Oral History of Gordon Bell, part 3

Interviewed by:
Gardner Hendrie

Recorded February, 2006
San Francisco, CA

CHM Reference number: X3202.2006

Hendrie: I'd like to welcome Gordon Bell to our third series of oral history tapings of his career. It's a long career and I'm not sure three tapings will do it, but you were mentioning some thoughts you had about Ken Olsen after you left and went to Encore.

Bell: Yeah. Right. I'm delighted that you've interviewed Ken and that he's talked about his risk management philosophy, philosophies in what he did and things like that. And I think one of the things I'd love to be able to do and we haven't done yet is to get all of his papers, if there's any way to get access to the papers because the memos and things like that I think will be very telling about Ken. In a way I always appreciated what he was doing. No, appreciate is not the right word. I kind of understood a lot of the motivation of what he did and some of the behavior. Now there are people who actually analyzed Ken and are much better at doing it, but from my point of view I appreciated him more when I left DEC. I was used to the DEC environment. If you were in a corporate environment and that's the only one that you've really been in….I'd been in large corporations as a co-op student and in fact I didn't want to go to work based on my experience as a co-op student at GE and the American Electric Power and looking at other organizations. So I had had this long period of being at DEC and then I went with a startup. Namely Henry Burkhardt convinced me to come help him and Ken Fisher start Encore which I did in '83. This is after I had the heart attack, and so it was at that point that I really got an appreciation for Ken Olsen because in fact we got into a situation at Encore where ultimately it was very destructive. There was a break between Ken Fisher and Henry, and I certainly sided with Henry and then ultimately left. But it was really on the basis of Ken not being good enough and also of really having an appreciation for . . . of arguing things out in an intellectual fashion by looking at the facts.

And I always used to say about decisions, "Let the data decide as opposed to you give me the information and I will make a decision." So these were the classic ways, thoughts that people get into normally to pass it up the chain of command and then it'll be decided at the top. Well, DEC never really worked that way. A lot of decisions were made by the top group but in fact it was pretty much on a consensus basis. By the time it got there it was clear what we should be doing and there sometimes were contentious things. So I really got a wonderful appreciation. And I said, "Gee, with Fisher he's about a three milli-Olsen in terms of management." It was a CPU power but it was also the question of how do you go about deciding things and also taking yourself too seriously as managers and the like. So I had a wonderful appreciation for Ken Olsen as a manager for a long period of time.

And now on the other hand I have also said that I thought there was a time -- and I can almost tell you exactly the day that they did it -- there was a time when he actually stopped being the president and got involved in issues, got too involved in issues. It doesn't mean that Ken wasn't always involved in issues, but in fact taking sides in issues, and so I'd say the ill fated PCs if you can call them that were Ken's. I think Ken felt too strongly about them as his projects and getting too close to them. And then that came about because he was interviewed by I think -- I don't remember where she worked but I do remember her -- and that's Julie Pitta who might have been Forbes or Business Week then. Ken went off on a rail of how CEOs are not getting involved in management, or she challenged him that you guys aren't involved, they're too aloof. So this was a gauntlet that she laid down, and then he started getting much more involved in stuff as opposed to sort of being very good at sort of seeing, sensing problems around the organization and raising those problems up so that they could get solved and then moving forward based on that and with no vested interest. But he had a wonderful set of sensors if you weren't the brunt of it. He had a lot of insight in terms of things that he wrote. He wrote a little series, kind of essays called parables, about management at various times and sometimes they were okay, sometimes they were a little bit offbeat maybe because I would very often take one and sort of say, "Well, you haven't really looked at this aspect", and so I was probably one of the ones to poke fun at them. And he was very bright and cleaver too. For example, he didn't like the way cabling was done, and cabling is always hard. Anybody who's got a PC understands that problem right now which hasn't gotten any better -- it's gotten worse. So he went into -- I think it was my office or something -- and looked at the back of essentially a DECMate with a terminal and some modems and stuff and it had a lot of cables in the back then and it was kind of halfway dressed. And then he took a great picture of this and made a big chart or a poster of it and sort of put on the top "engineering or marketing?" So here were the only two people that he could blame or something as opposed to who is the engineer who created all of this stuff? But given the modularity of the situation cabling is not any better but creates the problem. So that's kind of an example of sort of his kind of wry sense of humor on the thing and then I think he wrote a memo that went along with it. There were sort of a bunch of one-liners of excuses that any comedian would have been delighted to have written about this stuff so there was a lot of, I'd say a lot of good times there that we had. So let's go back to . . .

Hendrie: Do you know whether you have any of those memos or . . .

Bell: I have a few of the memos. I particularly have that poster because I always loved it. My regret is absolutely that I threw anything away having to do with correspondence with him and similarly things that I had. But on the other hand one, at one point Dick Clayton took me aside and he said you're emulating Ken. There's things that you do that emulate Ken and you realize that these things aren't productive for you and when you do them it's not productive for us. And he said you will get a lot further if you just sort of cool some of this. Because Ken could be very sarcastic and I could sort of outdo him in that, or I guess I could certainly keep up with him. And so I stopped doing that and certainly the team felt a lot better that I wasn't out to show that I was smarter than they were or cleverer I think or something like that. But anyway, the nice thing is I think I've scanned all of the stuff that I have and there's a lot of things in my files that have to do with Ken and sort of the interaction, but I don't have the other side. I think Ted Johnson actually probably has a lot of the stuff that he's got around but it would be very valuable because there could be a nice book on his philosophy and then also how it changed because it did change. Ken was influenced early on by General Doriot enormously. Ed Schein's book I think does a lot to explain it. I don't hold with the overall philosophy that it was the culture that made DEC undo itself. I don't believe that at all. In that regard, I hold Ken and the executive committee responsible and the board and then when Palmer took over it was clearly Palmer's undoing of the company, which I wrote in Schein's book. Schein let me have the last section of that book. I think it's pretty much toned down a lot in there to not name names, but anyway people say it's accurate but. . .

Hendrie: Should we loop back now?

Bell: Yeah, we can go loop back.

Hendrie: What year would we be going back to if we were going to start when the PDP-6 maybe was just a dream or something?

Bell: I can't remember what crystallized the PDP-6 because there was a PDP-3. There was a specification for a PDP-3 that Ben Gurley had put out and in a sense it was a double word PDP-1 almost, 36-bits and kind of a natural extension with index registers and almost in the style of the 7090 or 704, that series of machine, and so the PDP-3 spec was around. I think I talked about the PDP-3. Okay. This is one of those absolutely wonderful stories of creativity and invention that Harlan Anderson and I engaged in at one point. Anyway, the PDP-3 was, or rather DEC was, sort of going along and I think sales were not over 10 million or so. We had the PDP-1 and we had fortunately got the order from IT&T to make message switching equipment, and I was a project engineer and that's when we did the UART and so I got enamored with UART and telegraphy. So the company was doing well. It was small and struggling and all of that. We'd done the 4 and the 5. Let's see. Yeah, we'd done the 4 and the 5 hadn't been done yet, we hadn't worked on that yet, but the PDP-3 was there as a spec. It was a printed, mimeographed spec sheet and we had sold one. We offered it to Air Force Cambridge Research Lab and it took a couple years to get the order. Well, they actually bought it, and this is just as the company was changing and we were finding out how hard it was and what computers were all about. We still didn't know how hard software was, had no idea about software. Anyway, so we got this bloody order and we said "Oh, my god, how are we going to make this computer at this time?" The company was sort of stretched tight as a rubber band. We drove over to talk about the order with a guy by the name of Charlton Walter who was a speech UI kind of researcher and I can't remember what they were doing at AFCRL. But anyway we were going over there. It was on Route 2 and I can't remember whether I thought of it or Andy thought of it, but it was right as you would have gone right and up to 128 and you go straight there entering 2A. He said, "He wants a 36-bit machine, two 18-bit machines will do just as well and that'll be much more interesting for these guys, let's convert this into two computers." And sure enough, we went over there and I think we had all of five miles to go when we finalized the story.

Hendrie: To decide how you were going to do this.

Bell: How we were going to do it and invent it and we convinced them to do it. So we didn't have to build a machine at that point. Now it turns out a company actually did take the specs and they used the DEC modules and they actually built a PDP-3 but it was later than that. Ken and Andy sort of said "yeah, we need a bigger computer. We've been making these small machines but let's make a real computer." And so that was the 6. I was probably a party to all this, of let's make a real computer here and then I set about with a design. Of course, Wes Clark had designed the TX-2, a wonderful 36-bit computer.

Hendrie: There is a PDP-3 brochure -- I believe it's in the museum -- that is for the PDP 1, 2 and 3.

Bell: Right.

Hendrie: What was the 2?

Bell: The 2 to my knowledge . . .

Hendrie: Obviously, it never got built.

Bell: Never got to anything other than a reserve number for a 24-bit computer. But I don't even think it got 24 bits assigned to the 2, but that was kind of the idea of it, and frankly I don't recall the brochure.

Hendrie: I think it's a classic.

Bell: Yeah. That would be great, but the 3 was actually a manual one and one was built at a little lab in Waltham and I don't remember who did it. I actually saw it, <inaudible> but anyway so the 6. . . I don't know why we skipped. Well, I guess we got 4, 5 and the 6 was obviously there and then Alan and I started working on it

Hendrie: This is Alan Kotok?

Bell: I'm sorry. Yeah, Alan Kotok. And I think we were the only two initially assigned to it, and then as it sort of picked up steam we got several circuit people and I think. . . I'm trying to recall. Actually, I saw a memo by Clark Frazier in Mylifebits discussing the order code, so he influenced it. I think we set it at the 10. I think we were aiming at a 10-megahertz clock. Well, in those days, or in the DEC world, there wasn't a clock as we know clocks today. The 6 and actually the 10 were a synchronous logic that was sort of driven by pulse delay line and pulse and delay lines and so in this sense you have a flow chart of how the machine works. My flow charts are on the web that came from, I think it was Tom Knight from the AI Lab. They had 6 and 10s and he actually scanned the prints. So you can look at the prints and actually understand the whole machine, because it's a machine that I think we had half a dozen pages of flow charts and the flow charts basically defined every pulse that triggered every register transfer. So you could look and see what the instructions were doing and what's gated when and where and so on. And then the time when it was really crystallized was when I went to a conference in Monterey and gave a talk. John Leng who was the instrumentation engineer at Chalk River Atomic Energy in Chalk River gave a talk on pulse height analyzers using computers and so DEC had the first use. We used the PDP-1 as a pulse height analyzer and so the paper was on that, and in fact I have that paper. And then I remember sitting in the back of the room and drawing flow charts and things like that and working on the design. And in fact Dit Morse went with me on that trip too, and Dit took over the operating system for the 6. I subsequently fired Dit, and I think he's probably the only one that I can ever say I personally fired. But anyway, when we started working on the software there was this problem of Dit coming in at night and everybody else was working during the day. And then Dit would make changes and then nothing would work in the daytime and so he would leave a cryptic thing. I changed the assignment of all of the registers of this and so it was very sporadic in terms of what was really done. It was hard to make progress.

Hendrie: One might say not a team player?

Bell: Yeah. Right. It was all, "I'll decide it and I'll tell you when I get it decided," and so we were under the gun to get the operating system going and . . .

Hendrie: Was John Leng you said the John Leng that was at Chalk River, did he eventually join DEC?

Bell: That's the same John. John came in as a salesperson. I believe an engineer I think he was, whether he was at DEC Canada for a while but he ultimately came to DEC. He was in California for a time and then I think in the sales organization and then he came in and ultimately headed the PDP-10 group and then I think large computers was his main job. Yeah, so John was a very valuable guy.

Hendrie: What sort of ideas did you try to put in the PDP-6? Were you thinking about timesharing or any other uses of the machine when you worked on the architecture?

Bell: Right. I think that there was so much that was around at that time and I wish I had dates, and I probably do because I've got a bunch of notebooks and I really haven't tried to look at the notebooks in terms of when the first entries were and things like that. But what was sort of floating around at this time was [John] McCarthy and [Marvin] Minsky had described timesharing at MIT, and then Corby [Fernando Corbato] came in and built the CTSS using the 7090. And then I also remember a day or some days when there was a snowstorm, vast, a typical New England snowstorm, couldn't see anything, and Ben Gurley and I were at Maynard and [Ed] Fredkin and Minsky were at BBN and we were going to propose -- and in fact I think we probably did propose to some damn government agency or bureaucracy -- an idea for a large machine or something. There was a quote for a machine and it was Minsky who wrote a spec. The Stretch had just come out or was coming out, so it was in '62, '63 or something like that, and so basically over a few days we designed the most audacious computer we could think of. It was 36, 48 or 64 bits or something like that. It was a very large thing and it had tunnel-diode registers and everything, a large bank of fast registers. And the registers we determined were good and so we proposed that, and again I don't know whatever happened to the document but the document was in the form of a teletype printout that these guys had shipped to us. I think we may have just delivered the first PDP-1 -- probably not delivered the first PDP-1 timesharing system to BBN but that was one that we had eventually delivered. So the ideas at MIT and then the PDP-1 timesharing one at BBN were sort of the early, very first timesharing machines, and then Dartmouth went off and did their stuff in a different way with the GE machines. So those were the ideas. So timesharing was really, "Gee, let's make a timesharing computer," and that was kind of the thing in the sky, and in fact DEC had the first timesharing computer that was designed from the bottom up. Now Berkeley on the other hand claims this. Butler Lampson argues with me that in fact since we weren't swapping programs it didn't count. On our first machine we didn't do swap programs because we didn't have a drum or disk on the first shipments. We weren't swapping, we were multi programmed in, so you had everything that had resided in core. But then, I don't remember when the first drum or disc got put on a PDP-1, but it wasn't very long after that. I think it was '66, '67, '68.

