, allowing TI to produce 7 at about 10-25 million and not four, probably also helped the success of the venture. Hence another lesson.

.Lesson Thirty-two: A product created to solve a particular customers need will probably not be successful. If its only good for General Motors than it seems to be useless for almost anyone else. Have you ever heard of one-bit PDP 14 or Digital's GT40 or GT60 series? These were build for the unique user.

Whirlwind occupied this entire building in Cambridge. Yet, I like to think of it as the first mini-computer. It operated like one -- that is, it was personal -- even though the programmer had to walk into and not up to the console -- and it was interactive. Just a little bulky. In the early fifties, the engineers were anxious to try out transistor technology in order to significantly reduce the size from that of a building to a room. Whirlwind, with all its investment in operating programs, was dicarded for the TX-0, the first transistor machine, originally designed to test transistor circuitry and large memories.

Transistors were mounted in plastic tubes that

could be individually plugged in. The tops were color coded for easy checking and replacement. The whole machine was open and accessible -- like most lab computers. The transistors and boards w

Thus early lab computers and the tradition Von Neumann has been to treat them as human analogies. He applied the term organ to all parts of the computer and the word memory has stuck within this community.

.

with interactive program, the approach of IBM was to make "business machines". To IBM memory was, and still is, storage. The computer was boxed in and the worker gave it stacks of cards, often through a cashier like business window.

The 18 bit TX-0 was, and quite impressive -- so much so that you can see the Japanese had already taken cognizance of it for speech research. Note I was, and still feel, surrounded by the Japanese. (I'm leaning over the machine with my hand over my mouth -- holding in our secrets.)

TX-0 had an inadequate word length for accessing the 65K word memory that the machine was designed to test. Every three years the cost of a given size memory declines by a factor of two. Thus, each generation the machines have inadequate address space expansion to move to the next level.

.Lesson Thirty-four: Every three years another address bit is required throughout the system to address various memories.

.Lesson Thirty-five: A general purpose machine, including a language, should be designed for orderly extensibility, especially in address-size, or in the case of language machines, data types, otherwise the past machine will have to be emulated in successive generations because of perceived software investments.

Given that I've opened the issue of building successor machines that are compatible with or build on the past, I feel duty bound to state a lesson that RCA ignored and the Japanese eventually learned.

.Lesson Thirty-six: If you copy a machine, do it exactly -- not just closely. The test has to be that the software, including all user data and files can't know the difference between the original and the copy. Furthermore, if there is a desire to attract and then entrap a given set of user to your machine (or language), then build it with extensions that other machines don't have, but that your users will feel duty-bound to use.

.Lesson Thirty-four: Getting the right standards at the right time is essential. If a defacto standard exists, such as the IBM channel and Unibus, let it be. If a standard is needed, then go all out to create it so that others can avoid the hassle of having to invent in an area that will generally make work. Alternatively anarchy can reign until IBM makes an ad hoc decision, and then it can be accepted in a de facto fashion. I hope the forthcoming standard based on Ethernet will permit communication systems to be built.

Whereas early computing technology was marked by a change in the basic phenomena, now, it is a refinement of the semiconductor phenomenon. The end of the fourth computer generation is marked by the number of semiconductors on a single silicon chip. The fifth generation microprocessor with a single computer on a silicon substrate has emerged. The sixth generation will be limited by the time to make and refine a design and to find the next collection of ideas that generates the new structures. The estimate is seven years, which is also the time taken to get a factor of one-hundred times increase in the bit density on semiconductor memories.

With the fifth generation or perhaps near the end of it, we may see the beginning of the end of the computer. Production lines, cars, and typewriters will become more intelligent. Cams and levers will disappear in all electronic typewriters. The transition to all electronic transmission, storage and transduction of information is just a matter of time unless there is an infinite supply of energy for transmitting us and out paper.

Another positive feedback cycle exists for continuing to supply machines at a constant cost with increasing performance. The existing user-base metric is cost/performance or productivity. Given a substantial investment in costs of operations, increasing performance at the same cost gives the highest overall increase in productivity.

New computers and use evolve in three different ways providing three lessons:

One. Holding costs constant, improved and cheaper technology allows increasing performance and evolving use; hence

.Lesson Thirty-seven: The current economic mechanism favors evolution of machines in order to aid short-run productivity for existing users.

Two. Holding performance constant, a new structure can be developed based on decreasing costs. In this case use will simply become more widespread.

.Lesson Thirty-eight: These machines are based on new technology -- which will be old generation computing by the time they are on the market. As such, the machines are likely to make the same errors and go through the same evolution as their predecessors. They do evolve more rapidly than their predecessors because of the elastic nature of the market and because of the numerous design templates.

Three. Developing a new, larger structure with new uses emerges because of free resources. New technology permits change based on increased component reliability, speed, and density. Price always seems to be constrained to about $10 million and the achievement of overall system reliability.

.Lesson Thirty-nine: New uses come out of free and available resources not out of a computer system that has high throughput. This may explain why batch processing disappeared.

A final word of warning. All of us, including

N myself, want to design computers and languages as if we're the average man. This proves it to you, I'm absolutely average, so I can do it. Many of the pioneers, argued against high level language because coding and then punching one instruction at a time in row-binary was so trival that a machine should not be made to do it.

.Lesson Forty: Unless proven otherwise, in creating the design template, don't consider yourself the average user.

As I look forward, two goals could force the evolution of computing. These are energy self-sufficiency and economic self-sufficiency through production. Regaining a number one position in overall science and technology might then be a fallout.

There is still alot of computer pioneering to be done.

.Lesson Forty-one: No matter what a computer engineer tries to build it becomes a general purpose digital computer. And if you're not doing this then you're not a computer engineer.

ng: A 10 YEAR, DIRECTED RESEARCH PROGRAM AND NATIONAL FACILITIES AIMED AT PARALLEL PROCESSING

A Draft Outline*

&

Invitation For A Proposal(s)**

Gordon Bell, DEC

George Clark, Harris

Bruce Delagi, DEC

Sid Fernbach, Consultant to CDC

Bob Lillestrand, CDC

Red Phillips, Univac

13 August 1982

references to previous and ongoing work and bibliographic references have
. While we believe the general direction is correct, specific tactics
applications to focus on, will be subject to change with the final
We now solicit both conceptual and detailed critiques.

oposal must come from the program group dedicated to produce the results.
cit:

sites

individual researchers and a program director

applications and other research projects

TABLE OF CONTENTS

upport

Multiprocessors

Centers

ry

tical Global Questions

CDC's Advanced Flexible Processor?

dget

OVERVIEW

gan as an exercise by positing a computing environment we believe is years based on parallelism uncharacteristic of the single, von Neumann asking ourselves:

anything significant to understand and build this environment?

verwhelming:

ustrial research appears to be aimed at incrementally improving today's and processes; while

research is aimed at basic research and the mechanism of getting grants, g papers and Ph.D's.

this program is to develop the technology and build next generation ablishing several National Laboratories for computer science and arch within the U.S. military, academic and industrial community. This sential:

nse;

ve the declining computer and semicomputer part of the U.S. Information ng Industry which now constitutes and supports much of our economy and via exports; and

is for much of the 21st Century Industries.

chnology position in the computers and semicomputer industry is a As such, this necessitates these unique aspects of the program:

ation among national science, defense, university and industrial applied , often called technology, in a fashion not unlike the VHSIC program;

laboratories so that limited machine and people resources can be shared,
he VHSIC program;

fast network including access both for experimentation and to extend the
to other research sites;

tion of prototypes by industry for evaluation within the research
y;

gy transfer by industrial residents at the laboratories;

coupling of application (need), architecture, construction and use by
ion in order to rapidly engineer, build and test ideas. This speeds up
n of ideas to use by applying engineering resources earlier.

will be the hub of a goal directed research program aimed at new ly parallel computing structures. Parallel processing systems, ialized processors and hardware algorithms, multiprocessors, dataflow and high speed local area network based meshes will be built and utionary projections show a performance increase in processing of only a . 1) to 11 (Fig. 2) over the next 10 years. In contrast, the Japanese Research Project, is aimed at producing high speed and parallel factor of 100 to 1000 more computing power for conventional and computing systems by 1990 (Fig. 2).

al of the program is VLSIzation, the ability to transfer an algorithm, the computing environment, to VLSI limited only by the foundry time in grams are currently compiled. By it's nature, this structure adds lism to computing. The national facilities would also support the goal ould do a substantial part of the VLSI design. Research in the parallel ures we target will rely on accomplishment of these goals.

eneration is marked by concurrence of technology and needs causing a new ure and resulting in new use. We believe this driving need is for the mit, store, and process (understand) the same information as people, natural languages and images. Images are a major data type of this because of the links with people. The research need is driven both by hnology and by the potential of Knowledge Based Systems requiring much ce. These must be coupled with signal processing to assimilate voice and

d be organized in 3 phases, covering roughly a decade, in order to focus mely fashion. Generations have historically taken 7-10 years and consist specification and construction; followed by use and evaluation. The lation of the most powerful, high speed network of general purpose start the program in the use and evaluation phase. Results based on his facility would then be applied to produce new VLSIzed computing e end of this first phase. The second phase would apply these newer ing the basis for new designs in the final program phase.

MOTIVATION FOR THE PROGRAM

the combined Information Processing Industries is now declining relative there are many reasons for the decline, these are noteworthy and tivation for this program:

(and World) funding for basic and applied research is large. This m produces far more results than can be applied.

NO U.S. effort or policy aimed at systematically examining the basic results and refining them so they can be applied to products. The cost plied research on even a small fraction of the basic research is usually ter than the original work and is well beyond the scope of a single or a laboratory. Furthermore, most laboratories doing research can only eas to the paper stage because of the engineering nature of the final o build and test the idea. Thus, overfunding research relative to research means a "spilling" of knowledge that forms the basis of a ant industry.

panies have not worked collaboratively to develop these technologies of legal and cultural reasons.

ustry has been especially short sighted in its funding of this phase of . Now, many short term, mundane product opportunities (eg. another Z80 + ed personal computer) exist to attract resources resulting in further This is further fueled by the venture capital market and increased R&D its which in turn produce even more mundane products.

quate supply of people and equipment exist to carry out the work in and the research organizations.

ch program aimed at parallelism requires interaction and co-location with ommunity.

effectiveness of the Japanese collaborative research programs and emulate them. Both France and the U.K. have established programs aimed uter generation. Note the past and present programs in the Information

Information Processing- voice and vision

proved processing characteristics (eg. 64K and 256K rams resulted in a 2
d over U.S. industry)

puters- high speed technology

tronics- just established

Minicomputer for NTT- Fujitsu, NEC and Hitachi

neration Computer- Fujitsu, NEC, Hitachi, Mitsubishi, Matsushita, Oki,
ICOT Lab and 10 year program were established. The first phase builds
al Database and Prolog machines.

ea Network standards as part of the Fifth Generation.

eration research and technology program.

THE RESEARCH PROGRAM CONTENT

VES

ertaken with the expectation that the confluence of the disciplines of
ing applied to image processing, and knowledge engineering, and
g VLSI will prove fertile. It, and the resulting VLSIzation process,
derstanding specific algorithm and tasks and then VLSI'ing them, may well
cteristic of the next generation computing systems, which the Japanese
omputer Generation.* The establishment of a quasi-competitive, but
ram of research using common research facilities is intended to stimulate
standing of such systems and their potential application.

d at a fundamental understanding of parallelism and its application to a
s critical both to the growth of the computer industry in this country
enance of a preeminant US position in intelligence based military

USING THE FACILITIES: PHASE ONE

ocus will be on installing and applying parallel approaches to image
gic/circuit/process simulation problems, especially dataflow. We think
nderstand the range of dataflow from theory to practice across a wide
tions. In its simplest form, dataflow can be viewed as a formalized,
f pipelining that is conventionally used for graphics and image process.
ral form, dataflow looks appealing for logic simulation, signal routing,
array processing type tasks where a great deal of parallelism exists,
ploited due to the difficulty of expressing algorithms in conventional
s indeed possible that dataflow-specific machines <u>will not</u> exist, instead
es will enable programs to be written for large, multiprocessors. The
based on a high performance local area network to interconnect the
, including:

puters,

ntal machines (dataflow and conventional multiprocessors and
puters), and

AFP.*

rate with fixed microprograms to simulate several computer structures
ow computers. This will enable researchers to begin now and to
imits and use of dataflow architecture, for example. These efforts must
st of representative applications in order that the tradeoffs discovered
olve.

, believes that the current generation, number 5, is based on powerful
rs interconnected via local area networks. The Japanese are working on
tion, beginning in the late '80's.

to have real applications on which to "benchmark" various designs. The
ations cover some of the possible important military and industrial
ing electron microscopic image enhancement, automated assembly
et identification, digital system design and construction (eg. logic
ing and IC signature analysis). The actual applications should be made
proposal.

l results have focused on using a dataflow architecture to examine its
ork and facilities we envision are much more extensive and will be used
ays of computing.

ENT OF THE CENTRAL FACILITIES

hat the central research facilities will be enriched further over time by
ditional research tools, the fruits of the aspects of this program
ussed on realizing more powerful forms of processor interconnect and
ator based) intercommunication. It is expectd, further, that several
parallel solutions to specific application image processing problems will
in VLSI) and included in the central research environment.

RALLELISM: PHASE TWO

ase of the program here proposed, the principle results will include a
ng of the dimensions and metrics that describe the space of parallel
s, performance, programming expense, and reliability. The proposed
de a rich set of alternative realizations for parallel
ing from tightly coupled multiprocessors to conventional Local Area
not believe that the kind of interconnect for switching is a
itful area of study because it is really an economic issue that shifts
regulation, market demand, and supply. Thus, the goal is to provide
es for evaluation and use very rapidly, but not to research the
sibilities!

nd knowledge engineering efforts are expected to yield their most s in the last phase of the program. Significant milestones are ughout the research effort: discerning the computational (and data itives underlaying current rules-based expert systems languages, effective integration of image and symbolic information into a knowledge with the data management primitives noted above), realizing a VLSI f a highly parallel, post von Neumann computer structure for expert it out on (say) a SEM analysis problem, a fully automated VLSI design, n expert system for (semiconductor) process/crisis management (or threat econnaissance mission). These will, in turn, provide the understanding ond VLSI implementation of the expert system engine above.

ngredient of effective VLSI implementations supporting the research goals
we need the 1990's VLSI equivalent not merely of the Guttenberg Press but
machine and the automatic typesetter. The process would be completely
individual or small group. The most important element of this program
lopment of the capability for (fully) automated VLSI circuit design from
of parallel algorithms simulated on the parallel computing facilities
rst, this will likely be by means of both conventional supercomputers and
hine simulators running at the central facility.

sign capabilities will be made to stand the test of real use in VLSI
f (at least one) dataflow machine. The design of this machine will be
surement and analysis of simulated dataflow machines running applications
. These design capabilities will be also tested in VLSI realizations of
lysis dataflow algorithms and the mobile object identification and
s implemented previously. The culmination of efforts in image encoding
will be a special purpose VLSI processor chip that provides full motion
ng within the bounds of a 56 Kbps phone line, for example.

A FACILITY TO UNDERSTAND AND EXPLOIT PARALLELISM

lications usually result from having new, higher performance computers
n of problems that previously were computationally intractable.
eases in computing come from two sources: technology improvements and
elism. This program is aimed at understanding and exploiting parallelism
nce.

to parallelism in two ways.

dity processors allow the low cost construction of the most cost
stems. That is the Mips/chip of microprocessors far outstrips the
h performance ECL gate arrays.

zation is an inherently parallel process - standard algorithms are off

s to improve performance through highly parallel structures has been
pointing. We believe the major reason for this lack of progress is the
rsonal cost to build and evaluate parallel structures. This program
tic research and development on the following alternatives. In this
this fundamental hypothesis: in order for a new computer structure to
a user, and hence ultimately developed and exist, it must offer an order
rovent in performance over his current method of computation.

ESSING (AND VLSIZATION)

order of magnitude or more speed improvement has resulted from looking
times of particular work and then building hardware to carry out the
ation is a realization that this evolutionary process exists and is an
lize the process.

"off-loading" using special function hardware:

point hardware versus a software interpreter

, I/O Processors and I/O Computers versus interrupt and hardwired I/O

processors

gnal Processors

d (communications) and back end (disk, file and database) computers

g from a computation on a particular kind of data occurs.

n a requirement for a new computing structure. The function is then
specialized hardware that operates in parallel with the general purpose

ral purpose, very high speed system, the resulting, specialized
e totally simulated before they are committed to VLSI designs. In this
can interact with the structure in a quickly iteractive fashion instead
ch iteration for fabrication and system (re) integration.

computer class is formed, there are strong arguments to build
for performance reasons. Invariably, others build higher performance
the same time and deliver more power via the strictly sequential
processors were proposed by the early 60's, with Burroughs probably
irst one (B5000). By the early 70's Burrough's, CDC, DEC, GE, IBM and
uilt 2 - 4 processor multiprocessors. Unfortunately, these were either
etrical fashion, or at most they were used in an ordinary
environment. In no cases was parallel processing of a single task

nvestigated parallel processing of a single task with a 16 procesor
nd showed that for various tasks speed-ups were possible. By 1975 two 16

s were built by BTL and at CMU. The CMU system was predicted on the 11/40 a way to afford the construction, and speed-ups of up to 10 were ous algorithms.

lexible Processor is an ideal machine to investigate the use of and multicomputers since the interconnection among the computers is via local links (ultra LAN) and shared memory. It can be used in many ways,

puter multiprocessor;

cessor multiprocessor;

intrpreter for particular structures (eg. dataflow); or

ular, dedicated pipeline processing configuration (eg. image processing).

ries are building systems with up to several hundred microprocessors.

a multiprocessor, the successor to the S1, with 16 supercomputer class soon as the processor's available, it should be extended to the ase for evaluation, since the processors are both tightly coupled and nter processor communication mechanisms. This should be within the next

ring a 64 processor multiprocessor which requires investigation. We nd the installation of this machine in the facility in order to work on or problem.

tz, et al at NYU has proposed the Ultra-Computer, a multiprocessor with I microprocessors. Just as soon as we can operate a reasonable number of her, construction should begin on this very large multiprocessor.

that one can produce conventional parallel processors which should be up to a factor of four, for specially coded programs. A factor of 10 is

ere has to be a significant amount of research to make this automatically
es continue to indicate vast amounts of parallelism in algorithms that we
xploiting.

the optimistic (Fifth Generation) projection for computing power speed-up
cade could be accomplished simply and entirely by parallel processing
ssors and not by semiconductor and packaging technology if a significant
ied! Undoubtedly the dataflow language is an important part of this
ent, control and thereby exploit this form of parallelism.

been done formally with arrays of tightly coupled multicomputers where
uters (Pc-Mp pairs) operate independently and communicate with one
ng messages. By 1980, CM*, a multicomputer system based on the LSI-11
ith 5 clusters of 10 computers was constructed, and speedups of up to 30
r particular problems, including speech recognition. Because there is
tion among the computers, it is more difficult to predict the
 algorithm has to be carefully partitioned across computers rather than
emory.

FP, we believe that other multicomputers should be constructed and used,
se with several hundred computers. Here, we would support the
several, (say 6) different multicomputer alternatives.

CTURES

taflow computers have been proposed, only a half dozen computers have
 performance of dataflow computers is not understood, although the use of
and languages to express parallelism is promising. In particular,
 to be most useful in expressing signal processing operations. For
 is programmed using a dataflow-like representation for image processing
l computer modules can be assigned to various processing stages of say a
g task. The AFP also appears to be ideal to simulate static dataflow
d their application. It would be microprogrammed to be a general purpose
 using separate computer modules in a functional fashion: matching
, processing, and i/o.

ND CONVENTIONAL LOCAL AREA NETWORKS

rks, LANs, are systems which normally allow the physical distribution of
er components to cover a local geographical area (eg. a building, or
nctional servers roughly correspond to various parts of a shared system:
computing workstations/terminals), file servers, print servers, and
rvers. The communications is via message passing protocols. While the
ec LANs are relatively slow, they are well matched to today's, slow
nal computers and for intercomputer networking.

also posited that LANs can be used to provide high performance, parallel
too believe higher speed LANs are the backbone interconect architecture
structures. The higher speed, 100 Mbit/s LANs will be the basis for
functional computers in a hierarchy as shown in the facilities section

a-LAN as a major architectural component and standard for truly fast,
structures of this next generation. Note that the ring that
e AFP provides transmission at about 2 Gbits/sec for each computer node
e tightly connected computers. Thus, the AFP would be used for some
type of LAN-based architecture.

he hierarchy of three LANs is summarized:

AN	2 Gbits x p	AFP's processor intercommunication; as first basis for an ultra-LAN architecture
N	100 Mbits	Facility computer intercommunication and center to remote sites, forming a single cluster
	10 Mbits	Individual workstations to form centers

ING FOR KNOWLEDGE BASED SYSTEMS

widely agreed that Knowledge based Systems can exploit parallelism. For
ms, it is believed that many rules can be evaluated in parallel. The
aimed at first answering the question, and then simulating and
esulting structure. AFP might be used to simulate such a structure,
proach looks worthwhile.

THE RESEARCH PROGRAM FORM

RECTION AND RELATIONSHIP TO ONGOING RESEARCH

, together with a board of directors would contract the research in a
d fashion. While research of this type is not commonly done today in
, we believe it can and must be done effectively by a joint industry and
research laboratory efort. Industry can be effective at providing
ystems that have been traditionally absent from the research
effect, this is the major motivation for the proposal.

the research project is to provide a large infusion of computing systems
ing, more basic and unstructured work, including robotics.

d not be to change the nature of the existing unstructured research to be
nd goal directd, but rather to provide additional resources so that both
roject and unstructued work could co-exist and complement one another.

d be aimed at very similar research targets in order to get the benefit
petition". Similarly, several approaches would be examined within a
proach was successful in the mid-70's in speech research and should be
ction. However, the speech research resulted in few, commercialized
litary applications, because the research coupling between academic and
rch was poor. Unfortunately, the final transfer phase of research was
e the program ended.* It is this gap between basic research and
earch that the program is fundamentally addressing. It is interesting to
d an advanced development operating separately, but concurrently with the
he result is that NEC provides recognition products.

at a better model to follow is VHSIC. It is crucial that the
able to exploit the technology for commercial and military applications
nlike VHSIC, we believe that the work should be done at a few sites with
onnel.

CE

is research is a fairly close weave. The environments are, indeed, cipating that unexpected leverage and collaborations will yeild lts not included in the program plan. However, it is precisely the tructured program and the interrelation of its several work flows that to occur. The program office is responsible for the successive he fabric using resources as it can find them and coordinating efforts so build upon what came before.

nication with Allen Newell and Raj Reddy at CMM.

ce will set adequate standards so that ideas meet no unnecessary
en the workers and the worksites in this program. Early, stable
common rules, language, workstation, the network and the general
pport structure will be among the most important contributions of the
the goal is to use this commonality of interface to allow pyramiding of
eful not to pyramid risk.

applications to test ideas, and uses realizations of those ideas to
eneration applications. It even uses these applications themselves to
generation realizations fueling the next cycle. The central facilities
at application tools for realizing ideas, the realizations themselves,
ions for testing ideas all come together. This must all flow forward
leneck into a deadlocking interdependencies. The opportunity and
people to build on each others work as it becomes available is the key.
ncertainties inherent in this ambitious program of research, there must
ative paths so clever people can use their wits to find a critically
another's work or another's facility wherever it may turn up.

ce must have the ability to facilitate the construction of important
dboards so that systems can be rapidly built and evaluated. We envision
dustrial sponsors for this breadboarding.

ce is deliberately kept small to force most standards to be developed
with the groups doing the work. The program staffing for the parallel
ties is very light in the expectation that site personnel will be
host institution. The Budget Table, Appendix 3, provides a more detailed

ARIES

conceived in order to improve this flow of basic and applied research
research and eventually into products. The main beneficiaries are those
eas to <u>eventually </u>build products. Products will not come directly from

d, virtually everyone will benefit by the program:

technology will be drastically improved - thereby improving defense and
omy;

archers will be more effective and productive by having more meaningful

research will be published; and

ers will still migrate from the coupled programs, being attracted by
capital, and build higher technology products.

TECHNOLOGY

ve means of technology transfer is through the transfer of people.
will each have the right to place people in each project of the program.
hat assignments be for a three year interval and that the assigned person
onsoring organization prepared to produce the competitive products of the

perative working environment among the members of a project team,
perty rights for the work done as a team using the facilities of the host
be controlled by the policies of the host institution. However, each
will have the right to a non-exclusive license at reasonable terms.

the transfer will occur when the sites and industry collaborate on
sign that a site has specified.

chips produced as part of a research project would be licensed to the
rights" to chips and software produced as part of a research program are
at this time and vary among the institutions. This area would have to
tween the institution and the program.

for technology transfer include sponsor access to prototypes, published technical reports and invitations to program seminars.

held quarterly for program sponsors with invited speakers from vernment and industry.

kers the organizers of the seminars will have the freedom to draw on the pics encompassed by the program, including:

and image processing applications

gorithm research

ocessor architectural developments

software systems

ign process advancements

FACILITIES

REA NETWORKS

d be organized around at least central research computation centers
iety of production and experimental computing systems (nodes)
ia 100 Mb/s links and forming the central facility for a hierarchical set
ed, high performance, local area networks. The centers will be linked to
via the highest available links so that they could be used in a
n "as if local" computation centers.

contain supercomputers, AFP's and experimental computers.

proposing ARPA-net II. This must come into operation relatively soon,
terconnect the more remote research to the centers. High bandwidth, such
channels would be needed to avoid limiting the interaction between
e goal would be to provide only millisecond delays between processes
arated machines.

ITY

ts would be designing many VLSI chips, the facility would need a way to
he art VLSI chips from mask design. this could be acomplished by a
ttment of appropriate existing capacity to the needs of the program.

d start immediately and be coupled to existing computer science and
ring research facilities and programs. Facility selection is strictly on
intensity and quality of work in VLSI, image processing, parallel
. Either Lawrence Livermore or Berkeley Laboratories would be ideal
mputation center which would link to Stanford, SRI, and UC/Berkeley.
ncoln Laboratory could be the basis of an East Coast facility. Los
argest network of supercomputers and support computers including storage
tion. If a central site were Los Alamos, this would force the
installation of high speed links to other sites.

ERS

ry imcomplete list of application centers is included as an example of
e contracted by the program office to expertise centers throughout the

formance Interprocessor Or Communications Structure

. Illinois)

imulation And Parallel Algorithm Compilers

MIT, Berkeley)

n Automation For Parallel Computation

oln Laboratory, Berkeley)

----

nce/Map/Encode/Compress

Univ. Maryland, LASL, Lawrence)

tract/Target ID/Automated Inspection

RI, Univ. Texas)

Symbol Knowledge Representation/Expert System

MIT)

DELIVERABLES

assed is broken into three classes shown in the Deliverables Table.
s there are families of projects and finally the projects themselves.
about ten years broken into rough phase transitions at the end of 1985
ork in the first phase puts the research environment and work standards
elops the first generation tools and applications. The second phase
machine realizations that use the tools and runs the test bed
n this phase, the research facilities are enriched with the machines
ram efforts. These are in turn, the base of the second generation tools
. Finally, the third phase provides refinements and solves the hard
pended on the new understandings generated in the first two phases of the

DELIVERABLES TABLE

```
'82    '83    '84    '85    '86    '87    '88    '89    '90    '91    '92


 reconfigurable 100 MBy/s LAN       256cpu @ 100 MBy/s LAN
                256 cpu @ 10 MBy/s LAN
                                             1000 cpu @ 100MBy/s LAN


       simulator ok   hotspot analysis
                                     VLSI dataflow machine
           dataflow compiler


               parallel logic simulator running on Dataflow simulator
       VLSI parallel compiler              expert system for VLSI design



  pick 1 rules language       next generation rules language
common workstation (LISP?)
1,10,100 MBy/s LAN's    parallel rules VLSI     2nd implementation


        1 MBy/s NAN & gate
AFP  AI-VLSI support facilities            VLSI dataflow on 100 MBy/s
  I   II    III                      IC signature analysis array


                     256 cpu node on 10 MBy             4096 cpu node on 100
            SEM enhancement dataflow
```

Full motion video-conferencing in 56Kb/s

($500)

IC signature analysis dataflow

SEM scan analysis expert system

parallel rules language primatives

expert systems for
process/crisis mgt.

image/info=knowledge

MOTIVATING SUMMARY

or this approach is timeliness and effectiveness:

ARCH PROGRAM SHOULD START NOW

COUPLING OF INDUSTRIAL R&D AND APPLICATIONS WITH COMPUTER SCIENCE

UILD ON EXISTING RESEARCH PROGRAMS AND COMPUTERS

that we start now on the research program, as our computer science
n drifting these last few years as both industry and computer science
th gotten large, diffused, and independent of one another. Significant
rch outside of IBM, Bell Labs and Japanese companies is non-existent and
ling of basic and industrial research. For example, we believe there is
of Bell Labs work to the Japanese computer industry via NTT's, ECL, than
s and the U.S. Information Industry. Furthermore, both the academic and
rch communities are now poorly coupled to real applications. We believe
us of some of the existing research efforts into a goal directed system
ir productivity and enable the continuation of a vital Information
21st Century.

APPENDIX 1

SOME CRITICAL GLOBAL QUESTIONS (AND ANASWERS)

he establishment of national facilities the correct way to attach the problem?

ingle lab now has critical mass or focus in anyone area - currently all urces are difused.

lab(s) and programs operate together to do the work.

s, architects, and builders must couple.

act will this proposed program have on existing research facilities? ?

intent is to build on, and extend current facilities by additionla uces. We believe that this program is close enough to some of the ting.

ut the extra space required for these facilities?

on't know.

this effort help the basic problem of a shortage of qualified ers?

s hoped that a "program" will stimulate the demand to produce more arches over the long term.

t term, the focus should increase everyone's effectiveness.

ope to apply industrial researchers to the problem that are now difused
often operate as a sub-critical mass.

upposed to benefit from this proposal and in what specific ways?

tion on Program Beneficiaries)

a nationla crisis and exactly what is it?

tion on Motivation for the Program)

dence do you have to support the level of funding which is projected as
equate to achieve the goals?

really a draft outline for concrete proposals. From this we expect
sites to be established and operated in very targetted areas: such as
knowledge based systems, high performance parallel processing and
image processing.

actly is the overall objection of the program?

first sentence of this document)

APPENDIX 2

WHY USE CDC'S ADVANCED FLEXIBLE PROCESSOR?

nstrated high performance in digital image and signal -

. For example, a processor system can transform the every co-ordinate of picture in 1/30 second. Several systems are in operation today. It support software including simulators.

e design, build and then use. A machine as fast and general as AFP would 5 years to build. By using the current AFP as a general purpose e can gain at least 5 years on starting such a program from scratch. To ider the several data-flow projects that could use AFP today to simulate Since we need to evaluate these architectures by using them, we could enefits and drawbacks of these machines five years (or so) sooner by as a hardware simulation base.

ides a very fast, flexible, microprogrammed set of up to 16 computer rimenting in various parallel computing structures of various type. A oprogrammed processor provides the following capability:

0 Mops in 16 parallel, 16-bit arithmetic and logic units

grammed control

o 32 Megaword (256 Megabyte block oriented memory)

its/sec communication with neighbors in ring

processor and multicomputer structure are both provided since, the rs can be interconnected both to a common 32 Megabyte memory and to ors.

be used as a tool to study several different computer structures that we of the basis of the next generation.

o highly parallel, including having functional units with side effects,
ll not be imcroporgrammed to any great extent.

sion is that it would operate in several configurations, with fixed
behave as:

icroprogrammed pipelined, functional units within each processor. Four
n be initiated every 20 nanoseconds, although an average of seven units
in parallel for most problems. Because of the difficulty of programming
hly parallel structure, the most important benefit, or side-effect will
standing in how to do it effectively. Because the microprogramming so
pipelined, we believe a better understanding of dataflow techniques for
ng algorithms will result from the use. Nearly all high performance
are pipelined; hence, we believe AFP is a good vehicle to get a better
nding of pipelining.

ssor multiprocessor with shared memory and very fast interprocessor
munication. Here, the processors will be programmed to be particular
h C. If C could become the basis of the machine, then UNIX could be run.

6 Computer Modules microprogrammed for particular functions. AFP was
to be operated in this mode for image processing.

ow computer. This is a special case of item 3 whereby particular
s are programmed to behave as the various functionla units of a dataflow
.

special, parallel processing architectures using individual,
grammed processors as the functional units of the particular structures.
mode, AFP turns out to be a very good emulator of relatively complex VLSI

imental Ultra-LAN based architecture. To examine how computers can be
effectively and work together on a task, the AFP looks like an ideal for

APENDIX 3

ROUGH BUDGET

nses are estimated at approximately $18M/year running from 1982 through
is expensed as delivered. In general two or three "competitive but
roups are charged with each project family.

	#	Heads	Total	Expenses	($M)
	Sites	(ea.site)	Heads	Manpower	Equip
ructures	2	5	10	1	-
el Computation	1	1	1	.1	-
ign Automation	1	3	3	.3	-
g Environment	1	2	2	.2	15
ledge/	1	3	3	.3	-
	6		19	1.9	15

	#	Heads	Total	Expenses	($M)
	Sites	(ea. site)	Heads	Manpower	Equip.
ructures	2	5	10	1	5
el Computation	3	3	9	.9	-
ign Automation	2	5	10	1	-
g Environment	2	2	4	.4	10
ledge	3	5	15	1.5	-
Studies	1	3	3	.3	-
n Studies	1	5	5	.5	-
	14		56	5.6	15

	#	Heads	Total	Expenses	($M)
	Sites	(ea. site)	Heads	Manpower	Equip.
ructures	2	6	12	1.2	5
el Computation	3	3	9	.9	-
ign Automation	2	5	10	1.0	-
g Environment	3	2	6	.6	8
ledge	3	5	15	1.5	-
Studies	2	5	10	1.0	-
n Studies	3	5	15	1.5	-
	18		77	7.7	13

	#	Heads	Total	Expenses	($M)
	Sites	(ea. site)	Heads	Manpower	Equip.
ructures	2	6	12	1.2	5
l Computation	3	3	9	.9	-

ign Automation	2	5	10	1.0	-
g Environment	3	2	6	.6	5
ledge	3	5	10	1.0	-
Studies	2	5	10	1.0	-
n Studies	3	5	15	1.5	-
	18		77	7.7	10

	#	Heads	Total	Expenses	($M)
	Sites	(ea. site)	Heads	Manpower	Equip.
ructures	2	6	12	1.2	5
el Computation	3	3	9	.9	1
ign Automation	2	5	10	1.0	0
g Environment	2	2	6	.6	5
ledge	3	5	15	1.5	-
Studies	2	5	10	1.0	1
n Studies	3	5	15	1.5	-
	18		77	7.7	12

	#	Heads	Total	Expenses	($M)
	Sites	(ea. site)	Heads	Manpower	Equip.
ructures	2	6	12	1.2	5
el Computation	3	3	9	.9	-

ign Automation	2	5	10	1.0	-
g Environment	2	2	6	.6	5
ledge	3	5	15	1.5	-
Studies	2	5	15	1.5	-
n Studies	3	5	15	1.5	-
	18		77	7.7	10

	#	Heads	Total	Expenses	($M)
	Sites	(ea. site)	Heads	Manpower	Equip.
ructures	2	5	10	1	4
el Computation	2	3	6	.6	-
ign Automation	2	5	10	1	-
g Environment	3	2	6	.6	5
ledge	3	5	15	1.5	-
Studies	1	5	5	.5	-
n Studies	3	5	15	1.5	-
	16		67	6.7	9

	#	Heads	Total	Expenses	($M)
	Sites	(ea. site)	Heads	Manpower	Equip.
ructures	2	5	10	1	1

el Computation	1	2	2	.2	-
ign Automation	1	5	5	.5	-
g Environment	3	2	6	.6	5
ledge	1	5	5	.5	-
Studies	1	1	1	.1	-
n Studies	1	5	5	.5	-
	10		34	3.4	6

	#	Heads	Total	Expenses	($M)
	Sites	(ea. site)	Heads	Manpower	Equip.
ructures	2	5	10	1	-
el Computation	-	-	-	-	-
ign Automation	1	2	2	.2	-
g Environment	2	2	4	.4	3
ledge	-	-	-	-	-
Studies	-	-	-	-	-
n Studies	1	2	2	.2	-
	6		18	1.8	3

	#	Heads	Total	Expenses	($M)
	Sites	(ea. site)	Heads	Manpower	Equip.

ructures	1	5	5	.5	-
el Computation	-	-	-	-	-
ign Automation	1	1	1	.1	-
g Environment	1	2	2	.2	1.8
ledge	-	-	-	-	-
Studies	-	-	-	-	-
n Studies	-	-	-	-	-
	3		8	.8	1.8

TALK LISTING 9/14/78

FILE #

5-YEAR AWARD TALK

AMERICAN CAN TALK (TEXT FROM NSF)

BASIC LANGUAGE TALK

CAD SYMPOSIUM

COMPCON

COMPUTATIONAL SALES MGR. MEETING--MONTREAL

COMPUTER ORGANIZATION & ARCHITECTURE--COURSE

COMPUTERS IN EDUCATION--CHALLENGE

EDUCATION LECTURE--MANAGEMENT OF TECHNOLOGY

ESG TALK

IBM LECTURES - COMPUTER ARCHITECTURE

INTERACTION AMONG TECHNOLOGY, PRODUCTS AND USERS

IRVINE--THE 2ND DECADE

MCDOWELL AWARD

MELLON AWARD SPEECH

MINI TALK-1

MINI TALK-2

MINI + EFFECT OF SEMIS ON -11 DESIGN

MUSEUM SLIDE TALK--COMPUTER GENERATIONS

NAVAL UNDERWATER SYSTEM

NAVY TALK ANAPOLIS

N.E. TALK

NEREM

NET TALK

OBJECTIVES--HOW TO SET UP

PDP-10 MARKETING

PDP-11 TALK FILE

QUANTUM SCIENCE TALK

RTM'S

STANDARDS & PORTABILITY FOR SOFTWARE PROBLEM

TECHNOLOGY--3 YEARS

VAX-11 TALK OUTLINE

WHAT IS THE FIFTH GENERATION?

. PR

. TO ENGAGE US (CRITICS, COMPETITORS)

. TO LEARN TO DO RESEARCH

. TO LEARN KNOWLEDGE ENGINEERING AND OTHER AI-BASED TECHNIQUES

. TO GET BY-PRODUCTS FROM FAR-OUT GOALS

. TO REPEAT SUCCESS IN SEMIS AND SUPERCOMPUTERS

THE NEXT GENERATION WILL BE EVOLUTIONARY

. FUJITSU AND HITACHI HAVE RUN THE LIVERMORE KERNELS AT >2 x THE CRAY xmP USING EVOLUTION:

  . 25 YEAR OLD LANGUAGE - FORTRAN

  . 20 YEAR OLD ARCHITECTURE - 360/370

  . 25 YEAR OLD CIRCUITS AND SEMIS--ECL

GENERAL FACTORS IN COMPETING WITH JAPAN

- P = I x E (INTELLIGENCE, ENERGY)
- SOCIETAL VALUES
  - MEDICINE, LAW, POLITICS, ... , BUSINESS
  - SCIENCE, ENGINEERING, ... , MANUFACTURING
- TECHNOLOGICAL INFRASTRUCTURE
  - MATERIALS (SEMICONDUCTORS, MAGNETICS, ...
  - MECHANISMS
  - MANUFACTURING (CONTROL, ROBOTICS,
- MANAGEMENT - ESPECIALLY ENGINEERING AND MANUFACTURING
- QUALITY

. LONG TERM VERSUS SHORT TERM

COMPETING FOR THE NEXT GENERATION

. TURBULENCE DUE TO GENERATION TRANSITION

- NEW INDUSTRIES/PRODUCTS WITH MICRO

- VENTURE CAPITAL <--> ENTREPRENEURIAL ENERGY
  MANY, REDUNDANT, SHORT-TERM PRODUCTS
  THEREFORE, MUCH SHAKEOUT AND LOST EFFORT

PROBLEMS IN RESEARCHING THE NEXT GENERATION

- JAPAN IS BETTER COUPLED TO U.S. RESEARCH THAN AMERICAN INDUSTRY

- RESEARCH ON THE NEXT GENERATION IS HARD

.
UNIVERSITIES AND INDUSTRY ARE BOTH ILL-EQUIPPED!

