Oral History of Gordon Bell

Interviewed by:
Gardner Hendrie
Corrected by CGB May 9, 2008

Recorded: June 23, 2005
San Francisco, California

Interview 1, Transcript 1

CHM Reference number: X3202.2006

Gardner Hendrie: We have Gordon Bell with us, who's very graciously agreed to do an oral history interview for the Computer History Museum. Thank you up front, Gordon, for doing that.

Gordon Bell: Am happy to.

Hendrie: What I would like to start with is maybe you could tell us a little bit about where you were born, your family, what your parents did, how many brothers and sisters you had, just a little background as to sort of where you were during your formative years.

Bell: Great. I'm a fan of crediting everything to my parents and environment that I grew up in. I was born in Kirksville, Missouri, on August 19th, 1934. Kirksville was a college town and also a farming community, about 10,000 population, and had the Northeast Missouri State Teacher's College, which has now morphed itself into the Truman University. The town now has about 20,000 people. My father had an electrical contracting business and appliance store and did small appliance repair, and I grew up working at "the shop". My uncle was with him and had the refrigeration part of the business. And so I spent all my formative years at the shop and in that environment.

Hendrie: Did you have any brothers and sisters?

Bell: I have a sister who is six years younger -- she's a school teacher and was a primary school principal at one point. My mother taught fourth grade and so was a teacher as well. So I grew up in that kind of environment. Mother didn't teach while I was growing up, though. We lived kind of on the outskirts of the town. We had a couple of acres and had a great garden, so I still remember what real tomatoes taste like and all those other things you get when you have your own garden. You can't get good tomatoes in California until mid August when the dry farmed ones come in.

Hendrie: Yes, exactly.

Hendrie: What are the earliest memories you have of thinking about what you might want to do when you grew up?

Bell: That has been a really hard one for me to think of. I turned out to be one of the best dishwasher repair people in the area because it had cams and cycles and mechanical stuff then. I did house and building wiring, and at the time there was the REA or Rural Electrification Association or Administration. This was a federal program to wire all the farm houses in the country. So I went out and did a lot of wiring. I also installed industrial equipment and worked on all of that kind of stuff, e.g. appliances of all types, houses, buildings, and small industries like creameries, bottling plants, granaries.

In retrospect, I recall starting to earn a weekly salary of $6 at some early age, perhaps when I was 12. This meant having a savings account and being comparatively rich. In those days of not having strict child labor laws, it was easy to work. My father felt that I was doing a man’s work as an electrician at this age and should be paid accordingly. I do recall rewinding the stator of a motor when I was in the 5th grade. Today, you throw motors with burned out windings away.

Bell: In fact, I like to say I retired as a journeyman electrician when I went to MIT in 1952.

Hendrie: Well, now, tell me about the things you did, what you studied. You obviously learned a lot from just working with your father. What did you study in high school? Do you remember what courses you liked or...

Bell: Yeah. What was really important was having a really wonderful science teacher and a wonderful math teacher. I still remember them very fondly. At that point in time and in Kirksville Missouri, you didn't take calculus or it wasn't offered, but I took chemistry and physics and then geometry and, maybe, solid geometry and foundational stuff. Those were really critical to enable me to go to MIT. At some point, I don't know when, maybe when I was 12 or so, I thought I wanted to be an engineer. I had no idea what an engineer was. I had books that I sort of read -- books of knowledge and an earlier kind of The Way Things Work -- so I gleaned that somewhere, somebody figured out how to make these things work and that was the interesting thing, not repair. Repairing them was okay but, in fact, designing them or inventing them seemed like a lot more fun. So that was basically the course that I set fairly early, with no one telling me I should or should not be doing this.

Hendrie: Or having some relative who was an engineer.

Bell: Yeah, I didn't have any.

Hendrie: You had none of those role models that were engineers.

Bell: I had none.

Hendrie: You just sort of decided it.

Bell: And even the difference between a scientist and an engineer—I never sorted that out. In some respects I still haven't, especially since I was elected to the National Academy of Sciences in 2007.

Hendrie: Yeah. So where did you think you might want to go to school? I mean, we know, of course, you ended up at MIT but what were your alternatives or did you have...

Bell: I took the college boards and I think I applied to Case, Rensselaer, maybe Rochester, IIT and MIT. In all, a half dozen schools.