Hendrie: Wasn't there a swapping drum on the BB&N machine?

Bell: The BBN, oh, yeah.

Hendrie: You had worked on the design of that, didn't you?

Bell: Yeah. The BBN swapping drum was sort of a tour de force in terms of a swapper. I designed it, but thank god somebody else had to make it work because it's one of those things…well, you could put the logic down and I did the logic design of it.

Hendrie: The electricity didn't behave quite as well.

Bell: Yeah, right. You're reading and writing at the same time and so there's a lot of current going along and so you don't know whether what you're reading is what you wrote or just wrote or what, but basically in one drum revolution we were able to, quote, "do a swap," so we were able to read in the core memory and dump that core load into another part of memory the drum.

Hendrie: What were some of the concepts in the 6 that were not in the 1 in addition to obviously the word length and thus you could have a richer instruction set?

Bell: There was the idea of having general purpose registers. That's kind of the big thing that we had architecturally. We didn't know it at the time, but the Pegasus in the UK had a somewhat similar kind of architecture and then the 360 used general registers also. Ours in a sense were a lot more general. I started writing about the general registers at Carnegie Institute of Technology when I went there on the faculty in 1966. I was describing general registers to Professor Alan Perlis one day at lunch, when I came upon the idea of the PDP-11 general registers. The program counter was not in the registers, and so we put them it in the registers in the PDP-11.

Hendrie: But in the 6 . . .

Bell: In the 6 there was a separate PC.

Hendrie: You didn't have a thing like an accumulator or a special purpose in a B register or a multiplier.

Bell: Yeah. So all of those registers that were in one bank unlike a 7090 or the traditional earlier 701, almost a von Neumann machine, AC, and multiplier quotient register and the super index registers. Those were all uniform and I'd say architecturally that was the most interesting thing. When I think about the PDP-6 that's what people loved the most. There are people that still say yes, what is this, and they'll ask me about what this instruction does if I do that? And this is a quiz by the way. So there was that and just the way it was all kind of laid out and I remember doing that at the Monterey meeting of laying out instructions on a grid with all of the of 256, no, 512 essentially instructions, 9 bits of op codes.

Hendrie: 9-bits available.

Bell: Yeah. Right.

Hendrie: Was there the concept of a protected mode or anything like that?

Bell: Oh, yeah. This was the big part of the architecture. Timesharing we knew needed to have the protection, and I think the 7090 had a second bank of memory and so they ran user mode in that and relocation of programs. I don't remember whether it had a base and bounds registers or not.

Hendrie: I think Corby puts a bound resister or something for the 7090.

Bell: Yeah, or something so you had to stay in that, but anyway we've had full relocation registers so that you basically had your area of core and then you operated within that area. And in fact Tom Hastings at one point said of all the routines I wrote for the PDP-6 timesharing system there were only two. One was the core allocator and mover arounder so that it said I need some core, okay, well you can have it and then here it is and I moved things around so you could get it. And then the other one was the telegraph handler which was for people who think in go to's. You shouldn't have go to's. There were only go to's in the program, which is you can tell what is this character, what mode are you in, what do you do with this or that? And it was bad but almost impossible to debug and get right. I don't know how they're written today, but I think they're all probably table driven and they do the right thing. But that was that whole business of protection, that you had to have a protected thing, so that was really kind of why it was designed that way in the beginning. When the 360 came out in '64 I didn't quite understand. They didn't go far enough in my opinion about what it should be. We introduced several machines and I was probably burned out and decided to leave for Carnegie Tech. In retrospect, it was also because I didn't feel needed.

Hendrie: When was this?

Bell: That was in June of ’66.

Hendrie: What was the state of the PDP-6 then?

Bell: We had delivered 10 6s.

Hendrie: Okay.

Bell: And in retrospect I’ve got to say I probably was burned out at that time, certainly burned out. If you look at symptoms in modern day, it would probably be that, and I think the last thing I said was “Gee, we need another pair of registers and let’s put in two relocation registers.” One’s okay but two gets you a lot more and that way you could then have one copy of the compiler there and then let that operate on other parts of the program. And so there was that aspect. And by the way I wrote some of this in response to -- or not response but clarification -- of what Fred Brooks wrote. Brooks and Blaauw wrote a book about architecture, a very large, complete book about the philosophies of design. In a way the thing that I was always the most interested in was what I call the PMS structure, that is, the block diagrams and how those things operate together, kind of a plumbing view of the computer. And so there were two. So we had an I/O bus and a memory bus for doing all the I/O. All the I/O hung on the I/O bus. This was before the 5, and I think the 1 and the 4 were radial and at some point I said this is stupid, they ought to be bused, and the 5 I think was mostly bused and then the 6 was all a bus with I/O and memory. With the 11, we went to one bus for both I/O and memory, called the UniBus.

Hendrie: Of course with the 11 you got the UniBus.

Bell: And at Carnegie that fell out of when I was writing this book on architecture, and I said “Why do you have two buses?” So like the 11s general registers, the 11s came out of writing the book with Allen Newell.

Bell: In the 6 we had two buses, one for memory and one for the I/O. And by the way, at the time, you know, how we did memory was to make it all asynchronous because I couldn't pin anyone down on what memory I was going to have or its specs. Not the foggiest idea. "Well, I really would like to have an approximate two microsecond with one microsecond access." "Well, we don't know if we can do that." So, I said, "Okay, we'll just use PDP-1 memories initially, stack a couple of those together, and we'll start with that." And that was basically what we did for the introduction.

Hendrie: You could do a lot of debugging. Software and hardware with that.

Bell: Yeah, we can make a machine that way. And so, fundamentally that was some of the rationale. And that those were asynchronous and that you sort of put out a pulse and said, you know, here's the memory I want, you have it and then send it some address bits. And so, it was a very simple, clean interface. And I will attribute that idea to creating the add-on memory business. So, you know, we sold one, or I don't know which one we sold, but a very early one, to Stanford. And, of course, the first thing John McCarthy did was say, "Geez, I want a large, fast good memory here. I am going to go out and buy one. DEC is charging a zillion dollars for memory. Mr. Ampex, will you mind designing us a big memory for this? And we want it quick and cheap." And so we probably started the add-on memory business. I think that was probably the first add-on memory business that was going. It was easy to interface to.

Hendrie: Well, it was modular because it was on a bus?

Bell: Yeah, so it's easy. Sure, that's easy enough. The memory and I/O interfaces were published in the reference manual. And so this guy has fed us the right stuff. All we have to do is make a little interface for it. And that was the basic idea. And then on the I/O side it was a similar kind of thing so people could add on. And so this is very much in the tradition I had learned about almost starting with the TX-0, that, you know, the interesting thing was to be able to add stuff to the computer. It was necessary to be able to add peripherals like A to D. And so basically, the I/O was that way, too. And in fact, I think some of the interesting things, how I felt about the I/O, was that it would be programmed, that the transfers, I wanted to simplify the control of high speed transfers. And what we did was basically made the control of data words coming in and out under the control of the CPU. So basically, a device would say, "I want to transfer a word," and it would go to a particular location and say, "Here's my word." And then the block transfer instruction said, "Oh, it goes here." And it decremented a count and then incremented the place it went. So that was all block transferred on the basis of a special kind of an interrupt. So we had all that in there. And that was more of the philosophy I've always had about I/O and I/O processors. I don't like I/O processors because they invariably have a special instruction set, yet a program has to get involved, you know. A task can't do very much before it has to interrupt the CPU. And so my philosophy was, if the CPU is going to take a lot of effort, and you run out of CPU for that, then get another CPU. So, I like fungible resources versus dedicated things that don't do very much resourcing.

Hendrie: Now, was there, well, I think some people called it a DMA mode so that once the processor had set up a block where to start and how many words then it didn't have to get interrupted again and...

Bell: Right. That was how it worked, yeah.

Hendrie: That's how it worked, okay. So, then the memory would...

Bell: Yeah. So, all the DMA logic was executed by the CPU. And namely, you had the normal interrupt, which is interrupt and go to a location. But this interrupt said, "Take this word," and then it went and put it in the block transfer instruction. So there were two memory cycles, namely, one to pick up control and do the increment and put it back, and then one to actually make the transfer in or out. So, you basically lost a little bit of the computing power by having the registers part of the memory. But on the other hand, you saved all of the logic of DMA logic that was normally in the controller. By the way, we could do it either way, too, and have the control specify the location.

Hendrie: Sort of the philosophy to it?

Bell: Yeah. That was the philosophy. And you know, we had drums that were faster than that, and we didn't want to lose the extra time.

Hendrie: So you would do them as DMA?

Bell: Yeah, just put the registers over there in the controller and you set them up and they'd do a block transfer. And I think there were some other things, too -- I can't remember all of the innovations in the architecture. There was the equivalent of extra codes. And this is an idea that I'd gotten from the Atlas.

Hendrie: I was going to ask you whether you got any ideas from the Atlas.

Bell: Yeah. In fact, the Atlas to me was just a spectacular machine. And I remember visiting Manchester in ’62 or so, and then I saw it being built at Ferranti, and you know, in a very casual way. At one of the conferences, one of the joint computer conferences, I went to a little evening seminar and somebody -- John Fotheringham I think was his name -- was speaking about it. And he was from Ferranti and I was so taken by this machine. I mean, this was just so wonderful hearing about the one level store that first time.

Hendrie: I remember -- it's sort of an aside -- but I remember being in the audience at a talk about the Atlas and you getting up and asking a bunch of questions during the question period about it. Of course, I was at Computer Control, but that's how I figured out what DEC was probably going to build next.

Bell: But anyway, yes, I was really impressed by it. Then I went to Manchester and I was watching these guys build it. And, you know, again, I'd love to have these memos. I sort of said, "Gee, if these guys can build a computer like that, I mean, this is crazy, you know, we can do anything we want. We should be able to do this." But, because they were just, you know, they were just very casual, "When are you going to get this thing wired?" "Well, you know, it'll maybe be six months before we can turn the power on." And so they weren't in a hurry or anything like I expected from a commercial company. But at that time, I met the main people at Manchester.

Hendrie: Was there anything of the concept of paging yet?

Bell: No, that was...

Hendrie: In the 6?

Bell: No. Right, paging was what we wanted. Clearly after anyone who had seen the Atlas at that time, you know, that's the way to do it. And I can't remember, I think it was a KI before we got paging in because it had to do with the cost and the speed of registers and of having a paging register set.

Hendrie: Yeah. Being able to implement it successfully and at a reasonable cost.

Bell: Yeah. Because you're really stuck with getting a good fast memory that didn't slow things down. And in fact, you know, Seymour Cray basically had always said, "I'll never have paging in a machine." And he never did.

Hendrie: Well, he was never interested in the concept of time sharing?

Bell: Right. There was that aspect. But just managing even the stuff that he was doing, you know, it would have been so helpful to do that to help the OS in the Crays. And instead, what they did was just moved data around. So, you know, memory was always being shuffled around. And in fact, that's got some of the attributes of paging there. You know I met him and I think we might have said something about that. I asked him about that. But I almost knew the answer of, "I'm not putting anything in. It slows the machine down." And that was his philosophy.

Hendrie: Are there any stories you can tell me about, you know, just anecdotes that you remember when you were either designing or sort of getting the prototype built or debugging it.

Bell: Yeah.

Hendrie: There usually are things that happen.

Bell: Right. There were a lot -- two that had to do with software. And in fact, you only get your good stories from the software. Hardware engineers are just dull comparatively, so, you know, they're just too focused on just getting the job done. But software, there is a lot there. You get more interesting things. But, anyway, there were two about the compiler. This one was, I think, at the time the company might have had a revenue of ten million dollars, and we had five or six hundred thousand dollars tied up in a PDP-6 to go to Brookhaven. And so, this is sort of the last month of the quarter or of the year, and of course, DEC had always been profitable from the beginning so they were depending on our shipment. This was another amazing story about DEC in that it was profitable right from the start. But the ship included the Fortran compiler. And so it got to a point where I would . . . actually, I guess there's another piece of code I actually worked on, and that was the back end of the assembler. So we had a very simple assembler for it. So the compiler would spit out instructions and then it had to be converted to numeric codes. But it didn't have macros and things like that. So instead of running it through the macro assembler, which in retrospect probably we should have done, we had built a very trivial back end. The guy writing it was named Harris Hyman. And he was a bald headed guy. He was a civil engineer. And I think he had also written the macro assembler that nobody liked. All the MIT guys were always pissing on our assembler because of Harris, so he was not a beloved programmer. And so I started working on it. And I said, "Harris, you realize the whole company needs your assembler. We've got to ship this computer and get all the revenue for this thing and it's going to cost the company, you know, a half million," some damn number or whatever it was. And so, Harris said, "Ahh, take it out of my pay."