- LACK OF GOALS CREATES LOTS OF POOR PROJECTS

- LOTS OF FUNDING - FEW PEOPLE,

THEREFORE, TURBULENCE AND LOST EFFORT

- LOTS OF POORLY STAFFED, SUB-CRITICAL PROJECTS

- LARGE PROJECTS - LACK OF MANAGEMENT

WHAT IS THE FIFTH GENERATION?

- PR

- TO ENGAGE US (CRITICS, COMPETITORS)

- TO LEARN TO DO RESEARCH

- TO LEARN KNOWLEDGE ENGINEERING AND OTHER AI-BASED TECHNIQUES

- TO GET BY-PRODUCTS FROM FAR-OUT GOALS

- TO REPEAT SUCCESS IN SEMIS AND SUPERCOMPUTERS

THE NEXT GENERATION WILL BE EVOLUTIONARY

. FUJITSU AND HITACHI HAVE RUN THE LIVERMORE KERNELS AT >2 x THE CRAY xmP USING EVOLUTION:

  . 25 YEAR OLD LANGUAGE - FORTRAN

  . 20 YEAR OLD ARCHITECTURE - 360/370

  . 25 YEAR OLD CIRCUITS AND SEMIS--ECL

GENERAL FACTORS IN COMPETING WITH JAPAN

- P = I x E (INTELLIGENCE, ENERGY)
- SOCIETAL VALUES
  - MEDICINE, LAW, POLITICS, ... , BUSINESS
  - SCIENCE, ENGINEERING, ... , MANUFACTURING
- TECHNOLOGICAL INFRASTRUCTURE
  - MATERIALS (SEMICONDUCTORS, MAGNETICS, ...
  - MECHANISMS
  - MANUFACTURING (CONTROL, ROBOTICS,
- MANAGEMENT - ESPECIALLY ENGINEERING AND MANUFACTURING
- QUALITY

. LONG TERM VERSUS SHORT TERM

COMPETING FOR THE NEXT GENERATION

. TURBULENCE DUE TO GENERATION TRANSITION

- NEW INDUSTRIES/PRODUCTS WITH MICRO

- VENTURE CAPITAL <--> ENTREPRENEURIAL ENERGY
  MANY, REDUNDANT, SHORT-TERM PRODUCTS
  THEREFORE, MUCH SHAKEOUT AND LOST EFFORT

PROBLEMS IN RESEARCHING THE NEXT GENERATION

. JAPAN IS BETTER COUPLED TO U.S. RESEARCH THAN AMERICAN INDUSTRY

. RESEARCH ON THE NEXT GENERATION IS HARD

.
UNIVERSITIES AND INDUSTRY ARE BOTH ILL-EQUIPPED!

. LACK OF GOALS CREATES LOTS OF POOR PROJECTS

. LOTS OF FUNDING - FEW PEOPLE,
THEREFORE, TURBULENCE AND LOST EFFORT

. LOTS OF POORLY STAFFED, SUB-CRITICAL PROJECTS

. LARGE PROJECTS - LACK OF MANAGEMENT

PARALLEL COMPUTING

NETWORK - WITH LAN OR WAN INTERCONNECT

CLUSTER - WITH LAN INTERCONNECT

FUNCTIONAL - ONE PROCESSOR PER FUNCTION

CLOSE AREA NET CLUSTER - HIGH SPEED INTERCONNECT

TIMESHARING - ONE PROCESSOR PER USER

PARTITIONED - ONE PROCESSOR PER PROCESS

TRANSACTION PROCESSING - PROCESSOR PER STEP

FAULT-TOLERANT - DIFFERENT PROCESSORS ASSIGNED PER

STEP WITH REDUNDANT COMPUTATION

CONCURRENT-TASK - PARALLEL PROCESSING OF A TASK BY PARTITIONING FOR INDEPENDENT DATA

PIPELINED-TASK - PARALLEL PROCESSING OF A TASK

PARALLEL PROCESSING - PROCESSORS WORK ON ONE TASK

"IF A COMPUTER UNDERSTANDS ENGLISH,

IT MUST BE JAPANESE."

-ALAN PERLIS

THE FOURTH GENERATION

- EVOLUTIONARY USE BASED ON WORD, DATA PROCESSING, PROFESSIONAL APPLICATIONS. EMBEDDED COMPUTING
- INTER-COMMUNICATIONS AND PRODUCTIVITY NEEDS TO INCREASE USE
- WELL-DEVELOPED TECHNOLOGIES, INCLUDING VLSI MICROPROCESSORS, LANS, MAGNETICS, DISPLAYS AND STANDARD SOFTWARE
- NEW ORGANIZATIONS TO BUILD NEW COMPUTERS, BUT
- NEW USES THAT EVOLVE WON'T BE KNOWN FOR A DECADE

THE NEXT GENERATION: REVOLUTIONARY VIEW

- REVOLUTIONARY USE DEPENDING ON VOICE AND
  NATURAL LANGUAGE COMMUNICATION
- GREATER COMMUNICATION AND PRODUCTIVITY NEEDS
  INCLUDING ROBOTICS, SPEECH AND NATURAL LANGUAGE,
  EXPERT SYSTEMS FOR COMPLEXITY AND PRODUCTIVITY
- ROBOTICS, AND ARTIFICIAL INTELLIGENCE,
  FAST-WANS, ULTRA- AND VLSI AND PARALLELISM
- AVANT GARDE ORGANIZATIONAL COOPERATION
  BETWEEN RESEARCHERS AND INDUSTRY

THE NEXT GENERATION: EVOLUTIONARY VIEW

- EVOLUTIONARY USE. WIDESPREAD ELECTRONIC MAIL,

  ELECTRONIC-BASED LOGIC TO ENCODE KNOWLEDGE

- NEED TO HAVE INFORMATION AT "FINGERTIPS"

  (IN THE SYSTEM AND NOT IN PAPERS AND BOOKS)

- EVOLUTIONARY TECHNOLOGY WITH LARGER,

  DISTRIBUTED MEMORIES

- NEW COMPANIES. BUILD WITH EVOLVING TECHNOLOGY

A GENERATION IS THE CONVERGENCE OF:

- NEED (EG. THREAT OF ANNIHILATION, GREED)

  FREEING RESOURCES

- TECHNOLOGY AND SCIENCE

  THAT PROVIDE FOR BUILDING MACHINES

- ORGANIZATIONS TO BUILD NEW COMPUTING STRUCTURES

- USE TO CONFIRM A GENERATION (AFTER THE FACT)

PARALLEL COMPUTING

NETWORK - WITH LAN OR WAN INTERCONNECT

CLUSTER - WITH LAN INTERCONNECT

FUNCTIONAL - ONE PROCESSOR PER FUNCTION

CLOSE AREA NET CLUSTER - HIGH SPEED INTERCONNECT

TIMESHARING - ONE PROCESSOR PER USER

PARTITIONED - ONE PROCESSOR PER PROCESS

TRANSACTION PROCESSING - PROCESSOR PER STEP

FAULT-TOLERANT - DIFFERENT PROCESSORS ASSIGNED PER STEP WITH REDUNDANT COMPUTATION

CONCURRENT-TASK - PARALLEL PROCESSING OF A TASK BY PARTITIONING FOR INDEPENDENT DATA

PIPELINED-TASK - PARALLEL PROCESSING OF A TASK

PARALLEL PROCESSING - PROCESSORS WORK ON ONE TASK

"IF A COMPUTER UNDERSTANDS ENGLISH,

IT MUST BE JAPANESE."

-ALAN PERLIS

THE FOURTH GENERATION

- EVOLUTIONARY USE BASED ON WORD, DATA PROCESSING, PROFESSIONAL APPLICATIONS. EMBEDDED COMPUTING
- INTER-COMMUNICATIONS AND PRODUCTIVITY NEEDS TO INCREASE USE
- WELL-DEVELOPED TECHNOLOGIES, INCLUDING VLSI MICROPROCESSORS, LANS, MAGNETICS, DISPLAYS AND STANDARD SOFTWARE
- NEW ORGANIZATIONS TO BUILD NEW COMPUTERS, BUT
- NEW USES THAT EVOLVE WON'T BE KNOWN FOR A DECADE

THE NEXT GENERATION: REVOLUTIONARY VIEW

- REVOLUTIONARY USE DEPENDING ON VOICE AND NATURAL LANGUAGE COMMUNICATION
- GREATER COMMUNICATION AND PRODUCTIVITY NEEDS INCLUDING ROBOTICS, SPEECH AND NATURAL LANGUAGE, EXPERT SYSTEMS FOR COMPLEXITY AND PRODUCTIVITY
- ROBOTICS, AND ARTIFICIAL INTELLIGENCE, FAST-WANS, ULTRA- AND VLSI AND PARALLELISM
- AVANT GARDE ORGANIZATIONAL COOPERATION BETWEEN RESEARCHERS AND INDUSTRY

THE NEXT GENERATION: EVOLUTIONARY VIEW

- EVOLUTIONARY USE. WIDESPREAD ELECTRONIC MAIL,

  ELECTRONIC-BASED LOGIC TO ENCODE KNOWLEDGE

- NEED TO HAVE INFORMATION AT "FINGERTIPS"

  (IN THE SYSTEM AND NOT IN PAPERS AND BOOKS)

- EVOLUTIONARY TECHNOLOGY WITH LARGER,

  DISTRIBUTED MEMORIES

- NEW COMPANIES. BUILD WITH EVOLVING TECHNOLOGY

A GENERATION IS THE CONVERGENCE OF:

- NEED (EG. THREAT OF ANNIHILATION, GREED)

  FREEING RESOURCES

- TECHNOLOGY AND SCIENCE

  THAT PROVIDE FOR BUILDING MACHINES

- ORGANIZATIONS TO BUILD NEW COMPUTING STRUCTURES

- USE TO CONFIRM A GENERATION (AFTER THE FACT)

MINICOMPUTER COMPANY LESSONS

1. NEARLY 25% SURVIVED
2. ONLY 8% REALLY WON
3. ONLY 2% OF STARTUPS RETAINED AUTONOMY
4. MERGING WAS TRIED BY 10%

5. BUT ONLY 1/3 OF MERGERS WERE SUCCESSFUL

6. BEING A COMPUTER SUPPLIER DIDN'T HELP:
ONLY DEC AND IBM MADE THE TRANSITION!

7. IBM ALWAYS WINS... EVENTUALLY
(SYSTEM 3... SERIES 1)

MINICOMPUTER TECHNOLOGY (CIRCA 1970)

BASIC INDUSTRIES	MINICOMPUTER COMPANIES
POWER SUPPLIES	OPTIONAL
PACKAGING	ESSENTIAL
CORE MEMORY	OPTIONAL
SEMICONDUCTORS (MSI)	CPU AND MEMORIES
DISKS AND TAPES	PERIPHERAL CONTROLLERS
TERMINALS	-
	OPERATING SYSTEMS
	LANGUAGES
APPLICATIONS	OPTIONAL
	SYSTEM INTEGRATION

MAINFRAME TECHNOLOGY (1950, 1960)

BASIC INDUSTRIES	MAINFRAME COMPANIES
DISCRETE COMPONENTS	
TUBES, TRANSISTORS	PLUG-IN UNITS
MEMORIES	MEMORIES
	PERIPHERALS
	LANGUAGES
	APPLICATIONS
	SYSTEM INTEGRATION

MICROCOMPUTER-BASED COMPANIES

BASIC INDUSTRIES	MICROCOMPUTER COMPANIES
POWER SUPPLIES	OPTIONAL
PACKAGING	OPTIONAL
SEMICONDUCTORS	-
(MICROS, MEMORY, PERIPHERALS)	
CRT'S AND TERMINALS	-
DISKS AND TAPES	-
BOARD OPTIONS	OPTIONAL
UNIX & DIAGNOSTICS	OPTIONAL
LANGUAGES & DATABASES	OPTIONAL
LAN'S / COMMUNICATION	OPTIONAL
APPLICATIONS	OPTIONAL
	SYSTEM INTEGRATION

MICROPROCESSOR-BASED COMPUTER PRODUCTS

ON A DESK (PERSONAL COMPUTERS)

- SMART TELEPHONES
- TERMINALS
- HOME (AND GAME)
- PORTABLE PC'S
- WORD PROCESSORS
- PC'S
- WORKSTATIONS

DEPARTMENTAL AND GROUP-LEVEL COMPUTERS

- MICRO
- SUPER-MICRO
- CLUSTERED, FUNCTIONAL MULTIPROCESSOR
- SYMMETRIC MULTIPROCESSOR
- HIGH-AVAILABILITY
  - SINGLE COMPUTER VIA VOTING
  - MULTIPROCESSOR (N+1) REDUNDANCY
  - MULTI-COMPUTER CLUSTERS

STANDARDS

SPECIFIES INTERCONNECTION OF TWO (OR MORE) MORE PARTS

VANITY: EXISTS WITHIN A SINGLE COMPANY

DE FACTO: ORIGIN IS A SINGLE COMPANY, EVERYONE FOLLOWS

...SOME STANDARDS ORGANIZATION MAY BLESS IT

INDUSTRY: EXISTS WITHIN A SINGLE COMPANY (IBM)

NATIONAL: (EG. ANSI, JSA)

GOVERNMENT BUREAUS: (EG. NBS VDE)

INTERNATIONAL: (EG. ISO, IFIP, ECMA, CCITT)

PROFESSIONAL ORGANIZATION: (EG. IEEE, ASME)

CONSIDERATIONS FOR FUTURE COMPCON ON STANDARDS

- STANDARDS ESSENTIAL TO PROVIDE A PATHWAY FOR THE FUTURE BUT DON'T CONSTRAIN CREATIVITY,

- NATURE OF GOALS AND CONSTRAINTS FOR FUTURE STANDARDS

- ROLES OF VARIOUS ORGANIZATIONS: MANUFACTURERS, USERS, STANDARDS BODIES, PROFESSIONAL ORGS., ACADEMIA

- TIMING IN RESEARCH, PRODUCT, AND USE LIFE CYCLE

- MAININTAINING, EVOLVING AND DISCARDING STANDARDS

- ECONOMICS OF SINGLE AND MULTIPLE STANDARDS

A PRODUCT SEGMENTED INDUSTRY,
ORGANIZED BY LEVELS OF INTEGRATION
WHICH FORM STRATA DEFINED BY STANDARDS

- FUELED BY ENTREPRENEURIAL ENERGY
  RELEASED BY VENTURE CAPITAL FUNDS

- SOFTWARE FORMS NEW PRODUCTS AND USES

- SYSTEMS ARE DISTRIBUTED IN MANY DIFFERENT WAYS

GUIDELINES FOR STANDARDS

. SPONSORED... NOT JUST A COMMITTEE OR COMMITTEES

. REAL (IMPLEMENTABLE AND TESTABLE)

. UNIFIED NOT ALL POSSIBLE SOLUTIONS FOR ONE FUNCTION

. PRECISE, UNDERSTANDABLE AND APPLICABLE

. TIMELESS, AND

EXTENDABLE IN A RESPONSIVE FASHION

CRITICAL EXTENSIONS TO UNIX

. VIRTUAL MEMORY

. APPLICATIONS: REAL TIME, TRANSACTION PROCESSING, ETC.

. MODERN HUMAN INTERFACE FOR WINDOWING, GRAPHICS

. MULTIPROCESSING

. WIDE AREA NETWORKS

. LOCAL AREA NETWORKS (LAN) AND CLUSTERS (LANC)

HEURISTICS FOR STANDARDS

. EITHER MAKE THE STARNDARD OR FOLLOW THE STANDARD

. BUT, BE PREPARED TO REACT QUICKLY AND
  FOLLOW WHEN THE DE FACTO STANDARD CHANGES

. SET NEW STANDARDS AT YOUR OWN PERIL, AND
  ONLY IF YOUR NEW ONE IS <u>MUCH</u> BETTER

. CHANGE WHEN IT'S CLEAR YOU'VE GONE DOWN A RAT HOLE

LEVELS-OF-INTEGRATION: THE STRATA

SILICON WAFER:

STANDARD CHIP: MICROS, MICRO-PERIPHERALS, MEMORIES

BOARD: BUSES FOR PERFORMANCE, APPLICATIONS...

ELECTROMECHANICAL: DISKS, I/O, POWER, ENCLOSURES...

OPERATING SYSTEM: COMMUNICATIONS, DATABASES, I/O...

LANGUAGE: INCLUDING APPLICATION LANGUAGES

GENERIC APPLICATION: WORD PROCESSING...

DISCIPLINE/PROFESSION SPECIFIC APPLICATION:

```
procedure VENTURE_CAPITAL_ENTREPRENEURIAL_ENERGY_CYCLE

     begin

     while greed and not fear do

     write (business_plan);

     get (venture_MONEY);

     exit {job}; start (new_company);

     build (product); sell (product);

     sell (new_company); {for 100 times sales}

     venture_funds := venture_funds + sale_liquidity;

     end
```

UNDERSTANDING EVOLUTION TO LEVERAGE THE LEVERAGE

Civilization has always been concerned with building tools to leverage intellectual processes. Although a few tools are revolutionary, most are evolutionary. Virtually all revolutionary tools (machines) fail, usually for simple reasons. What are the heuristics for success and failure avoidance?

INTELLECTUAL LERAGE: THE DRIVING TECHNOLOGIES

Hope of conference: point out symbiotic relationship between new technology, and applications made possible by new tools and applications made possible by new technology, and the advances in in technology made possible by ever more powerful tools.

The notion of intellectual leverage also conjures up the notion of working smarter and not harder. I fear we don't always do this, also it seems to me one of our greates frailties is reinventing while not learning. In one of the Turing lectures, Hamming pointed out: we have to stand on each others shoulders instead of each others feet.

WHAT ARE TOOLS?

.machines ala VAX 780 that are the basis of all startups... all of which aim to replace the 780 the good news is that most of them will fail and hence the competition won't materialize. I feel like someone who has just equipped an army to shoot me. A capitalist is a person who'll sell you the rope to hang him.

.machines such as DA that do work, where we always seem to need the next one to design what we are designing

.networks, (Conway cited ARPAnet for multichip project last year)

.mail nets

.organizations

.mathmatics, notations, ways of communicating, note slang that allow the bright students to communicate (rent,dren or ren)

.a methodology and training

.goals and constraints: eg Supercomputers, targets and standards

TARGETS ARE KEY

Generations are now the target. Two scenarios.

This generation is different for what is a new industrial structure.

STANDARDS ARE THE LEVERAGE BECAUSE THEY HELP DEFINE THE TARGET

This generation is different. It is based on standards. Instead of a completely proprietary vertically integrated industry, it is a product fragmented, strateified by level of integration.

Driven by entrepreneurial Energy.

We have a very good example in the PC. Doesn't have to be the best. It was though. It was easy and strictly evolutionary. It came about by being an open architecture.

I would like to see truth in labeling as part of product description. If an interface will be kept proprietary or not. If not, what is the policy for evolution.

WHAT SHOULD STANDARDS DO?

play together

be easy to use and understandable

be around for awhile (ren retyped logo for prefix to infix

be evolvable: note Datpoint>8008>8080>Z80>8086>186>286>386
(Copy: national did, a good idea)

be real. ISO model is a model and not real. It's like quadrille graph paper with 1/4 squares and giving them a ruler marked in 0.1"... but at least it's in the same basic system.

be responsible and responsive

be few so designers know what to shoot at. Can't have 8 LANs or a new battery, every time a watch is built

be at right price

RESPONSIBLITY OF THE ORGANIZATIONS

Make them open by definition

Sponsor their evolution

Be able to test whether they adhere

Clear organization (name) in charge. with schedule.

No standard without an implementation

Standards meetings should contain implementers

CHIP-level... inability of foundry standards, communication of masks, etc. CIF established one. Responsiblity of the foundries. Absolutely chaotic.

ISPA- is by definition a major standard and this goes even lower vis a vis access to bit maps.   Responsiblity of the provider

Responsiblity for at least assemblers and mnemonics

What happened to CFA? DEC offered to give DOD the VAX. Didn't like it because they didn't control it. I'd rather have someone responsible controlling something than a committee anyday

We have a disturbing phenomenon around evolution to VM! Show the evolution of VA bits.

OS- UNIX now, with a new player that's hardly used to computer standards

Would like to have standards on my telephone so I can use they across different vendors. Telephones aren't standard!

Maybe UNIXco should be a seperate company, but may be the only profitable part of AT&T, given the price of royalties. I hope AT&T realizes the major lesson from UNIX: the reason it exists as a standard was that the price was right to universities: namely 0! The other reason was that it was small and simple enough for pedagogy, yet useful enough to attract and train students.

Finally, it evolved to be somewhat transportable and that was due to C.

Progress has been slow 300 b in 65, 1200 in 80.

issues: V.2 + 4.2 with incompatiblities between 4.1 to 4.2.

shouldn't UNIXco have been split off... maybe the only money maker in AT&T.

VENCO may get the UNIX crew out of AT&T

Crucial to have networking, VM, file, tp, multiprocessing, window

Would like a body for defining and then testing adherence to standards!

Misunderstanding that if one simply has a Unix port to a given machine, this is an acceptable system. NO. Much to do in compilers

LANGUAGES- It is critical that we start to address parallelism!

C- just another version of an assembler, after BCPL and before D

LISP- an absolute zoo. Keeps AI back. The Japanese were so confused they went to Prolog

DATABASES-

Spreadsheets

LANs

A paradox: they can't exist until they exist

When is it appropriate to simply acknowledge a de facto and move on? or to approve a new ad hoc one (eg. 1 Mhz PCnet)

We need a set of high level protocols. Gateways are too glib... can they be done? 12 protocols within HP, hence a committee to standardize and then you have 13.

802.3 CSMA/CD most developed with base, broad, thin and fiber media; Omninet is 1 Mhz, PC net is 1 Mhz

.4 Token Bus, especially on cable TV; committee has 3

.5 Token Ring- Apollo, Primenet, Cambridge Ring, IBM/TI why is this man smiling?

.8 Fiber

802.n = PBXen... most folks believe this is WAN. Wrong.

MAIL

3 standards plus lots of company ones

NOW THE BAD NEWS:

the US computer inustry structure will be overturned

Compcon: 70% Bay, 7% Japanese, 95% practitioners, 1.2Kp

committee wanted a discussion of Mail to improve technical excellence

---

ZINGERS

JAPANESE have won S/C race with x2 Cray xmP

Glamorous professions: science, business, engineering, not mfg eng

Glamorous businesses: systems not components, materials, products

or processes (eg robots)

SLIDES (*to do)

Ethernet and Unibus

Idea of decomposing (fision) from minis and mainframes

Idea of aggragation (fusion) from PCs and minis to build mainframe

*Evolution of address space

diversity of new machine structures

Generations, this generation, evolution to next, revolution to next

Correlate machine intros with memory intros

table of standards vs time

AREN'T STANDARDS OUR FRIENDS?

Gordon Bell

Chief Technical Officer

Encore Computer Corporation

15 Walnut Street

Wellesley Hills, Massachusetts 02181

Standards form the constraints necessary for the evolution NOT revolution, into the next decade of computing. Constraints save design time by narrowing the search for new products and processes. They also permit building on past work in a hierarchical fashion rather than having to start each new design with silicon. Standards provide a real intellectual discipline to leverage computing further and faster.

More than any other factor, the lack of standards impedes technological progress and lowers productivity. Redundancy in product development ties up critical resources on the reinvention of trivia with a shortage of resources for solving hard problems such as speech and video communication, intelligent programs, revolutionary machines and fully automatic production. Is there a shortage of engineers? or a shortage of leadership that lets engineers build disconnected, overlapping, low technology products?

As a professional society, the IEEE Computer Society can foster evolutionary change for the next generation. Revolution is different. It destroys an old order by new ways of thinking: the Industrial Revolution and the Information Revolution had major impacts throughout society. Having been designated an "official" Computer Pioneer <>[footnote: February 23, 1982], I'd like to call on the engineering establishment to see that key standards are set for the evolution to the next generation information era. These targets are the key toward productivity and maintaining a healthy information processing industry.

A brief review explains the critical role of standards in today's computer evolution, followed by ten observations (rules) that could make the standards setting and adoption process flow more smoothly. Five particular standards areas (silicon chips and wafers, microprocessors and their busses, LANs, UNIX, and LISP) are examined because they constitute barriers that are impeding technological progress. The final section proposes that a COMPCON be devoted to the sole purpose of understanding various standard and the standards process. The last paragraph describes the undesirability of standards.

STANDARDS ARE THE BASIS OF TODAY'S EVOLUTIONARY GENERATION

The first two computer generations were characterized by complete vertical integration. Each computer company or division designed and manufactured circuits, peripherals, hardware systems, operating systems, languages and applications and created unique, proprietary standards. Today, standards provide clear constraints for building products within a given strata and segment. For example a spread sheet industry has evolved common data format standards for programs built by different software companies. This not only allows competitive programs but the ability to interface with plotting and database programs--thus two other industries can form.

Today's generation is characterized by a large set of product segmented industries that are organized by levels of integration that form strata.

Strata are formed because of standards<1>. Entrepreneurial energy drives the industries and venture capital releases the energy.

Unlike the previous generations when processors and memories constituted a large fraction of system cost, microprocessors are comparatively small and standard. Thus, many more, creative computing structures are possible because of standardization! In essence, standards allow us to build with bricks not a collection of designer-created "pet rocks".

In 1984, about eight levels of integration form the industrial strata (Table 1). The bottom four are hardware and the next four software and applications. Each level has many product segmented industries. A given organization usually excels in only a few strata/product segments or types of systems which are a collection of many strata.

TABLE 1. THE LEVELS OF INTEGRATION THAT FORM THE INDUSTRIAL STRATA

DISCIPLINE AND PROFESSION SPECIFIC APPLICATION: eg. CAD (for logic design)

GENERIC APPLICATION: word processing, electronic mail, spread sheets, etc.

3RD GENERATION PROGRAMMING LANGUAGES:

Fortran, BASIC, C

OPERATING SYSTEM: base, communication gateways, databases

CP/M, MS/DOS, UNIX

ELECTROMECHANICAL: disks, monitors, power supplies, enclosures

8", 5", 3"(?) floppy; 5" winchester

PRINTED CIRCUIT BOARD: busses that are synchronized to micro and memory

S100, Multibus, PC Bus, Multibus II and VME

STANDARD CHIP: micros, micro-peripherals and memories

Evolution of Intel and Motorola Architectures which are synchronized to the evolution of memory chip sizes: 8080 (4K), Z80 (16K),

8086 and 68,000 (64K), 286, 68020 and NS32032 (256K)

SILICON WAFER: bipolar and evolving CMOS technologies
(proprietary, corporate process standards... require formalization
in order to realize a Silicon Foundry-based Industry)

EARTH: IRON AND SAND

SOME RULES FOR SETTING AND USING STANDARDS

A standard specifies precisely how two or more parts interconnect.

Table 2 gives a collection of suggestions (rules) that we might observe in regard to setting and using standards. They are not complete or necessarily consistent, merely some observations and personal predjudices.

TABLE 2. RULES FOR SETIING AND USING STANDARDS

1. Either make the standard or follow the standard.
2. Be Prepared to react quickly and follow when the de facto standard changes.
3. Change (the standard) when it's wrong.
4. Somebody (person(s), company, or companies) must be responsible for defining, implementing and caring for a standard.
5. Minimize the number of organizations responsible for a standard.
6. Almost any standard is far more important than a highly refined, optimum. To make progress we often have to regress.
7. Provide and plan for evolution, it's often the fastest way.
8. Make the standard based on real experience, not a design by committee. If you haven't lived with a proposed standard, don't adopt it.
9. A standard must be precise, understandable, applicable and useful at many levels of detail.
10. Only one (or a few) standards are needed for the same function; a standard should aim toward unifying a set of alternatives. Ideally, a standard should define the interface between sets of parts, not just two parts.

1. <u>Either make the standard or follow the standard</u>.

The IBM PC emerged immediately as a standard. It came at a propitious time - concurrent with a processor capable of accessing almost a megabyte of memory, the 64K chip, widescale availability of 5" floppies, and just prior to 5" Winchesters. The explosion in software for PCs came about because people could work on useful applications instead of reinventing and transferring old operating systems for hardware idiosyncratic PCs. The standard is fine for at least five years; BYTE Magazine's recent editorial about The Compatibility Craze<2>, was wrong.

2. Be prepared to react quickly and follow when the de facto standard changes!

Those who follow the IBM standards might remember the 360/370 transition. The Amdahl Corporation learned this in the early 70's when they had to scrap their 360 design as IBM evolved and introduced the 370. History may repeat when future IBM PC's obsolete current products by providing a fully upward compatible product with more capability such as virtual memory.

On the other hand, the Lisa and Macintosh designers<3> at Apple must be congratulated for NOT following the IBM standard for the PC. People and organizations such as the Amdahls at Trilogy who deviate from simple evolution in technology in order to make progress are essential, even though deviations fail or when successful, become the main line. Deviations that are meaningful just don't repackage old ideas or provide the same, simple function; they are large-scale well thought-out projects that provide much more capability and take us in a different direction.

3. Change (the standard) when its wrong.

IBM has finally adopted ASCII codes in a meaningful way after years of EBCDIC - Extended Binary Coded Decimal Interchange Code<4>, an evolutionary extension and remnant of the card era. This should provide mainframes with much translation work as the PC and mainframe communicate.

4. Somebody (person(s), company, or companies) must be responsible for defining, implementing and caring for a standard.

Ethernet is a good example. Xerox and DEC needed it as the backbone of their product strategies and Intel needed it to sell chips. Rarely has there been such an important interconnect standard as the LAN. How it is implemented is moot--the modulation (broadband versus baseband) and topology (busses, rings, trees, or centralized switches) have only minor impacts on cost and performance of the total system. Yet topological differences, the 802 series<5,6>, become ideological differences that consume everyone arguing. The work on building the real systems has been deferred by at least 5 years, with billions of dollars lost through redundancy, lack of productivity and poor communications.

5. Minimize the number of organization responsible for a standard.

Our industry lives in a chaotic world of de facto standards that emerge from particular companies (e.g. the S-100 Bus, dialects of BASIC), industry standards (i.e. IBM's standards), and vanity standards by individual companies which trap users into particular systems. Vanity machines--processors, operating systems, and languages, personal computers generated by every engineer and company trying to do its own thing with no real contribution to the state-of-the-art usually confuses the marketplace. In addition, many conflicting government and professional organizations across multiple, overlapped disciplines are involved in standards setting.

Perhaps the greatest problem in standards today is that too many groups are competing to set too few standards. We must have a way of selecting the best organizations and people to work on a standard, and to reduce the number of bodies involved because they only delay work and introduce noise. A standards process does have to recognize and respond to the great ideas which can come from outside the "official" body.

6. Almost any standard is far more important than a highly refined, optimum. To make progress we often have to regress.

Models such as the 7 layer open systems interconnect on which new standards may fit are often useful. Unless a model is followed by the detailed definition of the layers, it is only a template for everyone to create a unique, vanity standard and for writing advertising copy. Since implementation before standardization is a necessity, the seven layers might bettter be only 4, or even 9. Unfortunately, every real implementation that says it uses the 7 levels uses the levels like one uses a metric ruler to draw on 1/4" squared quadrille graph paper. The lines on the graph paper serve only as reference lines for the infinity of figures that one can draw using the ruler. About every 2.5 inches the two scales line up pretty well.

UNIX is an excellent example of picking an arbitrary standard that is below the state of the art and will require much evolution. The section on UNIX describes the desrability of the standard and the necessity evolution.

7. Provide and plan for evolution, it's often the fastest way.

The evolution of a real, arbitrary standard usually beats the ideal which never gets completed. The language, Algol is an excellent example of an ideal; it was, but wasn't completed or backed with appropriate implementations. As an ideal, it became a model for successor languages to build on.

With exponential change in virtually every dimension of computing, the domain of a standard should be specified a priori to understand when it should be extended. Many standards, such as Fortran, live longer than the sponsor thought or intended. National or international standards organizations can't arbitrarily pronounce a standard dead and ignore it as long as the standard has a large user population. Otherwise, new products make ad hoc extensions and no one is responsible.

8. Make the standard based real on experience, not a committee design. If you haven't lived with a proposed standard, don't adopt it.

The best way to insure reality is to implement several alternative interfaces before setting a given standard. The digital communications standard ISDN is an excellent example of a complex committee design, with no real test use, for what should be a simple, clean interface. No wonder this has taken a decade to design. If it is used widely in the next decade, a great deal of expensive redesign will be required.

It took almost ten years after a full-scale working model<7>, to develop the industry standard Ethernet (802.3). The upgrade over Xerox's first Ethernet provided almost a factor of 4 performance improvement. If the original had been used to get real experience then all local area networks could have been realized earlier, and not still be "several years away."<5>

9. A standard must be precise, understandable, applicable, and useful at several levels of detail.

You don't always have to be IBM to set a standard. In 1969, neither IBM nor any official group was interested in a standard for interconnecting computer components to form minicomputers. DEC's PDP-11 UNIBUS set the standard from the outset. Eight years after hundreds of engineers had designed hardware to attach to UNIBUSES, a really complete UNIBUS specification was finally written. The original specification provided a way of interconnecting different kinds of parts, not just a pair, and showed the way for this generation of buses and the future generation of micros.

Had a standards committee been involved in the original UNIBUS, it is doubtful whether the design would have been completed.

10. Only one (or a few) standards are needed for the same function; a standard should aim toward unifying a set of alternatives. Ideally a standard should define the interface between sets of parts, not just two parts.

Having too many standards is like having NO standards at all. The current plethora of 802 LAN standards, including digital communication switching, which is also a LAN, is a good example of too many, with no basis of experimentation.

Now, every company, consortia and committee try to get one more standard bus and LAN. In turn, members add features to ride every new bus and LAN specification. There are simply too many busses, LANs and riders. We need bus and LAN birth control.

CRITICAL STANDARDS FOR THE NEXT GENERATION

SILICON CHIP AND WAFER FOR CUSTOM SYSTEMS: THE SILICON FOUNDRY

The Silicon Wafer is an important level of integration that requires wide-scale standardization. Since semiconductor processes have traditionally been the corporate jewels of semiconductor companies, the wafer and chip is not a well publicized or documented level. Yet, it is safe to predict that the silicon wafer or custom chip is likely to be the basis of the next computer generation. Some computer systems will be a single chip with 1 to 10 million transistors. Of course, most chips will continue to come from semicomputer manufacturers as a "standard" or combinations of "standards" such as microcomputers, peripherals and memories.

Creative new products will come from the Silicon Foundry Industry that Carver Mead advocates<8> - and this requires substantial standardization. Weitek<9>, is an example of this new kind of company that takes algorithms and embeds them in silicon - VLSIzation. Another example is the workstation product, IRIS, from Silicon Graphics<10>. IRIS uses a dozen 75,000 transistor chips which Jim Clark calls the Geometry Engine and computes at a speed of 10 Megaflops--roughly equivalent to a CDC 7600 computer. In this way IRIS out performs, by a factor of several hundred, the other 150 workstations! One can envision radically new special chip-based systems which operate on pictures, voice, and mechanisms.

Due to lack of standards in foundries and CAD systems we are far from being able to realize the scenario of a Silicon Foundry Industry. Standards are essential for all user-specific gate array, standard cell or fully custom chips. It's distressing that we still have no standards for specifying gate arrays; custom PLAs and ROMs took too long to standardize. A few interfaces for this industrial structure include:

- specifications of structure and behavior, including simulation and timing at all levels
- physical information at all levels including for processing wafer masks (eg. CIF)
- control of foundry processes, especially if processing steps become optional
- chip test, including automatic generation of test data

. chip assembly and packaging including bonding and multi-chip interconnect

For CAD, the development of standard interfaces to languages and databases that are communicable via networks must be targeted. It might be desirable to standardize the specification languages; I can't identify any benefits of differences. Agreeing on interfaces doesn't limit the competitiveness or creativity of any CAD company or foundry, it means users don't have to learn many systems and languages for the same function, or convert data formats. Standards would let users mix and match different CAD systems in a completely flexible fashion. Syntactially idiosyncratic editors, timing verifiers, simulators, design rule checkers, etc. give no real increase in user power. Use would expand much more rapidly because buyers wouldn't be forced to make critical long term decisions, with no way to exchange data to other systems. This is completely analogous to the pre-Cobol / pre-Fortran era in the late fifties when all the users rebelled at every manufacturer providing a unique language. The rebels designed COBOL, the first, oldest, and largest used of the standard languages, because there was no reason for different languages!

In CAM, the user is also faced with a fuzzy and perplexing interface to the process from masks to tested components.

The foundries, CAD companies and users (e.g. the Microelectronics and Computer Corporation) could affect change now so we can have the next generation. A whole COMPCON could be devoted to describing alternatives and defining interfaces.

STANDARD CHIP: MICROS, MICRO-PERIPHERALS, MEMORIES

The semicomputer manufacturers have the responsibility for standards resulting from the Instruction-set Architecture. This lowest level of integration for computers is the input to a very high-gain "work amplifier" because it forces the creation of a range of unique busses, boards and systems, including operating systems and languages.

The micro is at the root of most of our redundant work. A micro's life is incredibly predictable, following a time worn path with respect to its ability to access memory. Frailey<11> suggested that there are about 20 measures of word length. I believe only one counts--the amount of directly addressable memory to a process, because it determines a computer's programmability and therefore its longevity. Of course, when considering performance, there are a few embellishments like data-types and implementation word lengths.

Unlike semiconductor process evolution, all micro users are dragged along as an architecture evolves and can relive history. For example we were dragged through the evolution of the stack, which started out in a Datapoint terminal, went on to become the 8008, the 8080, the Z80 (by another company providing us with an almost useful PC), then on to the 8086<12>, 186, 286 and more. As a user of these parts, I have been able to relive computer evolution for a third time. The good news is that the 286 may be the fastest micro--and it shows that evolution does work because there is finally a large, state-of-the-art address. It also illustrates the difficulty of design for compatibility.

In the late 50's, a system that allowed users to treat both primary and secondary memory as one was developed at the University of Manchester, using Ferranti's version of the university's second machine, Mercury. By 1962, the university had an operating breadboard with a 27 bit virtual address for Atlas<13>. (Atlas also had a number of other ideas, such as Extracodes that Bhujade recently rediscovered<14>.) Let's call Atlas the 0th time through. It was a university machine in the U. K. described in nearly ten papers, and Ferranti only built a few. The critical paper was republished in 1971 in Bell and Newell<15>, and again in 1982 in Siewiorek, Bell and Newell<16>. But if engineers read about it, they neither remembered or learned.

Having known Atlas, I went on to design two minicomputers with 12 and 13 bit addresses because I felt they were special and wouldn't evolve to general purpose use. Both had to be extended to 16 bit addresses almost before they were shipped. In 1964, the PDP-6, the forerunner of DECsystem 10 and the 360 were introduced and both could access about a megabyte. The DECsystem 10/20 and the 370 eventually ended up with 32 bits of address, complete with paging, just like ATLAS, but about 15 years later.

In 1970, the PDP-11 came out with a 16 bit address to solve the minicomputer addressing problem<17>. The first customer demanded a physical address extension to 18 bits. The virtual and physical addresses evolved to 17 and 22 bits. For several years DEC engineering spent thousands of hours trying to figure out how to address more memory.

Users spent much time encoding programs in small memories. In 1975, the VAX project was started to provide a 32 bit address with an embedded PDP-11 for compatibility. This cycle took about 8 years and was well documented<18,19>. Other East Coast minicomputers followed the same path for the second time around.

