Gordon Bell
Oral History

COMPUTERWORLD HONORS PROGRAM INTERNATIONAL ARCHIVES

Transcript of a Video History Interview with
Gordon Bell
Minicomputer Developer

Recipient of the 1995 Price Waterhouse Leadership Award for Lifetime Achievement

Interviewer: David K. Allison (DKA)
Curator, Division of Information Technology and Society, National Museum of American History, Smithsonian Institution

Date: April 1995

Location: Palo Alto, CA

DKA: You started your career as a Fulbright Scholar. How did this happen?

GB: I had been a co-op student at MIT working for large companies where there were seas of engineering desks, and so I was trying to delay going to work as an engineer. I visited Gordon Brown, the MIT head of the EE Department, who was an Australian. And he said: "Why don't you go to the University of New South Wales? They just started a department at their new eight year old university and they need somebody to teach computing and get them started in research." So Bob Brigham, my roommate, and I went to Australia as Fulbright scholars, taught a graduate course, and built a pretty impressive compiler for their computer. It was the English Electric Deuce, a follow on to the NPL National Physical Laboratory Ace that Turing designed. It was a very hard machine to program because its main memory was delay lines with 192, 32-bit words and programs resided on an 8 K word drum. It had card input, and you signed up to use the computer for short periods of time – it was used as a personal computer, albeit one you could walk into. We wrote a compiler to optimize programs and make it easier to use. It's 32 word, 32-bit memories could be displayed on a CRT, so you could interact with it.

When I returned from Australia, my thesis advisor, Ken Stevens head of the MIT Speech Lab hired me to the research staff. This allowed me to take courses, and work toward a PhD. I had little desire to get a doctorate because I had really just wanted to be an engineer. I needed a job because I had just gotten married and Gwen was finishing Harvard. So I followed that path. The lab was doing really fundamental and interesting work in speech understanding and I thought I could write a program to recognize speech. I wrote a program called Analysis-by-Synthesis that was a way to attack speech recognition or recognition of anything. Basically, you generate a synthetic signal from a model of speech production and then tune and compare that with the input to impute what the sound parameters might have been. The basic technique is still used for analysis. The 1959 paper still gets referenced. One of the students in the lab became a professor at Tokyo University is still pursuing the path and continues using the technique.

The more important thing to me about the MIT experience was the use of the TX-O, a machine that was designed by MIT's Lincoln laboratory and one of the very first transistorized computers. It was fast with a 6-microsecond core memory. And it was designed for interaction, real-time, and connecting things. We connected recorded speech through a bank of filters via an a-to-d converter. So it was both a real time and interactive machine. It was a personal computer used by one person at a time. It was basically a PC. It had only 16 kilobytes of memory and paper tape I/O. I designed a magnetic tape control because it needed to handle more data.

DKA: So you were really using personal computers from the start.

GB: Also, that's how I came to be a computer engineer. The tape control was designed from modules from a 1957 startup, Digital Equipment Corporation in nearby Maynard, MA. I looked at the small company in an old mill building and everybody was designing and building things just like I had always imagined engineering to be. Gee, this is how I thought engineering was! I can actually DO design and build something if I join DEC! They made products. My earlier co-op engineering assignments weren't very interesting to me. So I joined DEC in the summer of 1960.

DKA: Before we go into that, let me go back and talk a little bit more about MIT. I was curious as to whether you were interested in computing as a student there or that that interest grew or what had you hoped to go into when you first started working as an engineer?

GB: Okay, what was computing like? I took all the computing courses MIT offered in 1952-1957. There wasn't even a computing option. There was a course in digital design, courses in switching theory, numerical analysis courses, and several courses in machine language programming. I learned to program the IBM 650 and 704. MIT had a 704 or 709, and the 7090 didn't appear till 1960. MIT's Whirlwind was the machine that was the progenitor of real time, interactive, and air traffic control. I was fascinated with digital systems design and computers.

DKA: Interactive computing and SAGE?

GB: Yes, all of that. So that was the fascination. And the TX-0 was the machine that was attractive to all of us. So when I saw DEC introducing the PDP-1 as a follow-on to the TX-0, I wanted to be part of it.

DKA: But as a student had you had access to the TX-0 wouldn't this have changed things?

GB: It wasn't on campus until 1958.

DKA: When you came back?

GB: When I came back from Australia in 59 the TX-0 had just been installed. But no, there was not hands-on computing when I was a student, although we could sign up for some time on the IBM 650. It was only research associates or graduate students that had access to the machines because they were for research. The speech lab was a prime user.

DKA: And yet you knew that this was the area of engineering that you wanted to make your life?

GB: Yeah, it was the same way that I think of when everybody gets fascinated with computers. They are interactive and you are creating a living entity. TX-0 had a debug program to write programs on line, symbolically. And it was the fascination with the interaction that at least I found exciting. Because as a student I had run programs on the IBM 650 and Whirlwind but they were usually batch processed where someone else runs your programs and you get printouts, but it wasn't the same thing. Its conceivable I wouldn't have gotten into computing if I hadn't had the interactive experience.

I had online or personal experience when I was in Australia with the Deuce. And it was really used as a large personal computer, one person at a time that you signed up to use. That is the way machines were scheduled before batch processing.
DKA: One last thing I want to ask you about that you know that what seems like second nature to your experience on the TX-0 is a style of computing that is so far distant from what people think of now when they think of computing. Maybe you can just briefly describe what it was like to do something on the TX-0 with its oscilloscope and keyboard. Just what was it like to do something with that machine?
GB: Well, in a funny way I don't think it was that much different from today for programming. You sat and wrote programs like you do today with paper and pencil or directly into an editing program. I think people still do that or they should at least.

The great programmers I know like Dave Cutler still writes programs, desk checks it, and then compiles and runs them in a test environment. In that case the program was typed in using an off-line Flexowriter to create a punch paper tape. The tape was translated using a compiler or assembled and then loaded into the computer directly or via some kind of loader together with a debugging program that let you look at the program. The debug phase is virtually the same thing you have today but now it's more of a single system. The nice interpretive environments like Visual Basic are all-in-one environments for creation and debugging.

DKA: Now you started to talk a few minutes ago about the atmosphere at Digital when you first joined...

GB: And why it was that exciting?

DKA: I am interested in hearing why it was such an important company and is still such an important company in the history of computing. You might want to talk a little bit about that early phase and I'm sure you met Ken Olsen at that time and some of the other people there. Tell me about the atmosphere there.

GB: My badge number was 80 when I joined. What really struck me was that it was a startup in this mill building. In fact my office when I left DEC was still building 12, the ground floor, of a 3-story building that was pretty much the headquarters building. As a civil war woolen mill it was totally open, and the offices were made into semi-private offices by putting up partitions made with ordinary doors. It was quite open but yet everyone had there own private space unlike what I would call the aircraft company engineering offices of the 1960s with a sea of desks butted together where you looked at someone to your right and left and across your desk. Something about the seas of desks I guess bothered me about engineering, and what was attractive about DEC was that I was the second computer engineer. There were circuit engineers, but I was the second one that came to build computers.

DKA: But of course Digital didn't start to build computers when they started in 1957; they built the modules and they had just, I guess, at this time made the decision that they were in fact going to go further and build computers and that's why they begin hiring people like you. Tell me about the discussions that you had before you came on board.

GB: I don't exactly remember my first visit. I don't think I made very many visits, but I went out to buy modules and discuss a particular circuit that I didn't quite understand and how it worked. It was a circuit that had been invented at Lincoln labs. It did exactly what you wanted to do that solved a nasty timing problem and nobody else had one that was anything like it. It was an integrating single shot. You needed something like that to build tape units, or rather it made the design of my tape controller a lot easier to do, so I went out to talk about that and their tape read/write circuits.

I met Ben Gurley who was head of computer engineering and came from Lincoln Laboratory, like many of the early DEC employees. He had come a year before and had just built the PDP-1. I met everybody, the whole team -- Ken, Harlan Anderson, Ben and Dick Best, the chief engineer.

By the way, that is a title we have since lost. I think it's a wonderful title that people should use. Now it's the chief technology officer, but I think chief engineer is a wonderful and better title. I really enjoyed interaction with Ben and the whole crew and in fact they very shortly made me an offer and I immediately accepted it. DEC looked exactly like the place engineers should be in and work. The manufacturing was in the next building.

I had grown up in a small town and had no idea what an engineer was other than in my mind and had decided I wanted to be one at about age 10. I went straight from Kirksville, Missouri, against the recommendation of a college math teacher friend of my father's. He said you don't want to go to MIT, you'll be competing with all these guys from eastern prep schools. Why they all have had calculus and all you've had is algebra. I went anyway.

DKA: And so, but then you did not know what an engineer was, but you did want to build things and Digital gave you that…

GB: Yes, so I had it in my mind what an engineer was. I did many different things, including writing floating-point subroutines, designing tape controllers, and a drum controller for one of the first time-sharing systems that Bolt, Bernanek, and Newman had ordered. The main thing was that as an engineer I wasn't part of a huge hierarchy, but rather I had the responsibility for a product. I also wrote a manual on I/O control that I'm still proud of because the techniques and philosophy of how to do I/O using interrupts and direct memory access endured and influenced other architectures. I also helped establish DECUS, the DEC user's group, patterned after IBM's Share, to help get open and free software.

Inventing the UART to interface Teletypes. Architecting and designing the PDP-4 My first big project was the project engineer to make a telegraph line switch to replace IT&T torn-tape switching centers with a PDP-1. This gave me an appreciation for communications and for reliable telegraphy. But what I am most proud of is inventing the first UART or universal asynchronous receiver transmitter for bringing a communication line into the computer.

DKA: So you had some early experience with networking communications and computing services.

GB: Yes, that fondness for communication came right from the beginning.

DKA: Now you're well known for some early work on the PDP-4. I wonder if you might want to talk about the difference between the "4" and the "1" and why that was an important machine at Digital.

GB: Well the "4" was also an 18-bit computer like the "1" but it was not compatible with it. It was the first computer I had designed from scratch. I think I wrote in Computer Engineering, a book about DEC's computers, the importance of compatibility. The same thing could have been said about the PDP-1's lack of compatibility with the TX-0. Like virtually all hardware engineers, I didn't have an appreciation for software investment and architectural compatibility. But one's ego takes over and we reason that we can make a better order code or architecture.

This is why there were so many early computer architectures, and even now a large number of variants of digital signal processing computers. The “4” was the progenitor of the “7, 9, and 15”.