And then the other story is about one of the people that worked on the Fortran. The Fortran compiler was actually done by Peter Sampson. And Peter had written the first syntax directed compiler. A guy by the name of Ned Irons at Yale had written a paper about building compilers, the syntax directed compilers. And you had basically tables that you looked stuff up in. And that was how you did the compilation. So Peter said, "Oh, good, that sounds like a good project. I'll just go off and implement that." And so, the compiler was done that way. But, of course, there was one instruction halfway through, one comment halfway through, on the whole thing. No one knew how it worked. And the story goes that Larry Portner and Bill Segal, who then took over the software, had to climb in Peter's apartment and steal the source code because Peter was fairly mysterious about when he would appear. The complier had no comments, and we needed to get the original so we could actually start maintaining it. Because, you know, we haven't seen Peter and we've got a few problems. How do we get these problems resolved? And, you know, a syntax directed compiler had the properties that it would occasionally go into fairly hairy obscure loops. And one of the guys we hired to head marketing, and he had been a consultant. His name was Press Behn. And I don't remember where he'd come from, but he'd come from a consulting company and sort of said, "Gee, I am going to get into computing." And he was in charge of marketing. And then I can recall Larry Portner saying, "Yeah, Press knows we've got a syntax directed compiler. The problem is he doesn't know what a compiler is." So, you see, the software just has a character on its own.

Hendrie: It has a character of its own because of the characters that own it.

Bell: Right, yeah. Well, the other one was that I always say I had made the same discovery earlier than Cray did about parity. So we didn't have parity in the PDP-6 initially. I'm trying to remember whether those stacks were 19-bit stacks or not. I think they may have been. I'm pretty sure they were two 19-bit stacks of memory. But anyway, one of our first machines was at Adams Associates, which, by the way, they turned out to give back because it wasn't working well enough for them to go right to what they were going to do as a commercial service. They bought a big UNIVAC machine that had been out for a long time to do whatever they were doing, building a commercial time sharing system. And so I went into their office on the first floor of Tech Square -- and I think it was 1964 – and it was a good size machine with 4 16K word memories. It had I think four bays of memory, and then it had a lot of I/O. And so, I remember going in. I'd get the machine late at night and then kind of work all night on it and look for it to fail and see what was wrong, and putting in modifications and things like that. And then, looking down these bays of memory on the right and bays of I/O on my left, and blue bays this way and that way, and I said, "I have no idea which way to look when something fails. We've got to put parity in this machine." And so, it took us about a week to put parity in. And Cray had, I think, a similar effect, not on parity but error correction on the Cray 1's. And I don't know if he ever put parity in. I don't remember if it was in 7600 or not. But those are core memory machines. It wasn't in the 6600, but you know, it's one of those things of,"Oh, my God, you can't build a machine without parity." Cray is claimed to have said, "Parity is for farmers."

Hendrie: We just can't find it. We just can't find the problem.

Bell: You have no idea what's going on. You've got software, you've got hardware. In this we had I/O. And what is going on here?

Hendrie: You gave up.

Bell: Yeah. It just took me sort of one night with this machine to know that you have to have parity. I did learn fast.

Hendrie: That's great. Love it.

Bell: So no one...

Hendrie: All the theoretical reasons just disappear when you're up all night.

Bell: Yeah. All the probability, all the probability of a failure here.

Hendrie: Exactly. Well, that's a good story.

Bell: Yeah.

Hendrie: And so, did you ECO (Engineering Change Order) it into the design?

Bell: Oh, yeah, absolutely.

Hendrie: Did it ever come out of the 10 series?

Bell: No. No. You don't build a machine without parity. In fact, I don't remember when we got it in minicomputers. But, in fact, it took quite a bit longer to say, "Oh, yeah, we really have to have it." And so, you know, all the influence I had overall was generally don't build a machine without parity. You just can't maintain them. You just don't know what's happening.

Hendrie: Now, how successful was the machine? Were there a lot of people that bought the machine? Or do you remember how many?

Bell: I can't remember how many we made. I think it was in the order of twenty, but I'm not sure. I don't know whether it was that many or not. I think I wrote that down. I think that's in the DEC book. I know there were 40 PDP-1's, because 20 of them went to IT&T and 20 went to other people. But basically, I think we, you know, stopped selling them for a while, and it was we've really got to have a machine that we can make more reliably.

Let's talk about what it really did for the company or did to the company. First, you know, I looked at it as a failure because of the profitability of the thing. And we stopped making it early and all that. And basically, the other thing that was wrong with it was they were germanium transistors. And so the wiring was such that we really weren't able to cool it well enough to keep them...

Hendrie: Potential heat problems?

Bell: Keep it really reliable. And so, the KA [KA-10] was done with silicon transistors that were just coming in at the time. So it was a switch to silicon. But the big effect was that I said, "We can't afford to wire it because of the cost to debug it." You had this very long debug cycle when you're debugging wiring. So that meant cost in inventory.

Hendrie: When it's all hand wired, yeah.

Bell: And so, either you have to have a machine to check the wiring or you wire it automatically. And so I called up Gardner-Denver and got them in and said, “You know, we'd like to buy a machine.” And that was the beginning of the PDP-8. Because the main thing about the PDP-8 was it was mass produced using wire wrap. And that all came about because we couldn't wire PDP-6's.

Hendrie: All right. And so you decided you needed to have some automatic wiring technology?

Bell: Exactly.

Hendrie: And Gardner-Denver had those machines. They were using them. But I don't remember what they originally did with them.

Bell: I don't know what the cost was. And it was I think a machine that IBM had designed and UNIVAC was using them. But anyway, that was the genesis of, I'd say that was the real breakthrough for the minicomputer -- getting a wire-wrapped back panel so that you could really make them without hand wiring, fast and inexpensively, it was just what made it all go.

Hendrie: Yeah. Good.

Bell: So, I mean, yeah, that's...

Hendrie: Anything else about that story?

Bell: Yeah, I think that was it. I mean, this question of paging and stuff like that was knowing where you have to get to and then just waiting for the technology that's going to catch up. In fact, I was just involved in an e-mail flame, you know, e-mail going around now about Negroponte's $100 One Laptop Per Child computer. And I said, I don't think I invented it but it's, "No technology before its time." And you just can't do it. The world will not let you. You can influence it a little bit and provide a dream for market or something and somebody will try to match that, but as far as actually achieving it, the numbers have to do it. It's not any other way. You can't hurry evolutionary technology controlled by Moore's Law much faster than it already goes.

Hendrie: Exactly.

Bell: But anyway, it's one of those things.

Hendrie: Well, shall we move to the next? What did you do after you finished designing the PDP-6? You clearly were working on trying to get some of the early installations to work well.

Bell: And then, in fact, I went out to Rand. You know, the other thing is I couldn't get machine time, because you know, we were debugging machines. And so, at one point I went to Rand because we had delivered a machine there for their time sharing system they were building. And I had three weeks where I was there and I worked all night. I slept during the day, worked all night. So they would come in at 8 in the morning and start working and I would come in at 5 at night and so I worked 5 until 8, sort of 2 shifts, for 3 weeks.

Hendrie: This will cause burnout.

Bell: Yeah. And you know, I think I put in a minimum of one modification a day. And that was terminating resistors, looking at every signal in the whole thing, you know, that kind of thing. So it was kind of a thing that you need to do. And so, I caused literally a hundred modifications during that time.

Hendrie: But mostly just trying to get more reliable as opposed to not logically doing the right thing. Logic design was good and fine.

Bell: Right. It was all having to do with reliability, anything that would cause any failures.

Hendrie: Okay. Alright.

Bell: Yeah, and there were things like trying to increase the heat flow and stuff like that. So, you know, really understanding what was going on there in the machine.

Hendrie: Yeah, try to improve the cooling and all of those things.

Bell: But anyway . . .

Hendrie: The germanium transistors were against you.

Bell: Yeah.

Hendrie: Right from the start.

Bell: Absolutely.

Hendrie: So what do you do next?

Bell: Yeah. I think at that point in a way I didn't feel machines were challenging and I saw what DEC had to do, which was to make copies of their existing line of computers. And at this point I had discovered it's a software problem and that you've got to take advantage of that. And DEC should not be building any new architecture. It shouldn't be looking at architecture as a way to solve anything. So what that did was prompted me to sort of say, "Let me take a sabbatical." And what I wanted to do was go back to academia, and Ivan Sutherland and I talked about doing something together. And I wasn't that interested in graphics. And I think he was going to Utah. And so, I believe it was suggested I go to Carnegie and talk to them, because he was a Carnegie grad. You know, I ought to tell Ivan about that, too. I think that he probably was the one who said go there. And anyway, so I went. I had never met Perlis at the time, I guess. I hadn't met Perlis and Newell. I hadn't met anybody there.

Hendrie: Were Newell and Simon both there?

Gordon Bell: Newell and Simon, and Perlis were there.

Hendrie: All of them were there.

Bell: And in fact, the three of them had written a paper the summer I got there on "What is Computer Science," which is, you know, to me is a classic. It was a little paper for science about trying to define the scope of it and why call it computer science. You know, in a sense I think it was maybe the wrong thing as opposed to calling it computer engineering, because computer science has always taken a poke at…anything that calls itself a science has got a problem. But anyway, so I went for an interview there and was in the department. They offered me a job as associate professor. And Rod Williams was the head of EE. And then Perlis was heading the Computer Science Department. And then I started and said, "What I think I want to work on is computer-aided design." That's a great area. And besides . . .

Hendrie: There was very little of that at the time there.

Bell: Yeah. And I said, you know, "Computers should be designing these beasts because they can do a lot of things that we can do." And in fact, one of the things I did just before I left was I designed a little CAD package. In fact, we used it for one machine -- I think it was one of the 8's. And the program did all the drawings and made all the wirelist connections. So I had basically a preprocessor that you actually fed in the macro, and you know, you described the stuff, and then basically it sort of barfed out all of the drawings for the machine. So it would iterate. And so it was a very clever, you know, nice use of macros and connections and all kinds of things.

And then so I thought, well, I'll work on that at Carnegie. It'll be interesting to do something in that area. And so I kind of started for . . . I guess I was only doing that for literally hours. I went there to Carnegie and they had just gotten a 360/67 and the 67 wasn't doing time sharing yet. They had a funny batch system and they had a large core store of 8 Mbytes, an unheard of amount, and what you did for time sharing was everybody got a piece of core that you used to write your cards in. So there was a batch, a time sharing of key punches. So we all had our own little key punch. And then you'd submit jobs through the job queue. So I think I ran one job or something like that, and I said, "This is for the birds," you know. Maybe even before that, there was some key punching. And I had done key punching for a year in Australia and I said, "I'm never going to use another card in my life even though it is a virtual card. I don't like this. So I'll do something else." And then I got involved with Allen Newell, and then we wrote the book, Computer Structures. And that was sort of my main work. During this whole time, I consulted with DEC. So, it was '66 to '72 when I was at Carnegie, and then about '68, '69, DG had spun out, and then the 16-bit thread had emerged. And DEC had to ultimately switch from 12, a multiple of 6 to multiple of 8.

Hendrie: Yeah, exactly.

Bell: And I was involved in a little bit of that early. I got involved in what ultimately was the PDP-11. There was a guy, a project engineer on it, who said, "You know, we've got to build this machine."

Hendrie: Who was that? Do you remember?

Bell: Yeah, the guy's name was Cohen. He was, I think, a programmer and he was the project engineer. But by the way, I ought to go back one little step there. During this time, a 16-bit machine problem came up.

Hendrie: Now, are you consulting with DEC the whole time?

Bell: I consulted with DEC completely.

Hendrie: For the entire six years that you worked?

Bell: Yes, '66 to '72.

Hendrie: So you're never out of the loop or totally disconnected.

Bell: Right. I only had one other consulting thing that I did, and that was U.S. Steel. They wanted me to look at Burroughs. They had bought a Burroughs B-8500. So it's sitting off in Paoli or someplace like that. So I went over to look at it. And the deal was that the president of Burroughs wanted U.S. Steel to take it -- I guess they delivered the disks. You went in this room and there was just disks everywhere.

Hendrie: Yes, it's a farm, a giant farm.

Bell: And the deal was that the president of U.S. Steel was being talked to by Ray McDonald, the president of Burroughs. And the deal was they wanted them to buy this 8500 at a special discount because the engineers had gotten a new design. And they weren't going to really finish the 8500, but they were going to maintain the 8500 and then were going to build an 8502. And that's got some more ideas in it. And they could never get the 8500 running well enough to pass some of the tests. And so they wanted me to look at it. So, anyway, I looked at it. And there's a memoir I wish I'd had, but I wrote a report on it. I said, "Don't even think of buying this computer. First off, it doesn't work."

Hendrie: It isn't finished and it'll never be finished.