In 1971 the micro was born on the West Coast with the 4004 and 8008. These had 12 and 14 bits of address. The leverage of doing it right the first time was very high and the evolution ahead was clear. In 1978, the 8086 was extended to 20 bits and most recently to 24 bits of physical and 30 bits of virtual address. The cycle based on the 8086 has taken 6 to 12 years. It is ironic that information on addressing didn't travel from California to Oregon where Intel's 432 was developed.

Motorola's saga is similar. National took the high road and copied VAX without violating its patents to supply VAX-like chips - since DEC is a minicomputer, not a semicomputer company. Unfortunately, National didn't copy enough of VAX to make software transportability from VAX automatic. The National architecture is certainly an interesting alternative to VAX permitting transfer of VAX programs with minimal effort. If an exact copy of VAX could have been made, many billion dollars of software could have been made available, and many resources could have been freed for doing creative or otherwise productive work. With the micro the cycle has been repeated three times. The saga is not yet ended as we understand the ramifications of greater than 32 bit address spaces. Evolution to come.

The story surrounding the Computer Family Architecture--CFA, the Defense Department's version of VAX, Nebula, is far worse. An exact copy of VAX could have been made saving 10 years and many billions of dollars!

New architectures, especially those which have gone along well travelled evolution paths, have cost computing at least half of our resources and provided little or negative benefit. Now using C and UNIX to obtain machine independence is deceptively simple and misleading. A compiler for C or a compiler written in C is only a starting point for a product... not the end. An architecture pervades virtually every part of a system and its database. Even if C and UNIX can be standardized to a greater degree, the Instruction-set is still all pervasive. When an architecture should be copied, evolved or thrown out and started over is fundamental to the notion of standards because of the tremendous user program and data investments.

The issue of revolutionary, research architectures based on parallelism is another conference. A recent taxonomy listed 55 new evolutionary and radical computer system designs: 25 can be built, 15 may be built, perhaps 10 are worthwhile building and we have resources to build and evaluate at most 5! At a time when meaningful research requires large team efforts, only a few experiments can be performed. We need the results of a few critical experiments, not more half-done, toy projects.

BOARD: BUSES FOR VARIOUS PERFORMANCE, APPLICATIONS, ETC.

The board level is similar to the Instruction-set Architecture story, except that busses live longer. The various species of the IBM channel buses are now 20 years old and will continue for another 20 years in their current forms, even though many of the functions that a peripheral might perform could be handled with the same amount of hardware as that required to interface and drive the bus.

The IEEE sanctions these busses. The politics is hard to understand. Is a bus designed and sanctioned independent of whether there are any riders? How many more busses do we need or can we afford beyond the existing ones? Why aren't the onboard signals standardized in order to mix and match processor and peripheral chips?

LANs and LANCs ANOTHER KIND OF SWITCH

While riding busses, let's look at our most critical bus, the Local Area Network (LAN) used for interconnecting computers and terminals in a local area. A very few standards are essential so we can get on with building clusters - or LANCs - which few organizations understand experientially. The motivation of a LANC is the certain evolution of three types of clusters:

1. a single shared mini or large computer will gradually be decomposed into functional server components;

2. a collection of large computers must behave as a single system with a common database; and

3. the proliferation of PCs require intercommunication to form a single integrated system by aggregation.

LANs are especially difficult to design and standardize because they cross from the computer industry into traditional communications and cable television industries involving more disciplines and organizations. A sorry parallel can be sen in the slow formation of videodisk standards because both computer and television engineers are involved.

The IEEE 802 standards program is essential to LANs<5,6>. While it must be strengthened to include PABXs, a new set of numbers will be required just for all the new LANs. A description of the 802.X series follows.

802.3 was allocated for the CSMA/CD bus - Ethernet. Since there can exist lower cost LANs of this type, then groups took the basic idea and built incompatible, lower cost, non-standard versions. The same energy applied to cost reducing Ethernet would have made everyone win. With the recent announcement by IBM of a new LAN for PC's based on CSMA/CA, we have one more standard (and number).

802.3 can be transmitted on standard orange or yellow Ethernet cable. For those who like a simpler installation, lower cost and will give up distance, RGU 58 can be used - if you call it "Cheapernet." 3COM and Bob Metclfe, Ethernet's inventor, call this "Thin Ethernet." Codenol has a fiber optic transmission system using the same basic electronics! For those who like cable television technology, a modem permits the same controller to transfer Ethernet's baseband information on broadband. The purpose of all these media is to build and use LANs and not to wait for what is really quite an arbitrary choice of media that only delays the critical use. The controller / transceiver interconnect for Ethernet is becoming an important standard. With it, maybe any topology (including high speed PABXs), modulation scheme and media can be used!

802.4 denotes LANs carried on broadband with 3 incompatible data-rate versions. With the 5 Mbit per single channel pair version in operation, hopefully the other two will not materialize. General Motors is using its marketplace power to affect standardization by insisting that various computers demonstrate their ability to communicate at the NCC '84. If this works, maybe GM should take over some of the standards role from IEEE, ANSI, ECMA and NBS.

The ring came out of early work at Bell Labs and Cambridge University. Cambridge and its alumni invent about one new ring each year. Prime uses a ring, and since Apollo was founded predominately by Prime alumni, they too use another ring. Perhaps because rings can be built with large central controllers, IBM grabbed the ring, hence 802.5.

802.6 deals with metropolitan areas. Very high speed PABXs could provide the same function as the LAN, and hence should come under the 802 purview. It is imperative to have conformity at the higher levels. Is this 802.6 or do we need yet another standard?

Since one can obviously use fiber optics for building LANs we require a fiber standard, 802.8. Since several already exist, can't we either make one official or embed the media as an option within another standard?

The multiplicity of standards to switch information at modern, computer data rates causes us to avoid the essential problem of building networks and clusters. Almost every week a new incompatible LAN is announced. This is crazy!

The glib answer to the panopoly or lack of standards is gateways<20>. Virtually nothing is known about gateways except that building them is a craft - and roughly equivalent to conversions between high level languages (e.g. Fortran to Pascal). Building a gateway is about as easy as designing a train that can travel on different gauge tracks. (It may be fine if you can reach steady state, but it's the transition from track to track that is tricky.)

ELECTROMECHANICAL ASSEMBLY: DISKS, I/O, POWER, ENCLOSURES

The evolution of small disks and tapes has been very impressive and demonstrates the strongest case for standards. When Al Shugart started Seagate, his greatest concern was making sure a competitive second source industry with a common interface and form factor was available<21>. He used the same formula in creating the original 5 1/4" floppy disk form factors, standards and industries. The standardization process might be understood by studying this industry.

OPERATING SYSTEM AND UNIX

In 1966, a user could have a 300 baud Teletype using a phone line. By 1980 the speed had been raised to 1200 baud for a performance improvement of less than 10% per year, while the connect cost rose. This is roughly equal to the improvement for horsepower and cost increases in sports cars, not computers. By adopting UNIX a large part of our future systems development has been entrusted to AT&T (call it UNIXCO). Given the simplicity of UNIX, and need for much more rapid evolution, either a strong UNIX-compatible company will emerge, or IBM will take the responsibility for UNIX. It appears unlikely that UNIXCO will fulfil its role.

The UNIX phenomenon illustrates rule 6: Almost any standard is far more important than the "ultimate". Like many systems, the people who love UNIX are its many parents and those who grew up with it. The final clause of rule 6 also typifies UNIX: To make progress, we often have to regress.

UNIX evolved along these lines:

- UNIX was developed as a reaction by Thompson and Ritchie to MULTICS, the very large, joint MIT and Bell Labs project of the late 60's. It came out at about the time the book, "Small is Beautiful," was popular. They used a discarded DEC PDP-9 and evolved to use the PDP-11 in the early 70's.
- Since DEC didn't give away operating systems to universities, they used UNIX which was essentially free.
- No manufacturer provided source code to users. UNIX did.
- UNIX is by most measures a very simple operating system; to do useful work requires database access, special communications, and extra programs. Students and faculty could understand all facets of its internals and use because of its simplicity and availability. It was written in a very elegant, structured, high level assembly language, C, and as such could be modified. It was an excellent pedagogical tool. Universities embraced it and trained many students with it providing a large, future market.
- UNIX evolved to be used on other computers by being transportable. Provided a C language compiler was available, a team of people could move it to another computer system. Other early high level languages never quite succeeded in portability because of incompatible extensions to access the operating and file system. UNIX created the notion that it might someday be possible to have a complete system that was machine and manufacturer independent. Users like this idea.

- Chip makers with very small programming groups needed software and were use to adhering to standards. Small system manufacturers wanted system software and access to the DEC user base. IBM appeared to view UNIX as a way into the DEC's technical market. Thus, a standard is created which has almost everyone's support.

Much work is required to have a system that supports 80's computing concepts. UNIXCO must take the responsibility commensurate with their marketing. The notion of a standard is great. But it must be evolved more rapidly than any single manufacturer. It can, provided there is parallelism in the development using multiple organizations. If UNIXCO is the single company doing and blessing all the extensions, we have simply substituted multiple competitive companies with a single, behemoth! UNIX has to be evolved in a reasonable, not ad hoc fashion. This potential bottleneck may be the most serious problem we have in extending computing today.

Critical extensions include:

- higher reliability, greater performance and greater security;
- virtual memory; Berkeley, version 4.1 with virtual memory has been available nearly five years.
- special functions for real time and transaction processing; UNIX is being extended and adapted in incompatible ways by diverse companies. A clearinghouse to insure portability and compatibility of applications is required.
- a human interface and a modern human interface that is competitive with the PC or new PCs; UNIX was developed in the timesharing era using "glass Teletypes," as a result interaction is via one-dimensional, cryptic messages. Helpful, less cryptic interaction and multiple windows with fast interaction are critical today.
- multiprocessing; With the micro, multiprocessors are feasible, desirable and occurring.
- networks; Given UNIX's origin, modern communications capabilities for wide area networks should be demanded.
- fully distributed processing across a LAN to form LANCs. The University of Newcastle, Berkeley and others have implemented incompatible systems for fully distributed processing. Berkeley 4.2 is a good starting point.

LANGUAGE: INCLUDING EXTENSIONS TO APPLICATION LANGUAGES

The concern for UNIX, is paralleled by C, the the heart of applications portability, and must be standardized. A language for Artificial Intelligence (AI) is of great concern because of the next generation applications. LISP has been proven to be useful for AI applications.

LISP was designed about 1960, by John McCarthy. I was so enamored by LISP that the critical data access primitives were designed into the architecture of the DECsystem 10 in 1965 (still about the fastest LISP computer). LISP branched. One path went west via Bolt, Beranek and Newman alumni to Xerox, creating INTERLISP and its dialects. Many dialects evolved from the original MIT LISP: MACLISP, Zetalisp, NIL, SCHEME, TLISP, Portable Standard LISP and Common LISP. The later two vie for standards status. Franz LISP, GLISP, NIST are other dialects and extensions. Virtually everyone who works with a LISP compiler or interpreter creates his own language or extension. These languages are incompatible with one another and thus one can't benchmark, or extend the language in a compatible fashion using bootstrapping. Much work surrounding LISP is to make applications development easier. But given the number of dialects and extensions to ease development, is anyone working on applications?

In order to get on with the business of applying AI, we need some way of sharing information across the various different languages called LISP. A serious standards activity is long overdue.

In fact, the Japanese were so confused about LISP that they totally gave up and went to Prolog.

A COMPCON ON STANDARDS

Unfortunately, we can't go off and simply make rules for standards; instead a better understanding of the whole standards process is needed. A COMPCON devoted entirely to standards would:

- examine and prioritize critical standards; Certain standards such as LANs and electronic mail are relatively arbitrary and simply need to be frozen. On the other hand some care is needed to avoid constraining future creativity.

- establish responsibility and territoriality; Often too many groups are involved in setting goals and constraints, definition, review, test and implementation. Having fewer, competent designers always yields a far better system.

- establish goals and constraints; In many cases, efforts immediately digress to bit encoding without agreeing that a standard is necessary.

- understand the timing in the origin, present and future of a given standard; With the invention of new phenomena, it is pointless to discuss standardization until a breadboard has been made demonstrating utility. On the other hand several organizations have extended Fortran in incompatible ways to handle vectors because the standards group has considered Fortran a dead language; a standard is still long overdue.

- understand the effects and desirability of standards; Although most effects appear to be beneficial to both suppliers and consumers, the perception of almost every producer is that the "ideal state" is a monopoly.

- arbitrary "standards" to use in future radical research could aid rapid progress; For example, every dataflow computer has a unique, higher level language. Almost any dataflow language would let us encode algorithms and measure parallelism without having to build any special hardware!

SUMMARY

A model for the next generation has been posited that shows an ever increasing dependence on standards. The traditional levels of integration are now well defined through various industry and traditional standards. A set of rules which might improve the standards process were suggested. Five critical standards (silicon chips and wafers, the microprocessor, busses and LANs, Unix and lisp) were discussed in light of the next generation. Finally, a proposal is made for a COMPCON which would be devoted to particular standards and the standards process.

CONCLUSIONS

Having extolled standards now for sometime, there's a downside. A standard provides an interface, or target, by which similar systems can be compared.

> the focus and adoption of our standards has permitted
> Japan to become number one in computing.

In early 1984, the Lawrence Livermore National Laboratory kernel benchmark codes were run in Japan on the Fujitsu VT100, VT200 and Hitachi 810/820 at a rate of over 2 times a one processor Cray XMP<22>. The Japanese machines are evolved versions of a 20-year old architecture, the IBM 370, implemented with evolved 25-year old ECL circuitry, highly evolved 25-year old IC's, and expressed in 25-year-old Fortran. The Japanese used standards to increase productivity, they did not start by inventing a new architecture and the associated reprogramming. They built on the vectorizing compilers derived from the 15-year old Illinois' ILLIAC IV project.

The value of using standard interfaces, understanding the old and evolving it immediately increases output by freeing resources by higher productivity and lays down the real gauntlet for a new revolution.

DEDICATION

To the engineers who spend their own personal energy working on standards in an effort to create a more orderly environment.

1 M. A. Harris, Computer Makers Bet on Standards, Electronics, Vol. 57, No. 3., February 9, 1984, pp110-111.

2 P. Lemmons, The Standards Craze, Byte Magazine, Vol. 9, No. 1, Jan. 1984, p5.

3 P. Lemmons, An Interview: The Macintosh Design Team, Byte Magazine, Vol. 9, No. 2, Feb. 1984, pp58-80.

4 G. A. Brooks and F. P. Brooks, The Structure of the System/360, IBM Systems Journal, Vol 3, No. 2. 1964, pp 119-135. Reprinted in chapter 43<15>

5 J. Nelson, "802: A Progress Report, Datamation, August 1983, pp136-152.

6 R. P. Blanc, Local Area Network Standards, COMPCON SPRING 84, IEEE Computer Society, pp252-254.

7 R. M. Metcalfe, and D. R. Boggs, Ethernet: Distributed Packet Switching for Local Computer Networks, Comm. ACM Vol. 19, No. 7, Juty 1976, pp 395-404. Reprinted in chapter 26<16>.

8 Carver Mead, lectures and private communication.

9 Weitek, Floating Point Arithmetic Processor Manual.

10 Silicon Graphics, IRIS Product Reference Manual.

11 D. J. Frailey, Word Length of a Computer Architecture: Definitions and Applications, Computer Architecture News, Vol 11, No. 2, June 1983, pp 20-26.

12 S. P. Morse, B. W. Ravenel, S. Mazor, and W. B. Pohlman, Intel Microprocessors - 8008 to 8086, Computer, Vol 13, No. 10, October 1980 pp 42-60.

13 T. Kilburn, D. B. G. Edwards, M. J. Lanigan, and F. H. Sumner, One-level Storage System, Institute of Radio Engineers Transactions on Electronic Computers, EC-11, Vol 2, April 1962, pp 223-235.

14 M. R. Bhujade, On the Design of Always-Compatible Instruction Set Architecture (ACISA), Computer Architecture News, Vol. 11, No. 5, December 1983, pp 28-30.

15 C. G. Bell and A. Newell, Computer Structures: Readings and Examples, McGraw-Hill, New York, 1971.

16 D. P. Siewiorek, C. G. Bell, and A. Newell, Computer Structures: Principles and Examples, McGraw-Hill, New York, 1982.

17 C. G. Bell, R. Cady, H. McFarland, B. Delagi, J. F. O'Laughlin, A New Architecture for Minicomputers - The DEC PDP-11, Conf. Proc. AFIPS SJCC, Vol 36, 1970, pp657-675. Reprinted in Chapter 9<19>.

18 C. G. Bell and W. D. Strecker, Computer Structures: What Have We Learned from the PDP-11? Proc Conf. 3rd Annual Symposium on Computer Architecture, IEEE and ACM 1976.

19 C. G. Bell, J. C. Mudge, and J. E. McNamara, Computer Engineering: A DEC View of Hardware Systems Design, Digital Press, Bedford, Mass., 1978.

20 W. Stallings, Beyond Local Networks, Datamation, August 1983, pp 167-176.

21 A. Shughart, Private Communication, May 83

22 G. Michael, Notes of Conversation, LLNL, Benchmark Comaprison Between Highest Performance US and Japanese Machines for the 14 Standard Livermore Kernels, Jan. 1984.

GB8.20

4/8/84

SUPER-RANGE COMPUTER SYSTEMS

(Taking Maximum Advantage of the New Micros)

Economy-of-scale is invalid with today's computers **when** work can be done on multiple computers. Note the difference between today's mainframe and the technical workstation:

Computer	Speed	Mem. size	Price	Price/Mip	Price/Mbyte
	(Mips)	(Mbyte)	(M$)	(M$)	(M$)
IBM 3081	10	16	5	.5	.3
SUN WORKST'N	.1-.5	.5	.02	.04-.2	.04

BASIC PRODUCT DIRECTION

A **modular, super-mini sized computer**, being the most cost-effective for:

a wide price range 50 $K - 1000 $K; and

a wide performance range of 2 to over 100 Microprocessors.

The ultimate target is to be able to use ALL 100 Micros on a single problem, providing CRAY-power at a fraction of the cost.

BY-PRODUCTS

Several successful by-products are available on the way to the goal.

1. Fully independent machines can be best built in a much more cost-effective configuration than today's Workstations, Super-micros and Micro Clusters.

2. Highly reliable computers.

3. A large, multiprogrammed (conventional) computer which would be cost-effective for real time, transaction processing and timesharing.

THE ULTIMATE PRODUCT

Be able to process a SINGLE job in a Supercomputer fashion. Technical problems for science and engineering lend themselves to this form of parallelism. Parallelism is the basis for the high performance targets of the Japanese Fifth Generation Project.

FIFTH GENERATION COMPUTER FOR ARTIFICIAL INTELLIGENCE

BACKGROUND

A major impediment to the wider spread of of computers is the seemingly difficult problem of communicating with them in human terms. Recent advances in technology are now making it possible to consider building compuyters with liited aryificial intelligence, capable of performing routine wirk by dealing with people in natuaral language. However, the design of such computers is now only in the early stages, being severly limited by the few experts in the field, the high cost of experimentation, and the likelihood of several years before commercially profitable products can be delivered.

The Japanese government is underwriting a massive project to accomplish artificial intelligence at several levels, and posesed a threat to the US computer industry. While there is the possibility that the Japanese projects may fail or be sderuiusly delayed, they may be at the leading edge of artificial intelligence work being accomplished outside of a few centers in this country.

The objective of this R&D project to to intercept the Japanese work, and to deliver offshoot products ahead of avaialability of the Japanese.

The Project entails a hifgh degree of risk, has no assuarances that it can accomplish its goalsm on schedule of with the funds available, and may indeed result in loss of the entire investment.

If sucesful, the project has contracted with Encore Compyuter Corporation to commercialize any resuklting products and to market them in wlorld markets.

TECHNICAL FACTORS

Several independent disciplines are required to sucessfully solve the artificial intelligence problem:

1. Solve the problem of interactibg with people in natural languages;

2. Store and act upon massive amounts of data ("data bases");

3. Apply the general principles to specific tasks via a new technicak disciple - "knowledge engineering" - that varies from problem tom probem;

4. Advanced engineering to yield the potential for commercial viable products.

POTENTIAL APPLICATIONS

While there are a large number of potential applications for systems based upon artuificial intelluigence, the following have been shown to be both amenable to the approach and demandinbg of it:

1. Medical diagnosis - in face on increasing complexity of pharamcology, deepening and ever increasing medical knowledge.

2. Factory scheduling - now handled by computers but not in a mannner usuaklly acceptable by laymen.

3. Catstrophe control - modern process plants and power generating statyions cvan cause catrostrophic environmental damage in the even of unbexpected failure. Unaided human response is tyoppivallybtoo slow rto deal with these situations.

4. Advice Giving - a poterntially widespread application one of the major barriers to the widespread use of compyuters may be removed: using comnputers to act intelligently and to interact with people in a manner they are comfortable with.

We believe the Japanese Fifth Generation Computer effort based on logical programming for Artificial Intelligence applications will be successful. The next computer generation is characterized by tremendous speed through parallelism and ability to:

1. interact with humans in natural languages in various media including voice

2. store and act on large data bases (facts)

3. use knowledge engineering technology to solve a variety of problems (eg. expert systems for medical diagnosis, advice giving, factory scheduling, catastrophe control)

APPROACH

Build a series of products in an incremental fashion which are targeted at Artifical Intelligence:

1. basic workstation capable of interpreting both LISP (the current language of AI work in the U. S.) and Prolog (the basis of the Japanese work)

2. provide extensive hardware for voice and picture input/output

3. market the basic station into the emerging market

4. extend the hardware for higher speed execution

5. build a machine capable of parallel execution

CHALLENGES (TO COLUMBIA) IN GENERATING THE NEXT COMPUTER GENERATION

In June 1980 I helped Bill Miller and Joe Traub and others open Margaret Jacks Hall at Stanford. I hope with this convocation, we can open the CS building as successfully -- and maybe even somehow return to dedicate the Microelectronics and Computer Corporation, center in Austin someday which is another approach to applied research.

Even before the Japanese told us about the 5th Generation, I've been interested in CG's. This fascination surrounds the work on a C structures taxonomy essential for the C. Museum and understanding computer evolution. In the Museum we must have a way to contain and segment our ideas: by generations and by PMS structures whether they be components, computers or programs.

Richard DeLauer, of DARPA, claims we are working on the Nth generation, and I believe that the 5th Generation is already cast. LANS and powerful PC's are the main structures. So the issue now is what is the 6th Generation, what will it look like, and how can we continue to provide interesting computers?

Mostly, I think we all need to be concerned about the future; I'm going to dwell mainly on what we might observe from the past and present in creating it. I have been impressed with the Japanese evolutionary approach to engineering and how they had leveraged the world's Research. I also think they understand the notion of very long term process. On the issue of original research... it is crazy to think we are somehow creative and they aren't. Research is a luxury, not a necessity and they will teach us much. Now that they can afford it!

Last week AJP spoke at the CM, and in passing gave a number of his Pearls:

"If a computer understands English, it must be Japanese."

My concern is that the Japanese have already won. In the past, no one was interested in a race, contest or game. In fact our strength was that everyone was off inventing different kinds of games: board games, physical skill games, simple intellectual games like chess or complex ones like GO. NOW as a guerilla warfare army, we've been drawn into some sort of contest where we seem to be forced to compete! In short, we've been sucked into a contest where we have no knowledge of the rules, we have no notion of how to pick teams or whether the game is played with teams or individuals, whether more money or less money counts.

In the midst of all this, we have all types of forces moving people from institution to institution.

In a recent talk, Mike Dertouzous says there are 4 ways to beat the Japanese in the forthcoming new generation race:

. 100-200M to develop high speed computers with ai functions

. an open policy toward foreign workers in industry and academe

. tax credit for long range and in accord with national policy

. careful reexmination of antitrusts to permit consortia

He argues for forgoing the traditional short term gain at the expense of long term R and D.

While I concur, I am concerned:

. Where's a reasonable plan that would spend 100M? I've only seen one university based plan that's credible based on a record of accomplishment and with experience. What's worse... are there enough people to manage the research.

. open door is fine... but closed or open is probably irrelevent. Debating is time consuming and simply keeps us from working.

. there is no national policy or plan, so why have more R and D credit? I've seen this R and D credit go right to the bottom line to reduce spending in R and D and increase earnings. Similarly, most corporations aren't equipped to do either credible or useful research. Even A/D can be a conflict because so few managers understand the differences between Product Development, Product enhancement, let alone concepts of Basic and Applied Research. On the other hand, it's unclear to me that those engaging in research understand it that well either. There is no public understanding of these activities and clearly we can't manage the flow of ideas through the stages.

. antitrusts may not be the issue, figuring out how to work together and how to do these large, goal directed Research projects is hard and something that I'm afraid we don't know how to do.

We can learn from Japan about how to define, establish and then execute projects of this type. Here, I see the Fifth Generation effort as being 3-5 years ahead of us because they understand large scale, long term interacting processes and they have a plan that started in 1980 and based on the world's research. In contrast to their more directed approach, we have nearly 10 projects aimed at designing and building supposedly revolutionary but highly similar, single instruction, structured data machines here, MIT, Stanford and elsewhere which I believe will be pretty much a red herring...giving us only a few side-effects. I prefer to call these "structures for analogous computation." All, violate the historical notion of evolution since they start with a structure and not science and technology, but are loosely related to a problem. How many of these can we really afford (if the goals is to really manage them to completion with data gathering etc.)? Do revolutionary machines make sense? Are we prepared to run these 10 year, very high risk experiments? This involves incredible personal commitments. I hope to hell we can't afford them all, most likely we'll start them all and finish none. We should be able to learn from ARPA's speech research activity of several years ago, which I regard as highly successful when it was prematurely terminated.

WHAT IS A GENERATION? (Now that we know we need a new one!)

convergence of need that frees resources,

technology, science and ideas to build from, and

a basic structure.

Finally use will tell us that it's a generation after the fact. I can tell you lots about the first and second because of the lst 20 years but the others are

WHAT IS THIS ONE, I CLAIM THE FIFTH ONE WE'RE ENTERING?

Need is intercommunication, technology is MicroProcessor which in turn allows building: small shared, PC's , fault-tolerant structures, etc. and a new technology of LANs for intercommunication

This has created a product-fragmented, stratified by level of integration industry of many entrepreneurs!

A generation has a cyclic nature, much like a cyclotron. The concept to "do a machine" is injected into the accelerator at some stage... I'd like this to be needs driven to a large extent. Technology is the first stage, architecture and design are down stream, followed by the actual building. Software further accelerates the electron. Algorithms and use with critical evaluation (which we often ignore) provide the final stages... and of course by now, the particle has gone around once provided the people don't leave after the first six years. And now it is ready to be accelerated again and attain the critical energy level necessary to use or for going around again. For many generations, going around twice constitutes a new generation. The first time around a new structure is formed, and the second time around it is made useful and gains acceptance. Clearly the PC was like this: the very first PC, the LINC, cost about 40K in 65 and is in the Computer Museum, but not until 75 with the Micro was it really practical. It took about 3 trips around to reach an interesting energy level... which occured in 81 with the IBM PC. Now a trip around takes less than 2 years. This process is highly evolutionary with all parts of an industry providing energy to accelerate.

Note, the Japanese understand the notion of generations and evolution beautifully. The concepts of AI and AI workstations have existed for years in the lab. They started with plenty of cycles on a KL10, are making the very best hardware they can in a computation <?> to execute Prolog at a factor of 10-20x! In parallel, they're working on significant real applications and trying to develop the engineering discipline. Finally, they'll do evaluation on this, and will then go around again with a much higher performance station. They plan about 2 more trips around the cycle by 1990: use with critical evaluation, architect, build, deploy, then repeat the use and evaluation stage to start around again. Mostly I believe the important thing to do is start with use NOT architecture!

WHAT THEN IS THE 6TH GENERATION?

ai...and it will be wonderful or so we believe. One probelm is that I can only identify two "expert systems' that are in operation. Thus, it's hard to evolve a computer unless we have a model of what it is to do. A revolutionary machine is likely to fail--at least if it follows history. What would a new structure look like (usually a new generation has breadboards operating in one of the previous generations)... So where's a computer like this we can view?

For a revolution--I don't think we have a common view of the future. I believe the Japanese have a better view of one, albeit fuzzy. Therefore, our notion of a 6th Generation may not be realistic at all, if it violates the evolutionary, and needs-based notions of generations.

WHO ARE THE PLAYERS? THE JAPANESE, GOVERNMENT, UNIS AND INDUSTRY

Let me strongly urge particpation in the game plus definition of some new ones by everyone. I wish the effort were better directed though--much like the speech research project. Even a guerilla army needs some leadership. In the past, DARPA has provided much science for industry: Timesharing, speech (only partially completed, but more progress and better focussed than anything else)... it may be poetic justice that the person who cut this off now has to work in developing speech progress, it is also noteworthy that a DARPA researcher wrote the Speak and Spell product description and outlined the basic design, General computation by AEC to form Illiac I... and indirectly all of Cray's machines, Graphics, Packet Switching, and most recently VLSI.

Since the University played such an important role in the past, it's vital and even more necessary now.

Forrester, who headed MIT's Whirlwind, made several comments on building machines in Universities that still hold today:

"Experimental equipment merely for demonstration of principle and without inherent possiblity of transformation to designs of value to others does not meet the principle of systems engineering".

I've observed that this lesson should be a law that governs experimental machines: Unless a machine provides about an order of magnitude more power to the individuals who may use it than is available to them, there will be insufficient pull to attract users and test the basic idea. In other words, don't build toys.

WHAT CAN WE LEARN FROM BUILDING PAST MACHINES?

Harvard played a role in the beginning. Aiken was not particularly gratitious to IBM who actually built their Mark I, which at first glance one might consider to be an impossible machine, were it not for IBM's incredible engineering. In fact, this interaction proves to be grist for the computer history mills. None of the later Marks were near the state of the art in technology, and as influential. The most important effect was to train a large number of individuals who are influential in computing.

Columbia was influential too when Wallace J. Eckert got IBM to build the SSEC computer, a first, pre-computer generation machine composed of relays and vacuum tubes.

Eniac at the U. of Penn. was the truly revolutionary machine because it provided several orders of magnitude more performance than the Marks or the Bell Labs relay machines! Out of it came the stored programs concept. The work lead to Edvac, IAS, and the Illiacs directly and indirectly to the computer industry. MIT was evolutionary in structure, but revolutionary in technology with Whirlwind. TX-0 and LINC were even more successful. TX-0 took about - to design and then was in use over 10 years. The circuits were the basis of starting Digital. (Mostly, I believe a machine can only pioneer one thing.)

UNIVERSITY OF ILLINOIS MACHINES

Since I was just at Illinois last week, let me tell you what I learned from their machines and compare them with some of the observations of CMU:

Illiac I 9/52 operated

Concept-use: 4   Project start-use: 3   Concept-retire: 14

Use: 10   Use/lifetime: 33%

Conservative engineering for Aberdeen (Ordvac), copies were made for many other institutions. Design was spec'd at IAS. Implmentation project, not a total project of architecture, software, hardware, etc.

Illiac II 6/63 operated (3 years after 160, PDP-1, 1401, 1604, 7090, etc!)

Concept-use: 5.5   Project start-use: 3   Concept-retire: 9

Use: 3   Use/lifetime: 33%

New architecture, new technology, software took long. vehicle for timesharing, wiped out with commercial machines, Mistakes: Ge vs Si transistors, asynchronous logic (takes twice as long... something many folks still don't understand), didn't trust PCBs and used chassis, Memory was too small, too much to do yet it was far too conservative.

Illiac III isn't talked about. Use/lifetime: 0

Illiac IV Solomon 62, operted 11/75 @ 60 hours/wk

Concept-use: 12   Project start-use: 8.5-10.5   Concept-retire: 20

Use: 6.5   Use/lifetime: 33%

Processing rate: 250 Mips, Mp: 1/Mby, Mp.core: 2Mby MS: 139 Mby

Dan Slotnick, the designer of the Illiac IV a revolutionary machine commented last week:

Agreed that most machines come about through evolution and that's counter to the notion of original research which is suppposedly the basis of university rewards. The activity of building a machine for study entails a major amount of engineering, something that can be at conflict with science. ("Hamming once engineers science, math, than seat of their pants. Furthermore I am ? the conlift among the theory, engineering and AI parts of CS because it distracts and destroys.????

"Am convinced unis can't and shouldn't build machines. There are too many ideas. I used to have to stop the flow of ideas on interconnection every week. Too much bureaucracy. In a state uni it takes 90 days to get an IC. Too much democracy and too little discipline."

Larry Roberts who headed DARPA, claimed that it was absolutely clear that the machine should have been done with TTL and not ECL technology. People complain bitterly, but in the end, conservative technology seems to work out better. This is what I like to define as a tradeoff as either instructions per second versus instructions per month. The clock was 13 vs 25 Mhz and only 64 PEs were made instead of 256. I worried at the time that the clock for a simple machine by the same vendor was designed to run at 20 Mhz and actually ran at 10 so they may not make such an aggressive goal!

Contributions: got a number of good people working on parallelism at Illinois and elsewhere. Pushed the semiconductor ram somewhat faster than it might otherwise have gone. I IV did operate as the world's fastest machine for some problems and some time... until the Cray 1 came along in production. The fast rams were essential for the Cray 1. Most likely the biggest effect was to stimulate alternatives: TI's ASC, CDC's STAR and the CRAY 1.

With this, let me distinguish between 3 cases of machines: null, evolutionary and revolutionary. The null (make a copy of a previous Instruction set...), evolutionary, (do what is needed to enhance performance based on the knowledge of using the previous machines,) and finally revolutionary machines which are controversial. In the commercial world, the null is risky if the basis isn't there. The evolution such as the Cray 1, or VAX, taking advantage of all we know is probably the safest but still hard. The revolutionary machine is ... well revolutionary, and predictably bloody.

In 1949 Wilkes commented on the null case: "When a machine was finished and a number of subroutines in use, the order code could not be altered without causing a good deal of trouble. There would be almost as much capital sunk in the library of subroutines as the machine itself and builders of new machines in the future might wish to make use of the same order code that the subroutines could be taken over without modification"

Bottom Line about Illinois' machines

The Null Case, taking the IAS Instruction set turned out to be the most influential. Very good engineers were trained and theory of building machines posited.

Their evolutionary machine wasn't good enough. In fact, I believe that the tradition of providing vanity or proprietary instruction sets has cost computing (ie wasted more resources) than any other factor. There should have been significantly fewer machines. Watch what is happening with the IBM PC--finally there's some use, given there's a standard.

The revolutionary machine only had some side effects, but like all revolutions accompanied with much bloodshed. Unfortunately, like the case of Content Addressable Memories, Associative Memories, thin film memories, and CCD memories the world moves on an evolutionary trajectory, and rarely pursues two approaches for the same function!

Now they want to build a msmP at Illinois and the options:

1. Cheap labor of graduate students... brilliant, but unpredictable. Not recommended!

2. Professionals which create a second culture that is very hard to manage and basically unstable. But essential if you build the system. This is what has been done at the CMU projects.

3. Jointly with a company. A hardware/software split may be the right division of labor. This was used in the pc generations. Why

not do it again? It's being used at CMU with IBM for products. The Japanese companies build machines for the various universities, e.g. Tokyo.

4. As a seperate company outside the university and fueled by venture capital...now let's see if it's really venture. (TMC).

Now, let me go on to look at CMU's machines that were somewhat more evolutionary and which had more side effects and cost only a small fraction of Illiac IV to build.

CMU'S MULTIPROCESSORS

I have always been intrigued with multiprocessors, because an engineer likes to solve problems of performance by replicating a simple design intead of massive redesign. In fact I built an early 4P in 66, and have subsequently been involved in a half dozen other mP's. My only interest is trying to understand them so they can be applied to real use.

We started studying multprocessors at CMU in the late 60's, and I became intrigued with them when Bill Strecker's 1970 thesis showed how to compute the performance for p processors accessing a common memory of m modules.

This is the main reference work for multiprocessors, and I'll eventaully forgive the referees--in another 10-20 years--for rejecting the first paper because they didn't understand it or didn't think it ws relevant. There have been dozens of subsequent theses and papers on the subject, embellishing the topic, and they all reference the work. The Transaction just had an article on the subject. In fact, while I was in academe, I was finally successful in getting logic circuit switching theory mainly removed from the IEEE Transactions on Computers. Now, I find that switching theory is back where the object is to show how to switch a large number of processors (say 1000-10,000) to a similar number of memory modules. These papers have the same object: get someone tenure... the result is the same as the irrelevant circuit switching. Computing might go forward faster if we could simply grant the tenures and then have people go to work on the project. The miserable irony here is that I came home, looked at an interesting mP that's just come on the market and it has a switch that far outstrips the theoretical ones that could be operating in 4 years for the cases of interest. One researcher pointed out that he would get off the project of 32 if it couldn't be exended to 1000! These idiotic statements completely ignore the engineering nature of building a machine and mask getting on with the difficult job of building and perhaps impossible job of using the machine.

The issue is not the switch performance now or finding exotic switching structures simply: getting on with finding out whether multiprocessors actually work which is a combination of architecture, system software, language and algorithm design. I believe that if anyone can demonstrate that an ssmP of say 10 can work routinely in production, we can extend this to lsmP of 100 and then to 1000 rather easily.

In May 71, we proposed a ssmP of 16 processors for AI research which had a one gigabyte, very high bandwidth memory called C.ai. One of the students, an undergraduate, Tom McWilliams was in the seminar. C.ai roughly outlined the Stanford SI and SI, Mark IIA which is being built at Livermore. Unfortunately they became enamoured with building the world's most complex processor.

In August 71, a much simpler design was in place using the PDP-11 as a processor module. The project became known as C.mmp, a 16 processor Multiprocessor.

<u>C.mmp</u>

Concept-use: 5   Project start-use: 4.5   Concept-retire: 9

Use: 6   Use/lifetime: 66%

The project had 2 goals: a capablility based Operating System based on changing the PDP-11 and to examine the use of mP's. The addressing problem using the PDP-11 became a major issue and problem. Ironically, at least a few folks on the project didn't learn this. They went on to make the same mistake, plus a few others when doing the Intel 432. The project is well documented about what was learned in Wulf's book. Maximum speedups were hard to obtain. It is unclear why. I think because it wasn't used long enough!

Cm* a set of computer modules for building a msmP (50) in an open-ended fashion. First paper in Mar 73.

Concept-use: 4	Project start-use: 2	Concept-retire: >10
Use: >6	Use/lifetime: >60%	

This was an evolution on C.mmp, also, we forsaw the cluster of functional mP's that are present today and described them for adaptation in machines like Intel's multibus and Convergent Technology's Megaframe. It used the same OS concepts, even though any P could access any Mp, there was a preference to a local Mp, or that within a cluster of 10, and finally to memory outside the cluster. Thus, the machine is problem idiosyncratic. People began to understand this notion of the structure of computation and data with respect to particular physical structures. This is the key to these "structurally analogous computers".

There is still an incredible amount of science (and engineering) needed before these machines can work harmoniously in gangs of 50 without lots of work by anyone other than their trainers.

More interesting: evolution from C.mp in a project sense really paid off. Furthermore, the machine is still being used to collect data on parallelism. This is why it appears to me that CMU is so far ahead, say 10 years, in CS research.

For Multiprocessors, the progress has been slow. In each generation, I renew my optimism in the concept. I said this in the mid 60's with large computers and I said it in the early and mid 70's with minis, and now it just has to be true because the smallest unit is the very high performance processor with the characteristic that the smaller it becomes, the faster it goes.

Maybe there are reasons why mP have never been used:

the most likely, will we always find a simpler way using technology or instruction set to provide the same performance?

has engineering been too conservative?

the market not there?

too many other designs to try to avoid working on this?

too stogy and too compatiblity constrained?

or we simply don't believe users or compiler writers can cope? Clearly they can't if we don't try them. Happily there are several existing commercial machines at the small to medium scale level emerging with 4 to 32 processors, so maybe the technology will come.