The PDP-5 was really the forerunner to the minicomputer. It’s successor, the PDP-8 was what we think of as the classic minicomputer. Because of the way it was rack mounted, it was clearly a component to be incorporated with some other system. Other systems of the day were primarily stand-alone.

DKA: Well, I was going to ask you actually to contrast the series that came out of the “4” and the “8” and you’ve begun to do that. You might want to be somewhat more explicit about what that first line was targeted at, what were the innovations, and contrast that to the line that led up to the “8”, and of course we should talk about the 11 and the VAX. But I think the way to do it is maybe just be comparative about what were the objectives of each technical line and how those were achieved.

GB: Well, the “4” became a line that was designed to meet a couple of goals. One it was designed as a control computer for the Foxboro Control Company and needed to be lower cost than the PDP-1. One application I remember was to control a Nabisco baking factory. There was a lot of concern at the board because we might be liable if the computer stopped or dumped flour into the river. But the “4” used different circuits and we ran things slower and got economy not using all transistors. It used capacitor, diode, and transistor logic to run at a clock speed of 1 Mhz instead of the PDP-1’s 5 Mhz. In retrospect we should have used the PDP-1 order code. By running the “4” slower we reduced the price from $120 to $60 thousand. We also used a Teletype for the console because I disliked the modified IBM Selectric typewriters because they were unreliable, unlike the old fashioned, indestructible Teletypes. We were the first computer company to use Teletypes.

DKA: So a lot of the purpose of that whole line was to meet a market demand and the pricing.

GB: The “4” was cost and aimed at process control and real time data. It had several innovative features, for example any register could act as a counter and so it would allow you to collect data directly from external sources. Although it didn’t have index registers, certain memory registers were automatically incremented or decremented when accessed.

The “5” was an interesting story, too. One of the first applications that we looked at for the “4” was to control a nuclear reactor at Chalk River, Ontario. Ed DeCastro, a special systems engineer, and I went up there in the dead of winter to talk to them about their system. The “4” was doing the control and a special system that Ed was going to design was doing data collection. It had a rack full of counters, A-to-D converters and lots of buttons and switches.

So I said: “Gee, why don’t we make a tiny tiny computer to do data collection.” I think we started out with maybe a 10-bit computer. I asked: “What’s the smallest computer that can do the job?” It evolved from 10 to 12 bits. The analog conversion was done by using a D-to-A converter on the accumulator. That idea came from the LINC computer that Wes Clark had designed at Lincoln Lab for laboratory use. Wes influenced my thinking about architecture and I/O.

DKA: And the “5” led to the “8”. Let’s talk now about that transition from the “5” to the “8” because the “8” was such an important product in Digital’s history. Maybe you want to talk both about its objectives and why it became so successful.

GB: The “5” was built as a control computer, but the machine that was very important was its successor the PDP-8. The “5” occupied one or two cabinets whereas the “8” was less than a half cabinet. The net result is systems could be built that were significantly smaller. In many cases the “8” was put in other manufacturer’s packages.

Let me digress. The transition to make a PDP-8 really occurred because of another machine -the PDP-6. After doing the PDP-4, I went to work on the PDP-6 which was DEC’s big machine and the world’s first timesharing computer. We didn’t think it was that big, but it turned out to be quite a large machine with a 36-bit word length. It was patterned after the standard word length of the day, the IBM 7090 that came out in 1960. The PDP-6 was built using the original 5 Mhz and 10 Mhz modules that were interconnected using a hand-wired backpanel in two bays or 2 x 12 x 25 modules. Many women worked in the Maynard Mill to do the wiring. On the PDP-6, we found out that the many wires and corresponding wiring errors meant that it just took too long to debug, making it quite costly.
Now in retrospect we should have never plugged modules in. It should have all been checked even with people checking, but women did point to point wiring to build the machine. So I investigated buying a wiring machine from Gardner Denver. The original came from IBM, and Univac also used it. The net result was being able to produce PDP-8s in high volume and at lower cost. It allowed us to introduce the PDP-8 with its 12 bit word, 4 Kw memory, and Teletype for $18K.

DKA: And you began to really open new markets.

GB: Yes. In fact the idea of OEMs or Original Equipment Manufacturers came from the “8”. That is selling it to other companies who would resell it as part of another larger system, whether it’s a controller for a cigarette making machine or factory or a test instrument. So the “8” was really a transition to another way to market computers. Today, most of the adding on is software as in the thousands of Independent Software Vendor companies.

DKA: You might want to talk a little bit about the computer market at that time the "8" was introduced because it had gelled in a certain way and DEC was beginning to find its position in the market. How did that look to you at the time?

GB: We ought to look at the market in the mid 60s. This is right at the time when integrated circuits were being introduced in the mid 60s. The million dollar or so mainframe market was described as Snow White and the 7 Dwarfs -- IBM and its competitorDKA: Burroughs, CDC or Control Data Corp, GE, Honeywell, RCA, and UNIVAC. All targeting electronic data processing for large corporations.

So the minicomputer was a totally different kind of machine for a different market. The PDP-1 sold for $120,000. But it had only an 18-bit word. Who could use an 18-bit computer? Well, you can get a 36-bit computer by just doubling it up and mostly it works. SDS, Scientific Data Systems, was introducing 24-bit machines in the early 60s, but they were also young. DEC and Computer Controls Corporation were contemporary startups. So there were really only a few companies. First off, there were few competitors because you had to design your own circuits. DEC's basis technology was circuit design or as we would say now, barrier to entry. So a computer was just an assemblage of the logic circuits, built to interpret an architecture, and the software. From where it was as a startup, all that remained was to put the circuits together.

In those days, the software consisted of a bunch of independent routines. There was nothing like an operating system to manage the computer. When you ran a program you basically pulled together a bunch of software components and ran them.

DKA: So you had a core of innovative aspects to your company that nobody else really competed against. As you say, it's a full service at a certain extent but also at a level that was below in terms of complexity and price point what the other companies were doing.

GB: Well no, I would say we were at the same complexity level, but we were producing low cost, high volume machines and this allowed them to be used in a number of different markets. And because DEC had the modules, other companies could take the modules and build their own systems and write the program for an application.

This was the beginning of an era where the idea of standards was just beginning to happen. In languages, people said COBOL 60 will solve that problem for commercial computing and FORTRAN will solve problems for the scientific market. But the scientific calculator market was one based on wide words so you can't do science unless you have 36- or 48-bit word.

Also, those scientific machines were expensive with a memory of at least 32 K words. The PDP-1 with just a 4 K word memory was rarely used for calculations -- it had a scope and was a machine you interacted with.

So early in the '60s we said we've got to have a large word machine -- that's a REAL computer. MIT was building a timesharing computer based on the IBM 7090 so it was natural for us to look there. You're not going to have a 300-500 thousand dollar machine just for one user. So how are you going to do that? By timesharing one machine. So timesharing came out of the same era.

I feel so fortunate to be part of that period from 60 to 70 which is when minicomputers were born, timesharing started, integrated circuits introduced, and COBOL and FORTRAN. On the other hand, every decade I say: "Oh my god, the next decade is going to be much more exciting than what we've lived through." But in fact this was an exciting era.

Building a timesharing system meant lots of users on line, no restarting, and it can't fail. And it was the first time we took responsibility for significant software -- we're providing the software. Its not coming from the university or the users don't sort of glue it together. So that was our first operating system and it was introduced in 1965.

DKA: So that was really the beginning. DEC had achieved a maturity with the "8", and then I guess the "11" is the next big product that came out. You might want to talk about that transition.

GB: Right. What happened after this beginning was that in 1964 IBM introduced the System/360 and then that changed all the word lengths to be modulo 8 bits. Computer Controls Corporation had come out with the first 16-bit mini designed by Gardner Hendrie who I had known at Foxboro. Then a year or so later Honeywell bought them and promptly destroyed the company before they could become a threat. If you can just hang in there as a company, you've got a good chance of making it because others may self-destruct. For example, SDS was doing pretty well into the early 70s until Xerox bought them. That was a pure play for the founders --- gee, we're offered 900 million dollars for our computer company that we're having trouble with in a very competitive market. And so XDX was created and eventually written off.

So that era from 1965-75 was that transition to a 16-bit world using Integrated Circuits. Almost 100 minicomputer companies formed and eventually died with only HP surviving.

In a way, I can look back and say maybe I was burned out when I went to Carnegie Tech as an associate professor in what became the computer science department in 1966. I remained a consultant to the company. The PDP-6 begot the PDP-10, so that was going along nicely. The "5" and the "8" were established and growing, there were PDP-4 follow-ons, and so the company was doing very well. I didn't see that I was essential to the company.

Being a professor at Carnegie Tech that became CMU was a wonderful experience. Students were always there to question. Working with Allen Newell on Computer Structures that included notations for describing the behavior and structure of computers was simply great.

But there was this gnawing need within DEC, called the 16-bit computer and there was a group of people building and designing the PDP-X, which was an architecture of an 8, 16, or 32-bit machine. I wasn't there to catalyze it and what happened was the engineers and the management didn't get along. The machine was posited by Ed Decastro and Henry Burkhardt -- the guys who formed Data General. They put together a very nice proposal and management didn't buy it. There were a lot of bruised egos and a whole bunch of reasons that it didn't happen – maybe they tried to have it rejected. I probably shouldn't comment on the decision, except to say I was a strong supporter of the PDP-X. I said: "Build the X. It's a fine machine. DEC ought to be building this." And I think it would have been a lot cheaper had they done it, but they didn't. A team left and formed Data General and built its NOVA, that had no relationship to the PDP-X.

When they left, a project was started to redefine the PDP-X and it went through a long path of being defined and redefined and the guy running it had no idea how to design a computer. One of guys on the team was Harold McFarland, a student of mine from Carnegie who had worked at DEC in the past summer. The machine ultimately that emerged was PDP-11. The team had put together a machine proposal and then came to Carnegie to have it reviewed by myself and Bill Wulf, a fellow professor who eventually became the President of the National Academy of Engineering. We looked at it and we said: "Yuck! We don't like it, and Harold sort of pulled out another design from his notebook. It was basically a design that Harold and I had worked on while he was a student. The idea was formulated while writing the book Computer Structures with Allen Newell. The idea was an "aha" for very general registers and how they could operate as stack pointers, index registers, accumulators, and program counters. On the physical side it was centered around another "aha" or the idea of the Unibus, another concept that came from Computer Structures. These two ideas were really marketed by DEC against DG when it was introduced in 1970. Andy Knowles drove the marketing.