Bell: It's unlikely that it'll ever work, work reliably enough. And believe me, if it doesn't work in Paoli, it will not work in Pittsburgh. And you do not want this orphan because you can't do anything with the software question. And whether these guys go on and make another computer, that's their business. But given that you've decided not to build this computer, and that's not online, you can't accept this. And then, it was during this process that I met . . . I never got to the president of U.S. Steel, but I got to probably two vice presidents below him or something like that. You know, I think probably the highest was the CFO -- I don't know whether he even reported to him or not. But I had a hierarchy of vice presidents that were there that I talked with. And then, they were all, "Well, Mr. So-and-So is going to talk to Mr. McDonald about this thing." And I said, "Look, I mean, if you want me to talk to the guy," I said, "You cannot buy that goddamn machine. You cannot take this thing. This thing will take you under. You're not going to get anywhere with it." And so, I prevented . . .

Hendrie: A disaster at U.S. Steel.

Bell: A disaster, yea. But it was really funny, you know, that there are all these deals where us guys will get together. And so, this is why I have my contempt -- contempt is not the right word -- skepticism of top management and CEOs and getting together.

Hendrie: And deciding anything.

Bell: We'll get together on the golf course and decide this stuff. And you know, I always had similar kinds of reactions when I dealt with CDC. CDC was like that. We gentlemen will get together and make the decisions. And, you know, I went to one of their meetings once and it was Bob Price and oh, Bill Norris. I liked the guy. I actually liked these guys.

Hendrie: Bill Norris. You know, can we take a pause?

Bell: Yeah.

Hendrie: We've got to change the tape because I don't want to stop in the middle, just as you're starting another story.

Bell: CDC always was an interesting company. I had several interactions with them over the years. I was probably responsible for getting MCC started as much as Norris because we got together, had a big meeting, which did include the presidents of a lot of the companies. So we convinced them that we should try this consortium against the Japanese threat. Overall, I have never heard of whether anything good came of it.

Hendrie: Okay. Maybe you could just tell our viewers a little bit about what MCC was.

Bell: MCC was Microelectronics and Computer Consortium, which was an attempt to do pre-competitive research and it ended up being located in Austin. To my knowledge almost nothing came out of it. It tried to do a lot of CAD work. TI came in and wanted this. They were always big in CAD and they wanted to do that. But start-ups really had taken over the CAD area. I don't think you can do better than that. Doug Lenat's company, CYC, came out of there. Admiral Bobby Inman, who was head of the CIA was the first president.

Hendrie: Oh, yes.

Bell: He was, I think, the first director of it, but anyway, a really bright guy. But in a way, I guess maybe there was some good packaging that came out it, and I think companies did use that and it turned out the thing that was good about it was there was a way of getting people together and to discuss stuff in a kind of forum that was closer than forums that you get to, you know, conferencing kinds of things. It was a more intimate kind of thing and you could not collude -- I won't say it was even colluding or something -- but work on. Packaging was a good one because you were trying it -- "Well, you guys are going to use this or that" because you can't do everything. But anyway, I was instrumental in getting MCC started or rather to blame. In fact, that was right at the time that I left DEC. This was in 1983. So it was just getting started before I left. I think I probably would have killed, or I would like to think either made it work or killed it, if I hadn't left DEC. But anyway, that was one of the other interactions with Norris. And then we tried to do some architectural work around that, but then when Cray left CDC and I looked at what they were doing, I said I know what's happened there. Here's this renegade. CDC wouldn't have existed without Cray.

Hendrie: Right. Wouldn't have started without Cray?

Bell: Without him, and so here it is. I don't know whether Cray actually decided that. It was kind of a nice parting at least for Cray of saying, "God, I don't want to be part of these guys," and essentially leaving. I don't know. Actually, I think Norris actually put money in Cray too. Cray has clearly been a world resource, and so you get a company like CDC with a lot of engineers in it and they all want to be like Cray.

Hendrie: None of them are.

Bell: None of them are. And then when I was part of the National Science Foundation they asked me to give a talk on things. I walked into this computer and they said, “This is our brand new computer. It’s compatible with X or Y, one of the non-Cray, just a gray machine.” It’s a computer that wasn’t vector machines designed. CDC had never learned about software, never had the foggiest idea about software.

Hendrie: Yes, what could be done with it or what you could do with it.

Bell: Yes, and the fact that you have this pile of code and that represents a hell of a lot of money and so you don’t throw it away. If you’re going to do another one, you have to create another pile of code like that. And I looked at this machine and I said, “How fast?” And it turned out to be one plus MIPS. I said, “Why are you doing this? It’s an ECL machine. Do you realize that a MIPS chip is faster than the computer you just designed? One chip and it kills your machine.” By the way, it was the same story with Ken. There was an epiphany that Ken had at one point. It’s in the book by Ed Shein.

Hendrie: Which book?

Bell: Schein’s DEC is Dead: Long Live DEC book. Basically, he said you mean to say one chip is faster than the DEC 9,000 that we’ve spent $6 billion working on and has in two bays and has all this fancy technology? And yes. And so, that’s one of my reasons why they got in trouble - to let that happen. This was unconscionable on Ken’s part.

Hendrie: Other companies made the same error didn’t they.

Bell: In a way, I saw it at Cray too. Cray made the same damn mistake. Going off to Gallium Arsenide when in fact he should have been in the CMOS world.

Hendrie: Right.

Bell: But anyway, that’s a piece of technology that you have to see those curves coming and see who’s really going to head off somebody else, and if you can draw two or three points through each trajectory Y and it looks like it’s going to happen, get out of the way. There’s going to be an explosion. When I left DEC, or in fact by 1980, the data showed the story for the next decade. Even IBM probably made one too many ECL machines.

Hendrie: Get out of the way or on the train.

Bell: On the train. Then the other one at CDC was a similar case. This one going back to the management issue at CDC and I was at NSF. We had all these six computation centers. These were big com centers with supercomputers that were eating our budget like crazy. And before I’d gotten there, we had all these things at least on a temporary basis. So here I was trying to close centers and the guy working for me was opening . . . actually, I even think I may have fired him, but he left.

Hendrie: Who was that?

Bell: John Connelly. He had started all these supercomputer centers that we had to feed. So we had these things. We couldn't afford them all and the sixth one was at Princeton. They had two big CDC Cyber 205s machines based on the Star 100 and then their other company and I don't remember. . . oh, ETA was the spin-off from CDC that took them over. That was coming, and so I went to CDC. I went to ETA and I said, "What are you doing about software? Well, we want UNIX at NSF." They said, "No." And they had the most god-awful collection of software I'd ever seen, and it was because no one knew how to manage software or architect software and especially not do software. And so they had dialects and different languages, and we'd have to have all this garbage, run all this garbage. I said, "Throw all that crap out. We just want UNIX. You guys aren't getting us there." They said, "But we have all of our customers." I said, "You have no customers. I've talked to your customers. We've got some machines and there's DOE and guess what? I have talked to DOE and they are with me. I have their proxy and so get with the program. Get rid of those guys." And the guy running it was Thorndyke, a good hardware engineer, but knew nothing about software. Neil Lincoln was their CTO. I don't remember Thorndyke's first name.

Hendrie: Oh, yes. Was it Lloyd Thorndyke who ran it?

Bell: Yes.

Hendrie: Okay.

Bell: Good hardware guy. Didn't know anything about software, and they had an interesting technology to make a CMOS computer and to cool it with liquid Nitrogen and get another factor of 2, but the machine was flawed fundamentally because of the streaming from memory architecture. However they had a ridiculous acceptance test that they could never pass based on its terrible scalar performance. So I said you made this ridiculous test in the contract. You can never pass this test because the way the benchmark works, you'll fail all of the stuff. But anyway, then Princeton had the two machines. They were late in getting the ETA machine. They wanted us to buy the ETA machine. So Princeton's pounding on my table, "We have to bail them out. We have to pay for this." I said, "We're not going to do it." And so Erich Bloch, the famous Erich Bloch who was responsible for IBM manufacturing the 360, agreed. Erich was my boss and who's probably the world's greatest technical manager. Talk about a manager. He is one hell of a manager. Anyway, so he just sort of laughed at this whole situation and so I just held my ground. He got the call from Bob Price and Price said, "Erich, we men have got to get together and decide these things. We're going to have to fold ETA if we don't get this cash for our non working machine. We'll bring our folks and come and see you." And Erich says, "Well, I don't know if there's anything to talk about, but sure, come ahead." So at one point, Erich and I and Bob went off into the corner and we basically said, "No, we're not going to buy your machine. It doesn't work." This is not a grant. You agreed to sell us a working computer.

Hendrie: Yes. This is not a grant. This is not a grant to do experimentation.

Bell: Yes, and we're just not going to buy it. We're not going to bail you out and we're not going to get anything. That was fortunately the undoing of the Von Neumann Center, which shouldn't have been there in the first place. Ultimately, we got down to two centers and I think its now back up to three, but there's so much pressure to put lots of money into more computers.

Hendrie: Yes, and have these big super computer centers. Okay. Hold on. I'm going to just turn off for a second.

Bell: Yes. My last story about Price and the CDC folks. I mean, we did the MCC thing and I happened to be down there for a meeting. At one point, we wanted to get involved in computer-aided instruction I think on PDP-11s or something like that. Bless his heart . . .

Hendrie: This was the Plato System?

Bell: Yes, Plato. He had put all this money in Plato. He had a vision of what it was going to be, that is, the computer in Kansas City or in the cornfield in Illinois and everybody would be served by it. But the economics of course just weren't there -- telephone, all these things and the terminal, which is actually elegant but in fact not fundable. I mean schools couldn't afford it. So we went there one time and we wanted to put out some software. We wanted to use their language or content because DEC was in the education market. So I came with the team and they sort of pulled me aside and they said, "Okay, well the technical team is really talking about the issues and whether it can work and details like that. Let's us go over and decide." I said, "Hell, I don't know anything about this. Do you know anything about this?" They said, "No." I said, "Why don't we join the technical team and find out what the issues are and see if this is practical as we're designing something."

Hendrie: Oh, my goodness.

Bell: To me, it was what I thought of the classic CDC and how I think of, gee, we wanted to separate the business issues from whether we can do it or even should do it.

Hendrie: Whether it works.

Bell: Whether it works or something like that needs to be redone.

Hendrie: Wow. Alright.

Bell: Actually, I don't remember how we got here, but I want to go back to Carnegie.

Hendrie: Good. That is where we're going back to sooner or later.

Bell: Okay. So one of the things that happened, this is pre-PDP-11, but it's also very important relative to switching to 16-bit. The 16-bit switch was due to Ed de Castro, Henry Burkhardt, and the cast that were leaving. I was at Carnegie and I don't know what year it is. I hope I've got the specs somewhere, but there was a machine called the PDP-X.

Hendrie: Yes. This is '68 when they left.

Bell: Okay, and so Ed and Henry came to see me at Carnegie Tech. I had the specs and Ed and Henry came and we talked about it. Basically, they wanted me to sort of approve it. I said, "I think we've got to do it. I think it's exactly the right thing to do, and we should move ahead as quickly as possible." And that was my ruling on the thing and it got turned into a political thing. It turned out that Ed had been put under Stan Olsen. Or, no, I think even worse, Ed had been put under a guy who was put under Stan Olsen. So as an engineer, Ed was a very successful entrepreneur-driving guy. In fact, he had wanted to be a business guy because he had been at Harvard. I don't think he got his MBA there, but he had been around there. So he had that kind of interest in business. So as a project leader then, here's a guy put under another guy that he had no respect for.

Hendrie: Yes, there's the bottom line.

Bell: There's the bottom line on personnel. If you want to get rid of somebody capable that you may not care about just move them to someone they cannot respect.

Hendrie: That's a good way to do it.

Bell: That's a perfect way to do it. You couldn't ask for a better way. So he was put under John Jones. .But anyway, Jones was a student of General Doriot's. So John had been put under Stan with the sort of group vice president there at the time and Jones had done a good business thing, but it was sort of a marketing/selling into high energy physics and we already kind of had that market. So he didn't really create anything. So anyway, that prompted their leaving and forming DG, and then that created the necessity to have a 16-bit machine to compete with these upstarts that just started their company. And then it became sort of super high priority at a time when I think it was right after their announcement or something like that. I don't remember when it happened, but certainly it was a catastrophe within DEC when they announced. And so this created the need for PDP-11.

Hendrie: Was there any sort of like getting even with them for leaving? I mean was there any sort of anger, or was this a carefully thought, rational decision that there were already 16-bit machines there? I mean was he out of line?

Bell: No. I wasn't there. I wasn't inside at the time to get the feeling about it. I mean there certainly was this enormous hostility with them leaving like crazy. I mean this was a terrible thing. What can we do? How do we sue them or how do we do this or that? They've stolen our ideas, all this shit. And then I don't even recall whether I was asked are they building the PDP-X? I said, "No, nothing like the PDP-X." But anyway, I think my reaction was always the pressure of well, I'm sorry. We're going to have to make this big change and do it now or it's going to be more expensive later. And so DEC had to do it. There was no question about that.

Hendrie: It was the way the world was going.