If it does evolve, I would like to plead the case for universities to stay or finally get deeply involved even though you can buy them. Universities stayed remote from semiconductor research too long, and not until their involvenment was there the beginning of VLSI understanding.

As we work on parallelism, I regret that human organization theory can't help us except in an anecdotal fashion. More than a decade ago, Mel Conway wrote that people build computer structures like the human organizations they know. This explains why n people build n-pass compilers; IBM build hierarchically structured protocols like SNA; ARPA has to have a store and forward net independent of its users; DEC believes in democratic (anarhaic) structures like Unibus and Ethernet and multiprocessors and DECnet.

One researcher at Illinois commented that he could see merit in all sorts of physical structures like Illiac, Connection Machine, CAMs, Grids and Tree machines. It may be worthwhile trying various physical structures as you are doing here. To me these interesting physical structures may be premature because I don't think we have enough basic understanding of the notion of computational locality in order to map them into these particular, physical structures. I think this could be the basis of theory and building could be held off until the theory is built. Clearly they are not general purpose! Thus for the sixth generation, I would prefer to bet on highly tailored VLSI for performance like the geometry engine instead of these "general, highly special purpose computers." Therefore, the universities are crucial to develop the basis...my current bet for the 6th G.

If we could use human organization theory it might shed light on parallelism from structures that are connected together in exotic ways. It might also explain, like humans, why its difficult to get more than 6 processors to work together--unless totally top down directed with clear goals (like, take a beach or hill). (For now, I'm mostly only interested in the general case of multiprocessors because I don't know how to do it with the ultimate in connectivity... the memory, let alone by slow or restricted networks such as LANs, trees, hypercubes, etc.

At a time when Amdahl's constant of 1 byte/instruction has increased by at least an order of magnitude, I don't understand how something with a gop (giga-op/sec) can be content with a few megabytes! This kind of computation, I've called "Structurally Analogous Computation" because we're trying to make a physical analog of the computation. In a way, it resembles the very old analog computers that were patched together to solve particular problems such as network flows, simulation of all kinds, filtering of data as in a database, etc. I reiterate, I don't think there's enough basic understanding to do this mapping and hence build many machines.

OTHER PROBLEMS IN BUILDING REVOLUTIONARY OR EVEN INTERESTING C'S

Contrary to popular belief, I am quite concerned about the plethora of money which will mostly just cause excessive swapping and the erroneous, economies-based notion that money can be traded off for science ideas, and talent! The money comes from two sources:

1. The government. This acts to simply churn the small number of capable folks in universities and some labs, moving them from place to place. The nice effect is to raise everyone's salary. Yesterday's NY times contained a report of Aiken's quote.

Since the projects we're talking about are fairly large, they require professors to be very good project managers in a university environment designed for teaching. By being good managers, the reaction after a few years is simplpy: why work at somewhat lower pay and lack of freedom? (I enjoyed a very large pay cut to go into a university because I believe the issue was simply a tradeoff in the power/pay vs freedom plane. But with large projects, the freedom is diminished without the corresponding increase in power or pay. This provides a target for industry to scoop up kernels of the seed corn. In effect, the seed corn is really now poped corn. People have two choices: the established industry and becoming an entrepreneur.

2. The Venture Capital world which draws people from established industries and academe into what are often mundane or low tech products. For example, one high tech company started up in March and were shipping your generic 68,000-based UNIX product in 9 months, the standard gestation time. I recently saw a company of 4, build one board and assemble a UNIX product. Others build NOTHING at all but merely assemble.

Today the goal of a PhD is a chip, a program or algorithm, or system that is capable of starting a company. Recent examples include the Geometry Engine, the Timing Verify of Widdoes/McWilliams, the basis of the Valid Company, and the SUN Terminal, the basis of SUN Microsystems. So finally people can have freedom, fame and riches concurrently...but I doubt it.

Many folks believe that entrepreneurism is the way to beat the Japanese. Maybe it is because it unleashes such an incredible amount of focussed energy... but I wonder if the Japanese are going to feel threatened by 123 different kinds of ??? 68,000 based workstations! On the other hand--it is the basis for real applied R&D as with Amdahl's Trilogy Corporation.

I don't know what the final answer is, but we've got to get organized. Or in the words of Pogo, "we have met the enemy and he is us."

11/29/83 Tue 15:33:49

GB8.14ip, a program or algorithm, or system that is capable of starting a company. Recent examples include the Geometry Engine, the Timing Verify of Widdoes/McWilliams, the basis of the Valid Company, and the SUN Terminal, the basis of SUN Microsystems. So finally people can have freedom, fame and riches concurrently...but I doubt it.

Many folks believe that entrepreneurism is the way to beat the Japanese. Maybe it is because it unlgl(* 2(* 2$9.ju> +%% !+   #
#!#"###$#%#&#'#(#)#*#+#,#-#.#/#0#1#2#3#4#5#6#7#8#9#:#;#<#=#>#?#`#a#b#
c#d#e#k#l#m#h#i#j#f#g 0 < ? d!&!.!5!<!4!y"!"%"f"k"q"u"z

<title>5-YEAR AWARD TALK

<event/date>12/75,11/76,3/78

<>

<title>AMERICAN CAN TALK (TEXT FROM NSF)

<event/date>11/75

<>

<title>BASIC LANGUAGE TALK

<event/date>1ST TUESDAY-3/75

<>

<title>CAD SYMPOSIUM

<event/date>5/78

<>

<title>COMPCON

<event/date>9/74

<>

<title>COMPUTATIONAL SALES MGR. MEETING--MONTREAL

<event/date>8/73

<>

<title>COMPUTER ORGANIZATION & ARCHITECTURE--COURSE

<event/date>USC-LA-7/74

<>

<title>COMPUTERS IN EDUCATION--CHALLENGE

<event/date>MASS STATE COLLEGES-6/77

<>

<title>EDUCATION LECTURE--MANAGEMENT OF TECHNOLOGY

<event/date>SLOAN SCHOOL-4/77

<>

<title>ESG TALK

<event/date>8/77

<>

<title>IBM LECTURES - COMPUTER ARCHITECTURE

<event/date>8/72

<>

<title>INTERACTION AMONG TECHNOLOGY, PRODUCTS AND USERS

<event/date>DECUS-4/75

<>

<title>IRVINE--THE 2ND DECADE

<event/date>IRVINE-3/75

<>

<title>MCDOWELL AWARD

<event/date>10/75

<>

<title>MELLON AWARD SPEECH

<event/date>4/73

<>

<title>MINI TALK-1

<date>1/5/73

<event>

<>

<title>MINI TALK-2

<event/date>IEEE RALEIGH-7/74
<>

<title>MINI + EFFECT OF SEMIS ON -11 DESIGN
<abstract>EFFECT OF SEMICONDUCTOR MEMORY TECHNOLOGY ON THE DESIGN OF -11'S
<paper>Kaman & Bell
<event/date>10/73
<>

<title>MUSEUM SLIDE TALK--COMPUTER GENERATIONS
<event/date>8/75
<>

<title>NAVAL UNDERWATER SYSTEM
<event/date>12/76
<>

<title>NAVY TALK ANAPOLIS
<event/date>4/75
<>

<title>N.E. TALK
<event/date>1/76
<>

<title>NEREM
<event/date>12/74
<>

<title>NET TALK
<event/date>10/74
<>

<title>OBJECTIVES--HOW TO SET UP

<date>9/11/72

<event>PROGRAMMING TECHNOLOGY SYNPOSIUM

<>

<title>PDP-10 MARKETING

<event/date>8/72

<>

<title>PDP-11 TALK FILE

<date>67, 68, 69, 70, 73

<event>

<>

<title>QUANTUM SCIENCE TALK

<event/date>3/28/73

<>

<title>RTM'S

<event/date>IEEE-3/71, YALE-10/72

<>

<title>STANDARDS & PORTABILITY FOR SOFTWARE PROBLEM

<event/date>NRC-1/76

<>

<title>TECHNOLOGY--3 YEARS

<event/date>MID-ATLANTIC REG. MGRS.-7/74

<>

<title>VAX-11 TALK OUTLINE

<event/date>ANNUAL MEETING-10/77

<>

THE SILICON LEVEL-OF-INTEGRATION AND THE NEXT COMPUTER GENERATION

The silicon chip and wafer is an important level of integration that requires wide-scale standardization-- provided we believe that future generation systems will be formed from single chips or wafers. This could be the "real" Fifth Computer Generation.

Since semiconductor processes have traditionally been the corporate jewels of semiconductor companies, the wafer and chip level is not well documented, publicized or standardized. Yet, it is safe to predict that the silicon wafer or custom chip is likely to be the basis of the next computer generation. Some computer systems will be a single chip with 1 to 10 million transistors. Of course, most chips will continue to come from semicomputer manufacturers as a "standard" or combinations of "standards" such as microcomputers, peripherals and memories.

Creative new products will come from the Silicon Foundry Industry that Carver Mead advocates - and this requires substantial standardization. In effect, systems will be compiled onto a chip or wafer. Weitek, is an example of this new kind of company that takes algorithms and embeds them in silicon - VLSIzation. Another example is the workstation product, IRIS, from Silicon Graphics. IRIS uses a dozen 75,000 transistor chips which Jim Clark calls the Geometry Engine and computes at a speed of 10 Megaflops--roughly equivalent to a CDC 7600 computer. In this way IRIS out performs, by a factor of several hundred, the other 100 or so standard UNIX workstations! Similarly, conventional semiconductor companies search for standard chips that convert software to silicon. One can envision radically new special chip-based systems which operate on pictures, voice, and mechanisms.

Due to lack of standards in foundries and CAD systems we are far from being able to realize the scenario of a Silicon Foundry Industry. Standards are essential for all user-specific gate array, standard cell or fully custom chips. It's distressing that we still have no standards for specifying gate arrays; custom PLAs and ROMs took too long to standardize. A few interfaces for this industrial structure include:

- specifications of structure and behavior, including simulation and timing at all levels
- physical information at all levels including processing wafer masks (eg. CIF)
- control of foundry processes, especially if processing steps become optional
- chip test, including automatic generation of test data
- chip assembly and packaging including bonding and multi-chip interconnect

For CAD, the development of standard interfaces to languages and databases that are communicable via networks must be targeted. It might be desirable to standardize the specification languages; I can't identify any benefit of having syntactic differences. Agreeing on interfaces doesn't limit the competitiveness or creativity of any CAD company or foundry, it means users don't have to learn many systems and languages for the same function, or convert data formats. Standards would let users mix and match different CAD systems in a completely flexible fashion. Syntactically idiosyncratic editors, timing verifiers, simulators, design rule checkers, etc. give no real increase in user power. Use would expand much more rapidly because buyers wouldn't be forced to make critical long term decisions, with no way to exchange data to other systems. This is completely analogous to the pre-Cobol / pre-Fortran era in the late fifties when all the users rebelled at every manufacturer providing a unique language. The rebels designed COBOL, the first, oldest, and largest used of the standard languages, because there was no reason for different languages!

In CAM, the user is also faced with a fuzzy and perplexing interface to the process from masks to tested components.

The semiconductor companies and foundries (eg. Semiconductor Industry Association), CAD companies and users (e.g. the Microelectronics and

Computer Corporation) could affect change now so we can have the next generation.

An interesting first step would be to devote an entire technical conference such as Compcon to exploring this proposition that we need standards. What do you think?

Gordon Bell

Chief Technical Officer

Encore Computer Corporation

15 Walnut Street

Wellesley Hills, Massachusetts

GB8.27

STANDARDS: THE BASIS OF THIS GENERATION

Gordon Bell
Chief Technical Officer
Encore Computer Corporation

This generation is based on a compleley product fragmented industry that is stratified by levels of integration. Entrepreneurial energy is a major driving force. Short product gestation times and the rapid evolution require formal and de facto standards. What are the goals (product targets) and constraints (the standards)? What are the roles of the various organizations at the various levels of integration?

Post v N Computing: A 10 YEAR, DIRECTED RESEARCH PROGRAM AND NATIONAL FACILITIES AIMED AT PARALLEL PROCESSING

A Draft Outline*
&
Invitation For A Proposal(s)**

Gordon Bell, DEC

George Clark, Harris

Bruce Delagi, DEC

Sid Fernbach, Consultant to CDC

Bob Lillestrand, CDC

Red Phillips, Univac

13 August 1982

* Substantive references to previous and ongoing work and bibliographic references have been omitted. While we believe the general direction is correct, specific tactics such as the applications to focus on, will be subject to change with the final proposal(s). We now solicit both conceptual and detailed critiques.

** The final proposal must come from the program group dedicated to produce the results. Thus we solicit:

- o sites
- o individual researchers and a program director
- o applications and other research projects

TABLE OF CONTENTS

3. Rough Budget

OVERVIEW

This proposal began as an exercise by positing a computing environment we believe is attainable in 10 years based on parallelism uncharacteristic of the single, von Neumann machine and then asking ourselves:

> Are we doing anything significant to understand and build this environment?

The result was overwhelming:

1. most industrial research appears to be aimed at incrementally improving today's products and processes; while

2. academic research is aimed at basic research and the mechanism of getting grants, producing papers and Ph.D's.

The objective of this program is to develop the technology and build next generation computers by establishing several National Laboratories for computer science and engineering research within the U.S. military, academic and industrial community. This technology is essential:

1. for defense;

2. to improve the declining computer and semicomputer part of the U.S. Information Processing Industry which now constitutes and supports much of our economy directly and via exports; and

3. as a basis for much of the 21st Century Industries.

The declining technology position in the computers and semicomputer industry is a national crisis. As such, this necessitates these unique aspects of the program:

1. collaboration among national science, defense, university and industrial applied research, often called technology, in a fashion not unlike the VHSIC program;

2. National laboratories so that limited machine and people resources can be shared, unlike the VHSIC program;

3. a large, fast network including access both for experimentation and to extend the program to other research sites;

4. construction of prototypes by industry for evaluation within the research community;

5. technology transfer by industrial residents at the laboratories;

6. tighter coupling of application (need), architecture, construction and use by co-location in order to rapidly engineer, build and test ideas. This speeds up migration of ideas to use by applying engineering resources earlier.

These facilities will be the hub of a goal directed research program aimed at new VLSI-based, highly parallel computing structures. Parallel processing systems, including: specialized processors and hardware algorithms, multiprocessors, multicomputers, dataflow and high speed local area network based meshes will be built and evaluated. Evolutionary projections show a performance increase in processing of only a factor of 3 (Fig. 1) to 11 (Fig. 2) over the next 10 years. In contrast, the Japanese Fifth Generation Research Project, is aimed at producing high speed and parallel computers with a factor of 100 to 1000 more computing power for conventional and Knowledge Based computing systems by 1990 (Fig. 2).

Another major goal of the program is VLSIzation, the ability to transfer an algorithm, simulated within the computing environment, to VLSI limited only by the foundry time in much the way programs are currently compiled. By it's nature, this structure adds inherent parallelism to computing. The national facilities would also support the goal that computers would do a substantial part of the VLSI design. Research in the parallel computing structures we target will rely on accomplishment of these goals.

A new computer generation is marked by concurrence of technology and needs causing a new computing structure and resulting in new use. We believe this driving need is for the ability to transmit, store, and process (understand) the same information as people, including voice, natural languages and images. Images are a major data type of this research program because of the links with people. The research need is driven both by hardware and technology and by the potential of Knowledge Based Systems requiring much higher performance. These must be coupled with signal processing to assimilate voice and images.

The program would be organized in 3 phases, covering roughly a decade, in order to focus the work in a timely fashion. Generations have historically taken 7-10 years and consist of two periods: specification and construction; followed by use and evaluation. The immediate installation of the most powerful, high speed network of general purpose computers would start the program in the use and evaluation phase. Results based on application of this facility would then be applied to produce new VLSIzed computing structures by the end of this first phase. The second phase would apply these newer structures, forming the basis for new designs in the final program phase.

MOTIVATION FOR THE PROGRAM

The U.S. lead in the combined Information Processing Industries is now declining relative to Japan. While there are many reasons for the decline, these are noteworthy and represent the motivation for this program:

1. The U.S. (and World) funding for basic and applied research is large. This mechanism produces far more results than can be applied.

2. There is NO U.S. effort or policy aimed at systematically examining the basic research results and refining them so they can be applied to products. The cost to do applied research on even a small fraction of the basic research is usually far greater than the original work and is well beyond the scope of a single company or a laboratory. Furthermore, most laboratories doing research can only carry ideas to the paper stage because of the engineering nature of the final stages to build and test the idea. Thus, overfunding research relative to applied research means a "spilling" of knowledge that forms the basis of a significant industry.

3. U.S. companies have not worked collaboratively to develop these technologies because of legal and cultural reasons.

4. U.S. industry has been especially short sighted in its funding of this phase of research. Now, many short term, mundane product opportunities (eg. another Z80 + CP/M based personal computer) exist to attract resources resulting in further decline. This is further fueled by the venture capital market and increased R&D tax credits which in turn produce even more mundane products.

5. An inadequate supply of people and equipment exist to carry out the work in industry and the research organizations.

6. A research program aimed at parallelism requires interaction and co-location with a user community.

We marvel at the effectiveness of the Japanese collaborative research programs and believe we must emulate them. Both France and the U.K. have established programs aimed at the next computer generation. Note the past and present programs in the Information Processing area:

1. Pattern Information Processing- voice and vision

2. VLSI- improved processing characteristics (eg. 64K and 256K rams resulted in a 2 year lead over U.S. industry)

3. Supercomputers- high speed technology

4. Optoelectronics- just established

5. Standard Minicomputer for NTT- Fujitsu, NEC and Hitachi

6. Fifth Generation Computer- Fujitsu, NEC, Hitachi, Mitsubishi, Matsushita, Oki, Sharp. ICOT Lab and 10 year program were established. The first phase builds Relational Database and Prolog machines.

7. Local Area Network standards as part of the Fifth Generation.

8. Next generation research and technology program.

THE RESEARCH PROGRAM CONTENT

RESEARCH OBJECTIVES

This work is undertaken with the expectation that the confluence of the disciplines of parallel processing applied to image processing, and knowledge engineering, and implemented using VLSI will prove fertile. It, and the resulting VLSIzation process, that of first understanding specific algorithm and tasks and then VLSI'ing them, may well be a major characteristic of the next generation computing systems, which the Japanese call the Fifth Computer Generation.* The establishment of a quasi-competitive, but coordinated program of research using common research facilities is intended to stimulate a national understanding of such systems and their potential application.

The work is aimed at a fundamental understanding of parallelism and its application to a class of problems critical both to the growth of the computer industry in this country and to the maintenance of a preeminant US position in intelligence based military systems.

ESTABLISHING AND USING THE FACILITIES: PHASE ONE

The short term focus will be on installing and applying parallel approaches to image procesing and logic/circuit/process simulation problems, especially dataflow. We think it is vital to understand the range of dataflow from theory to practice across a wide range of applications. In its simplest form, dataflow can be viewed as a formalized, generalization of pipelining that is conventionally used for graphics and image process. In its more general form, dataflow looks appealing for logic simulation, signal routing, and conventional array processing type tasks where a great deal of parallelism exists, but cannot be exploited due to the difficulty of expressing algorithms in conventional languages. It is indeed possible that dataflow-specific machines <u>will not</u> exist, instead dataflow languages will enable programs to be written for large, multiprocessors. The centers will be based on a high performance local area network to interconnect the central machines, including:

- supercomputers,
- experimental machines (dataflow and conventional multiprocessors and multicomputers), and
- the CDC AFP.*

The AFP will operate with <u>fixed microprograms</u> to simulate several computer structures including dataflow computers. This will enable researchers to begin now and to understand the limits and use of dataflow architecture, for example. These efforts must be put to the test of representative applications in order that the tradeoffs discovered be relevant to solve.

* One of us (GB), believes that the current generation, number 5, is based on powerful personal computers interconnected via local area networks. The Japanese are working on the sixth generation, beginning in the late '80's.

It is essential to have real applications on which to "benchmark" various designs. The following applications cover some of the possible important military and industrial problems: scanning electron microscopic image enhancement, automated assembly inspection, target identification, digital system design and construction (eg. logic simulation, routing and IC signature analysis). The actual applications should be made firm with final proposal.

While the initial results have focused on using a dataflow architecture to examine its limits, the network and facilities we envision are much more extensive and will be used as alternative ways of computing.

PROGRAM DEVELOPMENT OF THE CENTRAL FACILITIES

It is expected that the central research facilities will be enriched further over time by including, as additional research tools, the fruits of the aspects of this program particularly focussed on realizing more powerful forms of processor interconnect and process (or operator based) intercommunication. It is expectd, further, that several realizations of parallel solutions to specific application image processing problems will be implemented (in VLSI) and included in the central research environment.

UNDERSTANDING PARALLELISM: PHASE TWO

In the middle phase of the program here proposed, the principle results will include a deep understanding of the dimensions and metrics that describe the space of parallel computing - costs, performance, programming expense, and reliability. The proposed facilities provide a rich set of alternative realizations for parallel computing - ranging from tightly coupled multiprocessors to conventional Local Area Networks. We do not believe that the kind of interconnect for switching is a particularly fruitful area of study because it is really an economic issue that shifts with technology, regulation, market demand, and supply. Thus, the goal is to provide various structures for evaluation and use very rapidly, but not to research the interconnect possibilities!

END POINTS

Expert systems and knowledge engineering efforts are expected to yield their most important results in the last phase of the program. Significant milestones are established throughout the research effort: discerning the computational (and data management) primitives underlaying current rules-based expert systems languages, establishing an effective integration of image and symbolic information into a knowledge base (consistent with the data management primitives noted above), realizing a VLSI implementation of a highly parallel, post von Neumann computer structure for expert systems, trying it out on (say) a SEM analysis problem, a fully automated VLSI design, and finally on an expert system for (semiconductor) process/crisis management (or threat evaluation and reconnaissance mission). These will, in turn, provide the understanding needed for a second VLSI implementation of the expert system engine above.

SINE QUA NON

As a necessary ingredient of effective VLSI implementations supporting the research goals of this program we need the 1990's VLSI equivalent not merely of the Guttenberg Press but of the linotype machine and the automatic typesetter. The process would be completely controlled by an individual or small group. The most important element of this program then is the development of the capability for (fully) automated VLSI circuit design from representations of parallel algorithms simulated on the parallel computing facilities proposed. At first, this will likely be by means of both conventional supercomputers and the dataflow machine simulators running at the central facility.

The automated design capabilities will be made to stand the test of real use in VLSI implementatins of (at least one) dataflow machine. The design of this machine will be based on the measurement and analysis of simulated dataflow machines running applications as noted earlier. These design capabilities will be also tested in VLSI realizations of IC signature analysis dataflow algorithms and the mobile object identification and tracking projects implemented previously. The culmination of efforts in image encoding and compressions will be a special purpose VLSI processor chip that provides full motion video-conferencing within the bounds of a 56 Kbps phone line, for example.

A FACILITY TO UNDERSTAND AND EXPLOIT PARALLELISM

New computer applications usually result from having new, higher performance computers allowing solution of problems that previously were computationally intractable. Performance increases in computing come from two sources: technology improvements and increased parallelism. This program is aimed at understanding and exploiting parallelism to gain performance.

VLSI contributes to parallelism in two ways.

> First, commodity processors allow the low cost construction of the most cost effective systems. That is the Mips/chip of microprocessors far outstrips the densest, high performance ECL gate arrays.
>
> Second, VLSIzation is an inherently parallel process - standard algorithms are off loaded.

To date, attempts to improve performance through highly parallel structures has been relatively disappointing. We believe the major reason for this lack of progress is the high real and personal cost to build and evaluate parallel structures. This program supports systematic research and development on the following alternatives. In this regard, we posit this fundamental hypothesis: <u>in order for a new computer structure to be attractive to a user, and hence ultimately developed and exist, it must offer an order of magnitude improvent in performance over his current method of computation.</u>

SPECIALIZED PROCESSING (AND VLSIZATION)

Historically, an order of magnitude or more speed improvement has resulted from looking at the execution times of particular work and then building hardware to carry out the function. VLSIzation is a realization that this evolutionary process exists and is an attempt to formalize the process.

Some examples of "off-loading" using special function hardware:

1. Floating point hardware versus a software interpreter

2. Channels, I/O Processors and I/O Computers versus interrupt and hardwired I/O

3. Display processors

4. Array Signal Processors

5. Front end (communications) and back end (disk, file and database) computers

A need, resulting from a computation on a particular kind of data occurs.

This need is then a requirement for a new computing structure. The function is then "off-loaded" in specialized hardware that operates in parallel with the general purpose computer.

By having a general purpose, very high speed system, the resulting, specialized structures can be totally simulated before they are committed to VLSI designs. In this way the designer can interact with the structure in a quickly iteractive fashion instead of waiting at each iteration for fabrication and system (re) integration.

MULTIPROCESSORS

Every time a new computer class is formed, there are strong arguments to build multiprocessors for performance reasons. Invariably, others build higher performance Uniprocessors at the same time and deliver more power via the strictly sequential approach. Multiprocessors were proposed by the early 60's, with Burroughs probably delivering the first one (B5000). By the early 70's Burrough's, CDC, DEC, GE, IBM and Univac had all built 2 - 4 processor multiprocessors. Unfortunately, these were either used in an asymmetrical fashion, or at most they were used in an ordinary multiprogramming environment. In no cases was parallel processing of a single task provided.

In 1966 Lehman investigated parallel processing of a single task with a 16 procesor multiprocessor and showed that for various tasks speed-ups were possible. By 1975 two 16 processor systems were built by BTL and at CMU. The CMU system was predicted on the 11/40 minicomputer, as a way to afford the construction, and speed-ups of up to 10 were observed in various algorithms.

CDC's Advanced Flexible Processor is an ideal machine to investigate the use of multiprocessors and multicomputers since the interconnection among the computers is via very high speed local links (ultra LAN) and shared memory. It can be used in many ways, including:

1. a 16 computer multiprocessor;
2. a 16 processor multiprocessor;
3. a fixed, intrpreter for particular structures (eg. dataflow); or
4. a particular, dedicated pipeline processing configuration (eg. image processing).

Several laboratories are building systems with up to several hundred microprocessors.

LLL is building a multiprocessor, the successor to the S1, with 16 supercomputer class processors. As soon as the processor's available, it should be extended to the multiprocessor case for evaluation, since the processors are both tightly coupled and have very fast inter processor communication mechanisms. This should be within the next three years.

DENELCOR is offering a 64 processor multiprocessor which requires investigation. We strongly recommend the installation of this machine in the facility in order to work on the multiprocessor problem.

Recently, Schwartz, et al at NYU has proposed the Ultra-Computer, a multiprocessor with up to 16,000 VLSI microprocessors. Just as soon as we can operate a reasonable number of processors together, construction should begin on this very large multiprocessor.

It's safe to say that one can produce conventional parallel processors which should be able to deliver up to a factor of four, for specially coded programs. A factor of 10 is possible, but there has to be a significant amount of research to make this automatically possible. Studies continue to indicate vast amounts of parallelism in algorithms that we have no way of exploiting.

We believe that the optimistic (Fifth Generation) projection for computing power speed-up over the next decade could be accomplished simply and entirely by parallel processing using multiprocessors and not by semiconductor and packaging technology if a significant effort were applied! Undoubtedly the dataflow language is an important part of this effort to represent, control and thereby exploit this form of parallelism.

MULTICOMPUTERS

Very little has been done formally with arrays of tightly coupled multicomputers where independent computers (Pc-Mp pairs) operate independently and communicate with one another by sending messages. By 1980, CM*, a multicomputer system based on the LSI-11 microprocessor with 5 clusters of 10 computers was constructed, and speedups of up to 30 were observed for particular problems, including speech recognition. Because there is less interconnection among the computers, it is more difficult to predict the performance: the algorithm has to be carefully partitioned across computers rather than distributed in memory.

In addition to AFP, we believe that other multicomputers should be constructed and used, particularly those with several hundred computers. Here, we would support the construction of several, (say 6) different multicomputer alternatives.

DATAFLOW ARCHITECTURES

Although many dataflow computers have been proposed, only a half dozen computers have been built. The performance of dataflow computers is not understood, although the use of dataflow graphs and languages to express parallelism is promising. In particular, dataflow appears to be most useful in expressing signal processing operations. For example, the AFP is programmed using a dataflow-like representation for image processing tasks. Individual computer modules can be assigned to various processing stages of say a digital filtering task. The AFP also appears to be ideal to simulate static dataflow architectures and their application. It would be microprogrammed to be a general purpose dataflow machine using separate computer modules in a functional fashion: matching store, switching, processing, and i/o.

ULTRA-, FAST-, AND CONVENTIONAL LOCAL AREA NETWORKS

Local Area Networks, LANs, are systems which normally allow the physical distribution of functional, server components to cover a local geographical area (eg. a building, or campus). The functional servers roughly correspond to various parts of a shared system: person servers (computing workstations/terminals), file servers, print servers, and communicatins servers. The communications is via message passing protocols. While the curent 10 Mbit/sec LANs are relatively slow, they are well matched to today's, slow terminals, personal computers and for intercomputer networking.

Researchers have also posited that LANs can be used to provide high performance, parallel processing. We too believe higher speed LANs are the backbone interconect architecture for new computer structures. The higher speed, 100 Mbit/s LANs will be the basis for interconnecting functional computers in a hierarchy as shown in the facilities section (Fig. 3).

We view the Ultra-LAN as a major architectural component and standard for truly fast, highly parallel structures of this next generation. Note that the ring that interconnects the AFP provides transmission at about 2 Gbits/sec for each computer node connected for the tightly connected computers. Thus, the AFP would be used for some studies of this type of LAN-based architecture.

The purpose of the hierarchy of three LANs is summarized:

Ultra-LAN	2 Gbits x p	AFP's processor intercommunication; as first basis for an ultra-LAN architecture
Fast-LAN	100 Mbits	Facility computer intercommunication and center to remote sites, forming a single cluster
LAN	10 Mbits	Individual workstations to form centers

PARALLEL PROCESSING FOR KNOWLEDGE BASED SYSTEMS

It has not been widely agreed that Knowledge based Systems can exploit parallelism. For Rule Based Systems, it is believed that many rules can be evaluated in parallel. The research will be aimed at first answering the question, and then simulating and evaluating the resulting structure. AFP might be used to simulate such a structure, provided this approach looks worthwhile.

THE RESEARCH PROGRAM FORM

ORGANIZATION, DIRECTION AND RELATIONSHIP TO ONGOING RESEARCH

A program office, together with a board of directors would contract the research in a fairly structured fashion. While research of this type is not commonly done today in computer science, we believe it can and must be done effectively by a joint industry and computer science research laboratory efort. Industry can be effective at providing facilities and systems that have been traditionally absent from the research laboratories. In effect, this is the major motivation for the proposal.

A major goal of the research project is to provide a large infusion of computing systems to support existing, more basic and unstructured work, including robotics.

The purpose would not be to change the nature of the existing unstructured research to be highly focused and goal directd, but rather to provide additional resources so that both the structured project and unstructued work could co-exist and complement one another.

The centers would be aimed at very similar research targets in order to get the benefit of "friendly competition". Similarly, several approaches would be examined within a center. This approach was successful in the mid-70's in speech research and should be the "model" direction. However, the speech research resulted in few, commercialized industrial or military applications, because the research coupling between academic and industrial research was poor. Unfortunately, the final transfer phase of research was terminated before the program ended.* It is this gap between basic research and applications research that the program is fundamentally addressing. It is interesting to note that NEC had an advanced development operating separately, but concurrently with the ARPA program. The result is that NEC provides recognition products.

We would hope that a better model to follow is VHSIC. It is crucial that the participants be able to exploit the technology for commercial and military applications propitiously. Unlike VHSIC, we believe that the work should be done at a few sites with movement of personnel.

THE PROGRAM OFFICE

The fabric of this research is a fairly close weave. The environments are, indeed, established anticipating that unexpected leverage and collaborations will yeild significant results not included in the program plan. However, it is precisely the existance of a structured program and the interrelation of its several work flows that will enable this to occur. The program office is responsible for the successive development of the fabric using resources as it can find them and coordinating efforts so work can easily build upon what came before.

* Personal communication with Allen Newell and Raj Reddy at CMM.

The program office will set adequate standards so that ideas meet no unnecessary boundaries between the workers and the worksites in this program. Early, stable agreement on the common rules, language, workstation, the network and the general computational support structure will be among the most important contributions of the program office, the goal is to use this commonality of interface to allow pyramiding of work - being careful not to pyramid risk.

The program uses applications to test ideas, and uses realizations of those ideas to build the next generation applications. It even uses these applications themselves to acomplish future generation realizations fueling the next cycle. The central facilities are the place that application tools for realizing ideas, the realizations themselves, and the applications for testing ideas all come together. This must all flow forward rather than bottleneck into a deadlocking interdependencies. The opportunity and expectation for people to build on each others work as it becomes available is the key. In the natural uncertainties inherent in this ambitious program of research, there must be enough alternative paths so clever people can use their wits to find a critically helpful piece of another's work or another's facility wherever it may turn up.

The program office must have the ability to facilitate the construction of important engineering breadboards so that systems can be rapidly built and evaluated. We envision utilizing the industrial sponsors for this breadboarding.

The program office is deliberately kept small to force most standards to be developed collaboratively with the groups doing the work. The program staffing for the parallel computing facilities is very light in the expectation that site personnel will be provided by the host institution. The Budget Table, Appendix 3, provides a more detailed breakdown.

PROGRAM BENEFICIARIES

The program was conceived in order to improve this flow of basic and applied research into industrial research and eventually into products. The main beneficiaries are those who use these ideas to eventually build products. Products will not come directly from this program.

On the other hand, virtually everyone will benefit by the program:

1. the U.S. technology will be drastically improved - thereby improving defense and the economy;

2. the researchers will be more effective and productive by having more meaningful work;

3. certain research will be published; and

4. researchers will still migrate from the coupled programs, being attracted by venture capital, and build higher technology products.

TRANSFERING THE TECHNOLOGY

The most effective means of technology transfer is through the transfer of people. Program sponsors will each have the right to place people in each project of the program. It is expected that assignments be for a three year interval and that the assigned person return to the sponsoring organization prepared to produce the competitive products of the late 80's.

To insure a co-operative working environment among the members of a project team, intellectual property rights for the work done as a team using the facilities of the host institution will be controlled by the policies of the host institution. However, each program sponsor will have the right to a non-exclusive license at reasonable terms.

A major part of the transfer will occur when the sites and industry collaborate on fabricating a design that a site has specified.

With VLSIzation, chips produced as part of a research project would be licensed to the sponsors. The "rights" to chips and software produced as part of a research program are indeed not clear at this time and vary among the institutions. This area would have to be worked out between the institution and the program.

Other mechanisms for technology transfer include sponsor access to prototypes, distribution of published technical reports and invitations to program seminars.

Seminars will be held quarterly for program sponsors with invited speakers from universities, government and industry.

In inviting speakers the organizers of the seminars will have the freedom to draw on the wide range of topics encompassed by the program, including:

- Pattern and image processing applications
- A. I. algorithm research
- Multi-processor architectural developments
- CAD/CAM software systems
- VLSI design process advancements

FACILITIES

HIERARCHIES OF AREA NETWORKS

The program would be organized around at least central research computation centers containing a variety of production and experimental computing systems (nodes) interconnected via 100 Mb/s links and forming the central facility for a hierarchical set of closely coupled, high performance, local area networks. The centers will be linked to several campuses via the highest available links so that they could be used in a clustered fashion "as if local" computation centers.

Each site would contain supercomputers, AFP's and experimental computers.

ARPA-NET II

In effect, we're proposing ARPA-net II. This must come into operation relatively soon, to be used to interconnect the more remote research to the centers. High bandwidth, such as several video channels would be needed to avoid limiting the interaction between sites. Here, the goal would be to provide only millisecond delays between processes operating on separated machines.

VLSIZATION FACILITY

Since the projects would be designing many VLSI chips, the facility would need a way to build state of the art VLSI chips from mask design. this could be acomplished by a multi-year committment of appropriate existing capacity to the needs of the program.

LOCATION

The program would start immediately and be coupled to existing computer science and computer engineering research facilities and programs. Facility selection is strictly on the basis of the intensity and quality of work in VLSI, image processing, parallel computing and AI. Either Lawrence Livermore or Berkeley Laboratories would be ideal sites for the computation center which would link to Stanford, SRI, and UC/Berkeley. MIT, MITRE or Lincoln Laboratory could be the basis of an East Coast facility. Los Alamos has the largest network of supercomputers and support computers including storage and image production. If a central site were Los Alamos, this would force the development and installation of high speed links to other sites.

APPLICATION CENTERS

The following very imcomplete list of application centers is included as an example of how work would be contracted by the program office to expertise centers throughout the country.

D
E
V
E
L
O o Higher Performance Interprocessor Or Communications Structure
P (CMU, Univ. Illinois)
M
E o Dataflow Simulation And Parallel Algorithm Compilers
N (Lawrence, MIT, Berkeley)
T
o VLSI Design Automation For Parallel Computation
T (MIT, Lincoln Laboratory, Berkeley)
O
O
L
S

A
P o Image Enhance/Map/Encode/Compress
P (Goddard, Univ. Maryland, LASL, Lawrence)
L
I o Feature Extract/Target ID/Automated Inspection
C (GM, GE, SRI, Univ. Texas)
A
T o Image And Symbol Knowledge Representation/Expert System
I (Stanford, MIT)
O
N
S

DELIVERABLES

The work encompassed is broken into three classes shown in the Deliverables Table. Within each class there are families of projects and finally the projects themselves. The program runs about ten years broken into rough phase transitions at the end of 1985 and 1989. The work in the first phase puts the research environment and work standards in place and develops the first generation tools and applications. The second phase includes several machine realizations that use the tools and runs the test bed applications. In this phase, the research facilities are enriched with the machines realized by program efforts. These are in turn, the base of the second generation tools and applications. Finally, the third phase provides refinements and solves the hard problems that depended on the new understandings generated in the first two phases of the program.

DELIVERABLES TABLE

	A											
	P											
DELIVERABLES	P	'82	'83	'84	'85	'86	'87	'88	'89	'90	'91	'92
	L											
	I											
Communications	C	reconfigurable 100 MBy/s LAN			256cpu @ 100 MBy/s LAN							
Structures	A			256 cpu @ 10 MBy/s LAN								
	T						1000 cpu @ 100MBy/s LAN					
	I											
Dataflow and	O		simulator ok	hotspot analysis								
Parallel	N						VLSI dataflow machine					
Compilers	S		dataflow compiler									
	|											
Parallel VLSI	E			parallel logic simulator running on Dataflow simulator								
Design Automat.	N		VLSI parallel compiler				expert system for VLSI design					
	V											
	I											
Program Office	R	pick 1 rules language		next generation rules language								
Work Standards	O	common workstation (LISP?)										
	N	1,10,100 MBy/s LAN's		parallel rules VLSI		2nd implementation						
	M											
Parallel Comput.	E		1 MBy/s NAN & gate									
Environment	N	AFP AI-VLSI support facilities					VLSI dataflow on 100 MBy/s					
	T	I II III			IC signature analysis array							
	|											
Image Enhance	D				256 cpu node on 10 MBy				4096 cpu node on 100			
Map/Code/	E		SEM enhancement dataflow									
Compress	V											
	E				Full motion video-conferencing in 56Kb/s							
	L									($500)		
Feature Extract	O			IC signature analysis dataflow								
Target ID/	M						SEM scan analysis expert system					
Inspect	E											
	N											
Image/Symbol	T		parallel rules language primatives				expert systems for					
Knowledge/									process/crisis mgt.			
Expert	T					image/info=knowledge						
	O											
	O											
	L											

MOTIVATING SUMMARY

The motivation for this approach is timeliness and effectiveness:

1. THE RESEARCH PROGRAM SHOULD START NOW

2. WE NEED COUPLING OF INDUSTRIAL R&D AND APPLICATIONS WITH COMPUTER SCIENCE RESEARCH

3. WE CAN BUILD ON EXISTING RESEARCH PROGRAMS AND COMPUTERS

It is essential that we start now on the research program, as our computer science research has been drifting these last few years as both industry and computer science research have both gotten large, diffused, and independent of one another. Significant industrial research outside of IBM, Bell Labs and Japanese companies is non-existent and there is no coupling of basic and industrial research. For example, we believe there is better coupling of Bell Labs work to the Japanese computer industry via NTT's, ECL, than between Bell Labs and the U.S. Information Industry. Furthermore, both the academic and industrial research communities are now poorly coupled to real applications. We believe that program focus of some of the existing research efforts into a goal directed system will enhance their productivity and enable the continuation of a vital Information Industry for the 21st Century.