Bell: I also remember that one of my father's golf buddies was a math teacher at the college. He tried to persuade me to go to the college and get a better foundation or, at most, then go to IIT or Missouri U or someplace else, but you really didn't go from the farm to MIT in one step -- that was just too much and it might break me and all of that. So I said, okay, my math is weak and I haven't had calculus like kids in eastern prep schools, so what are you offering? They had a college algebra course, so that summer I took college algebra by correspondence with him and, at that point, he may have decided that I was okay and that maybe I could actually make it to MIT. He didn't endorse the idea.

Hendrie: You could actually make it.

Bell: So, anyway, it didn't dissuade me from going to MIT. I think that's an important thing not to discourage people who want to do something. When you encourage them you help them versus saying, well, you really aren't that well prepared and you may not really have it. So start small.

Hendrie: Okay. So you picked MIT out of the ones you got into?

Bell: Yeah.

Hendrie: It was obvious to you that that was the best one and...

Bell: It was obvious, yeah. It was a little bit awesome to go there in terms of just the scale. Going from a class of 100 to a class of 1,000 was a big deal. And then just going to MIT, still comparatively small compared to large universities. I apparently was brand conscious.

Hendrie: Okay.

Bell: But it was a great experience.

Hendrie: So tell me some stories about your experience, when you first went to MIT. Had you ever gone there for a visit or anything? Or did you just go cold?

Bell: No, I just went cold.

Hendrie: Cold, from Missouri to Boston.

Bell: Well, in terms of preparation, at that point in time, you had to have an interview, and I drove to Kansas City to be interviewed by an MIT alumnus. I don't know whether there was a pass/fail associated with that or not but, anyway, I remember doing that. Then I think I actually remember going to another place to visit another MIT student at the time. He was a junior or something like that and just talked about preparation. These were not too far from my hometown but, anyway, that was kind of the preparation. Then my parents took me from Missouri to Boston, and the traffic in Boston, of course, is horrendous – and circuitous at that point. It wasn't as bad then, of course, but I remember being dumped off at MIT with my trunk full of stuff and joining a fraternity. For me, at least, that turned out to be a very good thing. It was, in a way, a smaller community that you didn't have in the dorm and you got more, I'd say, more intimate help and social interaction. And there were two guys there from Missouri and I was in their room. One was from St. Louis and I think the other was from Kansas City. I still correspond with one of them. He was a chemical engineer. They both got their Ph.D.s, both were very bright and so, actually, just being there and being able to get a little bit of extra help with chemistry was useful.

Hendrie: Okay.

Bell: But, anyway, I really had no trouble at MIT. I ended up with about a B average, and that's reasonably good and will get you into other schools and get you into the doctoral program and all of that. But, basically, I went into the co-op program because I wanted to understand what it was like to be an engineer.

Hendrie: Now, did they have a co-op program at this time?

Bell: Yeah, they were early -- one of the early co-op programs.

Hendrie: So now what year are you in when you do this?

Bell: That started after my sophomore year.

Hendrie: Now, did you specialize in anything?

Hendrie: As a freshman, you're just taking pretty much the same thing.

Bell: Then I went into electrical, course six (EE), as a sophomore, and then, if you had good enough grades, you could go into the co-op program. I decided that was for me and so I started the summer after I was a sophomore.

Hendrie: So where did you . . .

Bell: It was an interview, and even that experience was good -- interviewing, deciding where you want to work, all of that. Bell Labs, GE, and American Electric Power were in that, so I ended up taking a new program which was American Electric Power and GE. I ended up taking two semesters each at AEP and GE, and I thought I might have wanted to have been a power engineer.

Hendrie: Yeah. Okay.

Bell: So I ended up in Canton, Ohio, and I went around to all of the various divisions from residential to power transmission and power generation.

Hendrie: That's at . . .

Bell: American Electric Power.

Hendrie: Ohio Power was in Canton, Ohio?

Bell: Yes.

Hendrie: Was that the one you did that first summer?

Bell: Yeah. That was my first assignment, I think, and then I went to GE aircraft gas turbine testing where I actually made a device -- a multiplexer to measure experimental jet engine compressors. They had Leeds Northrup or basically an uncomplicated strip chart recorder, multi-channel strip recorder, and they were taking a lot more data than you normally get with one of those things. So I built a stepping relay device that multiplexed this multiple channel thing so that you basically got a lot more points on the paper.

Hendrie: On a lot more variables?

Bell: Yeah.

Hendrie: So you could . . .