Bell: Yes, and I don't remember when he went up there to DEC, but a student of mine, Harold McFarland, was the carrier of the ideas. There were two -- the Unibus and the general registers. So Harold was a designer, one of these students that in fact sit in class and design computers. So he was one of those and I worked with him on the design on it. Then Harold got into the group that was doing what originally was the DCM, Desk Calculator Machine. It was a code name for our 16-bit machine and I think Cohen had asked me, "Well how do you design a machine?" I said, "Well first off, you get some benchmarks," and so I think I gave him a bunch of benchmarks. It turns out that the machine was designed well. It will do those benchmarks, but you couldn't compile it in or do other things. It wasn't elegant at all -- not a nice machine. I think Cohen was having trouble with the hardware, so Roger Cady took over the project.

Hendrie: What had Roger done before?

Bell: I don't remember what his background had been.

Hendrie: But he had been at DEC for a while.

Bell: He'd been at DEC for a while and so he had credibility to get projects done. Cohen, on the other hand, had no credibility and wasn't able to do it. He'd been a software guy. So he really couldn't bring this project off. And then Jim O'Laughlin was working on the machine too, I think, for Roger. Yes, in fact, Jim was the logic designer. Roger was the engineering manager at this point. They came to Carnegie one Friday evening for the weekend to have a design review and get my okay. Nick Mazaresse was the VP for small machines, and we were asked to look at their design by Nick.

Hendrie: He's still doing that.

Bell: So they came and I said, "Okay. Let's spend the weekend looking at this computer." So I got Bill Wulf who had been doing some consulting for DEC in the compiler area and he was on the faculty and we got there and we sort of looked at the machine. We said, "This is not very good." And so, Harold pulls out this idea for the PDP-11. He said, "I've been thinking about this. What we really want is something along the lines of what we've been talking about. Here, I've got this thing." He started sketching it and everybody came around to this thing. And so I had to report in on Sunday nights to Nick. So I called Nick and he said, "How's it going?" I said, "Absolutely great." "Well do you like the machine?" I said, "Well, we've made a few changes to it."

Hendrie: I'm waiting for you to admit it -- we threw it in the wastebasket and we've started over.

Bell: We started over. Meanwhile, the Unibus part was okay.

Hendrie: They did have the Unibus part in there.

Bell: Yes. So that part was okay, but meanwhile, it turns out it was redesigned. So O'Laughlin had to do some very fancy redesign at that time to make it happen. So that was basically the story of the 11 architecture.

Hendrie: Now did O'Laughlin work at DEC?

Bell: Yes.

Hendrie: He was at DEC.

Bell: So here's the project. We just simply moved it . . .

Hendrie: Yes, but he'd been scribbling away all on his own, but he wasn't formally supposed to be doing this.

Bell: Oh, McFarland was part of that. He was the engineer working for O'Laughlin.

Hendrie: Okay.

Bell: But really an architect. He was playing the architect's role there. So he in fact was the architect of it.

Hendrie: Okay. Oh, that's a funny story.

Bell: Yes. It really is. So Nick's sort of pulling his hair out. I said, "Look, it'll be okay" and then Nick left. I don't remember when he left, but Andy Knowles, who worked for Nick and was a much more aggressive and fine marketer, came in and had the PDP-11 group. There was a time when I spent a lot of weekends there and that's when I met Andy. I know we talked about some of the details and how to market and what the features were and stuff like that. And then Andy did a beautiful job of marketing it on the basis of Unibus.

Hendrie: Unibus, yes.

Bell: Andy had these ads that had sort of asked "how much bus do you want?" He had the scissors so you cut this ribbon cable.

Hendrie: Yes. Very good.

Bell: And so it really slowed DG down at that point. And from then on, it was sort of good design with being very aggressive on the 11/45 and using MOS and bipolar. In fact, the bipolar I think probably set it up another level, and so the 11/45 was the one where "Ah, this is really a much hotter machine than DG has." Then we went to VAX after that.

Hendrie: Yes. Okay.

Bell: We started that project in 1975. It was on April 1 as I recall when we started it.

Hendrie: Really? Okay.

Bell: It came out in late '77.

Hendrie: Well let's roll back and do some more chronological work, shall we?

Bell: Sure.

Hendrie: Go back to Carnegie-Mellon. So you spent a lot of time working on your book there. You said you did some consulting. Did you do any teaching?

Bell: Oh, yes, I had the usual teaching load.

Hendrie: What courses did you taught?

Bell: In a way, I didn't do very much in debugging the book in terms of teaching, although there was some of that. But in a way, I may have missed the opportunity to do that. I was writing the book kind of in parallel, but using a lot of the examples that I had in the book to sort of crystallize stuff.

Hendrie: Crystallize, yes.

Bell: But no. When I first got there, I taught an EE course, the freshmen/junior EE and taught a year of that.

Hendrie: What was the course about? What was the name of it?

Bell: It was just Introduction to Electronics or Electricity.

Hendrie: Oh. It was fundamental . . .

Bell: It was a first year . . .

Hendrie: A first year electronics course for EE.

Bell: Yes. Electrical Engineering.

Hendrie: Electrical Engineering course.

Bell: Yes. Circuit theory, and Carnegie said it had a particular bent that it liked to do, which is to give the old form problems and then people have to come up with the best solution they can in that domain. And so the labs were very tricky labs. There was more than one answer. And so that was just getting me back - not getting me back because I hadn't really. . . I had taught a course in Australia, a graduate course in digital systems design, logic design.

Hendrie: Yes, logic and systems design course.

Bell: And so this was really back to basics. And then for the computer science part, I taught a lecture hall. I gave 200 students lectures in these big lectures halls on programming.

Hendrie: So it was about programming.

Bell: Yes, right. In that case, I said it had been an ALGOL place.

Hendrie: An ALGOL based programming.

Bell: Yes, and so I switched to FORTRAN.

Hendrie: Okay.

Bell: In retrospect, I'm not sure that was a good idea.

Hendrie: Well, it was certainly far better practical training.

Bell: Yes.

Hendrie: You could get a job with what they learned.

Bell: My rationalization was that as opposed to teaching . . . I might actually now have said, "Oh, we'll stay with ALGOL and then convert that to FORTRAN at the end" because so much of FORTRAN is the mechanics of how do you get output.

Hendrie: So these were not necessarily courses to teach people. These were introductory programming courses as opposed to a course to teach the theory of compilers and the theory of operating systems to people who are then going to go on and either teach it or invent the next generation.

Bell: Yes, and then I started the first two-semester -- and I don't think it was two semesters, I think it was one -- graduate course in computer system design and architecture. I think it was a single course because the graduate school was just starting. In fact, Carnegie didn't have undergraduate in computer science for a long time.

Hendrie: Oh, really. No computing CS undergraduate.

Bell: Right. It was all done in math and then the Computer Science Department would teach programming courses and the like. So we had a core. So this got put in the graduate computer science core. And there were like, I think, six courses. So as a core course, you had to pass that test, my test. There was kind of languages, theory, I don't know, AI. I don't remember what the six were, but the interesting thing was at the time . . . that would have been '67 probably when that all got straightened out.

Hendrie: Yes. You probably didn't do it the first year.

Bell: Yes, I might have. It might have been that summer, that spring of '67. But anyway, I think we were the first to have hardware. We might have even just called it hardware as part of the core curriculum for a Ph.D. and we had no Masters program. This was a Ph.D. program. And that was unheard of at that time because during that whole period I was on a number of electrical engineering committees where we defined all of the computer hardware, switching circuits, logic design - those kinds of courses. I think there were four of them blessed by an EE group called COSINE reports, but it was computers and electrical engineering. So these were some of the things that went on during that period. It was actually a very active period in terms of determining the curricula for EEs. Computer science, in a sense, was only started in ‘65-’66 as a word and Carnegie was right up there at the beginning. So it was really novel to have this graduate course, core graduate course, that you had to take. I like to look at it now as that was what had a great impact at [Xerox] PARC because the bright guys at PARC all were from Carnegie.

Hendrie: Is that right? I didn’t know that.

Bell: Jim Mitchell, who got his Ph.D. there. He was in my class. Ed McCreight who was at PARC. Dick Shoup who did the paint machine was in my class. And now, I work for Roy Levin who was also in the class and heads Microsoft’s Silicon Valley research lab and my boss. So you had all these guys.

Hendrie: I was going to ask you whether you had any particular students that you sort of remembered, that were your favorites. Did you do any thesis? You were the thesis advisor on some?

Bell: I was the thesis advisor to . . . the first one was Fred Haney. I don’t know if you know him. He was a Cal Tech grad and he is a venture capitalist in Southern California.

Hendrie: The name sounds familiar.

Bell: Bill Strecker. Bill was an early thesis student and the VAX Architect. Dileep Bhandakdar, who is one of the architects at Intel and now Microsoft. Dick Shoup. Anyway, Dick’s thesis was really interesting. In a way, you get some students and you’re their advisor but when you’re watching them do it you’re sort of saying, “Well, you’re going to get a thesis out of this. This is really interesting stuff, but it has absolutely no practical value at all.” I can’t relate to it. Well, it turns out that’s a wonderful example because guess what? He got his Ph.D. in 70 plus or minus a couple of years. And then low and behold in, I think, ‘92 or so, a guy I had met at a NATO seminar summer school I had taught with Gene Amdahl and John Grey from Scotland… but anyway, he contacted me and said, “Hey, I’ve started a company and by the way, I’m building the chip that I think you know about, which is . . .we ended up making exactly the chip that Dick Shoup designed in his cellular logic thesis.”

Hendrie: Oh, my goodness.

Bell: It’s exactly that.

Hendrie: Oh, wow.

Bell: It’s got a crossbar and it’s exactly that.

Hendrie: Oh, that’s funny.

Bell: Isn’t that funny?

Hendrie: That is. It's just like, what, 20 years later.

Bell: Yes. Exactly.

Hendrie: You say you have to wait to wait for technology to catch up.

Bell: Exactly, and in a way, I can't see more than ten years in technology. So to me, it was like never.

Hendrie: Yes, it was a real practical interest because you couldn't see it.

Bell: So I talked with him about well, yes, if you did it this way, you can implement this or that. So I played the game too. There were other people at the time I think talking about cellular logic and stuff like that. But in fact, I had trouble identifying with it because they were so far out and I couldn't imagine it.

Hendrie: Because you couldn't see how to build anything with it.

Bell: Because I couldn't see it.

Hendrie: That's good. I think we need to change the tape.

Bell: Yes.

Bell: Yeah, actually I really did enjoy teaching and the interaction with all the thesis students. I probably should have tried to learn more about things I didn't really understand that well. I mean, I could have gone that way, but I was really wanting to get the book done. I had three or four years, five years, to get tenure. The book came out in '71 and was a classic. But it took a good four or five years to get it done. And so it really went to sort of the top of the charts immediately, and so that was able to solve my tenure and promotion to Professor problem. I got promoted at that.

Hendrie: Is that right?

Bell: Yeah.

Hendrie: Because it was such a great book, as opposed to just writing a paper here and a paper there.

Bell: Right. Yeah. I mean, I wrote a few papers, too. But it is one of these bet it all on this, and it's either going to work or not.

Hendrie: Well, it was the subject that really interested you.

Bell: Oh, yeah. And so to me that was a natural thing to write about. And so in a way it was all I knew about the subject at that time. And in a way that's why I'm so adamant about the taxonomy at the museum, because I have thought about it so damn much. And now the problem is I can't see any alternative to it. I mean, it's like everything goes through this filter. It all fits in the structure.

Hendrie: When you were writing the book, you potentially had the opportunity to look at a lot of machines.

Bell: Oh, yeah.

Hendrie: That you really didn't know that much about.

Bell: Yeah. I looked at everything I could at the time that was interesting, or not even not interesting. I looked at everything in terms of all the research. I really knew the areas cold at that point. As I was leaving . . . and in a way I didn't decide to leave because I was going to do a sabbatical. I had it all arranged to go to Australia for a sabbatical in '73. And so at that point, I guess Win Hindle said, "Why don't you come back and run engineering? We just need somebody to take care of all the engineering." And so I basically. . . well, I'm packed anyway. I might as well go to Maynard.

Hendrie: Did you equivocate? Did you think any negative thoughts about doing that? Or you basically said, "Alright, that's what I'll probably do now."

Bell: I don't know. I can't remember, but I should somewhere have a sheet of paper that has this versus that, because I make those decision tables normally for something like this. That's how I normally decide things.

Hendrie: That would be a fascinating piece of paper if you could find it.

Bell: If I could find it.

Hendrie: Well, maybe you've scanned it and didn't know you've scanned it.

Bell: Yeah, that's possible. The problem is will I ever be able to find that piece of paper.

Hendrie: We'll Google it someday.

Bell: Can the computer find it? But anyway, so I came up and Ken said, "Here, let me show you where DEC is today." So we went out in his plane and flew around New England and up to Maine.

Hendrie: Oh, really.

Bell: Yeah.

Hendrie: To all the different plants.

Bell: Yeah. And he said, "Why don't you come back and run engineering?"

Hendrie: Okay. So obviously Win and he had talked.

Bell: Yeah.

Hendrie: Win carried the message.

Bell: Yeah.

Hendrie: But it was Ken's idea.