APPENDIX 1

SOME CRITICAL GLOBAL QUESTIONS (AND ANASWERS)

1. Why is the establishment of national facilities the correct way to attach the parallel problem?

   . No single lab now has critical mass or focus in anyone area - currently all resources are difused.

   . The lab(s) and programs operate together to do the work.

   . Users, architects, and builders must couple.

2. What impact will this proposed program have on existing research facilities? Programs?

   . The intent is to build on, and <u>extend</u> current facilities by additionla resouces. We believe that this program is close enough to some of the existing.

2a. What about the extra space required for these facilities?
   . We don't know.

3. How will this effort help the basic problem of a shortage of qualified researchers?

   . It is hoped that a "program" will stimulate the demand to produce more researches over the long term.

   . Short term, the focus should increase everyone's effectiveness.

   . We hope to apply industrial researchers to the problem that are now difused and often operate as a sub-critical mass.

4. Who is supposed to benefit from this proposal and in what specific ways?
   (See Section on Program Beneficiaries)

5. Is there a nationla crisis and exactly what is it?
   (See section on Motivation for the Program)

6. What evidence do you have to support the level of funding which is projected as being adequate to achieve the goals?

This is really a draft outline for concrete proposals. From this we expect specific sites to be established and operated in very targetted areas: such as parallel knowledge based systems, high performance parallel processing and parallel image processing.

7. What, exactly is the overall objection of the program?
(See the first sentence of this document)

APPENDIX 2

WHY USE CDC'S ADVANCED FLEXIBLE PROCESSOR?

The AFP has demonstrated high performance in digital image and signal - processing tasks. For example, a processor system can transform the every co-ordinate of a million point picture in 1/30 second. Several systems are in operation today. It includes various support software including simulators.

Traditionally, we design, build and then use. A machine as fast and general as AFP would require at least 5 years to build. By using the current AFP as a general purpose research tool, we can gain at least 5 years on starting such a program from scratch. To illustrate, consider the several data-flow projects that could use AFP today to simulate architectures. Since we need to evaluate these architectures by using them, we could understand the benefits and drawbacks of these machines five years (or so) sooner by adopting the AFP as a hardware simulation base.

The CDC AFP provides a very fast, flexible, microprogrammed set of up to 16 computer modules for experimenting in various parallel computing structures of various type. A <u>single</u>, AFP microprogrammed processor provides the following capability:

. 20 to 800 Mops in 16 parallel, 16-bit arithmetic and logic units
. Microprogrammed control
. Access to 32 Megaword (256 Megabyte block oriented memory)
. 2 X 1 Gbits/sec communication with neighbors in ring

A flexible multiprocessor and multicomputer structure are both provided since, the sixteen processors can be interconnected both to a common 32 Megabyte memory and to adjacent processors.

The AFP can thus be used as a <u>tool</u> to study several different computer structures that we believe are much of the basis of the next generation.

Because AFP is so highly parallel, including having functional units with side effects, we believe it will not be imcroporgrammed to any great extent.

The mode we envision is that it would operate in several configurations, with fixed microprograms to behave as:

1. Set of microprogrammed pipelined, functional units within each processor. Four units can be initiated every 20 nanoseconds, although an average of seven units operate in parallel for most problems. Because of the difficulty of programming this highly parallel structure, the most important benefit, or side-effect will be understanding in how to do it effectively. Because the microprogramming so heavily pipelined, we believe a better understanding of dataflow techniques

for expressing algorithms will result from the use. Nearly all high performance machines are pipelined; hence, we believe AFP is a good vehicle to get a better understanding of pipelining.

2. 16 processor multiprocessor with shared memory and very fast interprocessor intercommunication. Here, the processors will be programmed to be particular ISP, such C. If C could become the basis of the machine, then UNIX could be run.

3. Set of 16 Computer Modules microprogrammed for particular functions. AFP was designed to be operated in this mode for image processing.

4. A dataflow computer. This is a special case of item 3 whereby particular computers are programmed to behave as the various functionla units of a dataflow computer.

5. A set of special, parallel processing architectures using individual, microprogrammed processors as the functional units of the particular structures. In this mode, AFP turns out to be a very good emulator of relatively complex VLSI chips.

6. An experimental Ultra-LAN based architecture. To examine how computers can be coupled effectively and work together on a task, the AFP looks like an ideal for study.

APENDIX 3

ROUGH BUDGET

The program expenses are estimated at approximately $18M/year running from 1982 through 1989. Equipment is expensed as delivered. In general two or three "competitive but collaborative" groups are charged with each project family.

YEAR 1

	#	Heads	Total	Expenses	($M)
Program	Sites	(ea.site)	Heads	Manpower	Equip
Communications/Structures	2	5	10	1	-
Dataflow & Parallel Computation	1	1	1	.1	-
Parallel VLSI Design Automation	1	3	3	.3	-
Parallel Computing Environment	1	2	2	.2	15
Image/Symbol/Knowledge/ Expert Studies	1	3	3	.3	-
	6		19	1.9	15

YEAR 2

	#	Heads	Total	Expenses	($M)
Program	Sites	(ea. site)	Heads	Manpower	Equip.
Communications/Structures	2	5	10	1	5
Dataflow & Parallel Computation	3	3	9	.9	-
Parallel VLSI Design Automation	2	5	10	1	-
Parallel Computing Environment	2	2	4	.4	10
Image/Symbol/Knowledge Expert Studies	3	5	15	1.5	-

Image Enhancement Studies	1	3	3	.3	-
Feature Extraction Studies	1	5	5	.5	-
	14		56	5.6	15

YEAR 3

	#	Heads	Total	Expenses ($M)	
Program	Sites	(ea. site)	Heads	Manpower	Equip.
Communications/Structures	2	6	12	1.2	5
Dataflow & Parallel Computation	3	3	9	.9	-
Parallel VLSI Design Automation	2	5	10	1.0	-
Parallel Computing Environment	3	2	6	.6	8
Image/Symbol/Knowledge Expert Studies	3	5	15	1.5	-
Image Enhancement Studies	2	5	10	1.0	-
Feature Extraction Studies	3	5	15	1.5	-
	18		77	7.7	13

YEAR 4

	#	Heads	Total	Expenses	($M)
Program	Sites	(ea. site)	Heads	Manpower	Equip.
Communications/Structures	2	6	12	1.2	5
Datflow & Parallel Computation	3	3	9	.9	-
Parallel VLSI Design Automation	2	5	10	1.0	-
Parallel Computing Environment	3	2	6	.6	5
Image/Symbol/Knowledge Expert Studies	3	5	10	1.0	-
Image Enhancement Studies	2	5	10	1.0	-
Feature Extraction Studies	3	5	15	1.5	-
	18		77	7.7	10

YEAR 5

	#	Heads	Total	Expenses	($M)
Program	Sites	(ea. site)	Heads	Manpower	Equip.
Communications/Structures	2	6	12	1.2	5
Dataflow & Parallel Computation	3	3	9	.9	1
Parallel VLSI Design Automation	2	5	10	1.0	0
Parallel Computing Environment	2	2	6	.6	5
Image/Symbol/Knowledge Expert Studies	3	5	15	1.5	-
Image Enhancement Studies	2	5	10	1.0	1
Feature Extraction Studies	3	5	15	1.5	-
	18		77	7.7	12

YEAR 6

	#	Heads	Total	Expenses	($M)
Program	Sites	(ea. site)	Heads	Manpower	Equip.
Communications/Structures	2	6	12	1.2	5
Dataflow & Parallel Computation	3	3	9	.9	-
Parallel VLSI Design Automation	2	5	10	1.0	-
Parallel Computing Environment	2	2	6	.6	5
Image/Symbol/Knowledge Expert Studies	3	5	15	1.5	-
Image Enhancement Studies	2	5	15	1.5	-
Feature Extraction Studies	3	5	15	1.5	-
	18		77	7.7	10

YEAR 7

	#	Heads	Total	Expenses	($M)
Program	Sites	(ea. site)	Heads	Manpower	Equip.
Communications/Structures	2	5	10	1	4
Dataflow & Parallel Computation	2	3	6	.6	-
Parallel VLSI Design Automation	2	5	10	1	-
Parallel Computing Environment	3	2	6	.6	5
Image/Symbol/Knowledge Expert Studies	3	5	15	1.5	-
Image Enhancement Studies	1	5	5	.5	-
Feature Extraction Studies	3	5	15	1.5	-
	16		67	6.7	9

YEAR 8

	#	Heads	Total	Expenses	($M)
Program	Sites	(ea. site)	Heads	Manpower	Equip.
Communications/Structures	2	5	10	1	1
Dataflow & Parallel Computation	1	2	2	.2	-
Parallel VLSI Design Automation	1	5	5	.5	-
Parallel Computing Environment	3	2	6	.6	5
Image/Symbol/Knowledge Expert Studies	1	5	5	.5	-
Image Enhancement Studies	1	1	1	.1	-
Feature Extraction Studies	1	5	5	.5	-

10	34	3.4	6

YEAR 9

Program	# Sites	Heads (ea. site)	Total Heads	Expenses Manpower	($M) Equip.
Communications/Structures	2	5	10	1	-
Dataflow & Parallel Computation	-	-	-	-	-
Parallel VLSI Design Automation	1	2	2	.2	-
Parallel Computing Environment	2	2	4	.4	3
Image/Symbol/Knowledge Expert Studies	-	-	-	-	-
Image Enhancement Studies	-	-	-	-	-
Feature Extraction Studies	1	2	2	.2	-
	--				
	6		18	1.8	3

YEAR 10

Program	# Sites	Heads (ea. site)	Total Heads	Expenses Manpower	($M) Equip.
Communications/Structures	1	5	5	.5	-
Dataflow & Parallel Computation	-	-	-	-	-
Parallel VLSI Design Automation	1	1	1	.1	-
Parallel Computing Environment	1	2	2	.2	1.8
Image/Symbol/Knowledge Expert Studies	-	-	-	-	-
Image Enhancement Studies	-	-	-	-	-
Feature Extraction Studies	-	-	-	-	-

```
-----------------------------------------------
     3                                     8              .8              1.8

                                                                         GB3.S7.3
```

Gordon Bell
June, 1974

THE INTERACTION OF TECHNOLOGY WITH COMPUTER SCIENCE

Computer Science is unlike most other sciences. But it is like astronomy. Although both are based on mathematics, their development only came with technology. Galileo's telescope was the precursor of the science of astronomy just as the Von Neumann computer preceded computer science. It is impossible to conceptualize either science without its technology.

During the age of enlightenment, the modern university and astronomy developed concurrently. But by Von Neumann's time, the areas of knowledge were codified in the catalogues of the universities. Thus, the courses related to the computer were integrated into various disciplines. Generally academicians, who made serious attempts to be Renaissance men began to use the computer as a tool for integration of fragmenting fields of knowledge. Thus the first computer courses were to be found in a wide variety of departments ranging from philosophy to electrical engineering.

After two decades, in 1967, Allen Newell, Alan Perlis, and Herb Simon [1] synthesized the notion that the computer was deserving of its own science.

> Computer Science comprises the machines, the languages, the programs that operate them, and the basic algorithms that are applied over a broad range of user problems.

Furthermore, Computer Science is not only close to mathematics and engineering, but also to society--the users, who benefice and benefit from this science. Clearly, it is unlike automotive science, which has been discontinued at most universities. As it was a technology whose advance was clearly not helpful to society, computer science will continue because the technological development computers and their use hold value for all levels of society.

THE TECHNOLOGY OF THE COMPUTER

Computers have been genealogically described [see Fig. 1] starting from the first generation, with the start of each subsequent one becoming less precise.

The first, vacuum tube technology, began with the Cambridge EDSAC in 1946 and was used until 1959. The second generation was marked by the introduction of a single device--the transistor--as a component. After 1965, computer historians indicate a third marked less dramatically than the first two. Its beginning was the encapsulation of several transistors into a single silicon area to form one basic logic element. More recently, it has been stated that in the 70's computers entered a fourth generation, marked by extreme miniturization: a complete processor, memory, or other functional element with several thousand transistors, is placed on a single silicon chip with an area of 0.2" square (costing about one dollar). The number of binary digits (bits) stored on a single silicon die has been observed to be 2↑t, where t is in years since 1962. Such growth cannot go on indefinitely. If this is a new generation of computers, it is likely to be the last of this technology. Optical information processing may be used in the far future.

The latest computing machines are fabricated in roughly the same way as baking cookies. Although the recipes may vary, they will both become ubiquitious and uniquely recognizable but not necessarily notable. In fact, only humans can be fabricated with less process explicit control and sometimes less forethought.

As computers themselves are incorporated into other aspects of our everyday life--from the telephone system to transportation systems to monetary systems--their presence will be no more remarkable than the batteries or motors we rely upon. They will continue to develop and change, but many of the major innovations have been made which sharply differentiate one generation of computers from the next.

The sequence of development of stored program computer technology, has several parallels in the early mechanical computation, particularly as developed by Charles Babbage, an English mathematician. In 1823, he was the first recipient of the Gold Medal of the Royal Astronomical Society, for his work "Observations on the Application of Machines to the Computation of Mathematical Tables". Once, when involved with the checking of some astronomical calculations, he was quoted as saying, "I wish to God these calculations had been executed by steam." It is important to note that he thought instinctively of steam, the technology on the verge of becoming usable. For even though the steam engine was developed in early 1700's, and Watt made the first efficient engine about 1770, it wasn't until 1800-1825 that the steam engine was employed in the locomotive. This coincided precisely with the time when Babbage formulated his ideas on computing engines. Babbage was quick to adapt a variety of technologies. His second "Analytical Engine" used cards--an idea taken from the card-controlled loom of Jacquard--to hold a sequence of operations and its variables. It had an internal memory to hold intermediate results.

Stretch the Technology

The machine required more precise gears than could be manufactured at the time. This involved improving the technology. Although it was a sidetrack, Babbage detoured, and the result was a book, "Economy of Manufacturers and Machinery."

In a similar way, the premium on physical size, weight, and reliability by NASA has had a very positive effect on the improvement of logic technology, and forced the development of the third computer generation at a much faster rate than would otherwise have occurred. However, once stimulated, the market demand takes over and the process is self-sustaining. A current by-product of the third computer generation is the hand-held calculators which are available for under $50 and cost about $10-20 to build. they will be extended, to rival the standard computer costing over $10,000.

Starting New Projects With The Newest Ideas

Babbage's "Analytical Engine" is recognized to have the same structure as a modern stored program computer. But, neither the "Analytical Engine" nor the "Difference Engine" which preceded it were carried to completion. Each new machine idea captured his major attention so that none of his machines reached full operational status and did not have the impact they might have had.

The early computers constructed at various universities have run a close parallel. The most famous, and most successful was ILLIAC, the machine built at the University of Illinois in the early fifties which was a relatively pure engineering embodiment of the Institute for Advanced Studies--(Von Neumann) computer. About a half dozen replicas were constructed and used at various institutes and universities around the world. Then ILLIAC II was conceived with the goals of stretching circuit speed, using transistors, testing asynchronous logic ideas, and providing the university with a very large computer. Although it achieved many of the goals, it was too late to be useful to the engineering community. Academicians lagged significantly behind industry by the time of ILLIAC III, which was designed to test peripheral optical processing for the Atomic Energy Commission, which additionally needed a large computer which also had to be designed. Due to the large number of variables, ILLIAC III did not reach production operational status nor adequately provide a test for the optical processing part.

Finally, ILLIAC IV, which was to be the world's largest computer, was conceived in the early 60's as the Solomon computer, to be operational in 1969; but it is still to be scheduled for useful work.

In general, while many early university-built computers must be judged as unsuccessful in a production sense, nearly all have produced significant side benefits to the development of computing.

Government Funding For Computers

Babbage like many modern-day computer hardware designers needed funds beyond those available to him as professor in absentia from Cambridge. He found funds through the government earmarked to build a specific machine to calculate nautical tables. This early science-military grant was similar to the current ones and there was a difficult interface between Babbage and the government.

Similarly, the first generation of modern computers, which is often called the Von Neumann computer, was funded and used by the US Army for computing firing tables; thus it had a particular mathematical computation orientation. Much of later computation has been similarly funded because its expensive developments are only justified if the benefit is clear cut, direct, and seems to be very large. For example, the SAGE Air Defense computer was adapted and formed the basis for commercial air traffic control.

Diversion

There is a third and last parallel between Babbage's time and our own. Babbage found there were interesting peripheral problems to solve that drew him away from his central purpose. For example, he studied and devised a uniform cost for mail, independent of distance, and today computer scientists find themselves working on diverse problems (for example, NSF has more joint research programs with the Office of Computing Activities than any other program). While there are many examples of computer science working on peripheral problems (eg. speech synthesis and recognition, printing), the most progress can be observed with respect to the effect on digital communications. Computers have required reliable, low cost, high data-rate communication links for connecting user terminals to computers and for interconnecting computers (ie. networks). From the early 1900's until the late 1960's, teletypewriters and their communication links were limited to about 100 bits/sec. Most recently, the link capability has been extended to 300 bits/sec without special line conditioning and to 1200 bits/sec with line conditioning. Also, good high speed links are available at speeds of 2400 bits/sec to 9600 bits/sec and special 50,000 bit/sec links are eminent. Thus more information is transmitted with the same resources.

In a similar way, computers have instigated and been utilized in a store-and-forward fashion to build networks (Bell, 1974). For example, with the 100 site ARPA network computers can call one another for very short messages (eg. 1000 bits) and avoid the customary 20 second dial switching delay.

In fact, Babbage established a tradition which has remained in computing: he offered only promise over than existing calculation methods; the market (use) and its requirements were not fully defined so that they came after, not before, development; other people's problems were tackled to the detriment of his own work; his machines were not finished on time; he tried to build with technology well beyond the current state of the art; and he depended on government funding which was accompanied with promises, delays, and overruns on both sides.

Computer Industry Computer Use

One of the more advanced industries that uses computers is the computer industry itself. In fact, as a computer manufacturer, we are our best, albeit least preferred, customer; and perhaps this would seem natural. The obvious reason, familiarity, is not the main one. The growing number of computer scientists and engineers within the industry do, however, contribute to increased internal use as well as the external stimulus of need creating products which are marketed to a variety of external users. The users also stimulate improved use by encouraging competitive operating systems (and other programs). A by-product of better computer use is the understanding of resource allocation in a multiple commodity environment; this turn may be used eventualy in other disciplines (eg. economics).

Finally, computers can be used most easily by the computer industry itself for computer design and for software production, because the objects with which it deals (information encoding numbers and symbols) is directly represented by machines. Representing an automobile or a building within a computer, together with appropriate operations to manipulate them abstractly is more difficult. For example, a computer thereby can be simulated and even manipulated by algorithms. The abstraction of a hardware machine with memory, registers, an instruction set, and mechanism to interpret the instruction set maps directly into arrays, variables and a sequence of statements of conventional programming language (eg. FORTRAN). Contrast this with the notion of representing a primitive component such as a wooden beam, which has many attributes of relevance in house design (eg. length, and orientation), and an open ended set of operations (eg. sawing, painting, nailing, support of floors and walls) that are difficult to represent abstractly.

FUTURE COMPUTERS

Computer technology is improving at a yearly rate of 25 to 40%. In its 26 years of existence, computing performance has increased by a factor of 10↑5. While there are probably limits to such growth, it is not likely that they will be reached until after 1980. this means that every 2 years, the performance available for a given cost doubles; or as a corollary, every 2 years the price for a given level of performance is halved. In all cases, more computing is done for the same amount of money. Figure 2 illustrates the change in cost for minicomputers as a function of time for the last 15 years, and Figure 3 shows the cost of several minicomputers plotted against performance. If we assume a completely elastic demand for products, then each year, as the price is lowered by 40%, many new uses are possible--because useage is a function of the relative cost of computing. In some cases, this market demand appears to double, each time price is reduced by 28%.

The important number here is the 40%, because it really gives an upper limit to what we might expect in the future. The futurist who believes that a 1 to 2 million dollar computer of today will sell for $5000 in 1980 is, I believe, overly optimistic. At the 40% rate, or a more conservative 25% rate, this will mean a decrease by a factor of only 5 to 10, which means the computer will still cost $100,000. To give a more realistic example, Dennis and Smith (1971) of IBM projected a cost of about $100 for a 10,000 word minicomputer in 1980.

However, instead of using time to measure technological improvement, a much more precise measure is the number of computers produced. Since computers have been produced at an exponential rate, the time and number measures are identical. Technology, measured in this way, has been shown (Fusfeld, 1973) to be roughly of the form: TI=aI*b, where

T = Technological level of Ith unit produced
a = level of first unit
I = the cumulative number of units produced
b = technological progress constant

For computers, the technological progress constant has been about 2.5 (using memory size x memory rate as the technology level measures) which can be contrasted with (at most) 0.7 for recent automobiles (using horsepower as the technology level measure) and 1.06 for jet engines (using thrust as the technology level measure). Note that the technology level for computers is a factor of 10*15 improvement over 100,000 machines, whereas automobile horsepower improvement has been only a factor of 30 over several hundred million units with a trend to now decrease horsepower for fuel consumption reasons.

The numbers of computers have not yet begun to reach fundamental growth limits, unlike the numbers of men or automobiles. So far computers have barely reached the 100,000 level and, since growth rates are exponential, they will reach several million by 1980. But this will not be based on the waste technology or on changing styles so that the customer can boast of owning "the latest model". Computers can remain almost as current as the information given to them. Obsolete information is simply erased and does not pollute the atmosphere.

From an overall computer system design standpoint computer components are continually reassembled in new ways--older machines become secondary processors to new ones in a rejuvinating fashion--or else placed on the used computer market which is a thriving business. But computers have exceedingly long lives, based on parts that seldom wear out. Two machines, I designed in 1964, are still in operation today: one controls making cookies at the rate of a dozen freight loads of flour a day, and the other controls the nuclear reactor for a power station, both continue to perform their tasks better as they are finely tuned to the situation.

The growth in numbers of computers will be further spurred on by reductions in the cost of their components. Costs of the primary memory, which holds the instructions for processing the data held in the secondary memory, have been decreasing by 30 percent per year. A similar reduction is occuring in the secondary memory which is used to store the data bases. Although there is little advance as yet in the evolution of better whirling objects, such as drums and disks, this is more than offset by increasing capabilities to store information more densely, so that the net cost/performance ratio is decreasing at a rate of 27% per year.

Since the secondary memory is akin to file storage, computer memories, like the human mind, derive power by the amount of information they have stored and their ability to process it quickly. There are at least two competing technologies promising to provide a significant breakthrough by replacing the whirling mechanical parts with electrons.

The tendency is for the computer--as a piece of machinery--both to become more compact and cheaper. This is the dimension of its hardware. As a compliment, there are software developments in the art and science of using the computer.

COMPUTER SCIENCE

The development of computer science is based on created objects--the machines--not fixed natural laws like biology or physics. However, the underlying mathematics of representation and the theory of algorithms for mathematical machines preceded the development of the physical machines. An engineering discipline was concerned with the fabrication of the first machines to interpret the languages which expressed the algorithm. Schwartz believes that computer science is the search for algorithms that are machine interpretable.

When I first came to Carnegie Tech in 1966, Alan Perlis, who headed the Computer Science Department, apologized to me, an engineer, in the naming of the Computer Science Department, since he considered it a possible misnomer. It could as well have been computer engineering or technology, except for the restricted sense of the words. Sciences, like universities, appear to have a longer lifetime. If he had chosen engineering or technology, we would probably have experienced a later name change. Computer science encompasses parts of linguistics, philosophy, communication theory and mathematics, as well as engineering.

The science of computer programming has been studied in a way similar to the process of design. Design has yet to achieve scientific status--although it is clearly a candidate (see Science of the Artificial, Simon*). Anyone with some proficiency in a programming language can usually design a program with test data giving one run of valid results. But this is false security, as the program cannot be guaranteed for any other data. This kind of phenomena is clearly not scientific so that it necessitated creation of a branch of computer science which was first called software engineering and now is known as structured programming. This deals with knowing that a program is correct by building it correctly, instead of having to observe that it appears to give correct results.

The goal is to completely transform programming to a process based on scientific methods. There are still many programs and methods which are yet to be guaranteed. Similarly as new programs, machines, and machine languages are developed, there will be a need to formulate new algorithms. Some computer scientists feel the improvement and guarantee of algorithms and programming methods should be the sole role of the science. While this is certainly an important function, the application of science and technology to imorove new computers can also continue to offer benefits to the users. Both goals strive for the same ends but the mathematician-programmer considers the machine fixed while the engineer-technologist considers the machine a variable. As Alan Perlis has often said: "One man's constant is another man's variable."

I have given you examples of how the engineer has improved the computer. There has been similar quantum improvement in programming and some algorithms. Until recently, the time needed to compute the Fourier transform was proportional to n squared (where n is the number of sampled points). This usually amounted to about 1000 points. An improved algorithm reduced this time to be proportional to n log n, or by a factor of 100 for 1000 points. Assuming a continued 40% yearly improvement in machine performance, it would have taken the machine builders and technologists about 14 years to achieve the same performance. One societal benefit from this algorithm was that computers were then able to be utilized for interpreting electrocardiograms in actual time, ie. to continuously monitor a patient's heart in an intensive care unit and watch for anomalous behavior. In contrast, a highly trained person can carry on such monitoring for short periods of time.

THE COMPUTER (PEOPLE) GENERATION

Even now most people are not aware when they are conversing with or through a machine. A computer cannot only compute and remember (and forget), but it also has devices which can print on paper and film, be typed into, project on TV screens, sense TV input, speak, listen, and switch communications channels. This ability to interface, via various information carrying channels, with other processes and to carry out information processing and storage, gives it a chameleon-like quality, and makes computer science highly interdisciplinary.

Direct communication of people and processes will be completely different from the present standardized notion of computer use. The traditional image involves many needless steps starting with people who transfer data into hierglyphics on paper forms, others who copy the paper onto cards; yet others who present them to a computer for reading then observe the results of erroneous writing or keypunching, and repeat the process until they get an error-free output; this is returned to the researchers who if they find that the results really weren't what was wanted, must reinitiate the process from the beginning. This card carrying process has led to the notion that we feed the computer. That it eats cards, and somehow digests them and extrudes, presses, or reforms them into 11" by 14" sheets of paper--which are too wide to handle easily, let alone read. Computer scientists are moving away from transmitting information through long queues and error-prone channels.

The alternative, as experienced by the younger generation of users is direct processing through a familiar and usually friendly typewriter or TV-like screen which results in a stronger positive feeling about the machine as a companion and tool. Fortunately, the card reading machine has been decreasing while the number of typewriters and TV-like scopes per machine has increased significantly. But unfortunately, its page printing capability and its needless use seems to be on the rise. Until users are educated to communicate with non-printing terminals, and store data magnetically, it will continue to be necessary to print more information than is needed in order to get easy access to the desired information. The potential of the computer to replace the printer used for non-read storage is almost timeless if one considers the algorithms that can be used to describe how to compute encyclopedic data to fit each user's need. While the Guttenberg revolution may have resulted in raising educated consciences to possessing encyclopedias, the computation revolution allows man to possess a machine with access to both more precise and directly relevant information than the family set of encyclopedias.

At a more primitive level, in 1972, John Grason, Allen Newell, and I wrote, edited, produced, and printed a book on Digital Systems using a computer. After the first draft was typed on a terminal and edited from the stored text, the book was printed by conventional photo offset. By referring to various algorithms, the computer itself produced the table of contents, indexed the text, kept track of pages, justified the margins, layed out the pages, and finally printed the final version from a display on an oscilloscope using a high resolution Xerox printer on a dot-by-dot basis. This enabled us to make significant changes right up to the day before the copy was sent to the printer for photo-offset printing. Having previously been an author of a book which required 18 months at the publisher, shortening this cycle by 16 months was truly a joy. All told, we were able to have a text faster, cheaper, and with less pain by an order of magnitude.

FUTURE COMPUTER USE

With the current emphasis on better utilization of resources--particularly energy--computers are a significant part of the solution. Their own efficiency in terms of cost/operations has been decreasing at the high technological rates previously discussed. What is more important, computers contribute to the resource utilization problem in many basic ways:

1. Substitution of recyclable information storage material (eg. paper) by directly reuseable storage material (eg, magnetic tape). Much of what is currently published and distributed via paper (eg, interoffice documents, and even newspapers) can be held on magnetic storage media for common access by readers. Such information would never be printed unless needed at unlinked sites. Very little energy is needed for the distribution of information, virtually no energy or space is required to store it, and only small amounts of energy are required to display information.

2. Substitution of travel by communications. Given that physical media can become less a part of interpersonal communication, people can communicate indirectly (eg, via computer conference) at greater distances without travel. The ARPA network (Bell, 1974) is already being used in several scientific disciplines to share programs and data bases.

3. Better communication channel utilization by encoding information (using algorithms and computers). For example, a computer can be used to encode voice and picture information for retransmission at significantly reduced bandwidth. Various algorithms permit tradeoffs in computing time, information storage (at transmitter and receivers), and bandwidth.

4. Better storage utilization by encoding information (using algorithms and computers). Much tabular data can be generated from a small amount of basic data, permitting many ways of accessing and analysis.

5. Better efficiency of all mechanical processes (eg. automobile, assembly line, power distribution) by precise knowledge and control. In general, there is far more known about the control of processes than has been applied. For example, better control of a car's engine using a computer, provides better utilization of fuel with less pollution. Research vehicles have been constructed, and the actual introduction simply depends on the price and availability of the computer. (Recall that the 40% yearly improvement can be either in price or performance.)

6. Although other disciplines (eg. economics) have required algorithms for multiple resource allocation, these disciplines have been built up assuming linear relationships (eg. Leontif's input-output matrix) using statistical techniques (eg. multiple regression analysis).

   Computer systems absolutely require algorithms for efficient self-management (ie. accounting, pricing, scheduling, etc.). Hence, computer science has provided knowledge for computer measurement, modelling, analysis, and control. This knowledge may eventually change the scientific thinking in the other disciplines.

Through better technology and more sophisticated algorithms, computers will increase in their ability to permit the solutions to the class of problems we can now imagine. New, large, non-mechanical-random access memories of the late 1970's will have the greatest impact on all applications, and these applications are beyond the scope of the above possibilities and my own near term view. Computers can enter all aspects of society. New generations of machines are and will be placed directly in cars, homes, games, appliances, and almost everywhere imaginable. For widespread use, new generations of users--not just the trained programmers--will work, play, and communicate by machine. And finally, the future generations of computers will be built from the science that has developed as a result of the interplay of the algorithms and the physical machines that interpret these algorithms so that we will increasingly acquire greater machine intelligence with less energy, The human computer scientists and computer engineers and users have determined the machine, Don't blame the computer, If there is any enemy==he is us,

*1, SCIENCE MAGAZINE== Computer Science is the study of computers",

BIBLIOGRAPHY

Bell, C.G., MORE POWER BY NETWORKING, IEEE Spectrum, Feb. 1974, pages 40-56, Vol 11, 2.

Newell, Allen; Alan J. Perlis and Herbert A. Simon, WHAT IS COMPUTER SCIENCE, Science, Vol. 157, No. 3795, September 22, 1967, pp. 1373-1374.

Dennis, S. F. and M. G. Smith, "LSI: Implications for Future Design and Architecture", IBM Research Report, RC 3598 (#16278) Nov. 3, 1971, IBM Research, Yorktown Heights, NY, 26 pages.

Fusfeld, Alan, R., TECHNOLOGICAL FORECASTING, TECHNOLOGY REVIEW, FEBRUARY 1973.

SCHWARTZ, JACOB T., ON PROGRAMMING, Computer Science Department Report, Courant Institute of Math. Science, New York University, February 1973.

Simon, Herbert A., Science of the Artificial, MIT Press, 1969.

Diary of My Visit to Japan
Gordon Bell - Summer of '78

July 16th - Arrive Tokyo

My seat mate from Honolulu, Anthony Geber, Director of Economic Policy, Bureau of East Asian Affairs (State Dept., 202-632-9690) illicited some argument from me. (He opened.) We exchanged business cards (I'm practicing for Japan) as it's the only time I've carried cards. (This, according to Reischauer, is the thing to do.) At any rate, his concern is simply that Americans are too lazy to compete. Also it's too hard for us to go after their small markets. Mine is more fundamental -- Japan's growth versus return on investment; the availability of capital; Japan's trade barriers and language/cultural barriers; and the way the Japanese focus on winning in trade -- all serve to scare the hell out of me. Throw in our waste, vis a vis energy, too. I also attribute our regard for science over engineering and engineering over manufacturing as key. The fact that we no longer build, but use tape recorders (especially videotape), radios, TV, high quality cameras (we only built a few - Kodak 35), small cars, is cause for concern.

July 17th - DEC Office + Keio University

Here we lie, watching the news (in English) after a day of running around like mad. Our host, Yu Hata, picked us up at 9:00 AM, took us to the DEC office, gave us a one hour briefing on computers in Japan (a brief history), and then I gave a two hour seminar on DEC products and engineering organization. There was an hour of questions on everything from 50 hertz 100 volt power to multiprocessors. It's clear we have inadequate planning of products for this market. Engineering makes its plans clear. Who's got the responsibility? (GIA-Janzen, CSS-Holman/Martin/Watanabe, the VT100 product here - Halio, or DEC Japan or some P/L for character sets?) This is a mess! For starters, I say GIA had better drive this issue!

We went to a nearby hotel and had to have an international style lunch (versus Japanese) because the Japanese part was full. I intend to assimilate everything -- just like the Japanese, so the food is paramount.

We left at 1:00 PM for a ride to Keio University (CMU affiliate) where I gave a talk on minicomputer architecture, which prompted lots of questions. We left at 5:00 after interaction and a view of their predominantly batch 11/06. Professor Toroko is an assistant professor in hardware. We were shown around by Professor Nori Doi. They would very much like to visit DEC. There was interest in architecture, as they wanted to build a large multiprocessor. Funding is tight as they're a private university with no NSF, ARPA or real industrial support. The main professor wasn't there.

Toroko gave me some papers which I'm sending internally, and Doi gave me a paper on the Fortran they built for the 11/08 patterned after Waterloo's WATFIV. On the return, Hata finished his lesson on the Japanese computer industry vis a vis the 3rd pair of groups (Fujitsu - Hitachi, who make 370 compatible), (NEC - Toshika, who're looking for a mini in Honeywell and used to be with GE) and (Mitsubishi - Oki). Univac (Nippon) is a dominant supplier somehow, based on Mitsui's earlier impetus! (Sept. 76 Datamation explains this quicker and better. The article is attached.)

We were dropped off at the Okura Hotel, I wandered around checking out the baths, water, etc. and finally settled on a swim indoors with a sauna. This let me shed a kilogram quick plus earn dinner. We went to a nearby restaurant Hata recommended and got a reasonable meal for only $30 each. It was quite good, not great, but we muddled through as I almost drank the tempura sauce versus wait for food to dip it in. We returned at about 8:30 and I called Don Frost about our visit tomorrow to NEC. I'm set to see the Director plus the technical management. Since I have a strategy to get a large share of the market, I wanted to check it out. Don said I, objective technocrat, should try it out with them! Basically the theme is: Buy any or all of DEC hardware/software; use it as a standard (just as Fujitsu-Hitachi do with the 370); sell in any/all Japanese/world markets and build a huge 11-based computer business! The Ministry of Trade/Industry (MITI), who controls all, should absolutely love it. The only trick is to get them to invent the idea!

July 18th - NEC

We had enjoyable talks/visits with NEC. The main purpose was to assure them we'd support them in their effort with the distributed system for the IRS (NTAA) using our machines (DECnet, 11/70, IAS). This was needed because they may view us as a source of technology (DECnet, minicomputers, interactive systems). Actually, technologically they're quite advanced, but probably in the wrong direction as their machines are ECL-based. The high ends are a takeoff of the Honeywell ceramic modules. They've been affiliated with Honeywell for 10 years, and next year the affiliation will be reviewed again. (Honeywell doesn't offer them anything.) The high-end is 635 based; the mid is on some earlier (2000?); and the low is on their earlier machines. Their office machines (100-series) are based on their version of the 8080 -- note it's an upward-compatible (one-way) version!

Their factories (computer and TV) were immaculate. People seem to move around faster than in ours, with more to do. The designs and quality of workmanship were quite beautiful. They commented on our reputation for quality and reliability -- which I think we have ... but we have to get these better. It's the one sure way to sell in Japan. They like quality/reliability (probably our other customers do too).

I had visited a TV factory; it's like I expected. They make 1,500/day - 300K/year. They have to make subassemblies because of U.S. import quotas. Their 5 Japanese competitors for U.S. market now have U.S. factories, which means the issue of Japanese products (TV) is solely a capital, manufacturing and design issue -- not high labor, for example. Again, I remember buying and giving away a large Zenith portable color and replacing it with a Sony color! (The Zenith replaced an old GE B/W which was never particularly good.) So in essence, I believe our ability to compete with the Japanese is:

1. A product design: quality/functionality (they adore knobby gadgets just like we do)/reliability.

2. Ability to manufacture it cheaply (and in volume).

As long as we don't forget this, we have a market. When we do forget, we'll be a distributor, just like GE and Zenith! (Incidentally, I recall that as engineers we felt sorry for the 100 TV engineers on Zenith's research group...why didn't they design better products? Why not a tape recorder? No American company produces a VTR (yet it was a U.S. invention). This gets back to emphasis of research (science) vs engineering vs manufacturing. I hope we're doing the right thing by pushing more on engineering and manufacturing (to a lesser degree) versus research at DEC.

In the afternoon I met with a number of their people from Central Research. They're largely American trained, where the cost is lower and training is supported by U.S. government. One was trained on MIT Multics. We had no trouble in communicating! My earlier frustration that they wouldn't talk wasn't true. I did have to control the flow, otherwise they'd clean me out of information! The affiliation with CMU turns out to be good, because I can merely quote work there and stay out of DEC's work. The central (non-product specific) R&D is 100 people versus 50 for us...or they have 4 x the R&D per NOR since they have roughly 750M in sales!

They're building a very high speed COBOL engine, multiprocessor (just as we're fascinated by them for production reasons), and doing a mass store subsystem. It's hard to compare us because they're more into batch. They build bigger machines, but they'll soon learn as they build Honeywell's Level 6 mini under license. I sense they have a fairly muddy strategy, building product-by-product as ideas seem to be good. (With our high end VAX/10/20 machine, I think will be a long way to having a <u>clearer</u> product strategy -- although we'll have more products!) A person from R&D was amazed at VAX, and what it had, what it cost...he said all those ideas came from large machines. Surprise! I said this in a paper in 1971 on minis!

Speaking of ideas. The Japanese (and we) have about the same regard for ideas...they're useless until applied. Once applied, fair game to be modified, taken, etc., within the limits of the law and morality (e.g. patents). I think we need to state as a policy that we do want patent protection on ideas whenever possible, and that we'll take ideas from any source subject to moral/legal constraints! Here's what struck me:

1. The semi-automatic wiring machine that Stocky designed wasn't patented. We gave it away to be manufactured locally. It was manufactured here as a <u>copy</u> by a Japanese firm. (I suspect they've improved it and maybe we should look into purchasing them here.)