DKA: And you actually came back then to Digital.

GB: I'd been at Carnegie and then came back in 72 just as the next generation models were being planned. I was planning to take on a visiting professorship in Australia, but Ken said: "Come back and run engineering. We've got so much going on and nobody can control it."

DKA: When you left to go to Carnegie did you think that was the end of your time with Digital?

GB: No, I consulted for Digital and it wasn't until 72 that I saw the necessity to return.

DKA: But did you want to spend the rest of your career teaching and being academic?

GB: When I left DEC in 66, I knew that I was tired of building computers and I wanted to think about them. If you look at it historically, I sat out a dull period when small and medium scale ICs took over for discrete circuits. The first ICs weren't very big -- we were about the size of DEC's modules. But in '71, Intel's 4004 was introduced as the first microprocessor. And those weren't interesting to anyone who built a computer. They were used to build calculators, scales, and traffic controllers, but they were nowhere as powerful as a PDP-8.

The Intel introduction was characteristic of what I posited to be a Theory of Computer Evolution that is sort of a corollary to Moore's Law in 1972-75. It's what happens when there are just enough transistors on a chip to form a lower priced, new computer that can do something useful.

DKA: So you came back at a time when you could make that kind of transition at Digital.

GB: Yeah. I came back exactly to do one-chip computers or to do integrated circuit computers. Within a year, we were on a path to build an integrated circuit computer. I remember my first trip to Silicon Valley in the summer of 72. I met the Intel guy, my first meeting with Bob Noyce who invented the IC. I tried to get them to take the PDP-8: "Please won't you build this PDP-8 on a chip computer for us and make it a standard? We will buy chips and make systems and you can sell the chips to others."

It turns out this was a constant battle I had within DEC with Ken and most of the Operations Committee. I tried unsuccessfully to convince them to get other chip manufacturers involved in building chips for us. The situation occurred for all the computers, but eventually Intersil was allowed to build the "8", and Harris was licensed to build a small "11". However, we did get Western Digital to build an "11" that we sold. Unfortunately, they were not licensed to sell it, so while the "11" did well, it failed to become a standard.

DKA: Now you had seen innovation at Digital in many different stages. How would you describe the culture and the approach that Digital brought to design the VAX computers compared to the earlier phases of innovation? Was it just larger and more complicated? Was it a different approach? How would you characterize the evolution of the company?

GB: While I was there, the company didn't change very much, especially from a cultural standpoint. DEC was an incredibly open company during those years with free communication throughout the company. And so when we did VAX, and it didn't mean there weren't engineering camps and wars and politics, it was still open and people kind of knew where everybody stood. It wasn't a guarded environment or totally political. It wasn't protected. You knew what was happening. You might not like a project, but you knew what the other guys were doing.

There was a period of a year when we really stewed over the question of whether to extend the PDP-10 architecture and use all its software or build from the PDP-11. We actually built a small PDP-10. I let that process go on for a year. It was a process of examining what we should do from all angles, and especially talking to customers. At that point I ran all of engineering. There were product lines, or marketing lines that sold computers into various markets such as laboratories, education, industrial control, commercial banking, telephone companies, and to OEMs.

When I came back from CMU in June 1972 to run engineering I didn't get this responsibility. Ken assigned me to run memory and power supply engineering, probably the hardest jobs in the company. No one wanted to do it, and I knew very little about either. I didn't know about power supplies, I didn't know about memory, but I learned a lot more about circuits than I probably ever wanted to.

I had the title of VP of Engineering, and so I got involved in all the issues at the staff level. Throughout the company every marketing group had its own engineering, so what was happening was all these projects were getting formed with no coherence – especially in software that was sometimes used to differentiate the product lines. Then finally after about a year and a half I said, "Enough. I want all these engineers to report to me." I proposed to make it very simple. And that was the beginning of really pulling them together.

Our first strategic thing was to transition from the PDP-11. It just didn't have the addressing power to let us go on. We had built the 11/45 and the 11/70 and these were fine machines, but you could not program them because of the addressing limits. So the question became "Should we extend the 11 or should we take the PDP-10 which was already fine and use that as the base." We stewed over that question nearly a year in engineering. I don't remember what the catalyst was but at one point I said enough. We've looked at all the facts in every possible way, we're going to extend the 11 – not base it on the 10 – because all of our customers and our main line of business is 11-based. Just a few more than a thousand 10s were built. There's just not a good way to do the same things we're doing with the "11" using the "10" and its software.

So on April 1, 1975 I pulled a group together we called theVAX A group. VAX A was the mailing list and there were 6 of us. We took moved together on the 3rd floor of Building 12, almost at the same spot I had when I came to DEC in 1960. My main office was on the first floor with Ken.

DKA: Of course the company is now at that stage was much bigger …

GB: I think roughly quarter of a billion of revenue.

DKA: We can just talk about now how you brought that VAX A group together and that decision.

GB: So we determined we were going to extend the "11" and not work on the "10". I brought these guys together and we started doing the architecture work. Bill Strecker was the chief architect of VAX. He had been working on the idea, and had outlined the alternatives -- how much of the "11" do you want and how close do you want it to be to an "11"? We called the resulting architecture "culturally compatible with PDP-11".

I named the project VAX-11 or virtual address extension to the "11" to keep us on track. It was going to be an evolution on the "11". The way we dealt with compatibility was to put a PDP-11 in the instruction set to run all the RSX-11 software. This gave us a tremendous head start on software as well as a base. VAX ran a lot of PDP software for a long time, including many compilers. This allowed us to get all kinds of software done in another environment and then simply moved over rather than having to do it all from scratch.

This story was repeated at Microsoft when Dave Cutler, a member of VAX A, went to Microsoft to invent Microsoft's NT. He made that system also compatible with the PC hardware and all the apps. In that case, it was nearly impossible because of the lack of discipline and definition of the PC and the various interfaces because of the way the PC evolved in a chaotic, free market.

Microsoft was left to make it all these loosely compatible components work! I claim nobody but Dave could have done this.

DKA: So that was a strategy that appropriate for a company with an established base of customers, an established body of software that was an enormous investment, and yet was beginning to take advantage of the some of the new capabilities like the ICs and large scale integration.

GB: Yeah, especially larger memories. Remember VAX had to be built because the 11 ran out of address bits. RISC hadn't come in yet. Dave Cutler asked me a few years ago: "Why didn't we do RISC?" and I said: "Remember how much memory we had, how long it took for us to have enough memory, and how long we would have had to wait before we could build a RISC type machine because the RISC transition didn't occur until 1985. So we had a 10 years of "What are we going to do for revenue?" problem. During this time, and RISC is really not an architecture kind of question of "Oh god, you are stupid not to build this way!" but it's a question of what you can do in the compiler and the cost and availability of a memory hierarchy.

So it's not a religious or intellectual debate, as much of the RISC advocates phrased it. It's a plain old engineering question of memory cost and having large, fast memories for caches. To fundamentally make RISC work you need to have big caches because you are fundamentally running microcode in an open fashion. It used more bits per program. In fact, RISC versus CISC, ignores the fact it took about twice as much memory to say the same thing. And so I'll say VAX was the ultimate CISC machine. I maintained the goals and constraints of VAX and how it was going to be put together in a document called the VAX Blue Book, and it contains this whole question of micro programming – basically the idea was that we would put everything we possibly could into microcode to run faster and take less bits that the equivalent procedure calls. So VAX had instructions to queue for the operating system, an elaborate memory management system, and, of course, all the floating point routines. VAX also had decimal arithmetic that COBOL needed. It was probably the best COBOL machine every built, but the initial apps used it as a FORTRAN machine. A decade later, you would not do it that way. You would do these as subroutines that are called by ordinary programs.

DKA: So you really had a different kind of team to do the VAX in terms of your integration of all the engineers from the application areas and from a migration strategy. Let me ask you to put on your hat as an entrepreneur again - how would you characterize the working of that team in putting that machine together?

GB: It's the way I recommend engineering projects be done in an entrepreneurial setting --there were only 6 people in that group. We didn't want any more people. You can't deal with any more at the beginning of a project.

Every time that you are doing something new and different, where you haven't defined it yet, the worst thing you can have in a project is too many people at that critical startup phase. You have to manage that very slowly. That's why we were limited to only half a dozen people. We had NO marketing people. Every two weeks we had a group called VAX B that was a room full of about 25 people. The six of us communicated with a lot of other people, of course. But basically we worked together to define what it was going to be, and then the 25 would comment and sort of oversee us. It had only a couple of marketing people, and we used them to find out whether people needed this or that. The only customer we talked to was Ken Thompson of Bell Labs. He was hardly a customer, but rather a developer who was helpful in what we needed in order to run UNIX.

VAX was in the same architectural style as the PDP-11 and distinct from the IBM architectures. And a lot of that comes from how IO is done, and how to deal with multiple processors. A program could reach out and do something directly with the periphery was what made it powerful. And the 360 was the one where IO channels were always working, lots of protocol, lots of overhead designed for throughput at the expense of response time. My philosophy of IO was totally different than IBM's. Ironically IBM is finally coming out of all of this with the philosophy that DEC has always used, which is not having specialized weird computers doing IO. Just one kind that does it all. And then if you need more of those you put more of them in. It's much easier to do. But the mainframe kind of mentality of cascading many weird computers with their own instruction sets and software support is a pain in the ass. It's just not the way to do it.

I was consulting with Siemens three years ago about their minicomputer architecture. I asked about an elaborate communications option: "Well this is a board to do all the communications and protocols." I asked how much the board cost: "Well it cost 3000 dollars." It had two or three computers, following the old mainframe mentality of "we're offloading the main microprocessor." I said: "You realize that microprocessor is much more powerful than any one of these and cost less. That, in fact, what its doing is delaying doing the communication work. You've got plenty of cycles in the main processors, and you're creating an enormous number of bottlenecks and expenses, and the guys running the operating system are just tearing their hair out because they can't get at the I/O." I think that war has been won for simple, direct I/O, and using multiple micros. On the other hand, we are going around the loop again as each device becomes an independent computer and the entire system is now a network.

DKA: Now the VAX was an enormously successful product for Digital. How would you look at that phase in the history of computing and why that product reached out and was so enormously successful.