Bell: So I decided that that was probably what I would do, because the other thing, I mean, there were two things that were getting me at Carnegie. I was at Carnegie at the optimal time, because we had a big doctorate grant and at that time I was writing proposals and I was getting money from NSF and so I was succeeding in the academic game.

Hendrie: You figured it out.

Bell: I figured it out. But I didn't like it. I don't like writing proposals, you know. I mean, no, I didn't mind the proposals. It was the damn process, the reviews and all of the stuff.

Hendrie: You'd rather build something?

Bell: Yeah. And getting all the reviewers and dealing with all the bullshit, it was so much. "Oh, I don't really want to do this". And so I was prompted at that level. The other thing was the gnawing thing of. . . actually the thing I do have is my three axis diagram of why -- there's power, freedom, and money. And you can't have them all at the same time. And when I got to Carnegie, there was somewhat of a slight tradeoff between money -- slight tradeoff, but not that bad -- money and power and freedom. And so I had a lot of freedom, no power, or leverage, any of those words. But, anyway, so that was gnawing at me. The other thing that was gnawing at me was DEC has got to get into semiconductors, that it can't put chips on a board and be successful.

Hendrie: Yeah. You could see integration.

Bell: You could see that problem of what's it going to do.

Hendrie: Well the first microprocessors. This is '72.

Bell: Yeah. Intel. . .

Hendrie: That would be [Intel] 4004.

Bell: I don't remember if it was or not. But the first trip I made was to Intel and I was trying to get them to adopt the PDP-8. So there was a kind of group forming, a nucleus forming at DEC, and what are we going to do?

Hendrie: And, of course, you're consulting with them.

Bell: Yeah.

Hendrie: So you're tied in at this level.

Bell: Yeah. And what are we going to do in this area. So there was a thing of seeing the change, seeing that something big is going to happen. This is a big change. In one respect I can look about and say, my god, by getting out of this boring period from '66 to '72, nothing happened. More of the same.

Hendrie: Nothing happened. Just more of the same.

Bell: Yeah, right. But it was just better packaging, faster gates, faster circuitry. But nothing new to build from. So anyway, that was the thing to sort of say, yeah, I've got to come back and that's important to do.

Hendrie: Okay. Before we get into that whole phase, can we roll back and talk about your insights or what you were thinking when Harlan Anderson, when Andy left DEC, sort of that crises that occurred there?

Bell: Yeah. Have you talked to Andy or did you talk to Ken at all about that?

Hendrie: Ken wouldn't. He did not feel that it was appropriate.

Bell: No. That's right, because he didn't want to talk about it or anything.

Hendrie: He just wouldn't want to talk about it.

Bell: Yeah.

Hendrie: I do want to interview Andy at some point.

Bell: Okay.

Hendrie: Because I think it'd be good to get his perspective at DEC in the early days.

Bell: He's important. And also Stan Olsen.

Hendrie: Yeah? Okay.

Bell: Yeah. At the time, Andy was sort of nominally off the project, sort of overall head. Ken wanted Andy to take over the PDP-6 and run it. I wouldn't say head, because it wasn't that way -- Andy was sort of promoting the 6. He was kind of nominally dealing, you know, running interference for it. And then I think there was one period where he was talking to Ken about it, and said, "We have to do something about it, we have to decide whether we're going to make more of them or not." And there was a crisis there where Ken said, "Okay. We've got to organize. Andy, you cannot be a free agent in this company. A vice president is got to be in charge of something. You can't be just a guy not doing anything." This was actually a very important part of DEC when the product lines were established and all the books were rolled up around the product lines. So Ken was able to delegate the company in almost a divisional fashion.

Ironically, DEC's demise came when the company was dismantled and the profit responsibility was diffused.

Hendrie: Who was in charge of all sorts of things, but not in charge, not directly responsible.

Bell: Right. Responsibility.

Hendrie: Not where the buck stops.

Bell: Right, exactly. And so at that point in time, it was the quote. There was a famous Ken talk that was given for many years, and that was the product lines. It was when Andy was foot loose and fancy free, telling everybody else what to do, and had no responsibility. And then I assigned everybody a product line and product line responsibility and financial responsibility. Then that was the turning point in DEC.

Hendrie: Okay.

Bell: There was a PDP-1 or 4, the 18-bit group. There was the 12-bit group, special systems, modules, and so on.

Hendrie: 36-bit group.

Bell: And then there were the hardware product lines -- they were purely that. And then there were people that were kind of selling them into verticals, in a way a thing that DEC maintained for a long time. I think until I left it was pretty much that you had OEMs, and then you had verticals that were education, lab, business, Telco, and so on. So anyway, the crisis really had to do with kind of giving and taking responsibility to give them a financial responsibility. And by the way, that principle I attribute to the success of DEC when Ken did it. In a way, when I came back, I destroyed some of that -- kind of but not really. First off, these groups didn't have the manufacturing anyway. They got stuff there centrally. And so I was the remaining one of two. Well, now I've done it. I've taken away their engineering, because their engineers report to me and then they can concentrate on it. Now there are so many verticals and stuff like that. So the marketing guys could concentrate on the outside thing, and then people didn't have to concentrate on hardware or basic software, because I had all the hardware software resources.

Hendrie: When the reorganization occurred, is that when Harlan left?

Bell: Yes and it was the big change.

Hendrie: When that happened, did everybody then have their own engineering and everything?

Bell: In fact, I say one of the arguments for me coming back was that they all had their own engineering and were getting a huge number of special operating systems and eventually would have gotten more computers. And, in fact, that was one of their rights. And so, of course, it didn't become destructive. It took, I'd say, until I got back. Even when I left, there were areas -- this was in the early '80s -- of Ken wanting me to take all the remaining engineering in every group and have them report to me. And I said, "No. Like how did we in one year design 35 money losing, dumb terminals?" I said, "Well, they were done for marketing, you know. These guys thought they had to do marketing of the thing." And so I said I don't know whether some of them are valid or not. But clearly, I've never made anything by spending my life killing bad ideas. I mean, I'll kill all the bad ideas in my own shop, but I don't really want to go on a killing rampage. There's so much bad out there, that it will take me forever to get rid of all this crap. And so let it be because they think they knew that a typesetting terminal or this terminal or that terminal, or something for the lab, that's some identity. But anyway, when I left the engineering was assigned to particular groups and there wasn't a, quote, central engineering. The proposal to centralize engineering was one I made when I came back from Carnegie, because when I came back, even though people knew me, the company had grown quite a lot. And it was going like gangbusters, and the question is who's this guy coming in and now going to tell us what to do. So Ken assigned me two engineering groups initially -- power supplies and memories-- because those were the parts that were centralized or that could be centralized. Also, I knew nothing about them. It made me humble.

Hendrie: Yes. That obviously has very little customer interaction.

Bell: Right. And these were skills.

Hendrie: Input.

Bell: High tech skills.

Hendrie: Yes. And it's bad news to have everybody doing their own memory.

Bell: Yeah.

Hendrie: Or power supply.

Bell: Yeah. So anyway the other thing that was happening is when you have all these damn groups, DEC was going down the rat hole of, "Oh, I'm going to have my own operating system." And Univac had done that. I watched CDC do it, in terms of, "Oh yeah, we've got an operating system for this market or for that market." So at that time, there were a bunch of RSXs, and RSTS, DOS was in there, and RT was in there.

Hendrie: I think there were five.

Bell: DOS.

Hendrie: At least operating systems on the 11.

Bell: Yeah.

Hendrie: I don't remember how many there were on the 8.

Bell: Yeah. Well, the 8 never really caught that fever. There was almost just one, because the 8 was so constrained. OS8 was really important as the forerunner for CPM.

Hendrie: So simple.

Bell: And so constrained and you basically assembled the system to do what you want. You didn't have an operating system per se. You had a bunch of code that you set.

Hendrie: Code, the drivers, and things like that.

Bell: Then you assembled it. You loaded in to make it work.

Hendrie: Okay.

Bell: But anyway, so that was the situation with Andy. And what happened with Harlan Anderson. That was the blowup of his leaving, which is in '66, I guess. And having to do with this issue of responsibility and control. And then the product line and the P&L focus, I think was the main principle. And that principle existed until after I left, until slightly after I left. But when it was broken, I think that was the beginning of the demise of DEC, because we were straining under it all the time. Or not straining -- I mean, you could argue that, well, I can put that marketing in a cost center and then where do you want the profit rolled up?

Hendrie: Yes.

Bell: It's that. Okay. Do you want to roll profit up by systems and create, and these marketing guys are just overhead to the systems. Do you want it in the field where they get the profitability? And so there were like three places that it could be. And from what I can tell, it was the CFO, you know, with the two Jacks (Shields and Smith), who decided we are going to diffuse it all and we're going to roll it up in three places. And we're going to make everybody responsible for profitability. We're going to account it this way, we're going to account it that way, and we're going to account it three different ways.

Hendrie: Okay. And you have a feeling that . . .

Bell: I think that was DEC's demise because of the complexity or the diffusion of responsibility there that contributed to it. Now, I don't know the details of that. Ted Johnson, by the way, was the one who told me about this, that this wasn't under the cover because it was done without Ken's knowledge. "By god, we're going to run this company independent of how Ken wants to run it." And I'm not sure how Ken saw it at that point, anyway. In a way, when I left, Ken didn't have anyone to argue with. I mean, in a funny way, I wasn't there to do that.

Hendrie: People wouldn't argue with him.

Bell: Yeah, right. I mean, it was kind of how do you understand it. And then the fatal thing was actually putting marketing under Shields, who had the sales and support service, whatever you call it, putting those people together and then treating marketing as an expense, as a cost center and not a group that's got to go out and get the revenue and be responsible for the revenue. Marketing had been critical because it got the application software. However, the critical issue was the accounting for profit in a single place.

Hendrie: Yeah.

Bell: Because that's really where the company accounted for it.

Hendrie: That's where it used to be.

Bell: And to me there were two big reasons about DEC's demise. And that was one, and I think just all the engineering and the lack of direction and focus and understanding about products and engineering was the other reason.

Hendrie: Alright.

Bell: We're way out.

Hendrie: No. That's alright.

Bell: Let's go back.

Hendrie: Okay. So now you're coming back to work at DEC in 1972. You decide you're going to accept this, to come back. What did you do? Tell me about a little chronology of what happens when you get back there. Who do you have working for you when you get back there?

Bell: I had two groups -- power supply and the memory group. And then, of course, Ken thought he knew more about memory design than anybody, and power supplies, because he knew more about those than anything else. He always would harass me about, "Oh, the problem with our power supply guys is they don't know Maxwell's equations" or something like that. And I said, "Well, Ken, I think they're actually pretty good. I mean, they are doing very well at it."

Hendrie: Did they work?

Bell: Yeah. And, you know, we're making innovations. We were probably one of the first to switch to high frequency, high switching power supplies.

Hendrie: High frequency switchers, okay. Certainly innovative.

Bell: So I pushed them pretty hard. What are we going to do on cost per watt and power dissipation and all that stuff. And I think it was at one point I came back to the group. And this was probably maybe late '70s. Well, I think we've gotten Ken out of the power supply. He's now back in the wall plug. He's working on the power cable. This was one day when we had about three or four of the engineers that, you know, knew Maxwell's equations very well. And we're in talking to them, and it's at that point I think he decided these guys aren't so dumb after all. "I don't think I can contribute very much to the power supply."

Hendrie: I'll go find another place.

Bell: "I'll put my energy in the cable." And I was so proud of that day.

Hendrie: That's great.

Bell: He's back in wall plugs. It was funny, though. Yeah, so I did that. I don't remember what other power I had. I think I may have chaired the engineering committee or something like that. There was some other leverage to start to work on the architecture in the software area and getting involved in some of these big issues of how many operating systems we're going to have and what they're going to be, and sort of the overall software direction. So I got involved in that. I got involved in the strategy group or this or that. In '74, in February or March of '74, I proposed a Central Engineering. Oh, well, the other thing is I had all the power because I was on the operations committee.

Hendrie: You're the only person from engineering on the operations committee.

Bell: Right. So I had a title of VP of Engineering or R&D as our research group also reported to me. Then, I had only these guys reporting to me. And then I was looking over things that were going on in various projects. Andy had the PDP-11 engineers. John Leng had the 10, and that remained that way. I don't know. It wasn't for a long time. I ended up with those guys perhaps. I don't remember.

Hendrie: There's still a PDP-7/9 group somewhere.

Bell: Yeah, the 18-bit stuff. The 18-bit thing terminated with the decision to do VAX, which was a '75 decision.

Hendrie: Okay. Yeah.

Bell: Anyway, I proposed that all the engineers report to me in February of '74, at a Woods meeting in Bermuda.

Hendrie: Okay. A Woods meeting? What's that?

Bell: Woods meetings were the famous quarterly offsite meetings that we had.

Hendrie: And they were called Woods because they met in some woods at one time.

Bell: Yeah. Well, typically it would be in the woods. We'd go off someplace. I think a Woods meeting meant it was a two-day meeting, and I think maybe it was once a quarter as opposed to monthly. And so it was at that meeting in Bermuda that I basically wrote a proposal and probably undoubtedly Ken had said, "Propose that all the engineers report to you. You've got my vote, now you have to convince these other guys."