2. Their low cost Teleprinter was adapted from Extel.

3. Their high speed laser printer was mainly IBM based using some Honeywell ideas.

4. The CML logic on ceramic modules came from Honeywell - although they made them manufacturable.

5. They use Gardner-Denver wirewrap machines and Universal inserters.

6. Manufacturing tools seem to be adapted from Macrodata, Universal, Teradyne (the wirewrap/backplane tester).

7. Their printer came from Versetec, though in a different package!

8. Their Fax machines probably have similar origins!

9. Their new Spinwriter is an adaptation of Interdata's carousel - I have some printout samples. The quality may not be high enough for word processing use. They're stressing reliability, speed (to c/s) and quality!

10. Cables/connectors come from the U.S. (maybe under license).

11. There are copies of the Tektronix scopes.

On the other hand, aside from our development and dedication to interactive and real time computing, many ideas of our products came from someplace outside (e.g., DECtape, 3M tape, cassette tape, the RK05, the DECwriters, the CRT's, the cache) various CPU implementation organizations, APL, BASIC, COBOL, FORTRAN, wirewrap, various LSI and manufacturing tools). We did contribute to computer structures more. In many ways we resemble them.

In the evening we had dinner at a posh, continental style restaurant with Dr. Ishii and Mr. Kitamura of NEC. I reaffirmed our support to them to make the NTAA (IRS) project a success...without this MITI will clobber us and our name will be mud. This is merely a reaffirmation of the Operations Committee decision requested by Marcus, GIA, and DEC-Japan.

Ishii was relatively speechless when I laid out the proposition that they standardize on 11's and drop the manufacture of the Honeywell Level 6. This gets them a mini right now, without continued investment, and they can backward integrate as they see fit. This theme for the GIA nationalistic companies is the right way to approach the marketplace. Somehow, we have to convince them that we're sincere and believe it to be the way to get into computers. This "sales approach" isn't widely understood/used. We need to formalize it. Japan would be the ideal place to start.

In the afternoon we went to the NEC computer factory and I talked with a number of very bright people from their central research lab. Fortunately they don't understand minis .... or they put on a good act (they had xerox copies of our VAX documents). Research has 100 people for a company half our size (4 x the effort). We saw a TV factory complete with multi-height rack burn-in (which we should use for disks).

July 19th - Fujitsu

We visited the central lab of Fujitsu at Kawosaki (Mr. Kurosaki and Mr. Sato), and then went to Numazau near Mt. Fuji where the computers were built. Fujitsu is the most computer oriented of all the companies because their founder, who died a few years ago, built one of the first relay computers. They ran the relay machine for me at Numazau while it calculated several common functions. They're not especially profitable, but they make beautiful computers and have the necessary technology. We saw their newly announced M200 (1.3 - 1.5 x 3033) multiprocessor using a dual cross-point for reliability. It appears superior to both Amdahl V7 and IBM (neither of which believe in multiprocessors (on M200).

Yu Hata and I could have easily had an argument on the relationship between Amdahl and Fujitsu. My view is simple: at IBM, Amdahl had developed a significant set of ideas on how to build 360's/370's. He left there and further enhanced the ideas in the circuits, design aids, packaging, small components assembly and testing areas. He got into financing trouble and Fujitsu bought a significant amount in return for the technology. Fujitsu put up the capital for the factory and made the assembly line work - no trivial feat because there's so much small assembly work. Fujitsu's first machine was not better than Amdahl's, but they took a longer term view (they are not that profit oriented) and produced better design aids and semiconductors, etc., so that their circuit M200 will probably beat Amdahl's V7.

The workmanship and detailed engineering is really fantastic. They have a very good master-slice (gate arrays) and fast (8 nsec) RAMs. In the terminal work, they have an anechoic chamber to get noise level down. They have some color CRT's and a floppy based intelligent terminal and are working on high level forms languages to make them easier to use. Of course, their disks are reverse engineered copies of IBM's.

Overall, Fujitsu seems the most frightening because of their dedication to quality, and winning. They have the strongest engineering and so far haven't been interested in mini's (PANA FACOM is their brand - a joint venture of PANASONIC (Mitsubishi) and Fujitsu). Also, given their disinterest in profit, they'll be doubly hard to beat.

Probably more important, Amdahl understands IBM mentality and how they strategize. This clearly influences Fujitsu and MITI. In fact, I believe Amdahl influenced MITI, at least indirectly, to build the plug compatible systems!

In visiting the Fujitsu factory, we saw one of the floors of the factory was devoted to programming. They had set up something that was very much like an assembly line for programmers. I would love to have our programmers look at this kind of environment because, in effect, there was really a sea of programmers. Probably the most impressive part was that they had a great number of line printers all backed up to a conveyor; and as each line printer finished its output, it was cut and stacked. It was cut into the appropriate pile; the pile was put on the conveyor; and the conveyor ran it off. The whole thing appeared on a carousel so that in fact all the programming listings were delivered stacked automatically. Of course, there were no individual offices for the programmers, only a sea of desks.

I guess the other thing that was impressive about the Fujitsu factory was the very clean atmosphere. The custom of removing shoes is very helpful; this is done on entry to computer rooms, temples and tea rooms. It was the cleanest of all the computer companies that we saw. This really pays off when dealing with the large number of contacts, the small coaxial cable, and the way the multi-terminal integrated circuits are sorted at that point under the board.

The Fujitsu M190 and M200 computers also used color CRT's for controlling the computers. KIVIAT graphs are displayed on the consoles so that one can get an idea of what's happening to the various resources. They are used in real time display in the Fujitsu computers.

<u>July 20th</u> - Electro Technical Lab and University of Tokyo

We visited Dr. Nishino and Dr. Mori of the Electro Technical Lab, which is run by MITI. This is a Central Research group responsible for computer research (the nearest equivalent of ARPA). The lab in a sense looked like many government labs - a series of dusty old equipment with experiments, which can be put into service for visiting dignitaries; some good and some bad work; and a bunch of reasonably intense Ph.D's. I gave a talk on the VAX design and it illicited a number of interesting questions. They're doing a large number of computer structures related work, several projects on multiprocessors and on microprogramming, and various things on language translation. On Dr. Nishino's desk was a well worn copy of the Quantam Sciences forecast on office automation. I asked to see stuff on Word Processing but the stuff I saw was not particularly useful or impressive.

The ETL does have one interesting virtue in that it does very little hardware building. In fact, its main function is to fund various industry groups to do design for a lot of the Japanese minicomputers. Anyway the one that is the equivalent to the DG mini looked exactly like the DG framework, except the workmanship on the console was much better than Data General's.

We went to the Tokyo Hilton and fortunately had Tempura, which is sort of batter fried fish, shrimp, and vegetables (probably the easiest thing for Westerners to accept and digest). It was about our second Japanese meal, because all the other meals were given to us assuming that we could not eat Japansese food. We had sandwiches (with bread crusts removed, delicately made and presented) at the various companies and had continental food when we went out (especially the elegant NEC meal which was heavily influenced by French cooking).

In the afternoon we went over to the University of Tokyo where I gave a lecture on minicomputer architecture in a very formally decorated room (held about thirty). They apologized for the small crowd because it was vacation. I was with Professor Ashida and Professor Inose, both of whom had spent a great deal of time at BTL. Inose is the father of the time sort algorithms for ESS No. 4 time division multiplex switching, which he did about twenty years ago. The talk was supposed to take one and a half hours with a half hour of questions, but ended up taking about one hour with roughly forty-five minutes of questions. We went to Professor Inose's office, were formally received, and discussed various types of things. The two professors had to leave because they had a dinner meeting of some sort.

We were then shown around the large Hitachi machines by one of the students. It was the Hitachi 8800 and he lamented the fact that Hitachi now was making IBM compatible computers, which he considered inferior to the ones they had currently made. Their other line is almost IBM compatible, derived from the Spectre 70 unit, but has special supervisory call instructions which makes them incompatible. We looked around the computer, which is really a monstrous machine because it was made out of MECL 10K, I believe; but the machine was water cooled.

There was a four processor system, three fast processors and a slower processor. The load was not very heavy. We went over to look at the system resources and I ran a BASIC and FORTRAN program. The BASIC null program really bombed out so I have a feeling the null program took a good deal of time showing that they had some kind of interpretive compiler. The FORTRAN produced good quality code and ran very rapidly.

We left there about 6:00 PM for dinner with Yu Hata, his wife and Don Frost at Yu Hata's son's apartment. We spent a thoroughly enjoyable evening looking at his airplanes. Because he is an avid photographer, he got into building model airplanes for aerial reconnaissance photos. He also built some helicopters. All of this was indeed incredibly impressive. The airplanes are very detailed and take something in the area of six months to one year to build.

<u>July 21</u>st - Sony

We were picked up early at the hotel, checked out, and went to the Sony Corporation Central Research Lab where we were given a brief introduction to what Sony is working on. Other than that I was able to get no information from the Central Research Lab group. I asked about what was going on in the Systems Research Group, but the only thing we saw was a Sony TV tube (used for Graphics) for which I have the specifications.

They also demonstrated with characters but the interlace problem created incredible flickers. I asked about buying monitors but they said I would have to see Mr. Iwama. Having gotten no information from the Central Research Lab, we then went to Sony's Atsugi plant, where we saw the video tape recorder being made. In contrast to the NEC TV plant, the Sony plant did not do any burn in of parts but in fact used testing to ensure that the product worked when they were all put together.

A large number of the parts were done outside this plant and subassemblies were brought back for fabrication. In all the plants that we saw only about half of the work is done inside. The rest is done by subassembly or contract labor. In the factory only 40% of the 1,100 people were workers. Of course, this was reasonably high considering that in that factory about 250 out of the 1,100 were in the engineering group. This is where they made so many semis.

The semiconductor part used three micron channel width for NMOS. They were the first in Japan to use the Bell Lab license of the transistor, and Mr. Iwama, the President and technical person at the top, insisted that a large number of engineers be hired to do semiconductors and, in fact, he backed Dr. Esaki.

Sony has an electron beam mask maker, which they got from Japan Electric Corporation, which is a copy of the American electron beam mask maker. We saw one of the AM 2900 ion implanters. It was just the fourth or fifth installed there. They pride themselves in owning a great number of the key semiconductor patents and, in fact, have a 10,000 volt transistor patent which is very key to making all solid state TV sets.

We left the factory in time to have lunch with Mr. Iwama, who of course took us to a hotel where we had a western meal; but before this, we looked at three very interesting video recorder projects all of which we have become interested in.

The MAVICARD recorder, which I have brief information on and a carousel version that allows up to five other cards to be loaded automatically, is a scanned device and the card holds up to one hundred images. There was a small video disk which held ten seconds of video on a frame by frame basis, and could be used in freeze frame applications. That system will be introduced this year for sports teaching. I am asking Yu Hata to go ahead and get information on these products. The third device was a small tape recorder, a tiny video disk about three inches in diameter that can store only a few frames of video.

All these products I find extremely intriguing, and all we have to do is figure out how to couple them to DIGITAL recording. Iwama talked about the various forms of pulse code formulation for audio and video (they have got to get into it). We would automatically end up with tape and disks that will allow us to use the video technology in computers.

They make it a point in their advertising of trying to stay away from anything that other people are doing. One can see by their various products and images, just what their approach to life is. Their motto is, "research makes the day".

After lunch with Mr. Iwama, we drove to the train station where we got on the bullet train for Kyoto arriving in Osaka at about 7:15 PM. We were met by the software specialist and were taken to the Osaka Hotel where DEC Japan, Osaka Branch, were having their end-of-the-year party. There were about 75 people there. Don Frost gave a good speech calling for plenty of openness and then I followed up by saying how glad I was to be in Japan, about how impressed I was with the Japanese, and our need for quality.

We finally got back to Kyoto and the Tawaraya, an old-style Japanese Inn, at 11:00 or so. I was glad to lay on a mattress that was flat on the floor and very comfortable, after having lay too soft in Tokyo.

July 22nd and 23rd - Sightseeing at Kyoto and Nara

We had breakfast, Japanese style, in our room at about 8:30 AM and then Gen Narui and Miss Tomioka came for us to go sight-seeing. In Tokyo we had home-made coffee and fruit in the room to gain time, decrease interaction, write, and it's awfully cheap.

In the morning we went to the summer detached palace of the Emperor Shugakuin outside of Kyoto, which included many temples, houses and rice paddies in an extremely beautiful setting. We were very fortunate to get there, and because I was a visiting "dignitary", we were allowed to go. I was glad that neither Yu Hata nor Gen Narui had seen the palace so it was a treat for all of us. Miss Tomioko was in a traditional, elaborate, beautiful Kimono and kept being stopped by U.S. photographers at each site.

We took off on a tour, which was about a two mile walk in reasonably warm climate, up and down the hill in an almost Greek-like setting. Then we left for Arashi-Tei, a restaurant I think attached to a hotel that overlooked the Hozu River. We had a typical Japanese, probably nine course, luncheon starting off with beer because we were so thirsty after the walk. After lunch we went up the Hozu River and rode the boat down for about 10 miles back to the landing of the restaurant.

Off we went to visit the Nijo-Jo castle in the center of Kyoto. This was a castle of the Shogun, built to impress the Emperor to put him in business. However, neither of them spent that much time in Kyoto because they both lived in Tokyo. The castle was, of course, extremely impressive with moats all made of wood and bamboo.

We came back to the Tawaraya, cleaned up a bit, and went out to dinner at a very nice restaurant. It is hard to remember which is the most memorable part of it, given that there were so many courses. After dinner we went down the main street of Kyoto looking for various souvenirs.

I spent most of my time looking for a knife, having been intrigued with the possibility of slicing vegetables very thin which is one of the specialties of the Japanese salads. I found one, got a few other odds and ends as presents, some more ideas for presents, and returned to the Hotel about 9:00 or so, quite ready to konk out so I could go the next day.

On Sunday morning we were trying to sleep late, given that we were going to take off at 9:30, but our maid/attendant unfortunately decided that we should get up about the same time as the day before and we were out by about 8:30. We met Gen and Miss Tomioka at the railway station and caught the 10:00 o'clock express train to Nara. The train is run by a private company and was extremely comfortable and cool, as are all the Japanese trains. We all got to the Todaiji Temple at about 10:30.

We went on to visit the Taishi Shrine at the same location, walked around, and had a fairly heavy nine-course lunch at an old inn called Tonochaya. We were off by 2:00 and went to visit both the Toshodaiji Temple and the Yakushaji Temple. These were high points of our trip. We were met by a lady who is on the staff there. Miss Tomioka knows her very well and we had an incredible walk through the various temples. The latter temple was probably most impressive because a fire had destroyed the west temple and they are building a new one. We were able to talk to the engineer who is in charge of the new construction. He showed us around and we ended up going into the construction of the temple. It is made of wood with no metal and is about 30 - 40 meters tall. We also went to the site where the wood was being prefabricated. This is being done by a bunch of scholars and an old carpenter. The whole temple is, of course, designed to last 1,000 years and, with the care they are taking, should easily accomplish this. There are about twenty carpenters working on the building. It is thought it will take about three years, or about sixty man years of work, to complete this temple.

The superstructure of the building is built around a wood pole and the temporary structure is made of steel and is quite permanent. After we got through climbing around, we were taken up in another temple that houses some of the Buddhas. All these temples, of course, house Buddhas of various sizes and shapes. The first one houses the world's largest Buddha made of 12th century bronze.

We were presented with various photographs, gifts, good luck charms, and goods to help us on our way. We had tea and cakes with one of the monks at the temple before we left at about 5:15. We got the 5:30 from the station near the temple, transferred to the express at Nara, were back to Kyoto by six, and had dinner at seven.

The five of us had dinner at the Tawaraya -- eleven courses. It was a magnificent dinner starting with raw fish, vegetables, and soup. Along about the eighth course we were served with a very heavy tempura as batter-fried shrimp, vegetables, potatoes, and fish. I was hoping things would be over, but in came the next course, which featured the hibachi. Everybody had steak and various vegetables. Somehow I managed to get through that course, but skipped the next two because it is probably thought bad luck to have an even number of courses. We were all presented with small hibachis. We finished dinner at about quarter of nine which is not necessarily typical, because for some reason, even though food is very lovely and things are in small servings, the Japanese eat very fast. While I am here I am trying to eat slower than normal, otherwise we would finish the meal in probably an hour. I do enjoy the food and the time spent very sociably.

July 24th - Talks at Kyoto, Osaka, and Kyoto Sanyo University

I gave lectures at Kyoto and Osaka Universities and had dinner with people from Kyoto Sanyo University. (The tape is apparently lost in the Sydney secretarial pools).

July 25th - NEC

We visited NEC in Kyushu, which is on the island of Okinawa, a place where NEC makes almost 80% of its semiconductors. It is there because of the labor force and because of the supply of water. They make about 5 million pieces a month, 60 million per year (at 80%, this would give the total NEC IC's at 75 million per year). If each is selling for maybe $3.00, because they have a large amount of LSI, NEC's total sales would be at about a quarter billion dollars (which is what we think they are).

Mr. Iwao, Chief Engineer, took us around. He is actually the operator of the plant and is interested in high volume manufacturing. The brochure I took back has all of this annotated. They started there in September 1969, with only 400 million yen capital, or at today's prices, about $2,000,000. They employ about 1,750 people there -- 1,250 are direct laborers. They operate two shifts -- 5:30 AM to 1:45 PM, and then the second up to 10:30.

Their history there is one of starting out to do semiconductors for NEC's NTT telephone business, so they have a fundamental interest in quality. Subsequently when they got into the NMOS PMOS calculator, cash register, and computer business, they changed the emphasis to volume, which they have now. In doing this, they never left their concern for quality.

All products are burned in. The NTT products are sometimes burned in for as much as a week, and some products are only burned in half a day. Eight percent of NEC's total sales go outside. It is building as much as 15 to 20% of these sales for export. Probably a larger amount is to the United States, although we don't know. They are making all PMOS 4 calculators and cash registers, and NMOS computer memories, including the 4K plus 16K RAM. They are doing a lot of CMOS for watches, calculators and radio equipment. In addition this NEC plant makes the BIPOLAR CML logic for the high speed computers based on the Honeywell CML logic.

We initially had concern whether we could visit there and they reluctantly agreed to let us. The person who took us around was not that keen on having us, but was certainly cordial after we arrived. They try to keep their labor force flat. They have taken all of their plating and marking equipment for the two in-line packs to local shops outside. They start with silicon wafers, go through test, then ship. They have a very nice process chart. In fact, virtually like every Japanese company, we were handed a brochure that clearly described their whole process. In this case there are 15 steps. The 16th is shipment, which is by air in specialized containers. From a semiconductor standpoint, they used the 4" wafer on one line in a large two-story building (240 x 40 meters) -- they have about 4 lines and at the one end is the new 4 inch line.

In a small building they have the bipolar line which is low volume for all of the processing areas. The second floor is the pellitization through testing processes, exlcuding the part that is done outside.

Mr. Iwao wanted to know how this compared to TI and to INTEL. I could not tell him (probably because I don't understand semiconductors that well). Frankly I was quite impressed simply because of the incredible cleanliness and the well designed layout they have.

Again, the pressure of the Japanese custom of taking shoes off (leaving them at the door) to enter a building is really helpful to a semiconductor processor, because it means that you don't carry a lot of dirt around. All of the areas that were part of the factory were marked in terms of class. The workers and the back of the equipment was class F and then everything else was in class C. They had class B and class A rooms. They end up with a failure rate of 1% at burn in, so that they have a very high overall volume rate at customer acceptance.

They own mask making equipment in Tokyo, which is an EB machine. All of the work done by the design and manufacturing production equipment design is done in Tokyo.

NEC has processed SOS wafers, but is not interested in it because of the low volume, low yield, high cost nature of it. They are also looking at and made (it is not clear how) JIL parts apparently for the NTT. (NTT wants it.) Unlike many of the other semiconductor companies, especially Sony, NEC believes that it must bring all of the manufacturing equipment along. It has formed a wholly owned subsidiary tester company called ANDO. Of course, being very patriotic to Japan and themselves, they use the NEC M4 minicomputer, which is a conversion of the Varian machines. The manufacturing complaint, about the difficulty of maintenance of the tester, is traditional with every manufacturing group I have heard.

The 4 inch line is one area that we weren't allowed to see. In fact, he studiously avoided us looking at their wafer lines although there were windows into all of the other lines. In the case of the new 4NC wafer line, there were no windows and no hint as to what was inside. He did say, however, they use automatic aligners, and that through the process up to diffusion, everything was handled as a continuous process. I would guess they have as highly an automated function as

TI's we saw several years ago. Diffusion, and some of the other processes, are batch in production. He longs to have the whole thing be a continuous process.

I was incredibly impressed with the fact that there were graphs of everything everywhere and I suspect even some graphs on semi-log paper somewhere. The graphs were used to plot everything against everything else, so that they really knew what their process was doing and the output. In the case of the secret process I asked about, he said that it had considerable computer control and the main reason for doing this was to know what the various steps of the process were doing and what the productivity was. As a manager, since he is not given that much control over his own destiny, he is very concerned about productivity. He does move some of the simple parts outside, but also is concerned with automating as much as possible, and keeping the cost of all the labor force flat while maintaining the various steep increases in volume. He did not say when the 65K/RAM would be built but they are being produced now in Tokyo.

COMPANY CONFIDENTIAL

ID#0238

digital interoffice memorandum

Subject: Some First Impressions of Japan

To: OOD, OC, Dave Ballantine,
Jim Bell, Max Burnet,
Don Frost, Bill Green,
Carl Janzen, Ron Smart,
Dick Yen

Date: 24 AUG 78
From: Gordon Bell
Dept: OOD
Loc.: ML12-1 Ext.: 2236

Here are a few brief first impressions of Japan (having visited Sony, NEC, Fujitsu, five Universities and a Government Lab). As you see below, I'm impressed with their intense drive, technical ability and will to win. Also, I position my understanding of factors which support what I believe is a basic goal to dominate the computer market...just like they do other (especially consumer electronic) markets.

This is a one-sided view as to their ability to win in our market...I didn't see things to get in their way.

I was prepared to dislike the Japanese because they had been so closed and absorbing of our technology and work. I could not help but like them; they were generally open. Now, I fear them more than I was prepared to. Here's why:

1. As a group, they're (industry-government) the most competitive. It's really built into their culture and reinforced by training. The only reason they aren't competing in minis is they're still enamoured with competing with IBM and building mainframes e.g., Fujitsu's new M200 technically dominates the new IBM3033 and Amdahl V7 machines. (We must worry because of what they've done in quality cameras/optics, textiles, small cars, radio, TV, tape recorders, watches, calculators, their position in semiconductors and semiconductor-making equipment, typewriters, sewing machines, etc.) This also drives them to fast response and hard work.

2. They're excellent engineers and tend to be less NIH-oriented than us. This is derived from having less egos, although there is a strong group ego! Japan has acculturated customs, technology, etc. from everywhere for centuries. In the 16th century they apparently set up manufacturing of guns/gunpowder in 18 months once the Portuguese brought them in. Any good idea is fair game (subject only to strict patent technicalities). Having adopted an idea they want to understand it and improve it. (This can be seen looking at progress in all the above plus the research they do.)

3. The current computer manufacturers have a complete line of peripherals, and set of test and manufacturing equipment, taken from copying and improving counter-part U.S. products. [Just as we all learn something from touring another facility, so do they. Should we avoid having them visit our plants? We have to be careful about our discussion of technology!] Here, I'm somewhat ambivalent, because I think we should trade - buy/sell - with them. We (DEC - especially the 11, DG, HP) clearly influenced their minis.

4. All of the manufacturers have acquired their technology over a 10+ year history of dealing with U.S. manufacturers either as a joint venture or under license: Fujitsu (Amdahl/Siemens); Hitachi (RCA); NEC (Honeywell, GE, Varian); Toshiba (Honeywell, GE, Interdata); Mitsubishi (Xerox), Oki (Univac - actually joint venture); Yokogawa (HP); Nippon Minicon (DG). In all cases, the technology has been improved in terms of quality and manufacturability. For example, in the case of the Amdahl technology (that was at least started at IBM), I suspect Fujitsu is one of the few companies capable of manufacturing the miniature/hi-density PC Boards, backplanes and small cables.

5. They seem to be less oriented to technology for its own sake versus what it can do for them in the long run. For example, they moved more rapidly into gate arrays for their computers earlier. (Maybe Amdahl's influence). They clearly think both product and process together in what is a longer term view. (Here, let me reiterate: We must clean up our processes or they'll win by default. We can't make one shot products on a rigged up, ad hoc process). Again, here they're competitive and they orient the processes to 1. Quality first, 2. Volume second (for growth) and 3. Flexibility and turn-around in order to support the volume. This gets into:

6. The Japanese orientation is a strongly engineering versus strongly science-based culture! (We - the U.S. - do their research, wo why should they bother?) This comes about because of the competition through manufacturing novel products and their total dependence on export/manufacturing. For example, much of our federal training, funding, comes through the NSF, ARPA, and armed services for research. Their funding comes through MITI (Ministry of International Trade and Industry). There's rotation among design and manufacturing engineers. They do have good central research staff and their flow appears to exist to the development groups. They both think they're on the same team. -- In contrast, research in many of the large U.S. corporations is a vast waste, e.g., GE, Westinghouse, RCA and Univac; the work is usually behind the average development and totally decoupled. It's clear how TV, Radio and recording was lost, but the engineers had help because:

7. We (U.S.) have a higher regard to business training versus engineering training. They're in good shape because they don't yet have all the business schools. Therefore, instead of getting MBA's, their students get engineering masters. This not only makes them better engineers, but doesn't reinforce the notion that engineering is the route through to the management ladder, or that an MBA is automatically needed if one is to supervise people. The MBA, oriented at every dual-career person being president, and epitomized by the content-free case study methodology, focusses on the quick buck. This is in contrast to the Japanese concern for the long term (an overall theme).

8. They've read the Boston Consulting Group monograph and are volume (and growth) oriented, subject to the quality-first constraint. Knowledge of the learning curves is everywhere, even the government research labs and universities. Their needs and goals are manufacturing/trade/industry oriented. This also means, like TI that they're will to dump and lose money for the short term in order to gain the market. Although they put on a good act that their products won't be competitive when the yen is so strong, having gone from 300/$1 to 100/$1, it's a big ruse because:

9. Roughly speaking, they have systematically transformed American business from inventor-manufacturer-distributor to simply distributorships. This is in complete keeping with the goals of American business and the modern business school, Horatio Alger, such as RCA, GE, Chrysler, etc. No investment, no planning, no risk, these simply distribute products for the Japanese and roi, profits look fine. All a person has to do to be successful is buy the right product for resale. RCA/GE don't have to worry where the money comes from to pay the Japanese (or Arabs). On the other hand a group who can only run a distributor is probably fairly top heavy and can easily be replaced say, be a hard-working Japanese group. [A solution here is to make someone at the Commerce Department responsible for each area. This should include the joint planning with industry and the prohibition of current manufacturers from being importer/distributors. I.E., RCA would not be allowed to remain in the business and import. The responsibility for an industry has to be delegated!!]

10. There's no way we can re-enter various lost businesses now that we're just a distributor. The spirit, understanding to develop and manufacture are gone. It's too easy just to distribute. There are now no decent American TV, radio, Hi-Fi, or video recorder products/manufacturers for what are basically indigenous U.S. products and which the first invention or key patents apply! Somehow, these industries and companies have been grossly mismanaged. (I also blame the Department of Commerce - a faceless, leaderless nobody!) How? Why? It can easily happen to us!

11. They're more long versus short term oriented. Their history encourages this. They're capable of waiting us out in an area because we're so big bang (product) oriented and because they want long term business domination. NEC, Fujitsu and Hitachi, unlike Xerox, GE, Westinghouse, RCA, have all persisted with computers and now appear to be winning! This timeliness certainly affects their thinking on quality, and lastingness both in markets and products.

12. They believe computers are fundamental for the long term and they're prepared to wait. Machines are used in all products they build for export and they save labor - and labor is precious expensive in Japan as there are only 110 M people and 2% unemployment. They're considering raising retirement from 60 to 65 to get the extra productivity. They need computers to raise productivity! This is vital to their domination of manufacturing. (This is the opposite of the Australian attitude where there is high unemployment and a need/belief that computers must be eliminated. Australia is now almost totally dominated by Japanese products).

13. They are willing to give up profit for growth. For example, RCA is on a rug maker (or distributor), car rentor, book publication, TV Distributor etc., instead of an electronics company that really pioneered the T.V. Whereas there is extreme pressure on business for profit and return on investment, these factors are less in the Japanese companies. Sony is quite profitable, Fujitsu does relatively poor financially and I'd bet NEC or Hitachi computer divisions might even lose money. For now, they may still be buying in - clearly more acceptable than GE, Xerox and RCA. (This makes them doubly hard to beat...since they can lose money on every one and make it up in volume. They'll buy this business - DUMPING! and why not?)

14. Products are quality/detail oriented versus being the ultra-high volume, low-quality throw-away types. These are characterized by say, Sieko (versus Timex) and anyone of their cameras say, Minolta (versus Kodak or Polaroid which assume an idiot user with no concern for quality picture, but must have it now...again the time attitude). For example, while they make no instant cameras, perhaps due to patents, when they do they'll be quality.

    There are zero defect signs everywhere! In the Sony VTR plant there was no burn-in of the recorder. All subassemblies had been inspected and tested. When it was put together it worked.

15. They are compulsively clean. In an indirect way, this really helps the manufacturing of small, precise goods (e.g., cameras, LSI, high-speed computers, some disks).

16. Even though they have a concern for long term, they work the short term very hard. This may follow from the competitiveness/growth. They engineer for quick turn around, they have good processes and the engineers at these large companies work very hard. The official work week is 40 hours, but a more accepted pattern is 50-60 hours...particularly to maintain schedule or to win against IBM, Amdahl or Hitachi (if you're at Fujitsu).

17. As the head of our Osaka sales office put it: the Japanese live to work versus the American need to work to live. He claims this is instilled at birth and trained. Work is a central theme, and the companies go through extensive screening to hire for life - e.g., some companies only get graduates from certain universities. Housing is provided for the workers and they have what amounts to a lifetime contract. This is bad if a person's incompetent, it also means that it's hard to breathe different life into an organization. On the other hand, turn-over is low to non-existent and a team spirit clearly develops as the various members learn to work with one another.

    Their physical condition certainly reflects this work ethic too! On one hand there is a great deal of smoking, although a campaign is in progress to reduce it. However, nearly all Japanese are trim versus being basically overweight. Their diet (including excellent raw fish and vegetables) is conducive to trimness and better health, I'd guess. Although alcoholism is supposedly on the rise, the consumption in business I saw was certainly less than in the U.S.

18. The long term, quality products makes them build products that are hard to beat on a life-cycle basis. While it isn't clear they really consider all life-cycle costs, their cars now get good ratings - even though they may be designed to decay rapidly after say x years/y miles. In the case of computers, they always build multiprocessors because their customers invariably buy and want upgrades. Since IBM rents, the multiprocessor approach hasn't been developed. The multiprocessors they sell are also built for better Reliability, Availability and Maintainability. They seem to do a better job considering life-cycle costs than we do!

19. At a government/society level they appear to have their act together much more than we do. In both newspaper stories and in their products they seem to have clear, crisp ranking of goals and priorities. For starters, they know them, whereas nearly all our issues that start out simple become entangled as everyone (a new set of referees) enters the fray (e.g., human rights vs equal rights; full employment vs inflation, balance of payments; environment vs region vs country; capital vs labor; consumer protection vs business protection), but worse than a muddy set of design criteria is a muddy set of decision makers and an unclear decision process.

    Because of the need to export, there's very good support for engineering and many go into it. There are comparatively few lawyers, (factor of 2 per person), so the emphasis is on physical output rather than paper, and intergroup contracts, and bickering among semantic accountants.

20. INVESTMENT

    As a simple explanation, more money is available for investment because of lower taxes. This clearly affects their ability to invest in industry. They're supposed to be willing to pollute for profit...I didn't observe this. (Maybe they only kill whales outside of Japan and pollute other environments). Their environment is fine - though high density. On the other hand, taxes can be low because:

21. Their Government spending for military is far less. (Nearly non-existent). Although there is some fall out of our military spending for computers and related research, it's small compared to what it could be if there were more directed goals such as the Japanese export goals. It's not clear what these goals should be.

22. The Japanese don't have the waste, federal research expenditures, such as NASA and the Energy Department. (Here again, they can rely on us if there's any output.) These are big expenses and contribute little. The Energy Research seems to still be the old Atomic Energy Commission, but dressed in new clothes. The labs do about the same work, with essentially no output. (At least at the AEC, their goals were clearer, and we had a consistent flow of big computer bangs...plus a constant market that's motivated to provide computers.) Here, the Japanese do some nice work in regard to funding and managing research flow. Their labs buy versus develop in a vacuum with no way to get the flow.

23. MITI and other labs fund other laboratories and corporations to carry out research that's oriented to getting experience that will assist products. This not only provides a system of checks and balances, but provides an incentive. This minimizes what I call the "dusty-lab syndrome". Many of our government and federally funded labs were initially set up for a mission, and once the mission has been completed, the lab continues to exist. Since there's no real need, or mission, or review, negligible new work is output. (Recall visiting labs in which the dust is blown off the equipment for visitors and the same demo is run year after year. The same equations are on the board, with the same usually vague, unattainable, immeasurable goal for the research.) A buyer-seller relationship can help check this to some extent. Also this brings the groups together and technology transfer is more likely to take place.

    For example, NBS is setting up a lab to do standards research and industry is free to contribute interns to them - this is ridiculous! A more fruitful way to bring about the standards is to subcontract several approaches and have industry develop and report on them with NBS. In this way the staff is minimized at NBS (which will be obsolete and impossible to acquire) but they will get more quality through their buyer role. Foremost, there's a reason to interact.

    There's a good understanding about the research flow mechanism. They use all sorts of techniques -- organization, people-rotation, having many visitors to the U.S. Labs, buyer-seller, space, etc. -- but they do have the concern because of the limited number of people. We seem to have too many doing too little with no concern for output.

24. Overall, MITI appears to be very strong and competent! The goal of MITI/industry is a strong industry! This is in contrast to our Department of Commerce which appears to have the standard 9-5 bureaucrats, who are in it for either security or power - but with no real way to make anything happen. Nor is there any measure. I don't know what's missing that they have (just quality people - as Reischauer suggests, the right longevity, power, process, maybe they segment responsibility and measure results with reward based on performance (e.g., winning in a trade area)). In a few samples, I believe it's simple people quality, and the right process enabling them to accomplish something. Being responsible may be the key variable. Here, this suggests we could probably eliminate the Department of Commerce and have no real change except more output.

25. While there doesn't appear to be Japan Inc., there is clear collusion (planning etc.) among the government, and companies. They actually plan to win! This includes basic strategy setting among the players to segment and go after various markets (e.g., Fujitsu/Hitachi are 370 plug compatible). The companies can talk to one another and do, but certainly compete intensively with one another.

    Japan is quite a closed society and market. As the most powerful, homogeneous culture there is a long history of this. A quick trip, a pass through Reischauer's book, The Japanese, and an explanation of just a part of the tea ceremony make this vivid! Two years is a frequently quoted number to begin to understand this.

26. The language is a code to further segment. It's not clear how difficult it is to learn, but it's probably relatively useless without the societal understanding. We don't teach Japanese widely. On the other hand the technically trained Japanese have maybe six years of English in order to read the literature. (This is probably a good reason why we should use OEM's to enter the end-user commercial market versus translating the many manuals.) On the other hand, the lab, industrial, educational and engineering market may be open without extensive mail translation.

27. The tariffs support the establishment of any industries they target. Now the computer import duty has been reduced, but I doubt if this matters much since their industry is strong enough to withstand imports!

28. By the society and the emphasis on personal relationships (not clear they're any more than French) it's hard for foreigners to break into or sell, especially on a one shot basis. (It remains to be seen whether an American manager say, could set up and effectively manage a Japanese office.) "Doing business" together appears to be done over a long time period and is almost ritualistic.

29. There is an amalgamation of the Japanese within an industry which creates something that's often referred to as Japan Inc. (I think the Japan Club is a better name, because there's at least a show of competitiveness at the market level.) Not only is MITI supportive, they also appear to dictate. What's worse is they interact with industry in what appears to be a helping way as described above. For example, DEC products were in the Computer Engineering/Science Departments when the 11 first came out, but with a Japanese mini industry we really don't sell there. I'm sure it's because of their recognition of this market (also they discount heavily in the universities and consider it a prestige sale)...there may even be some special tax incentive. There is incredible pressure to buy Japanese products!

    The high cost of labor, limited population and full employment coupled with few natural resources, creates some interesting by-products.

30. The pressure to work is fed back, creating more work and output, since everyone is working.

31. Inventions are to labor-saving devices. I saw countless gadgets of this form. All the printers at computation centers had paper cutters on them with conveyors to bring output back to a single station. There are NO computer operators, tape mounters, etc. running around!

32. There's real concern for saving of physical resources too. At the computation centre, printout isn't automatic; it's queued and must be requested by badge reader, (also, lights - always fluorescent due to efficiency - are off in the computer room - the console is external with only one or two operators!) Of course small cars, taxis, a good train/subway are other indicators. The cars have bells that ring when the car is going over 100kmh!

33. There's measurement of and pressure for efficiency (i.e., work out/work in is high). In a taxi, there's an automatic back door opener so that the driver can load/unload faster. Of course, the factories graph everything. It feels like the notion of efficiency is taught to all.

34. Everything runs on time and at full capacity (trains, planes, a supply of taxis, buses and especially meetings, tours, etc.). This is in contrast to the habits we've gotten into on scheduling and performing at meetings! Also, Yu Hata did an excellent job of scheduling customers, manufacturers and sightseeing. I accomplished roughly twice as much per day as in another western country.

35. There's orderly queueing at each server. The Japanese appear to be the world's best self-queuers. There's probably some protocol for resolving races when two persons arrive to the queue at the same time.

36. There is a range of basically human/personal concern. While the subways and trains jostle people pretty badly (at high density), and there's no segmented smoker areas (and many smoke), there's great concern for the feeling/privacy/treatment of individuals. Perhaps I had special treatment, but on arrival/departure at every organization, we were given hot cloths for refreshing (it was hot and humid - but taxis and all buildings had A/C), and either tea (occasionally coffee - to be really considerate of a westerner) or cold juice. The hotels though incredibly expensive were the best I'd ever stayed at in terms of quietness, service and general treatment. This included a large, but very well run chain hotel in Tokyo. In Kyoto I stayed at a tiny (fifteen room) old style, old Inn, and only once did I ever see any other guests (at the front door). The goal is to make certain that guests are totally alone, with incredible attention to simplicity, design and detail (e.g., there was a cloth over the telephone because it didn't fit the room decor).

    Of course, the food is the ultimate in personal concern. Food served in seven, nine, or eleven courses varied from raw fish to pickled vegetables (e.g. potatoes) and flowers (lotus blossoms) with lots of seaweed and fish and fish eggs. There is western-oriented food like tempura (deep fried), hibachi grilled meat and fish and teryaki. At the first of the week we had western/continental/universal-style food because our hosts were concerned, and then we asked to have only Japanese food. We ate nearly everything (there's one kind of seaweed I found unpalatable). Of the sandwiches we had, the bread crusts were removed. There was much concern that the colors of the food matched - the physical looks were important.

    There are Japanese baths, and these are great too!

37. Products are designed for people with attention to detail. The styling happens to be also attractive to others, but their technical, gadget-orientation really biasses them to designing technical looking, knob-intensive products (hi-fi, complex watches, cameras, etc.). It's probably impossible to have them design a product like the polaroid one-step camera. (Emotionally, I doubt if the designers can do it based on the picture quality.) Color monitors were used to control the larger machines.

38. Contrary to expectations they are working the environment issue. There were U.S. environmental people there at a conference, and the Japanese were politely ignoring them...and taking their conference registration fees. Nearly all cabs are LP gas! Although they're physical comfort oriented, they do work the resources too.