GB: Okay, I'm going to tell you one other story about the VAX. We started April of '75, and first betas were introduced late '77 early '78. One or two of the first ones went to John Pople at Carnegie Mellon University – for his work in computational chemistry to replace the Univac 1108 batch system that he was being limited by. I insisted that CMU get the first ones as scientific users. Other early machines went to Lawrence Laboratories, and the NY Institute of Technology who had the leading graphics group. VAX was almost the first virtual memory machine. Bill Poduska, who founded Prime, had extended the old DDP-16 architecture from 3Cs and Honeywell to have a 32-bit virtual memory, but ours was a totally new architecture. And we found that all these users were just floored by the machine. There were a couple of other 32-bit machines, but the VAX really captured mind share of the technical community including computer science departments. With paging came the ability to run large programs, and it out performed every other machine except the large IBM 360s and Cray 7600 on floating point. "Give us more" was the reaction.

I made my first trip to Japan in the summer of 78 and talked about it. After that trip, our family spent three weeks scuba diving in Tahiti. During that time I conceived the VAX Strategy given in Figure 2, another "aha" as a way to focus all of our engineering effort on VAX and to reduce the plethora of computer models. We had plans to build new 11s and 10s upward and downward to compete with VAX, and the "8" was still being sold. I went back and said: "Folks, I propose the VAX Strategy to replace all of these efforts so that we end up with a single architecture. We will continue some of the machines for which there's a commitment." "We're going to make only VAXs. We're going to extend a couple of 11s that are in process, but we're not going to do any more. We'll extend he one chip "11" downward we were doing – and use that as a controller. Let's get rid of the PDP-11 that are aimed at competing with the VAX-11/780, let's get one or more semiconductor company to take it over and make it a chip that anyone can use." The reaction was, "We can't do that, the PDP-11 and architecture is the corporate jewel!" I said: "We've got to get somebody else to invest. We can't afford everything. People still hadn't come to grips with the notion of standards and the fact that the architecture needed to be a standard to survive against the Intel and Motorola chips."

In December '78 I went to the board with one slide describing how I envision this computing environment. I described how we can attack IBM and offer different styles and range of computers. Ironically, in 1975 I had written another article on the Theory of the Evolution of Computers that I just mentioned. Machines form in price bands and personal computers are now forming.

It was a three-tier model: the corporate centralized mainframe we called glasshouse computing; the departmental mini -- its put around in the various departments serving a department or single function -- and then all the computers for the desktop that we now call personal computers or PCs. And all of those levels are connected together by some magical interconnect -- which at that point wasn't Ethernet because we hadn't put the Ethernet deal together, but I knew we needed Ethernet and we had two or three alternatives internally.

We were also starting projects in cluster interconnect for connecting machines together using a new interconnection bus, CI (Computer Interconnect) in order to get more power similar to what Tandem introduced in 1975. Today, IBM has introduced its Sysplex and the UNIX variant companies are trying to build clustered machines. Again, 10 years after we had a good system! HP is still trying to introduce it and Sun is talking about it. How do you connect multiple independent computers? Well, DEC introduced that in '80. I'd say it was really solid in the 84/85 timeframe. So here these guys are introduced them a year or so ago, and it'll take them a good three or four years to get those products working. It's nontrivial connecting a bunch of computers to behave as a single computer.

So the big thing about VAX was really two things. One, was architecture. It was to be compatible up and down the line. Nothing different. The 360 did the same thing with a range of different powered models. The big difference was that VAX was aimed at different styles of use. The 360s were aimed at all the glasshouses, little glasshouses, big glasshouses, and huge glasshouses. But it was still the same kind of batch and remote job entry computing and with different operating systems. In the case of VAX, it was big glasshouses, closets, and desktops, and we wanted to be able to run the same programming. There's got to be one operating system. The 360 had different operating systems. We said no, the value is in the software. Its going to be one, we're going to run that image across that range so basically anyone can compute anywhere depending on do I want response time, do I want throughput, or do I want cross performance. And so that was the basic idea behind the VAX Strategy, which is more of this is all going to be tied together, this is all going to be a single unified architecture.

That whole thing lasted at DEC until the open system. In fact the day I left DEC in 1973, I said: "Look we've got VAX now, we've got exactly what I envisioned, the clusters work, we've got the one chip processors coming down the pike. They're not here yet, but we know what they'll do. Now you've got to get rid of it because of the whole business of open architecture." UNIX was there and that is a different story. I don't believe UNIX is open! UNIX is just another name for propriety operating system. But at least the threat was present, and DEC did it all very well until the UNIX open myth was established by SUN -- I think that was probably 89 or so.

DEC was riding high in 88-89, and then it got into trouble and this other factors set in. But it was simply that strategy. That's what made it all work and basically there wasn't anything to do. The lovely thing about the strategy was it was just one page with two or three pages of implications such as what we need to develop or stop, the work on networking, and a few pages on why it beats IBM and how it addresses the market issues. And that was the basic model for it. And there were events that happened after the first version in 1978 that had to be attended to -- the PC hit.

DKA: That was the next question. People have said Digital misunderstood what was happening with the PC … it missed the boat. Do you think that's legitimate?

GB: Oh I think that's totally legitimate. I think DEC totally missed the boat on the PC.

DKA: Why was that?

GB: Well, one reason was we were focused on VAX. During this period when we were doing VAX, Small Systems Engineering was working on personal computers. They weren't working on VAX, they were working on the PDP-11 extension, they were working on the Rainbow that was X86 CPM-based, and a PDP-8 for word processing. So we had three personal computer projects. But a strategy to have done a better job was exactly the same work that was needed to make VAX so coherent. I did that work and winnowed it down and was working on the VAX side. I ran the others and so you can blame me for the whole thing. But I had a little bit of help.

Ken was really running Small Systems Engineering. And Ken's big problem was that he really didn't understanding computing at a visceral level, at an economic level, and he also didn't understand the industry and what was happening. The industry was moving fast. I'd say if I'd been more involved, I probably would have sensed what was happening and you can bet we would have had an IBM compatible PC the day IBM had it running Microsoft MS DOS. Exactly the same thing. So I'll say, sure, that's what happened. But after a year, after two years, after three years the whole story was clear. I went back to DEC a year or so after I left in 1983 and talked with the Operations Committee, the half dozen people who ran the company and said: "Look, the war is over. You've got to be the strongest one in there. Get rid of all this shit. You can't support them. Be the best PC company out there." And that was totally compatible with VAX. The VAX had nothing to do with it. DEC was a big company, they could run and have a whole division. That is a great story of -- how do you allow entrepreneurial stuff to exist in a large company? How do you support it?

But they were still fooling around with the Rainbow. I mean that should have been killed. A year after the PC hit, it was so clear the game was over. And DEC never got it. They just didn't get it. And I hate to say it, but anyone should have gotten it.
Leaving Digital in the summer of 1983

Running the VAX and going to 10-12 billion dollars from where we were when I left at 2 or 3 billion took zero thought. There was no innovation at all in that evolution because it was all programmed, it was all determined, it was all set down in this one-page memo -- this is what we're doing. And personally the big reason that I left was because of the same reason I left to go to Carnegie Tech, I was tired. It really was a joy running these 6000 engineers and I loved working with them, but it really was a conflict between Ken and myself. And I thought my body was stronger but then I had a heart attack in 83, and that's what made me say this is too much. It's too hard for me to do things. Changing engineering and directing engineers wasn't hard, but fighting someone about this is the way its going to be wasn't worth dying for.

DKA: Too much stress.

GB: It was too much stress. And it shouldn't have been stressful at all. Who knows, Ken is an engineer too. He's just not a COMPUTER engineer. He's a power supply engineer. He's a wonderful packaging engineer. But he shouldn't have anything to do with computers.

DKA: Because of the detail …

GB: Because there's this stuff called software. There's this thing called the industry - how does the industry react, the understanding of the dynamics of it. He loves to package things and he's great at packaging physical design. He's done some very beautiful things, and he was successful before he personally got involved in driving the PC. After he got involved in it, we went through five vice presidents of the Small Systems Group designing the PC. At one point Ken said: "You've got to run this and have these people report to you." And I said: "Ken, I really want to get VAX stuff done. I can't really have six more people reporting to me." At the time I had at least 6 or 7 reports running the different sized groups and we were doing very complicated stuff. We were doing VSLI, we were trying to put a VAX on a chip, we were doing real hard engineering not just plugging a goddamn 8086 on a board. And the marketing and PC marketing stuff was in utter disaster during that time. It was legend. In fact, I can look back and say maybe the best thing was that they were all preoccupied with fooling around with the PC. The marketing guys that sat in the Operations Committee were all arguing about who's going to be able to sell this or that, who gets credit, and on and on. Meanwhile with Ken driving everything, they were all looking for credit, for pricing, and DEC was opening stores and all kinds of bullshit like that.

One of the things I remember was the Ethernet story and going to the Operations Committee for approving the announcement. I had let Ethernet go through and we were making the deal with Intel and Xerox. We went in and said: "Well, we're going to agree on a standard." It was no big deal, because I didn't want it to be a big deal. It was a big announcement. Bob Noyce, I and Dave Lidde from Xerox introduced it in New York, Amsterdam, and London.

DKA: By the way, on these interviews --how much personality should come in?

GB: Well I think this issue is important and it's an issue that does tie to personality. I think when it becomes significant in shaping… to a certain extent people want to know about the people. But my goal is to try to look at how personal preferences, personal decisions, strategic decisions affect the flow of the history of the industry. And I think the issue that you're talking about is clearly one where you had a company that took a certain strategy toward the small systems that ultimately was shown to be a failure, and its important to try to understand why that happened and how that happened. At a certain point I think that what you say is right when the strategy ... there was a while when it wasn't clear how much a company like Digital could control the market and could have its proprietary system, but as you say …

Ken was a fantastic CEO at one point but he changed, and I almost know the day he changed. I can almost contribute it to a woman -- Julie Pita, a Business Week reporter, who challenged him with, "Well, do you think CEOs are real leaders or are just sitting there?" And god damn it, he absolutely changed. He got a closeness and involvement to the personal computing and small systems that was his downfall.

Prior to this time he really was effective, he managed the company. He tried to manage engineering more than I ever wanted him to, but he was never in any of my space. He didn't know anything about ICs or their design, or computer design. He always focused on the physical stuff and he always focused on terminals and things that you could see or touch. He never got near questions like what does a program do, or what does a network do, or how to build them? But when it came to the package or the appearance he had strong feelings and there was a constant pain in terms of dealing with him. So trying to manage in this environment was a constant string of brush fires. I was loath to tell him what he wanted to hear and then do the opposite as the other VPs did. I was the only one who told him "no".