Hendrie: Okay. Alright. That's typically how it would work.

Bell: Yeah. And so basically I went there and it was typical of Woods meeting in Bermuda -- and we went to probably three of these in Bermuda – it was February and no one was there. Cold as hell, and it very often rained. The golfers would come down with their golf carts and maybe have one round and get rained out. And then so we all sat around in this cold dark hotel room and discussed some topics. And so anyway, the topic there was running engineering. And so I outlined how.

Hendrie: What you proposed.

Bell: What I'd do with that. There were other things going on that wouldn't necessitate this, because there were disks and all these other areas that were just sort of dangling on parts of the organization. So you had these more or less market basing organizations. Then you had these little appendages. "Where does Bob Puffer report?" "Oh, he reports to the guy making these computers, or this or that. He's doing this."

Hendrie: Yeah, where? Okay.

Bell: So that was another reason of putting it all together. So essentially I cleaned up the org chart to say that's all engineering. Now we've got a functional organization that parallels the manufacturer, pretty much treated like the manufacturing organization. And then you've got these more pure marketing organizations that go off and do things. And so the issue was you worked to get convergence between what I'm building and what everybody else wants, as opposed to when Andy Knowles had all the product line or PDP-11 engineers. Somebody wanted something in it. He said, "I'm not building that because I'm the PDP-11 guy. And it doesn't fit my market." Well, he doesn't have a market. He's the OEM.

Hendrie: Yes.

Bell: And so this was the way it was broken. Then in effect I had to get convergence about specs and agreement about what we're building, because that was part of the decision process. And so we had various groups that would look at the Product Line strategy. Is the strategy right and stuff like that.

Hendrie: Yeah. What's the next PDP like?

Bell: Yeah. And then in that whole process, the product managers remained in engineering.

Hendrie: Once upon a time didn't marketing have them?

Bell: They would sit with the various people in marketing. They weren't that far away from engineering at that point. Well, the product managers weren't that strong at that point in time. But when I took the engineering over, why, the product managers were put there in engineering. And I had always observed with highly marketing organizations that they put the product specs in a group that's parallel to engineering. And that you had to go up the chain to get specs approved, because it's the marketing guys. In fact, I coined the words "he who proposes does." That was an operative DEC word meaning, you know, the engineering manager is going to make the proposal and he's going to do the work. And then the product managers, in effect, were the tools to let you define what it is you're going to do. So the product managers would worry about getting the specs and then that all got thought out and the specs were resolved at the engineering manager level. Yeah, they're part of the team building of the PDP-11/30 or something like that or whatever. And so that close coupling of product and product market I regarded was critical, unlike other companies. Microsoft has a lot of that too. I mean, you still have enormous conflicts, but the conflicts are all isolated within each of the groups. Nothing that's sort of going up and disturbing the organization of this guy over in that building or not building what I told him to build.

Hendrie: And getting somebody else involved through a chain of irrelevant and unknowledgeable people.

Bell: And somebody else. Yeah.

Hendrie: Involved at a level that doesn't understand.

Bell: Yeah, exactly.

Hendrie: To make the decision.

Bell: And so that was a big principle. And then at one point, Andy had been moved into, quote, corporate marketing, whatever that was. And he could never get it defined, by the way, and it drove him crazy. But he proposed it once. And I swear every four years there would be a proposal that the product managers are not going to report to Gordon any longer. And I would go into the Woods meeting and for a day be harangued by everybody. "No. They are going to report to me." And I would come out and win these guys every time.

Hendrie: Yep.

Bell: But that was a strain, you know. So there was all of this, which, in fact, at the time I didn't think was a problem. But then when I had a heart attack, I thought, you know, maybe there is something to this stress stuff that my body isn't as strong as I think it is.

Hendrie: Yeah, exactly.

Bell: Yeah. I talked to a doctor, a cardiologist at Stanford, and he said, "Oh, no. There's nothing to stress, you know. We can't tell, you know. These arteries clog and then we unclog them and bypass them, so it has nothing to do with your stress."

Hendrie: It has nothing to do with your stress.

Bell: Yeah.

Hendrie: It has to do with your genes and your diet.

Bell: Yeah, exactly. That's all.

Hendrie: Alright. Good. So now back up. So now engineering reports to you and that's '74.

Bell: Right.

Hendrie: And this is '74. Now what else is going on in engineering? Are they just each group sort of cranking away on the next 11 and the next?

Bell: Yeah. The group's cranking away, but then in our deep bowels we knew that the 11 was dead.

Hendrie: Had the 45, for instance.

Bell: The 45 was out at that point.

Hendrie: Okay.

Bell: Or I think it was, yeah.

Hendrie: Somebody had come up with that idea.

Bell: Yeah. And so we said, wait a minute. So it was clear there was an undercurrent, the 11 is out of gas. We're just dying with the address space problem. Strecker and I wrote a paper about the problem.

Hendrie: Well, you went and hired Strecker.

Bell: Yeah. Strecker came to DEC about the same time I came back.

Hendrie: Yeah, that you got there, that you came back.

Bell: Yeah. And the question was really going to be do we extend the PDP-10 or do we make a new upward compatible 11? So there were task forces, all kinds of bullshit going on, for a year. And that was probably going on for almost a full year. And we came out of that with a decision to do VAX.

Hendrie: Okay. Now this was all while you were there.

Bell: Yes in 1974 and early 1975.

Hendrie: Yes.

Bell: Yeah. I was in the midst of that. I was in it, watching it. In a way, it's hard to really remember that whole story, remember how I felt, whether I really felt we're going to decide this. Let the damn thing go and they'll come to the right decision. I mean, there was a little bit of that, of everybody's got to fart around and do all this stuff, but they're going to come out with this point instead of me being able to say, "This is it, we're going to do it that way." This was a case of letting a process go on to get buy in even though I knew what the answer had to be.

Hendrie: Yeah.

Bell: But what was good in this one is Andy took the lead position. Andy had no responsibility for the 11 at this time, because he was off doing terminals and doing that OEM stuff. So he was kind of neutral and he took kind of a position of pulling people together and saying, "Okay, let's make a decision. Let's get all the facts out." So they worked on facts and, you know, getting it all, writing white papers and black papers and all kinds of papers. The usual thing that you might do to analyze it anyway you can or every way you can. And then, out of that, was a recommendation that we do the VAX.

Hendrie: Okay.

Hendrie: The year of sort of studying what should we do.

Bell: Right.

Hendrie: Maybe we should do this. Maybe we should do that.

Bell: Extend the 10 or extend the 11. And that was the time we came out of that and I swear it was the first of April '75, but I can't really get it out of my schedules.

Hendrie: Your notebooks?

Bell: Well, not my notebooks. Mary Jane's schedules for me of when we first pulled the VAX team together, VAX A. The group came together and then we moved up -- this was in building 12, which is the original DEC building -- to the, I don't know, I can't remember if it was the third or fourth floor, the top floor to be clear. It was to the top floor of 12 and it had been vacant. It was also the floor that I first had my office in.

Hendrie: Is this the building that Ken had his office in down on the first floor?

Bell: Yeah.

Hendrie: Down in the lower . . .

Bell: That was the original one.

Hendrie: Yeah, that was the original DEC building.

Bell: And then he moved. There was an old new building next to 12 that was ultimately linked to it, but, in fact, his office for most of his time was in building 12.

Hendrie: Okay.

Bell: In fact, part of this whole thing was he was in that corner of the first floor of 12. I was in the other corner of 12 where he could watch what went on in engineering.

Hendrie: Okay.

Bell: So, in fact, I think he wanted tighter control of engineering.

Hendrie: He wanted to know what was going on.

Bell: And so . . .

Hendrie: He can walk over any time and talk to you.

Bell: Yeah, right. And there was a conference room that used to be his office that was where the engineering committee met and also the Operations Committee met there until he had moved. So basically it was fairly easy to know what was going on. And so, when I'd have people coming in for meetings why he'd come in and talk, you know, so he knew what was going on in terms of . . . this gave him that feeling.

Hendrie: That feeling, yes. Would he sit in on meetings?

Bell: Oh, yeah, occasionally he would sit in.

Hendrie: Occasionally?

Bell: Well, it was more of not sitting in. It was more of, "Gee, I know you guys are meeting about this. Here's my thoughts. I want to talk to you about these things."

Hendrie: Okay.

Bell: So, he'd come in and . . .

Hendrie: Yeah, and talk to you about it.

Bell: No, well no.

Hendrie: Or talk to the people.

Bell: He'd come in and talk to the group about it, yeah.

Hendrie: Okay.

Bell: You know or he'd be pissed off at somebody and he would come in and attack them, you know.

Hendrie: In front of everybody else.

Bell: Yeah. In a way, he was very jealous of the engineering team, of myself, Bob Puffer, Dick Clayton, Larry Portner, and Grant Saviers because we were pretty cohesive. We had to be sometimes just to protect ourselves.

Hendrie: Oh.

Bell: Well, DEC what are you doing about the . . .

Hendrie: Right.

Bell: This type of thing.

Hendrie: Okay.

Bell: The VAX team moved up to the top floor.

Hendrie: Who was on this original team, who were the key people?

Bell: There was Strecker, myself, Richie Lary, Dave Cutler, Steve Rothman. I think Tom Hastings came in as more of a scribe because he was very good at sort of keeping notes or keeping the record.

Hendrie: All right.

Bell: Well, anyway there were about six or seven of us. Strecker had been in research and he had had fundamentally outlined a proposal for a byte oriented or an architecture we called culturally compatible with the PDP-11. I named it the VAX-11 to keep us on track in terms of it being an extension to the 11. It was called culturally compatible.

Hendrie: Okay.

Bell: Of the same style but more bits and that you'd recognize it as a PDP-11. And then the other decision was that we basically put a PDP-11 in it so there was a mode in so it just executed PDP-11 code in a part of the memory, and that was essential because we ran a lot of software there for a long time. It ran RSX unchanged, so it was the 11 and had all the calls for RSX. And, I think the big thing was we were starting out things like paging, page sizes and all those things, you know. People were doing a lot of reading and a lot of calculations and stuff like that. I like to say we didn't see the switch over as rapidly as possible to I'll say a cache-based RISC type machine, but in fact we're going to go the opposite, which is we'll put all of that code in RAM, because at the time it was just peaking where you could basically put anything you wanted into a microcode. It was the opposite of RISC.

Hendrie: Okay.

Bell: So we wanted to bind the machine. It was the most complex machine you could build. It was used for both business and science. So it was both for FORTAN and COBOL. It had COBOL instructions, just like the 360 and Burroughs B5500. This is one of the things that most academic architects never bothered to understand.

Hendrie: And then you put those in and you could have as much code as you wanted associated with any instructions.

Bell: Right.

Hendrie: So you could obviously have extremely complex instructions.

Bell: Yeah, but people who compared it said they were doing it with little simple benchmarks and I said, "No, you've got to look at how are we doing on COBOL."

Hendrie: Right.

Bell: You know because we compared ourselves with the 360 on that, which was a fairly complex machine that way. In fact, all the machines, all the general purpose machines had really two instruction sets in them and the 360 and the Burroughs machines, all of them kind of were the same way because you ran these very complex edit instructions. We had an instruction to put items on queues and take items off in queue. So, that was the decision. I've got that memo that serves as, you know, we're going down this path. You got to go all the way. You know, if there's any justification, well, you've got to go put it in because it's cheaper to do it that way and you get more of a bang because of the speed that you're executing.

Hendrie: Yes.

Bell: You're executing. . .

Hendrie: Executing micro code.

Bell: Right.

Hendrie: Instead of subroutines.

Bell: Right, and so that makes sense and so that was the rationale at the time.

Hendrie: All right.

Bell: And so we met for I think four or five months and at the same time there was the hardware group under Bill Demmer implementing. The 780 team had built a general purpose microcode or general purpose emulator and I think we started in April in that period of time. By September we actually had a running machine -- a fast, or relatively fast, running machine that we could check the microcode out on.

Hendrie: Okay.

Bell: Or it was being written such that you could see how it was going to run. So we basically had somebody that would execute VAX code at that point in time, and I think we built a few of those and that was used, you know . . .

Hendrie: Then the software people.

Bell: Software because we had a machine to run on.

Hendrie: Which would be much faster than pure, software emulation.

Bell: Yeah, there were two things on software. One is we're going to do as much as we can in PDP-11 mode so you can write code on the 11s. Now we can change the assemblers. We can change the wires. That can all run right now in PDP-11 mode if we have that, so we had all of that software runnable. You know, when you decide to do it that way then you can take what you've got and evolve from that, evolve that code, and that was kind of why we were actually able to come out with a machine and the first deliveries were less than three years from the time we started.

Hendrie: The time you started meeting to actually work out the specs.

Bell: In fact it may be even two years. I'm trying to think, '75 and the fall of '78 was when I put out the plan called the VAX strategy.

Hendrie: Okay.

Bell: And that was all prompted because I wanted to wait and see how VAX was going to be accepted, and among other reasons to get this support so that it would easier to replace all the machines and get rid of the PDP-10 and PDP-11.