39. They seem to do "bottom-up" product design versus "top-down" market planning as typified by the expensive heavy, multi-volume market surveys. These usually report history and extrapolate it in a self-perpetuating fashion. Using this approach, we continue to build heavy, gas-consuming cars because the market has historically bought them (given few alternatives). They look at the needs, and take existing ideas (designs) and improve them.

EPILOGUE

On arriving at Sydney, I was struck with contrast to dense, intense, humid and hurried Tokyo. I was ecstatic to get back (after twenty years) to a life style, people and place I really like:

Sydney's beaches are the world's finest; the weather's great; people spend lots of time out of doors with sports, strolling and simple gardening (versus the subtle and very complex Japanese gardens); work starts late, runs slower and ends promptly; and the food (universal/continental/western), beer and wine are drastically improved having moved away from the early English influence. I look forward to a last weekend stroll. I'll enquire about the best reef for SCUBA diving (on another trip).

GB:ljp

DIGITAL INTEROFFICE MEMORANDUM

DIST:

Al Bertocchi	PK3-2/A56	Dick Clayton	ML12-2/E71
Jim Cudmore	ML1-5/E30	Sheldon Davis	PK3-1/C16
Bill Demmer	TW/D19	Ulf Fagerquist	MR1-2/E78
Win Hindle	ML5-2/A53	Bill Johnson	ML21-3/E87
Ted Johnson	PK3-2/A55	John Kevill	ML1-3/E58
Andy Knowles	ML5-2/A53	John Leng	MR1-1/F35
Bill Long	ML5-2/A53	Julius Marcus	MK2/C37
John Meyer	ML12-1/A11	Ken Olsen	ML12-1/A50
Stan Olsen	MK1-2/A57	Larry Portner	ML12-3/A62
Bob Puffer	ML12-2/E38	Jack Smith	ML1-4/F31

Dave Ballantine	ME	Jim Bell	ML3-2/E41
Max Burnet	SN	Don Frost	PK3-2/S50
Bill Green	ML1-4/B34	Carl Janzen	AK
Ron Smart	AK	Dick Yen	TA

"On the management level," Dr. Hughes says, "many management development programs flounder when they are aimed at people who don't want any change or at others who know all the answers already. The opportunistic types see these programs only as a way to learn more about how the system works so they can beat it oftener.

"We are by no means saying that you should discard Dr. Fred Herzberg's theory that workers can realize a sense of accomplishment from the work itself. What we are saying is that it should be used for the right people. We divide people into types . . . but this isn't something out of the jungle; it's present right in your shop and front office. What it means is that you may have to have at least six types of management and communications to get to everyone."

SOCIOCENTRIC	EXISTENTIAL
15	25
5	almost none
10	10
Enables me to enjoy many friendships and support worthwhile causes.	By itself, not as important as how it is used. It gives me freedom and chance to be myself.
Ralph Nader's "Raiders"	Benjamin Franklin
Group meetings and participation. No stress or conflict. Highlights socially useful purpose of operation. Friendly supervisors.	Loose structure. Stimulate creativity. Spell out long-range goals. De-emphasize retirement plans and other "golden-handcuffs."
Harmonious working environment. De-emphasize merit pay and "climbing the corporate ladder."	Continuously challenging work. Freedom of choice. Job-enrichment programs.
If they see the company is hurting people, they will organize and rebel.	Increasingly found in management. Incompatible with tribalistic, egocentric, or conformist boss.

In conducting their seminars for Executive Enterprises Inc., New York, Dr. Hughes and Dr. Flowers are principally concerned with an awareness by personnel and management development specialists of the basic approach, rather than quick solutions. They do stress two immediate answers:

- To obtain results from any employee benefit program, management must sell the "benefits of the benefits," not concentrate on details.
- Most bluecollar workers consider their immediate boss their principal communications outlet and management symbol. Programs should be explained by him to workers in one-to-one or small-group sessions.

'Even somewhat immoral . . .'

Disagreeing with the personality-type-casting approach is Dr. Roy Walters of Roy Walters & Associates, Glen Rock, N. J., one of the original implementers of job enrichment programs at American Telephone & Telegraph Co.

"I have trouble pigeonholing people according to their personality type," he says, "and I think it is even somewhat immoral to classify them this way. Personality type has nothing to do with the way they do their work or their ability.

"You may have to have six types of management . . . to get to everyone."

"Another grave danger is that, while it is true some people don't want more responsibility, if you don't start giving it to those who do, you never learn what they can do, and you don't get any supervisory workers from their ranks."

Dr. Robert Janson, a partner in Dr. Walters' firm, admits that some elements of psychological-type-casting do appear in job redesign programs, but he considers them relatively insignificant.

"Where you set up jobs with different complexities, you must set up different work systems for all," he says. "It simply doesn't work if you push them into slots without any options. But this is only remotely related to their psychology and more to their ability and experience."

Six types key to approach

Dr. Hughes and Dr. Flowers contend that it is directly related to the workers' psychology, specifically to their psychological types as defined and developed by Dr. Clare W. Graves, professor of psychology, Union College, Schenectady, N. Y.

Almost no one fits 100% into a single classification, but most people are predominant in one, along with some traits of another of the six types:

- **Tribalistic.** Although accounting for only one-fourth of the population, this group includes some 40% of all hourly production workers. Most comfortable with an established ritual, members of this group take pride in working for a prominent company like being in a strong tribe.

They generally have little or no ambition to rise out of their group and interpret most of their job in the context of whether they have a good or a bad boss. They will attach themselves to a good one and often will go to him for assistance—even after he has been placed in a different job.

So the most effective way to motivate tribalistic workers is to provide them with a boss they will respect. Or, if you are their boss and can't relate to them, find their natural leader and get him

Which type are you?

To find out which type you are, answer the following questions. There are six choices for each question. There are no right or wrong answers; simply circle the statement you like best. If you have a second choice, place a check by it also. Answers on Page 36.

A family should . . .

A Stay close together and take care of one another.

B Let each person go his own way without interference.

C Provide guidance to the younger members on what is right and wrong.

D Help each other succeed in a career and see that the children get ahead in the world.

E Provide warmth and harmony among all the members and their friends.

F Permit family life to be like real life, with all of its good and bad points.

Freedom is . . .

A Not having to worry about money and other problems.

B Not being pushed around by people who have more power or money.

C The chance to work and live where I want and be a good citizen of the community.

D The opportunity to do what I want to do and to pursue success.

E The right of people to be themselves without prejudice and social differences.

F Doing what I like to do without denying others their freedom.

A good job is . . .

A Having a good boss regardless of the work.

B One that pays enough money.

C Knowing exactly what should be done.

D Where good work leads to promotion.

E Working with a good group of people.

F Solving interesting problems.

Laws are . . .

A To tell us what to do and protect us from people.

B Not important unless you get caught breaking them.

C Necessary to keep order in society and should be obeyed by everyone.

D Sometimes unnecessarily restrictive in getting things done.

E Useful if they promote social causes.

F Necessary to make any society function.

Money means . . .

A Paying for the things I need to keep going.

B Buying things that make me feel important.

C Security for the present and future and a good standard of living.

D Power and status and belongings that I have earned.

E Social distance and barriers in society.

F Freedom and opportunity to be myself.

Personal possessions . . .

A Are necessary for living.

B Make me feel like someone important.

C Come from hard work and should go only to people who deserve them.

D Are a sign of success and a source of pride.

E Are not as important as personal friendships.

F Are important only for what they mean to the individual.

A good boss . . .

A Tells me what he wants done and helps me do it.

B Is tough, but lets me be tough also.

C Sets clear policies and sees that people follow them.

D Helps me understand the objectives and rewards me when I achieve them.

E Is more of a friend than a boss.

F Sets goals with me, then trusts me to do the job the best way.

nonexistent. There are *tribalistic* presidents and *existential* floor sweepers.

And 70% to 80% of all managers alternate in their values between their professional and private lives, and their values may change as they age. Each of these could cause shifts in psychological groupings.

"It is quite common in this day of computers that a manager of a department or a branch office or region can be just as intimidated by having to do 'what the computer says' or 'what New York says' as a floor sweeper is by his boss," Dr. Flowers comments.

In motivating employees, he continues, "the principal problem is that most programs are developed by management types who think everyone wants what they [management] want. It is 'mirror management,' and it simply isn't true to life. They don't all want authority, responsibility, a private office, or a retirement plan.

"For instance, quite a few companies that took out their time clocks for hourly workers have gone back to them. Many workers like them. Typical attitudes we found were: 'How do we know you're computing the pay right without them?' 'Why doesn't everyone punch it?' 'We're supposed to be here at 8 a.m., and this shows we were here.' "

Many people would rather be told what to do and get their achievement kicks on their own time, he adds. They have different conceptions of loyalty to a company; it's a place to work and nothing more.

Six employee types and what makes them tick

	TRIBALISTIC	EGOCENTRIC	CONFORMIST	MANIPULATIVE
Percentage of managers, based on studies of 1,707 supervisors	10	10	20	20
Percentage of hourly personnel	40	10	40	5
Percentage of general population	25	10	35	10
Key identifying attitude on reasons for importance of money today	Buys groceries, pays for rent and other things I need to keep going.	Buys things I want and makes me feel like somebody.	Allows me to save for rainy day, have decent standard of living, and aid the unfortunate.	Is a measure of success in my job, my company, and my community.
Archetype person or character in this class	Kamikaze pilot	Archie Bunker	Queen Victoria	J. P. Morgan
Most effective management climate	Good boss, no decision-making, rules to follow, plenty of security, pensions. Regular pay, no piecework. Work groups of ten maximum. Short-cycle work.	Freedom of action to a point, but clear line of authority. Piecework pay. No intangibles or deferred compensation.	Rewards for seniority and loyal service. Rules and procedures for everything. Organization charts and career planning.	Keep light rein. Allow innovation. Give status symbols, decision-making authority.
Most effective motivators	Good boss, steady pay. Job content irrelevant.	Hard cash; leave him or her alone.	Regular advancement by seniority. Clear procedures. Efficient management system, appraisal reviews.	Opportunity to wheel and deal. Options in pension and retirement. Money. Status symbols.
Comment	Class exemplified by company uniform in Japan. Women tend toward this group. Little desire for advancement. Will resist transfer. Recession brings converts.	High turnover. Always discontented. Recession brings converts from other groups. Highly suspicious.	Size dwindling rapidly due to broadening horizons of mass media, but reinforced by tradition. Found in bureaucracies.	Sex, poker, religion are all games. Flexible ethics. Best as salesmen.

digital INTEROFFICE MEMORANDUM

TO: Brad Vachon

DATE: October 1, 1971

cc: R. L. Best
Nick Mazzarese
Stan Olsen
John Holman
Fred Gould
John Eggert
Ken Olsen

FROM: Gordon Bell

DEPT:

SUBJ: The System's Designs Sweepstakes: RTM's (alias) PDP-16 for Special Systems

I just procured a copy of your memo (from a source who must remain nameless) regarding the above subject, and urge Fred Gould and John Eggert to also see about getting copies.

Let me small voice disagreement:

1. Field Support - Sounds like we're always committed to only M and old R, B, W, etc. series modules. Since K series aren't applicable to the types of systems you build, isn't that a straw man? Field people can and are being trained. They won't have to, if you don't use them. They learn about many options and an RTM designed system, is by definition, well documented because the system is constructed from the documentation. I would bet that the few systems they have in the field are better documented and better designed (in that their behavior is well-defined) than those of special systems.

2. Information - The current handbook is also aimed at your engineers. How many of your engineers really understand the fundamentals of the transistors and circuits of the AND gates they use? Now we know the error of our ways - Combinational (e.g. AND) gates and FLIP FLOPS are small, bad components. They encourage poor design by forcing finite state machines, i.e., sequential circuits to be built using them. Even though there is a morass of theory on the subject, and we educator-types encourage learning about them), they have always been a very poor foundation on which to build (assuming all you're using is people for designers).

   The PDP-16 components (alias RTM's) give a reasonable level of components, but foremost, they make it virtually impossible for a designer to screw-up. This has to be an ideal situation for a manager. --

3. Cost - The PDP-16's were applied to the wrong problem. Look in my paper or the handbook for a section on "when to use PDP-16's." They don't solve all problems for all people --- we also can't build KI10's out of them either or production PDP-8's, etc. --- In your area, you are always given a computer; hence, it's unlikely that starting from bare PDP-16 modules, we can solve the problem by first building a computer - cheaper. Besides, let me see the data on the PDP-8 vs. PDP-16 system solution for a 2780.

4. Technical Restrictions - I don't think that parallel buses will be needed for your applications, if they are, then use them. They're explained in the manual.

   Both programming and diagnostics just have to be easier with PDP-16, because the states of a control you build using them are better defined than with the ad hoc sequential circuits that designers now use.

PDP-16 - The Perfect Answer to All the World's Problems

Yes, you will have problems with them. Some more modules will be needed to make combined computer-special systems with them to be really smooth to build. There are lots of things that can't be built with them. We only make the following claims

The Great System's Designers Sweepstakes

According to our discussion on September 23, about company design approaches, let me propose the following rules for the great system's design sweepstakes:

1. The PDP-16 will monitor RFQ's for special systems that come into your group.

2. They will select one such design (without stating which one).

3. They will build their design (while you build your design).

4. When both designs are operational, a comparison [1] will be made on the basis of:

   a. Overall clarity of design documentation.
   b. Calendar time
   c. Quotation time

----

[1] By me or by Ken if he can be torn away from science.

d. Design and checkout time (costs)
e. Parts and production cost
f. Spare parts cost
g. Ease of writing software to checkout design
h. Cost of checkout software.

The weighting function is to be scheduled later.

I assume this is a relatively fair way of conducting an evaluation. Also, I would hope you might also take one of their applications and do a similar test.

bwf

digital INTEROFFICE MEMØRANDUM

Gordon Bell

TO: Ken Olsen
cc: Nick Mazzarese
Stan Olsen
John Holman

DATE: September 15, 1971

FROM: Brad Vachon Brad

DEPT: Computer Special Systems

SUBJ: RTM's (alias PDP-16) for Special Systems

Generally the idea of RTM's as building blocks is good. Initially we were frustrated by the lack of individual module documentation. We had only PDP-16 sales literature and engineering sketches to work with.

As of now, we are still not keen on using RTM's for the following reasons:

1. Field Support- All "special products" are warranteed and field supported. RTM's are neither stocked in the field nor understood by field service personnel. (Similar problems with K-series).

2. Information- The current handbook is not aimed at a systems engineer who really must understand the module blocks before using them.

3. Cost justification- Our one attempt to design a 278Ø replacement proved that a PDP-8E would do the job better and cheaper with more flexibility.

4. Technical restrictions- We should discuss these in detail but the restrictions fall into concerns about mixing modules to get control and timing functions, diagnostics and programming, using more than one RTM bus, etc.

John Holman and I would welcome the opportunity to discuss this in detail with you.

BJV/pgh

digital

October 4, 1971

Mr. C. Gordon Bell
553 Briar Cliff Rd.
Pittsburgh, Pa. 15221

Dear Gordon:

We will be happy to give you the data on the 2780 design using RTM's. John Holman, the manager of our Projects group, will gather this data and send it to you this week.

John and I will be available to discuss other possible applications of RTM's for Special Systems applications. John is most familiar with these applications since it is his group within Computer Special Systems that is responsible for the "traditional special systems business" i.e., special interfaces, controllers, etc.

Fred Gould and I will also get together this week to discuss alternatives to the design race that we discussed briefly with Nick.

Best regards,

Brad

Bradstreet J. Vachon,
Manager
Computer Special Systems

BJV/pgh

digital INTERFOFFICE MEMORANDUM

TO: Gordon Bell

DATE: October 6, 1971

FROM: Brad Vachon Brad

DEPT: Computer Special Systems

SUBJ: PDP-16's for Special Systems

Today we met with Fred Gould and John Eggert to discuss the advantages of using the PDP-16 approach for Special Systems. During the conversation, one fact came out that overwhelmingly convinced us to start using this approach today! It appears that there is no requirement for drafting (in the conventional sense) using this approach. Most of the documentation is done by computer. Even if this is not strictly true, even a partial savings in drafting costs (and delays) would be of major importance to us.

We are going to do the following:

1. Select 5 or 6 new jobs for review by Fred and John for PDP-16 approach feasibility.

2. Start at least two of our engineers working on the selected projects.

3. Monitor costs and elapsed time in the usual manner.

4. If we don't run into any major problems we should be using this approach predominately.

Judging from the enthusiasm of Fred and John, I expect if we do run into major problems, we will be able to work together to solve them.

cc: R.L. Best
John Eggert
Fred Gould
John Holman
Nick Mazzarese
Ken Olsen
Stan Olsen

BJV/pgh

digital INTEROFFICE MEMORANDUM

TO: Bob Savell, Jim Cudmore, Jack Courtemanche, Fred Gould, Pete Durant, Roger Cady, Don White, Jim O'Loughlin

DATE: November 1, 1971

FROM: Gordon Bell

DEPT:

cc: Nick Mazzarese
Stan Olsen

Subject: The Use of PDP-16's for Constructing In-House Test Equipment

Although we have been discussing the PDP-16 to make computer special systems, it is likely that our own test equipment is an excellent applications area too. Here you are making only a small number of the devices, and you want to make them as quickly as possible, and they should be right. Possibly you also have to modify them. I suspect that while they can't be used for everything, especially where significant speed is required, they will turn out to be highly useful. There are undoubtedly more modules that are needed here to provide switch input, light and audio outputs, but they can be added in using the basic framework and would also be modular.

You will have problems with the modules, no doubt, but they do do the following:

1. It is difficult for a designer to construct a system with a behavior he did not specify. (This is completely contrary to sequential circuit design which is predicated on encoding states into a collection of flip flops. An additional flip flop,usually to get one more state, actually doubles the number of states.)

2. The behavior of the system is defined relatively formally in a flow chart. How many designs are now produced by DEC which have any higher level, semi-formal behavior descriptions? Now our designs are all defined by text + the thing (i.e., Logic diagram).

3. The design aids are such that once the design is specified in a high level input format, all the lower level details are carried out automatically.

4. So far the PDP-16 group has designed some fairly large systems in a small time. Although these claims have been confined to their performance, the purpose of this memo is to open up the issue to your group as a whole.

Let's try designing some test equipment with them.

bwf

digital INTEROFFICE MEMORANDUM

SUBJECT: Register Transfer Modules

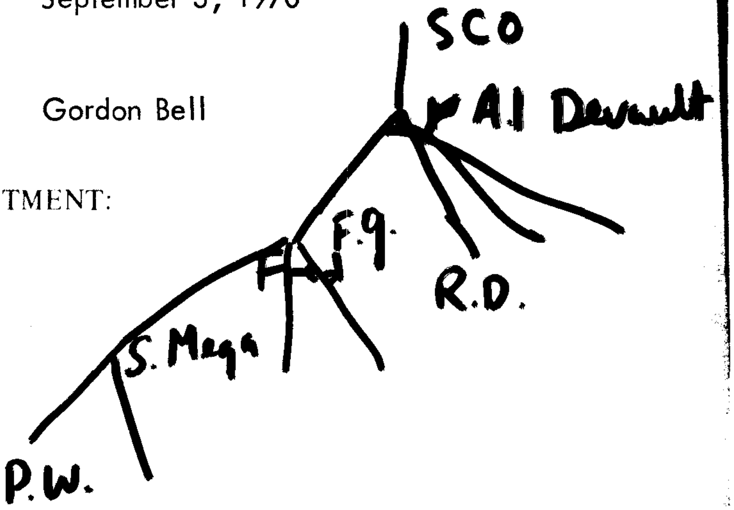
DATE: September 3, 1970

TO:
Fred Gould
Stan Mega
Bob Vannaarden
Pete Williams
Dick Best
Russ Doane
Jack Courtemanche
Al Devault
Marty Gordon

cc: Stan Olsen

FROM: Gordon Bell

DEPARTMENT:



This is an appeal to keep the fire under RTM's. After Pete returns from Houston, let's have a one day conference to decide the plan over the next few months. I would like to talk with Al Devault and Jack Courtemanche before this meeting in order to try some benchmark designs first. Possibly we should talk with Computer Special Systems, too.

Conferences and Schedule

It would be nice to have them shown at the FJCC, but failing that, a really big blast at SJCC is still fine. There are some obscure conferences, like COSINE, for all the university logic design teachers.

Papers

John Grason and I have started a paper on RTM's for the 1970 SJCC. The deadline is October 9, 1970, so we need to know the co-authors from DEC. The proposed title is Register Transfer Modules: Application Constrains, Design and Use.

Present CMU effort and intentions

We have a reasonably large effort going now here at CMU. This ind udes: Gordon Bell - Professor (part-time); John Grason - Assistant Professor - probable project director; Paulo Corullupi - engineer - full time (he and Grason are well on way to eliminating the single shots in K. simple.); graduate assistant - documentation; Sashil Rege - graduate student; one graduate student - research; 3+ undergraduates - senior research project.

We're planning a weekly two hour seminar as a vehicle for keeping everyone on the project informed. The immediate projects are:

1. Proposal to NSF for funding as a research project. We're currently operating from ARPA, department funds, plus the $25K that we had from NSF to buy the equipment in Nov. 1970. (We're constrained to spend it then.)

2. SJCC paper - announce RTM's and our intent to research.

3. RTM descriptions - We appreciate that everyone has been busy, but we'll have to try to generate documentation so that we know what we're all doing.

4. Class notes - Grason will do these this term in conjunction with the class he's teaching in logical design. He intends to introduce paper, RTM's very soon in the class. These should be of a form which could be useful by next summer.

5. Formal description of designs. We will probably start out using my languages, ISP, and PMS. We'll let these evolve to the appropriate representation for expressing designs.

6. A diagram to wiring diagram converter. We'll try using the DEC ADS which runs on our PDP-10 as the language and pictures for expressing designs. This is a relatively simple task so we'll probably do it early this term to include in the proposal.

7. Simulator. A simulator would use the diagram generator output. A user would specify input test data, and the thing would be run and give test output.

Applications Areas

Here we, (DEC and CMU) need to get going seriously. The Hycel project is significant because it shows that systems can be built with them --although with it we have a real question of whether they are economical or not.

It's clear that we have a great system for training people in computer operation and design. Are there any other markets? How are they identified? The markets that show the most promise are:

1. Special systems with relatively simple control logic, but too small to require a core memory. (Hycel is perhaps too big -- the production economics of PDP-8/E kills us here.)

2. Emulating old and other computers I've had a try at this - Data General and H.P. -- I think we just may have something here. The old computer market (e.g. PDP-1, 1401, 1620) also could be quite interesting . Here, work at CMU could be most important because the key problem in emulation is getting a good description of the computer to emulate.

3. Interfaces between computers and special systems. Perhaps RTM's are the way to build highly specialized boxes that connect to computers. (We should talk to Computer Special Systems to get an idea whether this is possible. Since the cost of their systems is mostly in the engineering, we might just win by being able to cut engineering time from a month to a week or less.)

   Out of necessity we have an 11 Bus to RTM converter to use 11 peripherals. For attaching special systems to an 11 we need an RTM to 11 Bus converter -- also 8 Bus to RTM.

Packaging

What's going to be done here? This should get worked out soon. We need some ideas badly.

The CMU Research Proposal to NSF

We see the need for a research project to work at this level of hardware design. Such an effort would look at topics like:

Design of individual modules (combinational and sequential circuit design).

Design of a "set" of RTM's designed for a particular class of tasks.

Microprogrammed controls for RTM's.

Emulation of a given class of computers using RTM's.

Comparison of RTM with Clark's Macro Modules and Patil's Pipeline Modules.

Design automation (language, graphics, simulation, manufacturing, etc.).

Introducing these modules into the logical design culture: texts, analytic means, etc.

bwf

```
*****************
*   digital   *
*****************

TO: ENG STAFF:                          DATE: MON 15 FEB 1982 6:55 AM EST
    JACK SMITH                          FROM: GORDON BELL
                                        DEPT: ENG STAFF
                                        EXT:  223-2236
                                        LOC/MAIL STOP: ML12-1/A51
```

SUBJECT: TASK FORCES, COMMITTEES; NOD; C-I T/F; PRODUCTIVITY REV.

I just read the minutes of two meetings of a task force called Customer Installability. It is not a task force it is a sewing circle consisting of 21 people! If there weren't 3 people there who I know have real work to do and have done good work, I would ask that we simply dismiss the whole group.

The minutes contain no real information on the subject. We already have a spec on what CI is, and we have to do some work on products to get it. This is not the work of a committee.

My point, I would like you to come forward with a list of the various committees and task forces, etc that are working within your group during the productivity review. I don't want to look at them, but I expect you to have, and I want to know that you understand what's going on in your area.

I believe 1/2 of these people could be let go from DEC today and our productivity would take a sharp rise. If this is the case, I would like to have their names and since we have the reputation for never firing anyone we can put them in a new group I propose we start called NOD (No Output Division) where they won't take time from people who have real work to do.
PS
I'm quite serious about NOD. Since it is so difficult to get rid of people, I want to make us at least not have them mixed in with the workers and suck up good people's time.

15-FEB-82 06:55:06 S 31987 BURT

THE TECHNOLOGY OF THE COMPUTER*

Gordon Bell

DIGITAL EQUIPMENT CORPORATION

Computers have been genealogically described [see Fig. 1] starting from the first generation, with the start of each subsequent one becoming less precise.

The first, vacuum tube technology, began with the Cambridge EDSAC in 1946 and was used until 1959. The second generation was marked by the introduction of a single device--the transistor--as a component. After 1965, computer historians indicate a third marked less dramatically than the first two. Its beginning was the encapsulation of several transistors into a single silicon area to form one basic logic element. More recently, it has been stated that in the 70's computers entered a fourth generation, marked by extreme miniturization: a complete processor, memory, or other functional element with several thousand transistors, is placed on a single silicon chip with an area of 0.2" square (costing about one dollar). The number of binary digits (bits) stored on a single silicon die has been observed to be 2↑t, where t is in years since 1962. Such growth cannot go on indefinitely. If this is a new generation of computers, it is likely to be the last of this technology. Optical information processing may be used in the far future.

The latest computing machines are fabricated in roughly the same way as baking cookies. Although the recipes may vary, they will both become ubiquitous and uniquely recognizable but not necessarily notable. In fact, only humans can be fabricated with less process explicit control and sometimes less forethought.

As computers themselves are incorporated into other aspects of our everyday life--from the telephone system to transportation systems to monetary systems--their presence will be no more remarkable than the batteries or motors we rely upon. They will continue to develop and change, but many of the major innovations have been made which sharply differentiate one generation of computers from the next.

The sequence of development of stored program computer technology, has several parallels in the early mechanical computation, particularly as developed by Charles Babbage, an English mathematician. In 1823, he was the first recipient of the Gold Medal of the Royal Astronomical

Society, for his work "Observations on the Application of Machines to the Computation of Mathematical Tables". Once, when involved with the checking of some astronomical calculations, he was quoted as saying, "I wish to God these calculations had been executed by steam." It is important to note that he thought instinctively of steam, the technology on the verge of becoming usable. For even though the steam engine was developed in early 1700's, and Watt made the first efficient engine about 1770, it wasn't until 1800-1825 that the steam engine was employed in the locomotive. This coincided precisely with the time when Babbage formulated his ideas on computing engines. Babbage was quick to adapt a variety of technologies. His second "Analytical Engine" used cards--an idea taken from the card-controlled loom of Jacquard--to hold a sequence of operations and its variables. It had an internal memory to hold intermediate results.

Stretch the Technology

The machine required more precise gears than could be manufactured at the time. This involved improving the technology. Although it was a sidetrack, Babbage detoured, and the result was a book, "Economy of Manufacturers and Machinery."

In a similar way, the premium on physical size, weight, and reliability by NASA has had a very positive effect on the improvement of logic technology, and forced the development of the third computer generation at a much faster rate than would otherwise have occurred. However, once stimulated, the market demand takes over and the process is self-sustaining. A current by-product of the third computer generation is the hand-held calculators which are available for under $50 and cost about $10-20 to build. They will be extended, to rival the standard computer costing over $10,000.

Starting New Projects With The Newest Ideas

Babbage's "Analytical Engine" is recognized to have the same structure as a modern stored program computer. But, neither the "Analytical Engine" nor the "Difference Engine" which preceded it were carried to completion. Each new machine idea captured his major attention so that none of his machines reached full operational status and did not have the impact they might have had.

The early computers constructed at various universities have run a close parallel. The most famous, and most successful was ILLIAC, the machine built at the University of Illinois in the early fifties which was a relatively pure engineering embodiment of the Institute for Advanced Studies--(Von Neumann) computer. About a half dozen replicas were constructed and used at various institutes and universities around the world. Then ILLIAC II was conceived with the goals of stretching circuit speed, using transistors, testing asynchronous logic ideas, and providing the university with a very large computer. Although it achieved many of the goals, it was too late to be useful to the engineering community. Academicians lagged significantly behind industry by the time of ILLIAC III, which was designed to test peripheral optical processing for the Atomic Energy Commission, which additionally needed a large computer which also had to be designed. Due to the large number of variables, ILLIAC III did not reach production operational status nor adequately provide a test for the optical processing part.

Finally, ILLIAC IV, which was to be the world's largest computer, was conceived in the early 60's as the Solomon computer, to be operational in 1969; but it is still to be scheduled for useful work.

In general, while many early university-built computers must be judged as unsuccessful in a production sense, nearly all have produced significant side benefits to the development of computing.

Government Funding For Computers

Babbage like many modern-day computer hardware designers needed funds beyond those available to him as professor in absentia from Cambridge. He found funds through the government earmarked to build a specific machine to calculate nautical tables. This early science-military grant was similar to the current ones and there was a difficult interface between Babbage and the government.

Similarly, the first generation of modern computers, which is often called the Von Neumann computer, was funded and used by the US Army for computing firing tables; thus it had a particular mathematical computation orientation. Much of later computation has been similarly funded because its expensive developments are only justified if the benefit is clear cut, direct, and seems to be very large. For example, the SAGE Air Defense computer was adapted and formed the basis for commercial air traffic control.

Diversion

There is a third and last parallel between Babbage's time and our own. Babbage found there were interesting peripheral problems to solve that drew him away from his central purpose. For example, he studied and devised a uniform cost for mail, independent of distance, and today computer scientists find themselves working on diverse problems (for example, NSF has more joint research programs with the Office of Computing Activities than any other program). While there are many examples of computer science working on peripheral problems (e.g., speech synthesis and recognition, printing), the most progress can be observed with respect to the effect on digital communications. Computers have required reliable, low cost, high data-rate communication links for connecting user terminals to computers and for interconnecting computers (i.e., networks). From the early 1900's until the late 1960's, teletypewriters and their communication links were limited to about 100 bits/sec. Most recently, the link capability has been extended to 300 bits/sec without special line conditioning and to 1200 bits/sec with line conditioning. Also, good high speed links are available at speeds of 2400 bits/sec to 9600 bits/sec and special 50,000 bit/sec links are eminent. Thus more information is transmitted with the same resources.

In a similar way, computers have instigated and been utilized in a store-and-forward fashion to build networks (Bell, 1974). For example, with the 100 site ARPA network computers can call one another for very short messages (e.g., 1000 bits) and avoid the customary 20 second dial switching delay.

In fact, Babbage established a tradition which has remained in computing: he offered only promise over then existing calculation methods; the market (use) and its requirements were not fully defined so that they came after, not before, development; other people's problems were tackled to the detriment of his own work; his machines were not finished on time; he tried to build with technology well beyond the current state of the art; and he depended on government funding which was accompanied with promises, delays, and overruns on both sides.

Computer Industry Computer Use

One of the more advanced industries that uses computers is the computer industry itself. In fact, as a computer manufacturer, we are our best, albeit least preferred, customer; and perhaps this would seem natural. The obvious reason, familiarity, is not the main one. The growing number of computer scientists and engineers within the industry do, however, contribute to increased internal use as well as the external stimulus of need creating products which are marketed to a variety of external users. The users also stimulate improved use by encouraging competitive operating systems (and other programs). A by-product of better computer use is the understanding of resource allocation in a multiple commodity environment; this turn may be used eventualy in other disciplines (e.g., economics).

Finally, computers can be used most easily by the computer industry itself for computer design and for software production, because the objects with which it deals (information encoding numbers and symbols) is directly represented by machines. Representing an automobile or a building within a computer, together with appropriate operations to manipulate them abstractly is more difficult. For example, a computer thereby can be simulated and even manipulated by algorithms. The abstraction of a hardware machine with memory, registers, an instruction set, and mechanism to interpret the instruction set maps directly into arrays, variables and a sequence of statements of conventional programming language (e.g., FORTRAN). Contrast this with the notion of representing a primitive component such as a wooden beam, which has many attributes of relevance in house design (e.g., length, and orientation), and an open ended set of operations (e.g., sawing, painting, nailing, support of floors and walls) that are difficult to represent abstractly.

FUTURE COMPUTERS

Computer technology is improving at a yearly rate of 25 to 40%. In its 26 years of existence, computing performance has increased by a factor of 10↑5. While there are probably limits to such growth, it is not likely that they will be reached until after 1980. This means that every 2 years, the performance available for a given cost doubles; or as a corollary, every 2 years the price for a given level of performance is halved. In all cases, more computing is done for the same amount of money. Figure 2 illustrates the change in cost for minicomputers as a function of time for the last 15 years, and Figure 3 shows the cost of several minicomputers plotted against performance. If we assume a completely elastic demand for products, then each year, as the price is lowered by 40%, many new uses are possible--because useage is a function of the relative cost of computing. In some cases, this market demand appears to double, each time price is reduced by 28%.

The important number here is the 40%, because it really gives an upper limit to what we might expect in the future. The futurist who believes that a 1 to 2 million dollar computer of today will sell for $5000 in 1980 is, I believe, overly optimistic. At the 40% rate, or a more conservative 25% rate, this will mean a decrease by a factor of only 5 to 10, which means the computer will still cost $100,000. To give a more realistic example, Dennis and Smith (1971) of IBM projected a cost of about $100 for a 10,000 word minicomputer in 1980.

However, instead of using time to measure technological improvement, a much more precise measure is the number of computers produced. Since computers have been produced at an exponential rate, the time and number measures are identical. Technology, measured in this way, has been shown (Fusfeld, 1973) to be roughly of the form: Ti=ai↑b, where

T = Technological level of ith unit produced
a = level of first unit
i = the cumulative number of units produced
b = technological progress constant

For computers, the technological progress constant has been about 2.5 (using memory size x memory rate as the technology level measures) which can be contrasted with (at most) 0.7 for recent automobiles (using horsepower as the technology level measure) and 1.06 for jet engines (using thrust as the technology level measure). Note that the technology level for computers is a factor of 10↑15 improvement over 100,000 machines, whereas automobile horsepower improvement has been only a factor of 30 over several hundred million units with a trend to now decrease horsepower for fuel consumption reasons.

The numbers of computers have not yet begun to reach fundamental growth limits, unlike the numbers of men or automobiles. So far computers have barely reached the 100,000 level and, since growth rates are exponential, they will reach several million by 1980. But this will not be based on the waste technology or on changing styles so that the customer can boast of owning "the latest model". Computers can remain almost as current as the information given to them. Obsolete information is simply erased and does not pollute the atmosphere.

From an overall computer system design standpoint computer components are continually reassembled in new ways--older machines become secondary processors to new ones in a rejuvenating fashion--or else placed on the used computer market which is a thriving business. But computers have exceedingly long lives, based on parts that seldom wear out. Two machines, I designed in 1964, are still in operation today: one controls making cookies at the rate of a dozen freight loads of flour a day, and the other controls the nuclear reactor for a power station. Both continue to perform their tasks better as they are finely tuned to the situation.

The growth in numbers of computers will be further spurred on by reductions in the cost of their components. Costs of the primary memory, which holds the instructions for processing the data held in the secondary memory, have been decreasing by 30 percent per year. A similar reduction is occurring in the secondary memory which is used to store the data bases. Although there is little advance as yet in the evolution of better whirling objects, such as drums and disks, this is more than offset by increasing capabilities to store information more densely, so that the net cost/performance ratio is decreasing at a rate of 27% per year.

Since the secondary memory is akin to file storage, computer memories, like the human mind, derive power by the amount of information they have stored and their ability to process it quickly. There are at least two competing technologies promising to provide a significant breakthrough by replacing the whirling mechanical parts with electrons.

The tendency is for the computer--as a piece of machinery--both to become more compact and cheaper. This is the dimension of its hardware. As a compliment, there are software developments in the art and science of using the computer.

*taken from a larger article submitted to American Scientist, entitled: THE INTERACTION OF TECHNOLOGY WITH COMPUTER SCIENCE.