DKA: So that could work when you had somebody that could make the right decisions down in the organization, but when you had people that weren't strong enough to stand up to him and he didn't trust them, bad decisions could result?

GB: When I left he was involved in all decisions and there were plenty of people to deal with. People were constantly gaming Ken in terms of how you deal with this man. And after I left there was sort of a triumvirate running DEC - the head of engineering/manufacturing, Jack Smith, and Jack Shields running all the marketing, sales, and service organizations. Ken had by all of his cunning ended up having these two guys, both of whom were disasters, in their own ways, being the team to lead DEC into a significant battle.

DKA: DEC's relationship to the PC. You talked some about the fact that yes they had …

GB: DEC had the three programs going - using the PDP-8 for word processing, building a PDP-11 that would be a standard or be its architecture, and then using the Intel architecture. The later was the favored one because you could make the lowest cost machines. And in fact that was an era right after we had been using the Z80 to make PCs running CPM. And then there was a follow-on to it. Somebody favored using the Z80 or Z80 follow-on that was the 8088 -- and that was the Rainbow -- and we had the PDP-11 that was the main line.

The PC was different than other machines because it was the first time a standard got established outside of the company, and you did have a single architecture as opposed to the traditional past of a vertically integrated industry. You have the software, the hardware, the chips, and you have the whole line and then you dominate the industry. The PC wouldn't have taken off without the standardization and stratification of horizontal levels of integration. If there had been IBM and then if DEC had been successful with either MicroVAX or PDP-11 and that had all been stable, the PC industry would be nothing today. Because you wouldn't have had the volume that you have and the single standard that you have that Microsoft defined for software. Microsoft and Intel. Forget IBM in the whole thing, they were just the catalyst. In fact everything that IBM did since the first PC has been rejected - the micro-channel and OS2 is no competitor.

Just looking at the variants of UNIX tells us that proprietariness doesn't work … one of the things Ken got right in the mid 80s was to declare "UNIX is Snake Oil". With unique variants the manufacturers keep high prices, but they get no applications market, and customers have to do their own thing on variants. Unfortunately, people bought snake oil.

DKA: But DEC had been successful by as you say having a vertical domination, and the notion initially to maybe extend this to the PC market wasn't crazy … but never realizing when the game was over …

GB: The game was over a year after IBM announced and everybody started making IBM compatible PCs. There was a compatible industry, the whole market went sort of straight up, and software was forming around it. The game was over and anybody could see that. But these guys didn't see it. In fact they still had the ego to say; "Oh, we can come back in there." And everything they said was always wrong. I told them we might have a chance if we got a better bus, we got a better interconnect, make that all standard and make that all available. Their attitude waDKA: "Nope, that's ours. How do we charge for that?" And the irony is that we taught IBM how to do all of this with the Unibus. It was a standard, others connected peripherals to it and we had no compulsion at all to inhibit them because the market grew accordingly. But yet with the PC or the PRO, we didn't said: "Hey lets make that standard and let anybody who wants to make peripherals." But rather: "No, that's ours!" It was a control issue, a proprietary issue.

When we were just about to announce Ethernet the Operations Committee looked at the announcement and said: "Wait! Why are we giving this to the world?" And I said first off we weren't giving it to the world. We got it from Xerox, we participated in the evolution of it, Xerox owns the Ethernet patent, and we evolved the standard beyond that. We were just part of it, it was not our ownership, and second is we wanted this to be a standard. If everyone is out there is connecting using different kind of wires, how are they things ever going to play together or get others to spend money to install the wiring in the first place? They said: "Well, we want only our computers on it." I said: No, you don't want only your computers on it because everyone's got their own telephone system. They are all different.

It was this whole paradox of standards being a double-edged sword. You've got to have them and yet you want control. You can't have it both ways. Unless its de facto ala IBM mainframe software and Microsoft. Microsoft does it totally by market dominance. And that's the ideal. Because from a standards standpoint the worst thing going is having a standard that's just a "government standard" that really isn't good. It gets there by a big committee process. It doesn't hold at all and its very hard to maintain the standards. But de facto with a single vendor driving the standard is ideal, because than you can drive it as fast as you can and that vendor determines it together with the market placing their demands to improve things. I personally think the Microsoft standard is the best way to evolve computing. The PC wouldn't have happened without that interface layer - every application guy puts his software to that standard. And then similarly that's why we have a thousand or so PC vendors.
DKA: You had had - you Digital - had not quite the same clout, but a significant clout with your minicomputer line …

Transition from proprietary minicomputer standards to proprietary UNIX standards
GB: We had a de facto standard. Yes, VAX was a standard. A whole software industry had strung up around the VAX, the AS400, IBM's MVS. That was the day a single hardware company could set a standard and that would become the de facto standard for an industry. But in the case of VAX there were no competitors, no alternative suppliers. In the case of IBM mainframes, there were Amdahl, Futjisu, and Hitachi, - they were all alternative suppliers for platforms. They all had to use IBM operating system software, of course. Because that's the interface layer, just like Microsoft sets the interface layer for the PC. But what has made the computer evolve so fast is when you can establish these interface layers.

DKA: So again asking you to put on your hat as somebody who looks at entrepreneurialship. This critical time when Digital should have been going through a change in approach to the market and yet failed to maybe see the opportunities that it should have seen. How does that look as you look back on it? What were the critical errors and mistakes that were made, when seeds were laid for the kind of trouble the company got into years later?

GB: Okay, there was the whole PC question. That's one that should have been very, very clear because you had Compaq forming, you had the system guys like HP out there, and the standards were absolutely established. The industry was set and DEC should have been the dominant PC supplier. That's what I can never come to grips with it – why that didn't happen? And DEC is now getting to be strong in PCs. I mean they've gone up and down with it. When I was at NSF, Ken sent me a particular PC and I said this doesn't look like a PC. Well you've got to do this and that. And I said wait a second, I gotta do nothing. I get software from these floppies and you're either a standard, you're compatible or you're not . If you have to tell me about you CAN do this, forget it, I don't want it. I'm not going to do anything except turn it on. You've got to enter into a market where it's all the same.

DKA: So that was one error, but there were others things …

GB: That was one error, but the big error, the big thing that happened to DEC subsequently was failing to deal with UNIX. We had a very strong UNIX group, but allowing UNIX to compete across the board with VAX/VMS, wasn't allowed. There wasn't a way to do that. UNIX was sold as a last resort. And that could have been a reasonable strategy. But DEC was always very paranoid about that. About whether they wanted those things out there or not.

Next, I think what really got DEC into the most significant trouble was the way it dealt with the transition from to RISC and to a 64-bit address. Dave Cutler had an architecture called Prism that he had designed at the Seattle lab. That was all done, the manuals were done, people were working on chips, and the program was going along well.

Meanwhile, MIPS came to DEC and said: "Gee, you're not there with RISC or your one chip VAXen, you need a RISC machine for your workstations. Why don't you build a workstation on RISC?" And DEC did, introduced it, and said: "Oh well, we'll stay with MIPS." Then they killed the Prism project and Mr. Cutler left. They killed it, but Ken didn't know that it wasn't dead. It was still alive in the semiconductor group and it sprung up as Alpha. And so that came back several years later.

Meanwhile other people within the company were looking at building a fast MIPS architecture machine including a group in Palo Alto which built something called BIPS – a billion instructions per second processor. In fact they have one. They had one about three years ago. So all of those projects never came to market.

And that's why I said when I left the company that you've got to get rid of VAX, you've got to go open. The companies that I then started and worked with were open systems companies. They were all UNIX. But it was deciding to go to Alpha or deciding to do Prism, then killing Prism and going to MIPS, and then coming back to Alpha and killing MIPS again. DEC could have survived any of those decisions. It could have stayed with Prism, got it out there a year earlier, and been significant in the marketplace. It could have switched to MIPS, and I think that would have probably been the best strategy. But coming in late, having to build these very fancy FAB facilities to get the performance was really costly.

And today, there is no way I see that DEC can afford to be a semiconductor supplier or microprocessor supplier when they have to build their own, use their own, FAB facilities. So that was a significant error in judgment and decision making. On the other hand, the world is better off because Dave Cutler went to Microsoft and built NT for a much larger market.

Another error in judgment was building the last ECL-based machine - the 9000 - that was introduced. The machine was really late, and the transition from ECL and CMOS had already taken place. The 9000 should never had been started, even though I have to admit being responsible for signing the original development agreement with Trilogy, Gene Amdahl's follow-on company. It was a big, hot, package mega-engineering project that was really going after the IBM kind of SLT technology, a very difficult technology that came out of Gene Amdahl's project. But again that was one that should have been stopped because the company burned a lot of money and a lot of resources that didn't get them anywhere. And it also got them thinking of big mainframe like structures as opposed to moving into multiprocessors. Cray Research and Cray Computer also failed to make the CMOS transition and it cost them their lives as the premier supercomputer company. In 2000, three Japanese vendors supply vector supercomputers to the world.

But multiprocessors were my favorites, too-- since the first PDP-6. When I left, we had an advanced development project to put 64 Microvax chips in a single, multiprocessor computer. It then went from an "AD" to being a development project and then back again. If I'd stayed... my faith in shared memory, multiple microprocessors or "multis"

DKA: You would have pushed that one.

GB: Yeah. That would have been the way to go because if you want to be in the mainframe business then that's the way to go mainframe because that's the model we have today. In the company that I left DEC to start – Encore – we introduced one of the first "multi". That is a 20-processor VAX-like architecture machine that ran UNIX. And it ran circles around any of the UNIX boxes or nearly every other computer. Today what you see is the downsizing market -- Sequent uses 20 processors, DEC has a 6 or 8 processor Alpha, Sun with 20 processors and HP with 12. You hear IBM saying they're going to introduce one. We did that.

Our first product at Encore came out ten years ago – we made our first delivery in 1985. I wrote an article in Science in 1985 and declared that multiple microprocessor, shared memory computers is the only way to build a computer. This was completely prophetic. But the irony is that we had that project going before I left DEC and it never saw the light of day. It wasn't pushed. People didn't understand the commercial marketplace as opposed to the uniprocessor, because transaction processing and databases all work fine with that multiprocessor structure.