Hendrie: Now was that an idea in the back of your mind from the very beginning?

Bell: Yeah.

Hendrie: But that didn't mean it was time to try to sell it.

Bell: Right. It was too hard to do at the time. I mean, to me the machine -- all these machines -- were fatally flawed because they had to all be modified. The 10 had its problems in terms of address space. The 11 was groaning all the time and it was a miracle that we were able to actually cover all the 11 variations -- I mean, that software could actually be written that would run across all these machines.

Hendrie: Yes, okay.

Bell: There's a paper on that in the book on DEC by Ron Brender who is a compiler guy and it's called "Turning Cousins into Brothers and Sisters" or something like that. Yeah, that dealt with the fact that all the 11s were different.

Hendrie: Okay, good.

Bell: Yeah, well these machines are alike but there are all these variants. How do you do floating point? Well, however this engineer decided, so it was all the tricks you played. But anyway, I'm trying to think of where I am. So, you know, to me the VAX strategy was the other thing. It was very interesting about that and that was knowing the 360 story well but in fact having a different model of computing than IBM had. The model I had was a tiered model of machine classes. So I want to talk about Bell's Law because it explains a lot about the strategy.

Hendrie: We can do that but not right now. That's toward the end of the story.

Bell: Right, but the part about the VAX as opposed to the 360 was that, in fact, they were different styles of computing as opposed to you sell big machines to big companies, little machines to little companies, but they'd look exactly the same only they have different operating systems. It's just how much work they can do in that environment.

Hendrie: Uh huh.

Bell: And you make them as compatible as possible and so 360 . . .

Hendrie: So, you do as little software as possible.

Bell: Yeah, so the 360 had a number of operating systems depending on how big your company was, and what I wanted was a single operating system that the only thing that was different is what you might run on that and how many people were attached to it down to single user VAX, up to a cluster.

Hendrie: Yes.

Bell: So, anyway, the idea of getting rid of all the other machines and having just a single line was one that I came up with sort of after the VAX. I went to Japan and Australia in the summer of '78 and talked about VAX. I gave a talk in Japan. Then I went to Australia and gave some talks there, then came back to Tahiti where my children Brigham and Laura were with Gwen. We were on a sailboat for three weeks diving.

Hendrie: Okay, yes.

Bell: And so basically it was then I outlined all the VAX should do and why you would do it and I don't think we had really gotten . . . I'm trying to remember. Clustering was introduced to cover the high end and allow us to compete with bigger mainframes but using smaller machines.

Hendrie: Yeah, when clustering came.

Bell: Clusters became the major backbone of it. I'm trying to think when clustering came in because that was part of the HSC (shared disk controller) and we'd have actually common shared disks.

Hendrie: Yes.

Bell: Which by the way was a reason that Oracle got off the ground. The idea of that and a disk computer was probably a little bit later than '78 but was basically the idea of a single line that's going to be sort of central computing, traditional minis in departmental computing, and then personal computing when we could get VLSI.

Hendrie: Yes.

Bell: So, that was the idea and that we didn't need to build because we were building a whole pile of different PDP-11s that were being done. We had 100 quad-processor 11/70s that were in production.

Hendrie: A hundred quad processors?

Bell: Yeah.

Hendrie: Oh, my goodness.

Bell: And so those were all scrapped.

Hendrie: You actually were making them and selling them?

Bell: They hadn't been introduced yet.

Hendrie: Ah!

Bell: This was an 11/74.

Hendrie: Oh, my goodness.

Bell: And I recall going to the operation committee every few weeks and it was every Monday and I said to the guys who thought they needed it, I asked "Are you sure we need to introduce this computer?"

Hendrie: Yes.

Bell: "Can we not do it, introduce this? It's costing us money to get it there but it's going to cost us more if we put them in the field, and you realize that that means you're not going to be able to sell them VAXs for a long time."

Hendrie: Yes.

Bell: "And that when they decide that they need to have VAXs they're going to have to look at other alternatives."

Hendrie: Right.

Bell: And so this was kind of periodic. Every couple weeks I would say that. And then one day I went in there and they said, "You know I don't know that we need those machines in order to get our revenue. My customers are starting to get interested in VAXs and we just introduced it and they're liking them." And I said, "Let's stop."

Hendrie: Oh, wow, okay.

Bell: And so we did.

Hendrie: Stop it before you get this turkey out there that then we have to support.

Bell: Yeah, it's a support issue.

Hendrie: It's a horrible support issue.

Bell: Yeah, it was a nice fine quad processor machine and all of that. So I came back in September 78 and then wrote basically the VAX strategy memo. Here's what we want to do. I don't remember what date the first one was. The copy I have is I think a modification and it's like January '79, and it ended up being approved by the board in a board meeting in January '79.

Hendrie: Wow, okay.

Bell: But Ken went through various kinds of questioning like, “You're going to get rid of all those computers? What about those customers, you know, what are the 11 customers going to do?” And then the 10 guys were absolutely adamant against it. “You can't do this to us!”

Hendrie: I know.

Bell: The 8 guys . . .

Hendrie: We have the world's most loved machine here, the 10 guys say.

Bell: And the 8 guys were unaffected because I said, "Look, you guys, go off to wherever and as long as people want to buy 8s, build and sell them 8s as long as it's highly profitable."

Hendrie: Right, as long as you can make money.

Bell: Yeah, it's a money question.

Hendrie: Because we don't have a big software burden.

Bell: Right.

Hendrie: We don't . . .

Bell: But we're not going to build these big 11s, you know we're not going to build an 11/70 follow-on.

Hendrie: Yes.

Bell: And by the way we're not going to do these VLSI, LSI -- I think we had a couple of LSI-11, some more chips that were in process.

Hendrie: Or you may have had a single chip LSI or something like that.

Bell: Yeah, some of those, we had the T-11, we had a few of those things that were out already, the one chip 11.

Hendrie: Yeah, you did have LSI at that time.

Bell: We're not going to change that.

Hendrie: We're not going to do more of it either.

Bell: We're not going to do more of it and so that totally changed things. In engineering there wasn't a lot of -- I mean there was some conflict there but not a lot. It was more from, more worries from the ten group. I remember John Leng coming up at a big corporate meeting and John looked at me and said "You've got to see the movie A Bridge Too Far".

Hendrie: He was convinced the risk was too great.

Bell: He was convinced that you can't do it, and so I basically spent those four months convincing everyone. Andy was absolutely there helping drive the decision within the rest of the company.

Hendrie: Okay.

Bell: So, he was convinced.

Hendrie: So now was the initial VAX out?

Bell: Yeah. That made it possible.

Hendrie: Okay, the 780?

Bell: The 780 was out and then we were doing . . .

Hendrie: Was the 750 in the works?

Bell: The 750 was in the works and then we needed a really higher end follow-on. Oh, yeah, I was trying to think where we were. We hadn't started Venus yet I guess.

Hendrie: Okay, now which machine was Venus?

Bell: That was the 8600.

Hendrie: Ah, okay.

Bell: And it was due out in '81 or '82 or something like that and it ended up getting out in '84.

Hendrie: Okay.

Bell: It was really a disaster. I said, you know, it makes the guys at DG who did Soul of a New Machine look like geniuses. The DG effort I thought was just terrible hack engineering, and I said Venus makes that one look great.

Hendrie: I see.

Bell: Yeah, it was that badly done, badly executed. No, it was just absolutely poorly done and that was one where I ended up being the project engineer for about three months when it got in trouble.

Hendrie: Really?

Bell: I was running here and running there and running this group while running engineering, and then I always had a very strong associate, assistant, whatever. Portner did it for a while, Puffer did it for a while, and then I had staff guys. The guys that reported to me all ran the parts anyway and so I remember going in to a project review of that. There was a couple times I would go into a meeting and Ken would say "Gordon, why don't you leave right now and go fix that problem" and so literally I came into the operations committee one Monday morning -- this is after Friday the 13th review of Venus -- and said "Folks, this project is in deep shit." The whole processor engineering management and technical staff was there. I started at the project review. Demmer was there, all the engineering guys. We were reviewing it. And the guy reporting on it said, "Well we're having a little trouble with this or that." And so I went around and I said "What do you guys think about all this?" I said "Doesn't anybody see this is a f******* disaster? I have never seen anything this bad, you know, this project will never work. The way it's being designed, it can never be built or never work."

Hendrie: Yeah.

Bell: "You can't do it." And so I came in that Monday morning and, you know, I didn't sleep that weekend.

Hendrie: What am I going to do?

Bell: What's going to happen? How are we going to solve this problem? And so it ended up with Ken saying, "Why don't you just go out there and run it?" So I spent three months in Marlboro reviewing and hiring.

Hendrie: Really? Now was Kotok working on this machine?

Bell: No. I pulled Kotok in to work on it. Kotok had left the 10 by now and I don't remember what Kotok was doing. He was probably dabbling with the phone system or something like that.

Hendrie: Yeah, he was doing something. He was having fun.

Bell: He was having fun. And so I think I sort of talked to a bunch of the guys and they had all these four . . . I mean the guy running it had a guy working for him that was actually running it. He was building a new 10 and he was responsible for Venus, so the person was responsible for both high-end machines.

Hendrie: Now who is this?

Bell: Ulf Fagerquist was the VP in charge.

Hendrie: Oh, Leng isn't there anymore?

Bell: John had left at this point. Rose Ann Giodarno had the 10 marketing responsibility.

Hendrie: Machines, okay.

Bell: Large machines. There was an ECL 10 being built and then there was the 8600. I asked how many boards are in this machine because there were boards or how many boxes? He didn't know. And so I developed a principle there -- if I can ask any question anywhere in the line in that chain and they answer they don't know then these people are unnecessary.

Hendrie: They don't know.

Bell: If they don't know, if I know more than they do . . .

Hendrie: They are unnecessary.

Bell: They're unnecessary because, they know nothing.

Hendrie: Exactly.

Bell: It's a good principle of engineering. I looked at his org chart and he had staff meetings and there were these two guys that were doing all the work and then there was the HR and finance and a this and a that and all these people that had no responsibility to actually get the work done. So I said, "Oh shit, what have we created here?" So, it was quite, you know, a real eye-opener. Well, not really, but it was all good experience.

Hendrie: Yeah.

Bell: Oh yeah, I know how to run a project but the thing was all screwed up. I mean it was and it really had to do with you actually can't make a machine without simulating that complexity. That was a transition. You can't make a gate array or anything that's got hard-coded chips in it without full . . .

Hendrie: Without full simulation.

Bell: . . .proving that it works.

Hendrie: Yes, so there were basically lots and lots of logic flaws in the thing.

Bell: Yeah.

Hendrie: Well this was not an electrical engineering problem where it was a worst case design.

Bell: Right.

Hendrie: This was a software and logic problem.

Bell: Yeah and so they hadn't gotten to the point, it was like okay the problem with this thing is Motorola. They're not going to give us the chip and they were in there smoking. "Oh, Motorola, what are we going to do?" So, I said "Good sign, okay, it's probably not Motorola's problem." So I said, "Let's go to Phoenix." So we went to Phoenix, talked to the guys there, came back and I said, "You got your answer. It's not their problem. We can do our work and Motorola is just fine. Leave them alone. They'll deliver. It’s us. We don’t know what we are doing.”

Hendrie: Just order the right chips.

Bell: Yeah, because when I went into the meeting with the disaster it was like, oh my god, we've got to feed these guys chips because we got all these slots. And I said it all started when the team needed to fill the Motorola production line and they had nothing. "You can't feed them anything. You don't have anything that's going to work."

Hendrie: Yeah.

Bell: And so what are you going to feed them? “Well, this might work or this might work and then we can put in a board. Then we can test it.” I said, "Oh, no, you can't do it."

Hendrie: You can't do it that way.

Bell: Yeah, you can't do it that way.

Hendrie: Oh, wow.

Bell: And then we immediately created in that instant a simulator project and so we basically went back to ground zero and had to create a simulator. We had one that was sort of on the back burner that actually a student of mine at Carnegie -- well you probably know him, Tom McWilliams -- was working on.

Hendrie: Oh, I know Tom McWilliams, yes. Was he a student of yours too?

Bell: So, anyway, and he was at DEC at one point.

Hendrie: Oh, I didn't realize that he was doing simulators. Cool. Alright.

Bell: So anyway I was hoping to get through VAX today, but . . .

Hendrie: Why don't we wrap it up. It's six o'clock.

Bell: Yeah, but the whole VAX thing was really fighting to get the strategy. That was a four-month thing and Ken sort of hung back there and sometimes he'd be up on it. Sometimes it was bad. And so I went to the board.

Hendrie: He wasn't sure.

Bell: He wasn't sure absolutely and that was the reason to play it and he just sat there listening to all the arguments.

Hendrie: Okay.

Bell: And how we can answer them and that was the right thing to do.

Hendrie: Well, all right, thank you very much for spending your afternoon.

Bell: Right.

Hendrie: I think we do have at least one more session in the works if you can deal with it.

Bell: Oh, yeah, right, we're this far, you know. I never like to stop a project that's this far along. You've got an investment here.

Hendrie: Right.