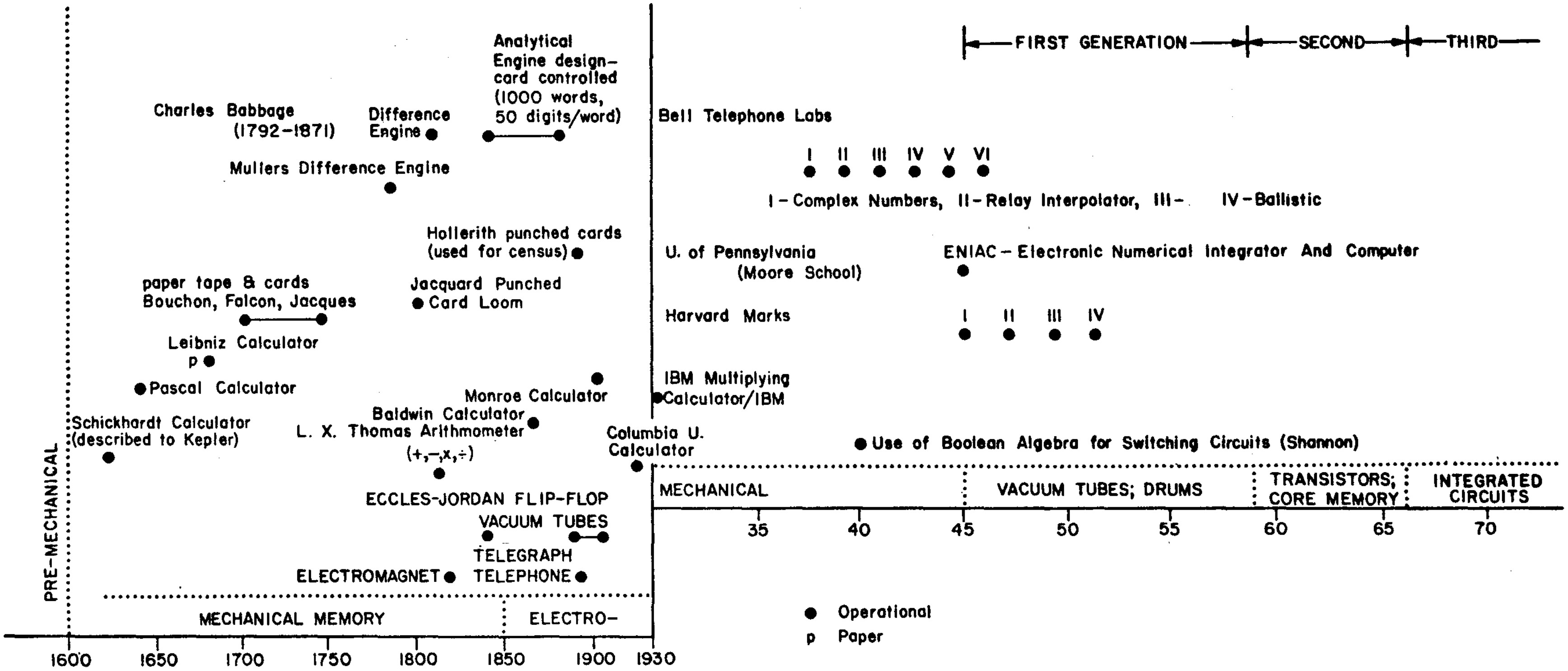


FIGURE 1 COMPUTER GENEALOGY

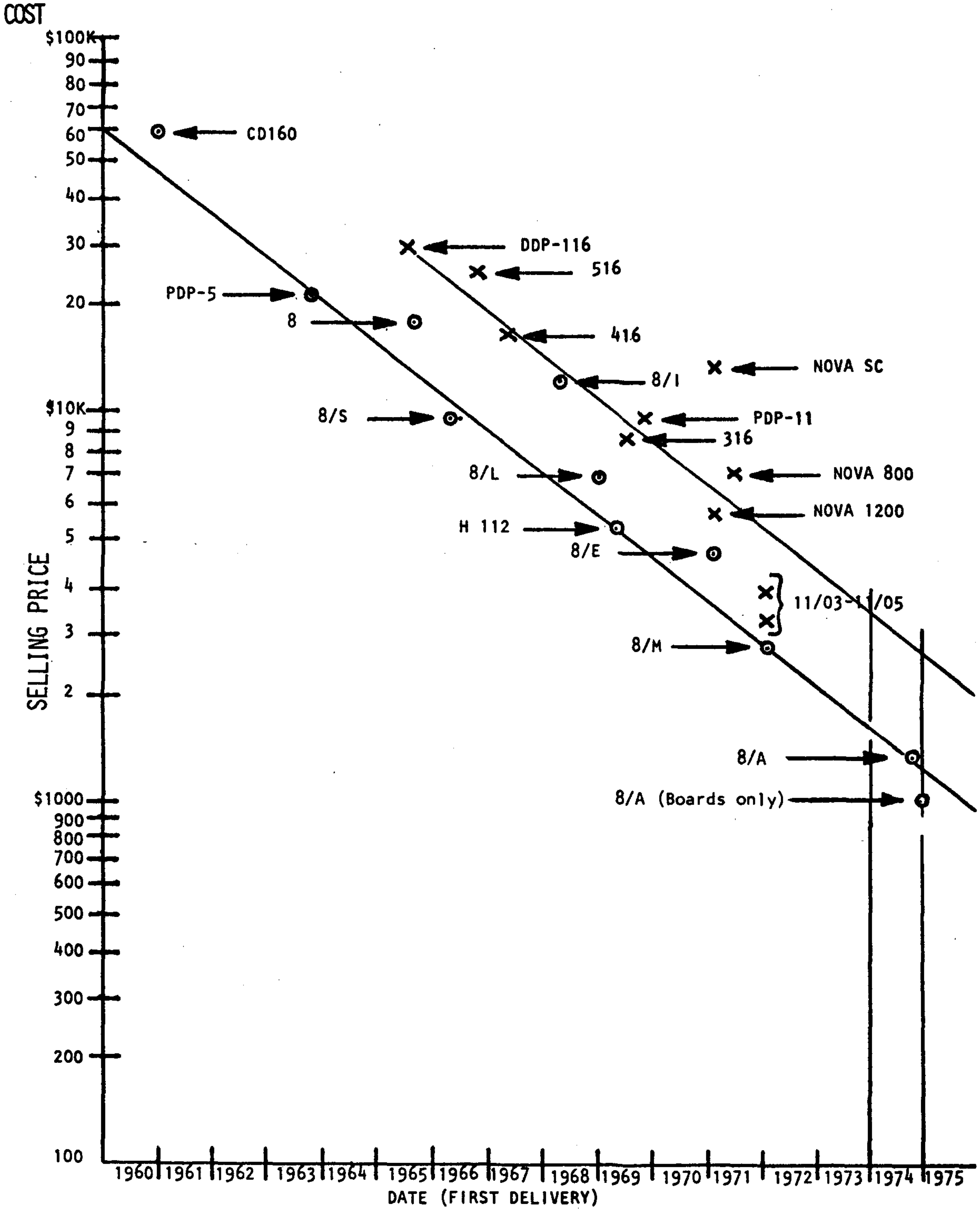


FIGURE 2 SELLING PRICE VERSUS FIRST DELIVERY DATE FOR MINICOMPUTERS

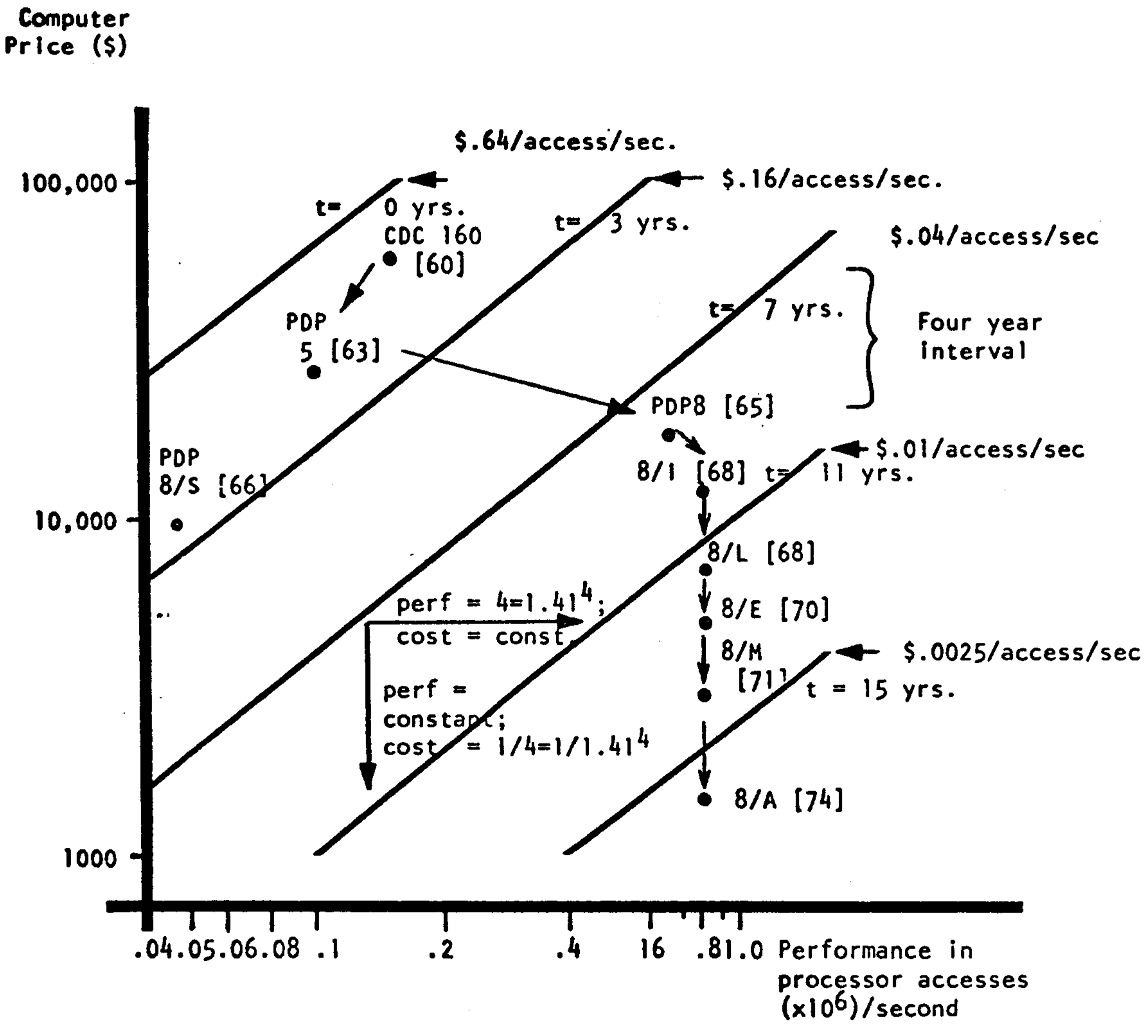


FIGURE 3 MINICOMPUTER PRICE VS PERFORMANCE FOR VARIOUS TECHNOLOGIES. LINES OF CONSTANT COST/PERFORMANCE ($/ACCESS/SEC) ARE PLOTTED FOR EACH FOUR YEAR'S (FACTOR OF 4 = 1.41^4) ASSUMING IMPROVEMENT OF 41% PER YEAR

BIBLIOGRAPHY

Bell, C.G., MORE POWER BY NETWORKING, IEEE SPECTRUM, Feb. 1974, pages 40-56, Vol. 11, 2.

Dennis, S.F. and M.G. Smith, "LSI; IMPLICATIONS FOR FUTURE DESIGN AND ARCHITECTURE", IBM Research Report, RC3598 (#16278) Nov. 3, 1971, IBM Research, Yorktown Heights, NY, 26 pages.

Fusfeld, Alan, R., TECHNOLOGICAL FORECASTING, TECHNOLOGY REVIEW, Feb. 1973.

Gordon Bell
April, 1974

WHY DO WE NEED HARDWARE DESCRIPTION LANGU~~AGES?~~

In writing the book, COMPUTER STRUCTURES, Allen Newell, and I used 2 notations to describe and analyze computers, PMS, a 2-dimensional representation, is for the block diagram, physical structure (processors, memories, switches, terminals, links), with extensions for lower level structures such as logical diagrams, ISP (for Instruction Set Processor) describes the instruction set precisely,

The notations have been used in several other ways;

0, The ISP descriptions in the above book, have been hand translated to programming languages for simulation,

1. Michael Knudsen built a program PMS for use in computer structures design; the system computes reliability and performance parameters, Extensions will compare machines and test valid computer configurations,

2. Mario Barbacci built a programm which accepts ISP and carries out various design activities for a specific set of register transfer level modules,

3, ISP has been extended for register transfer systems (hardwired and microprogrammed control structures), design although its need is unclear,

4, ISP was used to describe the DEC PDP-11 in its design phase, and in the programming manual, Since this description is supplementary to the conventional text description, users of the manual have not damned the description, nor are we overrun with letters of praise, Through lack of support, descriptions of future DEC computers will probably be more conventional--with no formal descriptions--simply to save trees and cost,

5, A set of Register Transfer Modules, called RTM's (PDP 16), were built by DEC, PMS was used for describing structure, while a flowchart form of ISP was used for control, Here we needed and use description languages, including software for processing the designs (including simulation),

All of the above uses (except 3) stem from need,

I believe there is little need for the conventional 1-dimensional hardware description languages typified by the plethora of register transfer languages. These seem to come from the need to invent a language and write a simulator. I have seen little actual use, even the texts that posit and promote these language inventions give no real (not toy) machine designs. For logical design, block diagram symbols for the elements (gates, flip-flops, etc.) and the corresponding logic diagrams are better than a 1-dimensional text (eg, Boolean Algebra) or a description language to conceptualize designs. The diagrams are sometimes converted to a 1-dimensional form for logic simulation, but the register transfer language is unsuitable for describing the logic and doing the design. For register transfer descriptions, flowcharts (again 2 dimensions) are usually preferred for showing hardwired and microprogrammed control flow. Again, these flow diagrams are compressed into 1 dimensional text to assemble microprograms into binary words, and occasionally for simulation. When the conversion from flowchart form occurs, it is easy enough to use or modify a conventional assembler; and for simulation, a conventional software register transfer language such as ALGOL, BASIC, FORTRAN, or PL/1 is adequate (and preferred because it's better known and such a program executes substantially faster.

If the 1-dimensional, register transfer language is not for the logical designer, the machine designer, microprogrammer, or system software writer who uses a simulation of a machine, then who is it for? Students, who should know that systems can be represented in various ways? Why can't they use a programming register transfer language (eg, Fortran)? Until graphic displays are more universal, these languages will fall short of the block diagrams and flow charts currently in use.

There is need for machine representation for design and checking aids, automatic compiler, and systems program writers, comparing, designing, and configuring machines; but these have not been in the domain of the typical hardware description language designer.

A NEED FOR HARDWARE DESCRIPTION LANGUAGES?

I believe there is little need for any more conventional hardware description languages typified by the plethora of register transfer languages. These seem to come from the need to invent a language and write a simulator. The problems addressed by these languages have been adequately addressed by existing languages. Also, they are still at too low a level to address the problems that are met in real design. I have seen little actual use, even the texts that posit and promote these language inventions give no real (not toy) machine designs.

Taking the concept of a language in a broader sense, a graphical representation or "graphical language" has proven to be more useful than a conventional programming printed (or text) language. For logical design, block diagram symbols for the elements (gates, flip-flops, etc.) and the corresponding logic diagrams are better than a text (e.g. Boolean Algebra) or a description language to conceptualize designs. Similarly, the flowchart remains the tool of most hardware designers. This has yet to be incorporated in a formal way within hardware description languages.

Finally, conventional software register transfer language such as ALGOL, APL, BASIC, FORTRAN, or PL/1 are generally adequate because they are better known, available, and execute substantially faster. With these languages, a system designer can (at last) think in terms of hardware-software tradeoffs. However, such languges may have to be extended to express concurrency, time delays, and other hardware constructions.

There is also a need for machine representation for design and checking aids, automatic compiler, and systems program writers, comparing, designing, and configuring machines; but these have not been in the domain of the typical hardware description language designer.

There is a need for work leading to better hardware description languages, but until the work is done, there are many languages available to use.

Gordon Bell
October 14, 1974

Digital Interoffice Memo

Subject: World Model of Computing Based on Amount of File Memory

To: Distribution

Date: 30 NOV 76
From: Gordon Bell
Dept: OOD
Loc.: ML12-1 Ext.: 2236

(F/U 12/14)

The attached graph is a guess, note no scales on distribution, as to how problems are structured based on memory data base sizes. Note, by coupling this with an earlier price model where cpu primary memory size determines the function (or amount of multiprogramming (and a memory hierarchiey model)), these two graphs could be used to impute the potential market size for all systems.

What's your estimate for shape of curves?

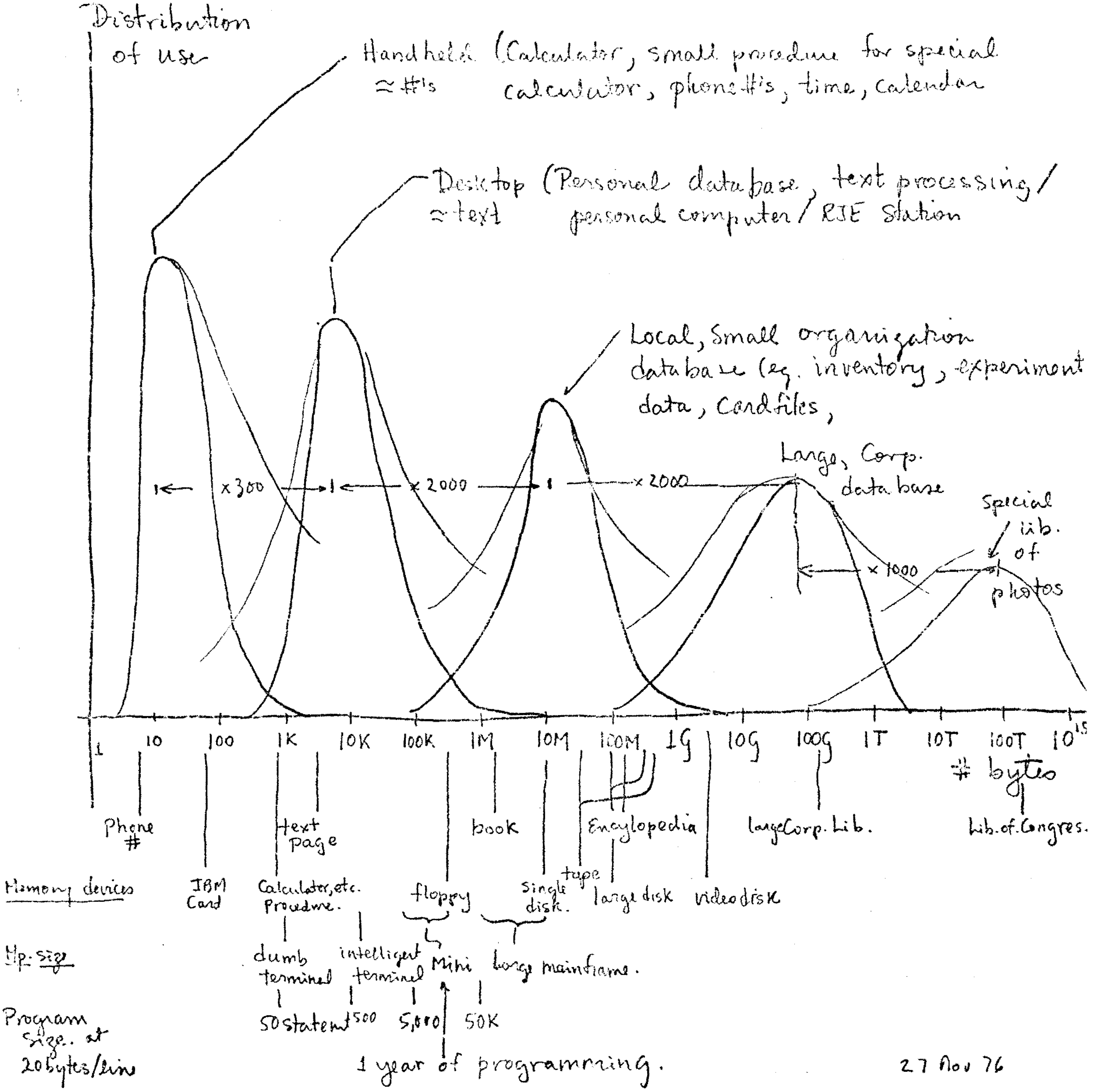
What are they centered about? Use? Technology?

GB:ljp

Attachment

Distribution
OOD
Brian Croxon
Stan Pearson
Grant Saviers
Steve Teicher
Jim Bell
Mike Gutman
Bob Peyton
Bill Strecker
Mel Woolsey

Fig. — Distribution of Use for various memory sizes.



27 Nov 76

ESTABLISHING NATIONAL HIGH TECHNOLOGY INDUSTRIES (E.G. COMPUTERS) Page 1
G. Bell 10/30/81
Last Edit: 10/26/79
Latest Edit: 11/7/79 PAPER2/2

Advice to Brazilians

Written to Brazil

cf

Gordon Bell
Vice President of Engineering, Digital Equipment Corporation
Professor of Computer Science and Electrical Engineering (on leave)
Carnegie-Mellon University

with Gwen Bell
Social Science Editor, Pergamon Press

Computers are at the center of current industrial growth because they are tools, for modern communications and manufacturing, and components of nearly every industrialized product. For example, in the automobile industry computers, are used in design, in operating the organization, in manufacturing control, and for use within the car itself. Over the last thirty years, computers have evolved at a rate more rapid than any other object in the history of civilization. But national economic policy-making still relates to industries that change very little. Effective policies for innovative, knowledge based industries have not been formulated because the key to policy-making, the computer and its constitutent technologies have not been understood.

Although the computer industry is high technology, it is also appropriate technology. Computers do not despoil environment, degrade populations, or use undo amounts of energy. Furthermore, they are critical to the development and understanding of national and local resources. Analysis of such complex issues as the proper utilization of the Amazonian forests, the control of agricultural pests, and ecological management of wildlife, depend on computational power. Without using such tools, it is easy to make wrong choices, since the full range of alternatives are not clearly understood. But, even of greater importance, is the significance of the introduction of a computer industry in order to maximize the talents of the population.

The computer on a university campus -- and a computer science department -- are becoming as central to learning and development as mathematics and physics departments. Computer science fundamentally affects all engineering, management sciences and ultimately social sciences.

In establishing a computer industry, the foundations of the new technologically based society are laid, and the development of an information-based economy can be established. Without a computer industry any country is doomed to a backward, poor economy. Fear, greed, and strong national interests have led to policies for establishment of computer industry that have had counter-intuitive effects, slowing down rather than hastening the process because assumptions have been made based on conventional instead of rapidly evolving, high technology industrialization. In order to describe appropriate policies for the development of a computer industry, the nature of computer evolution is first explained.

The Ongoing Computer Evolution

The computer has evolved more rapidly than any other man-made object as measured by improved price for a given computer or by improved performance at a given price. By either of these measures, the rate of improvement is 20 to 30 percent per year. This means that if the price of a given system at year

(t) is 1.0, at year (t+1) it will be 0.8, and at (t+2), 0.64, and (t+3), 0.51. Every 3 years, the price halves. These measures have held constant for 20 years, and do not even take into account inflation. For example, in 1970 a boxed, PDP-11/20 with 4096 word memory sold for $9,300; in 1976 a large scale integrated version was introduced that sold for $1995. This price reduction amounts to a yearly price reduction of 23% compound annually. In contrast, nearly all consumer goods, such as automobiles, have increased in price at least an inflationary rate! The consumer price index and the prices of constant function computers are plotted in Fig. 1.

Computers performance has improved roughly eight orders of magnitude during the 30 years of their life. Logic and memory technology generations mark this improvement:

1. Vacuum tubes/drum, electrostatic memory 1948-1958;
2. Transistors/core memories 1958-1966;
3. Integrated circuits core memory 1966-1974;
4. Large scale integrated circuits for processor-on-a-chip computers/integrated circuit memory 1974-1980;
5. Very large scale integrated circuits 1980-

Computers not only have become all pervasive but have affected many other rapidly evolving technologies, including semiconductors, magnetic recording, conventional electronic sub-assemblies, xerography, video display, process control and conventional communications. Computers also require "software" technologies ranging from operating systems which administer computers to conventional computer languages in which application systems are written.

Computer components are ordered by what is known as levels of integration. The highest level is the application and the lowest is the physical device used to hold or process information. The structuring of the levels in Table I starting from the top applications (use) or starting from the bottom with physical devices form the basis of developing a computer industry either through backward or forward integration, respectively.

TABLE I

LEVELS OF INTEGRATION

Application software -- a particular use;

Application components library software -- generalized parts for building various applications quite often implemented as a special language (e.g. the COGO language for civil engineering design);

Basic software -- includes standard programming languages (e.g. BASIC, COBOL, FORTRAN), data management; network, and operating system software for building any application;

Basic Hardware System -- all the hardware of the system (e.g. an IBM 370/148);

Computer Component Hardware Options -- packages of boxes or cabinets, such as disk and tape units, terminals, memories, processing units, etc.;

Printed Circuit Modules -- holders of the semiconductor circuits;

Semiconductor circuits in a single integrated circuit array package- the smallest physical component from which the computer is formed.

Two additional concepts are important in understanding computing: architecture and implementation. Computer architecture is the interface of a machine as seen by a program or programmer, and is an important standard because it provides the user with a constant base for writing (investing in) programs; examples include the IBM 360, and the DEC PDP-11. Although there is a temptation to change the architecture to fit the technology or the application, changes in a given architecture and among several architectures produce relatively small changes (a factor of 4) in cost or performance. Most changes in architecture have occurred by having to change the amount of memory that a program is able to access. Address size (roughly corresponding to brain cavity size) is the single parameter of architecture that must be attended to, because it is the limit on how large the computer can be.

Implementation is the building of a particular computer model, (e.g. an IBM 370 Model 158) using a specific hardware technology, that operates according to the architectural standard. It is important that there be many implementations of a given architecture over a wide price range (typically 100) so that a user may select the appropriate price and performance model for a given application. In a similar fashion, it is important to implement various models of the same architecture with evolving technologies over several computer hardware generations so that user software investment and training is protected.

Since computers can supplant every other form of information processing, storage, transmission, switching, control, and processing, broad technological expertise is required. Software engineers not only must know the subject of the application (e.g. manufacturing, civil engineering, communications) deeply and unambiguously so that it can be made algorithmic but they also must know the computer almost as well. The lack of a standard architecture requires learning different machines. But with the agreement on standards, the potential for both substantive problem and computer understanding can be greater helping to overcome the current manpower shortage of software specialists currently limiting computer use.

Policy Dimensions

Nearly all governments establish policies which they feel will allow computer industries to form and thrive. There are many alternatives ways to establish a computer industry. Three important questions usually are the focus:

Who's allowed to supply computers?
How is the supplier allowed to obtain the product?
What is the supplier allowed to supply?

Although there are a wide spectrum of answers, concentration is placed on the extremes of each dimension. A total of 2 x 3 x 3 (18) alternatives are spelled out followed by scenarios of three policies.

Who's allowed to supply computers?

From a nationalistic viewpoint, the most sensitive, political and seemingly important dimension is who will be the supplier — foreign or multinational firms, government, monopolies, and private individuals etc. From the viewpoint of the final result, this decision is relatively unimportant except as it affects the selection of the manufacturing and computer standard policy.

Government ownership of computer industries have so far been singularly unsuccessful (e.g. the Eastern Bloc), and hence not even worth considering as an alternative. The issue of whether there be local or multinational ownership also seems less important, except as it relates to limit the ability to use a particular standard. Hence, only the two extreme cases will be considered:

P1. Free market; and
P2. Government sanctioned monopoly.

In the former case, any group who is willing to live within the various governmental constraints is free to engage in the market. A free market is likely to introduce many standards, hence cummulative learning about applications standards will be minimum, unless each product is segmented into a particular application.

A government sanctioned monopolistic supplier is often given a particular market segment. Because of the complexity of computers with many components evolving at different rates, the static nature of governmental decision making to arrive at costs and prices often mean that the equipment decided upon is not only obsolete but also more costly than newer machines. The computer industry is no exception to the rule that monopolies are likely to be most costly, with the lowest rate of evolution.

How's the supplier allowed to obtain the product?

Like all organizations, countries are especially concerned with how products are obtained. The pressure to obtain state of the art products, independent of where they originate, is seen to exacerbate national balance of payments problems. The main issue is how much of a computer should be imported and then these imports could be reduced in the future.

While it is extremely important and nationalistic to believe that, de facto, a computer should be manufactured locally. Appendix 1 spells out how a strategy of local manufacture can require more imports for component parts than if a later model product were fully imported! Hence, importing finished goods is the best alternative for many countries.

Given, that local manufacture appears feasible in terms of market, manpower and capital, then either forward or backward integration can be used to evolve to full manufacture of all levels (so far only attained by U.S. and Japan). Forward integration requires a strong technology base, and for computers has been demonstrably impossible except in the U. S.

Therefore the alternatives for Manufacturing are:

M1. Import complete products;
M2. Backward integration, starting with the user application; and
M3. Forward integration, starting with components.

What is the supplier allowed to supply? (Standards and the Product)

In the section on the On Going Computer Evolution the architecture and implementation concepts that underlie the product were described. There are three alternatives describe a range of standards control:

S1. Use established, de facto "industry standards";
S2. Design and evolve a unique, indigenous vanity architecture; and
S3. Use any product in a non-standards fashion.

In the first case, there is a wide array of hardware and software components in the marketplace that can be potentially used (i.e. bought, copied, licensed, etc.). The standards are de facto because of their strong marketplace position and as such there is an alternative source of supply to the originating organization that provided the computer. (The so called plug-compatible industry). The temptation to build a "national computer" is so great, and the results so deleterious, that this special case is described in a section below. The third case ignores the standards question and permits free use of what ever products happen to make it into the environment. It's effect is less clear because the market can structure to automatically provide "the standards" by rejecting the non-standards.

Establishing a Computer Industry via A Forward Integration Manufacturing Approach

The conventional manufacturing approach for establishing an industry usually cuts off external supply by sanctioning various firms to operate as monopolies and builds up internal manufacturing via licenses, joint ventures, and favored manufacturers. It is a "bottom up" process. Only essential components and manufacturing equipment are imported. This approach is most successful when the evolution rate is low (i.e. automobiles, tv sets, radios) or where the ultimate goal is world market domination. However, in either case, the essential first step is the manufacture of all components, raw materials, and in some cases the equipment to manufacture components. For example, the Japanese who for 25 years have had the goal to dominate in computer manufacturing, lacked the internal manufacture of critical components (e.g. semiconductors and magnetics) as well as software technology until recently.

In essence, Japan switched from forward integration to backward integration to become successful. If Japan, or any other country, starts with the goal of internal computer manufacturing to limit imports, the flow may well become increasingly more negative due to increased reliance on critical outside component and software suppliers. For example, had a country engaged in manufacturing transistorized and MSI based calculators in 1975 with imported semi-conductor components, in 1978 it would be possible to obtain the complete calculator for less than the imported component cost because with each new generation radically different parts are used. In a similar case relating to computers, more imports of raw materials occurred to build expensive disk memories and computer systems that were obsolete on completion.

Virtually every country that has operated a protected, computer industry (except recently Japan) has paid a significant price in terms of both imports and price to users. Computing with obsolete computers, costs each country scarce resources for maintainance and operations deferring critical economic and applications gains. Only the elite company owners directly benefitting from government sanctioned monopolies have profited while the country looses both technologically and economically.

Simultaneously establishing the manufacture of the critical base components, test equipment and component manufacturing equipment for a high technology product like computers is probably not feasible. Even manufacture of the lowest technology parts (e.g., printed circuit boards) is hazardous because these components may limit the final product as described in the calculator and computer components examples above. Manufacture of high technology components depends on the existence of all levels of integration listed above it in Table I. Neither state-of-the-art (i.e. cost effective or least cost) disks nor semiconductors are manufactured away from their design groups who require fertile environments (including large, modern computers) for innovation.

Since the critical resource for the manufacture and use of computers is educated manpower, their effective allocation has to be central to any effort in establishing a computer industry. If these limited numbers are utilized for manufacture, then there are few left to do the critical systems applications jobs that are necessary in the manufacture itself. Reversing the allocation starting at the highest level of integration and working down then establishes a firm base for local manufacture.

Establishing An Industry By An Indigenous, National Vanity Design

Most attempts to design, produce and sell computers either fail to meet their market and profit objectives or fail to be cost-effective over competitive alternatives. Worse yet only a small number of computers evolve over a long term and remain viable and available such that a user's investment in software is preserved. Vanity computers designed for special purpose (e.g. Military) have been shown to be particularly cost ineffective over their standard, commercial counterparts such that the distinction is finally disappearing. Although there are mahy reasons why military computers are so poor, one clear one is the long procurement cycle that guarantees the thechnology has moved a full generation during design and negotiations.

The temptation is especially great because the art of computer design (architecture) is fascinating. By not adopting standards untold resources are required to engage in hardware and software design that could otherwise be applied to implementing computers based on standards, or be applied directly to applications.

There is virtually no chance that a computer can become a standard without a very large user base (market) and a commitment of multiple implementation over a range of price and time. Furthermore the architecture must be evolved in a compatible fashion to teach the technology.

An Example of A Monopoly Based on Backward Integrated, Non-Standard Compilers

The poorest method for establishing a high technology industry is by government sanctioned, local monopolistic companies. Appendix 1 describes the scenario of such a case. Here, a monopolistic company selected a high cost, non-standard, low performance basically obsolete computer for license from a North American Company that might have failed except for its exports to its foreign "licensee". The company then promised their government the following three-year, three-stage backward integration process: importing finished goods, putting together sub-assemblies, and finally building sub-assemblies from imported circuit components under license.

After five years, the local "manufacturer" was still importing finished goods, and no progress has been made toward local manufacturing. The computer was fundamentally unable to accomplish most of the tasks that were promised. The local company has a monopoly that has cost the country roughly a factor of two or $34M in imports over the promised commitment and $50M over what could have been accomplished under a policy permitting competition which would encourage local manufacture. Also, the users have paid a factor of 5.5 times or $400M extra for equipment because of the monopoly. Also, user costs including applications programming, maintenance, and operations (e.g. power and air conditioning) are several times greater for technically obsolete equipment. Certain applications are not possible, and where possible but not available, local effort has to be expanded in doing applications.

Rather than using monopolies to establish industry, government approval of imports based on the import cost would stimulate local manufacture. The incentives to all would be clear and the system would adjust rapidly! Alternatively, simple duties of any percentage, would probably work as well, avoiding bureaucracies, hassle and loss of productivity!

Establishing A Computer Industry Via A Backward Integration Manufacturing Approach Based on Industry Standards

A policy that encourages using state-of-the-art computers to be applied in a standards-setting fashion will ultimately result in appropriate local computer manufacture. The selection of widely applicable computer standards is essential, otherwise the situation is exactly what countries that do not have a computer industry fear--manufacturer domination and non-utilitarian machines.

There are four criteria to use in the selection of the de facto standards:

1. Maximum range of applicability - germane to evolving and necessary applications. Leverage of internal resources can be gained by selecting the most appropriate machine family based on application programs.

2. The conventional metrics for cost, cost/performance, and address size. A trade-off for larger address-space may offset short-term gains in cost-performance with smaller address-space.

3. Large family range of machine models from micro to mainframe. The utilization of one family versus a variety of machine-types maximizes the learning in terms of physical implementations, architecture, and software across all system ranges from the processor-on-a-chip (often called a microcomputer), through dedicated systems for special purpose use (often called a minicomputer) and to a large, shared, central system serving many users and managed by a staff (often called a mainframe). This helps achieve a critical mass of local experts.

4. Be available from numerous suppliers in a "standards-based" fashion. Ideally, each machine in a range would be the "defacto standard" machine. A de facto standard has the following characteristics: a large fraction of installed units; well-defined system interfaces that manufacturers, users, and third-party suppliers understand; and many supply sources so that a user can build up systems by assemblying components via numerous fashions.

Countries following a standards-setting policy are assured of having the latest models available, alternative competitive sources of supply and a method of intercommunications that has lasted and will last over time since it is understood and used by many different groups. Until Japan adopted the approach of building computers to the IBM standard, its machines were uncompetitive (even in a closed market) and were ultimately withdrawn, requiring user program conversion. The backward integration path was finally followed, interfacing with many manufacturers to license computer architecture know-how and hardware technology. Ultimate success in Japan depended on five factors: an open market; use of the industry standard; selective licenses (versus licensing the non-standards); engineering near copies in a "reverse engineering" fashion; and the growth of its large internal market.

The Backward Integration Steps (Top Down)

It is the "standards" approach that provides the method for backward integration into local manufacturing via the following steps:

1. Import complete systems and assign them to critical applications. This will help attract back any computer scientists and engineers attracted by the charisma of exciting problems using state-of-the-art computers who have left the country in the so-called brain drain. During this period it is important to take advantage of the training systems now developed in North America, Europe, and Japan, just as the Japanese took advantage of these systems when their industry was embryonic.

2. Enlarge applications specialties to include special systems interfaces. Special hardware interfaces could be provided by users, the applications industry, and manufacturers. This would create the base knowledge for the ultimate design and manufacture of computers.

3. User and applications industry would begin to import "standard" alternative manufactured computer options (e.g., memory modules, disks, terminals) to minimize systems costs. Systems would form from components by having local final assembly and testing. By tnis time a critical teaching and research mass will have been reached at a significant level internally so that the appropriate computer scientists and engineers can be attracted and held. Training, research and development will be primarily nationally based, maintaining the continual need for international cross-fertilization of ideas. However, these critical nationally attained skills are necessary since many computing applications are culture-based.

4. A secondary supplier industry would develop based on both buying lower level components (e.g., integrated circuits, disk drives) and interfacing to further reduced imports. Computer options would start to be manufactured locally both based on foreign and local designs.

5. Component manufacture may be possible when the market materialize.

If a user-directed, backward integration policy were implemented, one might see the beginning of stage two within one or two years, followed rapidly by stage three. Finally local peripheral interface designs marking stage four could occur as early as four years from the time of the policy adoption. Stage five is a Function of Market Size.

During all periods the number of computers, useful local applications, and most importantly, computer scientists and engineers, who provide a strong intellectual base for the industry, would grow. Simply trying to assemble imported, likely-to-be-obsolete components with the forward integration policy defers the applications that build up a critical mass of manpower, applications, and computers.

CONCLUSION

The different approaches toward the goal of establishing a national computer industry have varying risks, costs, and benefits as clearly outlined in the appendix. The conventional approach to establishing an industry is manufacturing-directed licensing, starting with components in a bottom-up fashion and is most risky, expensive, and has been uniformly unsuccessful for computers. The ultimate goal of indigenous design following the bottom-up approach called "forward integration" is implicitly predicated on slowly evolving standard components. In contrast, for rapidly evolving or high technologies with snort component life, a "top down" user-based approach, starting with the application, categorized as "backward integration" is probably best. The computer industry falls into the second category since many rapidly changing disciplines and technologies are required for building and using computers. By initiating a policy based on the second approach a country can establish an appropriate computer industry provided it is based on standards. It will become self-sufficient quickly, and with less imports than by taking the forward integration approach. Futhermore, it can be shown (Appendix 1) that the forward integration approach can require more component imports than a fully assembled computer because the technology evolves so rapidly!

Backward integration necessitates the selection of one or more standard computer families. However, is desirable to not segment and control the market by size because emerging distributed processing systems are built more easily from a single general architecture. Furthermore, since computer prices decline rapidly, a computer characterized in one class now will enter a new class in a few years. The adoption of an "industry standard" allows rapid take-off in computational ability and the selection is based on four criteria: 1) wide range of available applications programs enabling immediate effective use; 2) cost-effectiveness and expandability as shown by various metrics including address space size; 3) availability of a family range from micro- to mainframe computer so that a small number of architectures (hopefully one), maximizes training, permits alternative computing styles to fit various problems, and results in a maximum cumulative learning curve; and 4) compatibility and accessibility through numerous suppliers for peripheral equipment and software.

Appendix 1 - Case Study of Santioned Monopoly, Free Market Import and Backward Integration Strategies

In one country a sanctioned monopoly was set up to license and manufacture what was basically an obsolete computer. The following analysis of a five year period shows, in principle, the situation and compares it with scenarios based on different policies. It neglects any user loss of productivity because of poor computers, and any import duties (since the licensee was given duty free status).

Four cases are compared:

1. Monopoly (actual) - no manufacturing was achieved and licensee only imported finished goods.

2. Monopoly (plan) - the monopoly was to have imported finished goods the first year, put together sub-assemblies the second year, and assemble the sub-assemblies from components the third year.

3. Free Market Import - No controls, are assumed. The most cost-effective system is imported.

4. Free Market Import with Backward Integration - Case 3, except a policy (e.g. duties) which gives preference to minimizing import content is instituted. In the second and third year sub-assemblies are put together locally and in the fourth and fifth years sub-assemblies are built from imported components. The base design can only be changed each two years for new components!

A summary of the results of the four cases using various costs, markups, and market data is described below:

	Total Import (M$)	Local Mfg (M$)	Cost to Users
Monopoly (actual)	78.2	0	488
Monopoly (plan)	44.2	34.0	488
Free Market Import	44.1	0	88
Free Mkt Import with Mfg.	32.0	17.2	83.3

For the study, the market is assumed to grow at 50%/year using the following units: 295 (first year), 443, 666, 1000, and 1500 (fifth year).

The following markups for sales and service are assumed:

Monopoly (actual)	6.25
Monopoly (plan)	6.25
Free Market Import	2.0
Free Market, Local Assembly	2.5
Free Market Local Assembly and Sub-Assemblies (using imported components)	3.0

It is further assumed that the following local content is possible:

Importing Finished Goods	0%

Importing Sub-assemblies	25%
Importing Components	5Ø%

Case and Year	Import Cost/ Unit (K$)	Total Import Cost (M$)	Local Mfg. (M$)	User Price (K$)	Total Price (M$)
Monopoly (actual)					
1	20	6	0	125	36.9
2	20	8.9	0	125	55.3
3	20	13.3	0	125	83.3
4	20	20	0	125	125
5	20	30	0	125	187.5
		78.2			488.
Monopoly (plan)					
1	20	6	0	125	36.9
2	15	6.6	2.3	125	55.3
3	10	6.7	6.7	125	83.3
4	10	10	10	125	125
5	10	15	15	125	187.5
		44.2	34.		488
Free Market Import					
1	20	6	0	40	11.8
2	16	7.1	0	32	14.2
3	12.8	8.5	0	25.6	17.0
4	10.2	10.2	0	20.4	20.4
5	8.2	12.3	0	16.4	24.6
		44.1	0		88.
Free Market Import With Staged (each two years) Local Manufacture					
1	20	6	0	40	11.8
2	12	5.3	1.8	30	13.3
3	12	8.0	2.7	30	20.
4	5.1	5.1	5.1	15.3	15.3
5	5.1	7.6	7.6	15.3	22.9
		32.0	17.2		83.3