DKA: Tell me about that …

GB: So there was another missed opportunity that would have solved all their problems. It would have cost peanuts compared to the 9000 and it would have gotten DEC as the dominant downsizing supplier instead of SUN and HP.

DKA: Well tell me about this transition. You left. You had had a physical problem with your heart attack. You had been under stress and you were ready to try something new. You wanted to go back to doing something entrepreneurial? Is that what you expected when you left or you didn't know?

GB: I didn't know. Ken Fisher said come and join Encore. Henry Burkhardt the founder of DG said: "Yeah, lets do something fun - we'll get some money and we'll go start companies. Or people will come to us and we'll start companies." I asked what my responsibilities were and Ken said: "You have no responsibilities. I don't care if I ever see you." That sounded fine by me. There was a plan, however, for what Encore was going to be, and Ken wanted me to look over the technical part of that plan. Aside from that I wasn't doing a line engineering job.

Anyway, that plan didn't work. The next two or three plans didn't work. But what finally worked was we acquired a group – from DEC – building a 20 processor system called the Multimax and that was introduced in '85. It was a smaller version of the 64 processor. It wasn't from the AD group doing the 64 processor so it didn't take anything intellectually from DEC, but being in the DEC engineering environment the guys probably knew about it. This group designed the Multimax. We founded several other companies as part of Encore.

DKA: And what happened to that machine? I don't know the history.

GB: Encore is still selling it, ten years later. And Encore still exists. They're not a large company, and they go in and out of profitability. The irony is that we built a complete entire computer company at Encore. We had Multimax as the server, and it was scalable from 1 to 20 so it covered all of DEC's lines, except the low end, and then we built a concentrator for bringing terminals into the environment, and we also built a CRT terminal that allowed you to have multiple windows - it was a 21-inch terminal, like today's modern X terminals. We built X terminals three to five years before X terminals, before there was an X protocol in fact. From Multimax, we proposed Ultramax, a 1,000 processor shared memory multiprocessor consisting of an interconnected hierarchy of Multimaxes as part of DARPA's Strategic Computing Initiative. I don't know whether Ultramax ever worked.

But the tragedy was that the marketing people within Encore didn't know how to deal with any of the products. The first thing I said was this terminal has got to be an OEM terminal, we've got to get it out in volume. We had established a small entrepreneurial group, a few guys designed and set up a production line for the terminal. It was a beautiful terminal, probably the best terminal that's ever been built. It never got anywhere because the guys that we had in sales from Encore had come out of a Prime field sales force and they only knew how to sell big boxes.
So this began my era of serious questioning of anybody who has the title of marketing or sales.

And that's why I wrote so much about them in my book and the seriousness of marketing and selling. These people didn't have a clue about how to market or sell products. That was one problem, but there's a more difficult one of people in organizations. There are people who can deal with the whole birthing process of starting something new, but the vastness of large organizations is the creation of a steady state. Someone once mis-quoted that programmers were like light bulbs – you unscrew one and put another one in.

As far as I'm concerned, modern corporations, are just filled mostly with light bulbs. You know -- I need a bigger one, I need a new manager, do I have a 100 watt manager? I unscrew one over here and put a new one in, and this one burns out and you throw it away, or you get rid of them or you move them into the dead light bulb box. Because the company is in steady state. We've got to change the process a little bit it because it isn't working very well, and mostly in engineering its "I've got to get rid of some cost." We do something and sure enough the processes are all broken, usually based on what you can do with computers. You find out there's a better way of doing the process. But the vast part of the organization is steady state. It's there forever. You can take away the input or output and it'll still be there. These people will still come in and be in the offices.

Being an entrepreneur, starting something from scratch, is totally different. And people just can't, just don't like to do that. And when we started Encore we brought these very expensive, light bulbs in and they wanted to sell stuff to the people they already knew in big companies. Basically, you hire a salesperson and their address book or contacts. Well we didn't have anything to sell the big companies. Or what we had to sell, they hadn't seen before. That's just as bad. "Gee, I've got to have something that competes with this." Well we don't have anything, this is better, this is different. "Well it's not a competitive."

This problem is addressed in my book, *High Tech Ventures*. Most of these products are new, you've never seen this product before. What do you do? How do you do something when it's never existed. How do you build an organization that's never existed? How do you build a product that's never existed? How do you get this all to happen? And it's very, very tricky. I know how to do it outside of large companies. Doing it inside an existing company is very hard and a problem that I have given up on.

DKA: Can it be done inside one's company? Well, we've got to… let's answer that question on the next one.

GB: 3M is the only one that seems to be able to create totally new products and divisions. However, we should look about whether they create new products that sell to new customers or new markets.

We were talking about entrepreneuring at Siemens, and how you do it. They've got a new CEO and he's gone through and tried to change things. And they've got 50 or so divisions or business unit's that have started, and these guys are director level - one level down - and are supposed to be the change agents that try to do it. But it's unclear to me that a company vastly more bureaucratic than any U.S. company, can change.

This was at a time when Jim Gray and I were talking about scalable computers that can be made from PCs. I said, "Look, computing is going to be vastly different and you're not going to maintain the margins that you have today." After our meeting they are deciding to write a manifesto to the president and say this is not going to make it, there's too much change, we've been steadily unprofitable and we are not going to be able to get out of that. I told them: "Look, two more ratchets on Moore's Law or six years and you're out of it, you'll be so far out of it that it's not going to do you any good. You can't compete. Here's the way the world is now. You just don't get it. It's not the old style of business where you can control everything from the government, technology, to your customers. Why do you need a 1000 people working on UNIX? Why? They're not adding value, they are just adding cost, and down stream it's costing your customers an enormous amount."

DKA: So really, taking and dealing with those evolutionary changes in your product lines particularly in this field becomes enormously difficult.

GB: Yes. The best news would be if the person running the company understands the whole thing. He understands, I suspect, viscerally that something is happening. I don't know what the guys beneath him know, how old they are, what they think. There's tremendous denial. Every time I look at what's going to happen in the future, I can't believe it's going to be this way. What's the implication? The implications are vast. And the cost structure has changed so much. And that's what cost DEC so much because they evolved to have a very big cost structure. Their numbers or rather ratios had totally gotten out of control. Anybody should have been able to see them because they had the lowest productivity in the industry. Every part of the company got bloated when VAX was going well and now there was just no way to off load the costs.

The irony is I was just talking with another DEC alumnus at *InternetWorld*, and the president and founder of the company and said: "I have stock in your company, Ascend, from a venture fund investment." He said, "You know, I used to work for you." And I said that's wonderful. I am so proud of the people who came out of engineering that have started companies. The number of people who came from DEC marketing and started companies I think is nil. Especially the one's that have been successful. I can't think of a soul, because I think the difference was the way the DEC marketing organization had to operate as integrators across the company. Really what it trained was politicians. Those poor guys had to go around and lobby with me to get their product, they had to lobby with manufacturing to get resources or the right people, and they had to lobby with sales to get sales time. So what have you got? You don't have entrepreneurs, you've got politicians. Lobbyists. And that's why they've done so poorly after they left DEC.

DKA: Now you went from Encore to a very different kind of position going to Washington. And I guess that was looking at entrepreneurship or looking at new ideas and trying to drive it. Tell us what you were trying to do at that position.

GB: I look at it as another startup. Eric Bloch was the director of NSF and had come from IBM. He had been responsible for manufacturing the IBM 360. I had met him when he was the catalyst from IBM, with Bob Noyce, to establish the SRC (Semiconductor Research Consortium). His charter to me: "Pull all of these various parts of NSF that do computing researc together and create the directorate for computing – we'll call it CISE for Computer and Information Science and Engineering." That was a tremendously exciting thing to do. I loved it. It was a nice size group – about 50. Our budget was $120 - 130 million. I don't know what the budget is today, probably $2 or $3 hundred million. That was just a great time, -- to get the various divisions in place and to establish their direction and priorities.

DKA: But the culture was a very different culture. You worked in private industry and now you were working in government…

GB: I don't know what it's like now. And I don't think I could have dealt with NSF under anybody but Bloch. He had already been there for two or three years and changed NSF already. He really had influenced that organization enormously, in delegating responsibilities, cutting through bureaucracy, everything. NSF doesn't have a departmental boss, it isn't under the Department of Commerce, so we didn't have a lot of hierarchy. There was no hierarchy above us. It had a board of directors, the National Science Board. So in a sense, it was only a thousand person organization. So it was really quite small. And I'd say entrepreneurial, too, at that time, even though every congressman and senator tried to influence the outcome for their constituents.

DKA: But your goal was to define an area but also to define a strategy or help come up with a strategy. Why don't you talk about what that was and why you thought that was an appropriate strategy for computing at this time.

GB: Right. In fact I had a lot of push back on it. The first thing was just get the organization in place. The supercomputing centers were part of that, thank goodness, and one of the goals was to integrate supercomputing into computer science, which to a certain extent I totally failed at along with every successor running CISE. But I did influence supercomputing and spent a lot of time just working on the program, pulling it together, and building a strategy: "Folks, we're all going to run UNIX. We need standardization because it is a question of programs. To use supercomputers you've got to have a vast array of applications. I want to integrate that into the computer science community where the folks all speak some dialect of UNIX."

They had been running a homegrown DOE operating system at the San Diego and Illinois centers. First off, we are not spending any money evolving and maintaining a piece of code that the Department of Energy maintains. It's stupid. Get rid of it. There was a lot of resistance. I said I want compatibility up and down the line so I can take a program from an SGI or a Sun and run it on a super or minisuper from Convex. Another thing I asked for: "I want you to support a whole set of new and diverse kinds of computing facilities. We need to get into massive parallelism. This is after we've got stability." I wrote a lot of policy papers about the future and the need for flexibility.

The supercomputer guys told me initially that I was the guy who destroyed supercomputing with VAX because everyone bought their own. "You didn't provide enough capacity. We've got to have supercomputers." So they got this pile of money together. I said, but people liked those computers. Now don't you think you should tolerate smaller computers such as the Convex instead of only large centers because people really didn't like going to the centers. And there was that whole dilemma of how is it going to be funded, all the politics. I got into a lot of those issues, but couldn't get at all of them because of the politics. When I came to NSF, the guy that had been putting the supercomputing centers in place was still trying to start new ones. I said we don't know anything about capacity. We don't know what the demand is. Why do we need to do all of that? Let's wait for this to build. Besides, for supercomputing, it is better to have more resources in one place if you really want a supercomputer, rather than lots of little ones.

But he was playing the Washington trick -- the way you get power is through budget, the way you get budget is to get a program started. The reason we've got such a horribly unbalanced budget is because of the bureaucrats, who in fact get something funded, then their constituents say: "Hey, you can't cut this, I'm dependent upon this." But the demand for supercomputing has fallen off, continues to drop, so there is a smaller number of users than when I was running it, in part because smaller computers get faster, more rapidly than larger ones.

I also tried to deal with the question of who is going to pay for all of this. I wanted the scientists to pay for use. I don't believe that computing ought to be like air. It's not free. You've got to pay some token amount for use. And if you're not willing to pay something, then what's wrong? There's something wrong if you won't take some of your budget money, or if you have budget money, you might rather buy a workstation. I wanted a lot more flexibility in terms of getting an economic model of supply and demand to work.

The other thing was the John Von Neumann Center at Princeton had been established and to use the new ETA 10. It should be noted that at the five super computer centers three had Crays, one had an IBM, and one was to be an ETA (CDC owned company).

I refused to approve their budgeted expenditure because ETA didn't deliver its machine. And this was a totally novel concept within the government. How can you cut a center's budget? Congressmen, senators, staffers were all calling my office. I said: "This is not a grant, this is a contract. You have no machine so why would do we pay?" Well CDC needs the money. Ok, when CDC can deliver the machine, they get the money. CDC never delivered. And Erich backed me up.

DKA: It's certainly not like the government way of doing business.

GB: Oh that couldn't happen today. It couldn't happen with anyone other than Erich Bloch running NSF. It is how it should be.

The main thing I did that I think was really important concerned the NSFNet and how it became NREN or the Internet we have today. The net was established as part of and reported to the person who ran the supercomputer centers division. I came in and said: "Networking is going to report directly to me as a new division and not to the supercomputer division. The network is independent and distinct from the supercomputer centers"

This was based on my experience at DEC. The other part of the VAX Strategy was that we had built super networking technology called DECnet, by having a network group. It wasn't part of the computer guys who said: "We'll simply put UARTs in our computers and connect them to each other – we'll do the networking." Where is the network and why do we need a group to make links? Well the network is all of those lines and links, and it's especially all the code that makes the collection of computers work as one. So I did the same thing and said: "NSF needs a strong, independent networking group. We're going to build a network. We're starting all of that."

And so I'd say I am most comfortable with my Washington experience leading networking. We said we were going to take a lead position, the Gore Bill came out in 1986, and NSF was given the charter to lead the group on networking across all the government agencies. And then again I would like to say that NREN (for National Research and Education Network) is the only thing I can cite as inter-agencies ever doing together and agreeing on. We got everybody together from all government agencies, industry, and academe and put a plan forward in February of 1987, that was a three-phase plan to provide bandwidth. And why this is really fresh is I gave a keynote talk at InternetWorld '95 in April.

It's the role of serendipity. Most everyone think that the Internet just happened overnight. But it didn't. We had a three-day workshop of 500 people in San Diego talking about networking. We had industry - what's bandwidth going to be like? All the government agencies - what are the needs? To the academics - what can you do?

On the final morning, after listening to the previous two days, another "aha" occurred that was fundamentally the NREN plan. I drew it on a single overhead that everyone understood.

I basically said: "Here's the plan. We're really have nothing now. Our networks are overloaded and really don't work very well. Phase Zero: We get ourselves together. We make the network solid. So without a system running no one is going to believe you about the future. Then we're going to go from 56 kilobit's today in the backbone to 1.5 megabit's in 1990 using T1 and then we go immediately to 45 Mbits. In 96-97 we'll start to field test the first gigabit nets. The later stage is research, the earlier network is strictly engineering."

I called them Internet 1, 2, and 3 in a recent talk that I keynoted at InternetWorld 1995. One is ARPAnet, running 56-kilobit prototype for email. Two is what we've got today, which was really mail as a reliable delivery, the worldwide web and a prototype for three. And here's what three iDKA: telephony, audio, video, and video conferencing. It can't be ubiquitous without fiber optic speeds, there's not enough capacity. And that's three to five years down the pike. Meanwhile we can have a lot of fun with what we're doing with Internet 2.

Interestingly, the goal of ARPAnet was not mail, mail was not even conceived of. It was remote log in to other systems and sharing files. The plan didn't say anything about the application in our goals, we didn't say anything about worldwide web. We had no idea. It was proposed to be used for supercomputing. Well, all the networkers knew it wasn't supercomputers. There was no demand. We knew that supercomputers needed bandwidth, they needed to communicate, but when you really force people to use them they would prefer their own machines. I talked to various folks at DOE about this dilemma. If you really want to get a lot of power together why don't you have Los Alamos run all your computers. You've got plenty of power, you have it together. The networking is just fine. In supercomputing there is no reason to have more than one computer in the center of the earth. In fact there is every reason not to except for the de-attachment you get. You get some attachment of these people coming together. Leading the NREN effort across all the agencies that created the network plan was the other thing I did at NSF I'm proud of.

And then in the computer science area I proposed: "We are going to focus on parallelism. So what is the challenge? We've got to get out from under our thinking about computing and we've got to go parallel." I put forth a taxonomy and it hasn't changed. The irony is I had advocated the computer science department's working on networks and workstations. That's where all the power is, so why don't you guys exploit that? Well they didn't even hear any of that. And now Berkeley and Wisconsin have nice research efforts aimed this way.

DKA: So it's slowly coming back.

GB: Yes, slower than I would have liked, but in fact people are going the right way at least. So I think the push to parallelism and saying this is going to be the dominant focus of the work which in some sense was complementing what ARPA was doing by funding all those parallel machines.

DKA: Gordon, when you were receiving an award this year for innovation, and we've talked for quite awhile now about the astonishing career you've had in various aspects of that innovation as down at the bench level making innovations, at the management level overseeing innovation, in the academic sector studying innovation, writing about innovation, in the policy arena funding innovation, trying to pick directions, I wonder as you sort of survey that what concluding thoughts you have or what thoughts you have on the future of this innovation and particularly this industry – how it should be conducted, where it should be conducted, what your experience tells you about where the industry should be going and how it should inter-relate to government and to other bodies that work together. It's a big question and you can sort of take it in any direction, but it's remarkable when you think about the various perspectives that you've had and it's hard to think of anybody whose seen it -- not just seen it but participated.

GB: Seeing is one thing, but being in it is the other. And that's what I've enjoyed. I've really enjoyed every one of the environments. When I was at Carnegie I thought boy this is really great, and then running engineering at DEC was wonderful, NSF, and now dealing with entrepreneurs. It's the stimulation or encouragement of people doing that. I've been very critical about certain government aspects of the way they encourage it, which is really more a reflection -- at least in my view -- of human nature than anything else.

I'm strongly anti big programs like the Advanced Technology Program of the Department of Commerce. I'm willing to spend the money, but it's the way it's couched. I would put it out as a loan. I would even let the government invest in some way in venture kinds of things, earmarked in a certain way to support work coming out of universities. One paper I wrote focused on this question of what policies have worked for funding innovation when applied to supercomputing system. And the only two good heuristics that I can cite are: university research begets ideas and companies and its great to fund research as such; and universities, government labs, government need to be purchasers NOT developers of innovative equipment. Those are the only two things that will work. I'm really against funding companies and especially large companies for building things that are going to be the next whatever it is. Because so many times those programs end up as programs that the companies don't have the nerve to cut out themselves and that there's no way to commercialize. So I would almost require a way of commercialization. I always worry about commercialization, about why are we doing this.

You know, I have a very different view about science than most every policy maker in the U.S. I said this once in Erich Bloch's staff meeting and he thought it was off the wall, sarcastic and almost anti-research funding. Ed Davis once said I was getting cynical, but I say: "No don't take it that way. I'm giving you a model of human nature and don't think of it in those terms. Just understand how people behave within the bounds you set."

Some times I think that scientists are like a bunch of gold miners. If you're in a new field, a new gold field, and you put these gold miners out in it and they're digging up gold all the time and they've been at work for a year or so and you walk out there and all that's laying there is just this gold. And the problem is nobody really wants the gold, they want it refined and made into something. It has no intrinsic value as gold. And the way we fund science very often is: "Oh we have to fund science because we've got to find new gold."

I also worry about the economic future of the country. And I'm so different about what I think what's wrong with it. I look at the price of the yen and it's heading toward 80. I remember when I was in Japan in 1978 it was 200 yen per dollar or even more and then 100 when I was there a year ago and it was exactly a spot on 100. Then a year before that I think it was 135. I can't see any way to have the current system work with massive trade imbalance. I don't think this country works if we so gullible to the issue of wanting free trade but yet nobody plays by free trade rules. I mean all the deals that we have in free trade. I'd almost rather say: "Well here, just take our money and go deal with it." And I don't know if there is any other way. And economists seem not to understand this.

I read something in *Business Week* recently that it's the government, it's a balanced budget problem. Turns out a balanced budget problem is only half of it because we blow a couple hundred billion there. But the other thing is this trade thing is so serious, and our economy goes up, and the economists can't figure out why the trade is getting so bad. Well, it turns out the economy is good, we have more money to buy things, and what do we buy? What do we make that nobody else can make right now? And the only thing I can think of are Intel PC chips. Everything else is made offshore as a fundamental thing. I mean, the whole car issue has sort of stabilized in a funny way, the car guys are happy because they're making a lot of money, the Japanese are happy because their cars are more expensive and they've made deals with the American auto makers and you probably can never figure it out anyway because of the onshore/offshore. But if you worry about ownership it turns out that we have lost so much here, and it's a funny thing, but science may be to blame.

DKA: Because we funded science but we lost innovation.

GB: We've placed so much emphasis on science, and so much of Washington is controlled by it. Basically science is good. How can you be against science? Well I may be when it is unbalanced. Because how are you going to convert that into gold? How are you going to convert that into commerce? Because if all you do is mine the gold and leave it on the ground, then somebody else is going to make the jewelry. The Japanese are extraordinary at making jewelry.

DKA: So we have the gold but . . .

GB: So we've got the gold, but then a couple of these miners go in their cabins at night and they make little trinkets and things and they say: "Hmmm, that's pretty good stuff!" And because they don't really have an avenue for making lots of trinkets, they have to show off their trinkets to earn respect. They only get points, by the way, if they show it off to everybody, because scientists only get points for the mining of knowledge not the utilization of knowledge. So they say: "Hmmm, that's pretty good!" and the Japanese say: "Yeah, that's pretty good. Mind if I make a few million of those?" Then its off and running and we miss the whole market thing. Our balance of trade is just extraordinarily bad and I don't see any way to turn it around.

DKA: I just wonder if that's why you now are working with individual entrepreneurs.

GB: Yeah it's that. I only work in environments that I can influence, can affect, that I can bring something to the party about. So why I'm working with individual startups is because we can see these ideas and we know how to do it. I want to see it come in, and I want to see it be an enterprise so that this thing gets revenue and we'll affect the balance of trade. That's fundamentally why I do it. Because in a large company it's very hard to. I can go back to work in a large company and influence and do it, but it's more fun this way. In a sense I'm not changing very much of what I did at DEC. At DEC I had this universe of 6000 engineers and somebody would say: "Hey, I've got this new idea to make a mail system or word processor or new interconnect to make all our software connect and work together." Those things were all done in a sense as encouraging entrepreneurial efforts.

The more I've gotten away from large organizations, the more I feel that this organizational hierarchy has to be totally supportive up and down. It starts with the CEO and it goes down from there. Why am I such a fan of Microsoft? Look at Gates, Allchin, Maritz … go down the line of people running the company. Everyone link in the management chains is filled with great people. Why I like them is they're smart, they know their business, they know technology and they know what they're doing and they've got this mission of creating this industry and wanting to put it out there. And I haven't seen that at other companies. Until Microsoft, I though DEC had the greatest engineering organization, but Microsoft is substantially better.

DEC is doing a lot of interesting Internet technology and products right now and they an advanced development group in the bay area, but it is managed by an incompetent. I don't see that they are going to figure out how to do it as a business. My partner Heidi Mason and I offered to help. We'll look at those things and help put a process in place so you can make this thing entrepreneurial and test them. We may try, if they take us up on it, but they may not want to hear what we say. I just like to see ideas come into existence. I guess everything that I'm working is like that.

Have you heard of our other little project to produce some historical videos?

DKA: No.

GB: Ah, they are fun, too. It's turning out to be entrepreneurial. We have videos from the Computer Museum's film and video library and some from the Smithsonian. Anyway, we have put the first one together -- it's a four-tape series on the first computers. The first one is on the first four computers - Zuse, Atanasoff, Stibitz and Aiken. The next one is on ENIAC, EDVAC, and that line. The third is on the MIT, IBM and early DEC machines. The final one is the English machines. Four one-hour videos get us up to 1955 or so. I funded the first one, and it's all using original material. I'm the narrator gluing the pieces together. The ACM has come in as a partner. And then we're going to try to get some other folks. We've got the Los Alamos MANIAC film which is a really excellent one explaining what computing is all about, the four boxes, all the classic stuff. They did a very good job of making color and 16-millimeter sound.

DKA: Before we wrap this up, I wonder if there is anything else that you want to... any final reflections or regrets or anything that you want to … we've sort of gone through the whole thing and we've heard a lot of information, but just wondered if there is anything else that you want to close with.

GB: I guess I really just get a kick out of seeing new uses for computers – seeing our machines reach their potential and helping the people, especially entrepreneurs who are driving the new applications. I don't have to take it all the way to the end. In a sense I looked at the Internet and web -- it was flashy, neat and all of that, but I rarely see any surprises. Well did you know this or that? Well I didn't think about it that way. Once you've got the infrastructure, anybody can generate – well not anybody - you can generate most of what follows. The network was one that was like that. I like to put things in place and let things take off, given the infrastructure. So I guess that's what I enjoy. How do you do things that can then enable other people to do a lot with it, whether it's a component to use as a minicomputer, or a network to use to build this or that.

I've started working with Jim Gray who I just met six months ago, and we're having a wonderful time. We're talking about an architecture we call SNAP – Scalable Networks and Platforms - which is a dream of how to build world-scale computers out of an ATM or worldwide network and a collected set of computers. The ideas are gradually unfolding. We're giving our talk and content to anybody who wants it. And so we're using that as a vehicle to say we're in the architecture business, we're building this great computer, only we're not doing it at all. We're just coming out of our heads and having other people say: "Gee, these are good ideas!" And then somebody takes an idea here and there, and this is really a platform -- how does this all work.

In switching to clusters, I gave up my 30 year belief in multiprocessors. They are just too hard and too expensive to build in a scalable fashion. There are too many reasons why we just can't get there with them starting with they take too long to build and are likely to obsolete when introduced. Furthermore, unlike clusters, with every change in model, the whole system has to be re-designed. Clusters can evolve and accept nodes over a period of several technology generations.

DKA: That's the next dream.

GB: Yeah, it's a dream, but we're already influencing others. We went off and determined that we needed a switch – a System Area Network switch to interconnect things. Well, we went to Tandem and said: "Hey, you've got a pretty good switch!" But then Intel's got theirs and somebody else has one. And we say: "Wait, to build the kind of computer we need, you guys have got to standardize this. You can't hold it to yourself. How are you going to make this a standard?" And so we're off trying to get this switch in place so that anybody can build these computers in a wild way. We're having a lot of fun with that.

DKA: Well that's great. That's a good place to stop. Great, this has been very interesting. Thank you for taking the time to do this. Let's look at the plan you promised on your computer.

Epilog and More Things to Say in May 2000 After the Interview

GB: I feel fortunate in being able to create two strategies that were successful plans for implementing technologies and products over a 15 year time scale. Ironically, the VAX Strategy may have been a reason for Digital's demise because it let them not think about the market while they were busily implementing new VAXen and selling them.

The NREN Bandwidth plan was equally useful. It is interesting to look at the original figure from the first report because of its simplicity and usefulness as a strategy and plan.

Everyone always talks about strategies, but rarely do you see one that actually works that you are able to learn from.

"aha's"

I have been fortunate to create more than one significant "aha" in my lifetime. I recall the first small one was the invention of patented switching circuit used for the memory cycle of PDP-4 that was a generalization of the flip-flop made by cross-coupling n-NAND gates to make an n-state device. This allowed me to understand and feel exactly what an "aha" was.

Two "aha's" came from writing Computer Structures with Allen Newell: the general registers idea and the Unibus. Although they were inventions, the ISP and PMS notations for computer structures were also important, but the "aha" cannot be recalled.

The VAX Strategy was another "aha" that occurred while vacationing in Tahitii. The NREN Plan came out of the very stimulating meeting at San Diego in February 1987 at an interagency workshop organized to explore the technologies and needs for an NREN in order to respond to the "Gore Bill" for an information superhighway for supercomputing.

"Prize Power" to Mark and Stimulate Technology Progress: The Gordon Bell Prize

In 1987 while at NSF, I agreed to give Gordon Bell prizes of $2,500 per year for advances in application parallelism. ARPA's Strategic Computing Initiative was starting to bear fruit, but applications were significantly lagging. The first prize given in 1987 was to three researchers at Sandia National Labs for applications of a 1,024 node Ncube computer. The lab publicized the award of the prize to call attention to their very significant accomplishment. Since the first prize achieved a performance level approaching about ½ Gflops, performance has increased to over 1 Tflops in 1999 using up to 10,000 processors. The annual prize of $5,000 will be given at least until 2007.

In May 2000, Jim Gray and I began awarding $10,000 prizes for network performance to understand and stimulate this area that is the basis of large scale distributed computing. The first prize was awarded for two Windows 2000 PCs transferring data at 770 Gigabits per second. They operated between Virginia and Washington, passing through one dozen nodes that are part of the Internet 2 Backbone.

From this experience, I strongly advocate the use of prizes to mark and stimulate technological development. Unlike awards and medals that are given a posteri, prizes stimulate effort and mark progress.

I enjoy betting with technologists about future progress. Currently I have a perfect win record. The secret is to bet against optimist, but use knowledge of the marketplace and other factors to insure a win.

In 1990, I bet Danny Hillis, the founder of Thinking Machines that by December 1995, the majority of technical computing measured in floating-point operations per month and costing more than a million dollars would NOT be done on computers with more than 1,000 processing elements. I relied on the fact that traditional supers would supply much of the capacity or that because of cost, there would only be a few 1,000 node computers and that lower priced machines would carry out the bulk of the computation. The loser was required to write a paper. Danny has yet to write the paper.

At the 1995 InternetWorld, I bet DEC's VP of Marketing for Internet products $100, that SUN would be the dominant supplier within a year and that DEC's position would be nil. The VP reneged, but was still with the company 3 years later in the same position.

At one point, mega-manager Bob Allen, CEO of AT&T decided that his company had to buy the computer company NCR. A friend of mine, Rob Wilmot, former Chairman and CEO of the English computer company, ICL volunteered that he had played a role in the acquisition. I criticized him for playing fast and loose with our national assets by getting them tied up with such a losing deal. I bet just $100 that within 3 years the deal would have gone bad and AT&T would have to divest the company. Rob paid.

October 1993 several members of the Microsoft TAB bet the ultimate optimist, Raj Reddy and Professor Ed Lazowska that there would not be significant video on demand in service by 1996. Raj bought dinners.

Raj Reddy made two other bets in 1993 (decided in 2003). In 10 years, there will be production model cars that drive themselves. In 10 years, we agree that AI (Artificial Intelligence) has made more of an impact on society than the transistor or IC.
On March 1997, Raj Reddy Jim Gray, and Dan Ling believe that at least 10K Workstations, located in at least 10 sites, in at least 3 states will be able to communicate with one another over an end to end path operating at least at a 1 Gigabit per second rate.

On November 1997, two bets were made with Nicholas Negroponte:
1. $1000 even odds. That by December 31, 2000 there will NOT be 1 billion web users.
2. $1000 5:1 odds. That by December 31, 2001, there will NOT be 1 billion web users.
This is measured by people with one or more addresses that can access Internet, but only one user is counted no matter how many addresses each has. Intranet users who do not have the ability to access the web aren't counted. IP addresses aren't counted.
August 1999, $1K bet with Herman Hauser, Chairman of Amadeus. More LEP (Light Emitting Polymer) displays will NOT be sold than LCD displays in Q4 2004.

A Bet I Expect To Lose
April 1996: I optimistically bet with Jim Gray that half of the PCs will ship with videophone capability by April 2001. In an April 2000 talk, Bill Gates said he expected all future PCs to have cameras.