Bell: Because these were jet engines. They had big wind tunnels and so they instrumented everything and were trying to get more pressures and temperatures out of the test. So I remember doing that. Then one of the programs was at Syracuse and that was in heavy military electronics. I worked on a radar A to D converter so that you send the radar bang out and you want to find out after a period of time what the return pulse height was which gave an indication of target size. Return time was the distance.

Hendrie: Alright.

Bell: So I did that. And then I had an assignment with AEP in New York to make graphs about using power to melt ice on transmission lines. We wrote a program that ran on IBM's 650 at their Madison Avenue Headquarters.

Hendrie: You were getting a lot of experience about what the real world was like out there.

Bell: Yeah, exactly.

Hendrie: And what people did who were engineers.

Bell: Right, exactly. And just having that variation in these different environments was really very good. AEP was downtown New York, and I decided I didn't like to commute to New York, so I learned that as well.

Hendrie: So what did you do at the power? I'm just curious what you did at the power plants.

Bell: I spent a week each in all of these different programs.

Hendrie: Are these different power plants?

Bell: Well, I had a couple weeks at a power plant. In a way, it was bad because I didn't do anything. It was just observing what people do, and I felt like these people were trying to babysit and entertain me. I think in one place I almost got kicked out. I don't remember what it was. I mean, I wasn't very interested in meter calibration, so I think I started building a meter or something. I took a bunch of parts and went off and the guy goes, "You're not supposed to do that!" I remember getting reprimanded a bit about not doing what I was supposed to -- that I was just there to listen and not to do anything.

Hendrie: Not to do anything, yeah.

Bell: Anyway, the other thing I learned was that GE was a sea of desks, and you just work on this little part of the circuit and you work on this part of the radar. And I said, no, this doesn't look like very much fun to me, because you were really pigeon holed. So there was a sea of desks and then the boss was over in the corner. There was a little bit of hierarchy because you have a cluster of desks and your group was in this cluster and they had, I don't know, 50 engineers in this room. We played bridge at noon.

Hendrie: Now, was the boss on a raised dais like Japan?

Bell: No, no.

Hendrie: Like they do in Japan?

Bell: He wasn't on a raised dais but he had his own glassed in area. And so, it was a bit too hierarchical and too segmented, and I didn't feel a lot of comfort in there. And so, in a way, I didn't want to go to work. Here I was, I had programmed myself to be an engineer, and this doesn't look like a lot of fun to me because there was very little responsibility.

Hendrie: Yes, of course.

Bell: And so the rest of the MIT experience was just fine. I'm going to my 50th class reunion next year.

Hendrie: So tell me about some of the other courses?

Bell: Yeah, that was the co-op part but the other . . .

Hendrie: You stepped through, you got to be a junior.

Bell: The other thing was that this was before computers. I graduated in '56 with a bachelors and I graduated in '57 and had a bachelors and masters.

Hendrie: It was a five-year co-op program?

Bell: Right.

Hendrie: Did you go to the co-op program at all? Did you just work during the summer or did you work during some semesters?

Bell: You worked during semesters, too, so it alternated on where you were at a particular time.

Hendrie: Okay. So it was a classic one.

Bell: Yeah.

Hendrie: Like Northeastern has run for years.

Bell: To me, it was great, yeah. And U of Cincinnati, I think, invented the co-op program.

Bell: But, anyway, it was very nice because you got this understanding about what it's all about. I took essentially all of the digital courses that they offered, all the computing courses. I'm not sure that there was a program, per se, or a track or anything like that, but that was kind of the one I was on. They had, for example, acoustics, power, electronics, control tracks and various other programs.

Hendrie: Do you remember who was teaching any of the digital courses at that time?

Bell: Yeah. There was a guy by the name of Al Susskind who wrote a book on A to D converters. He taught that course. Caldwell taught switching theory. Hennie was a grad student and wrote a book on switching theory. So there were those people. Probably the most memorable courses I had were by Ernest Guillemin, who was pretty much the inventor of circuit theory. He essentially founded circuit theory, electrical circuit theory and analysis. He was probably one of the most memorable teachers I had there, and a great lecturer, just fabulous. And then I remember a mechanics and statics and dynamics mechanics course that was taught in mechanical. Everybody said they hated it. It turned out I loved it. I mean, it was just sort of all these things moving around and you isolated bodies. I always remember that more than almost anything because you'd get some very complicated structure and you'd have to find the force on a beam or something like that, and the guy would say, "isolate a body". I loved it and it was probably the most important advice for an engineer — solve problems by parts. And then I took two semesters of mechanical drawing or the equivalent of drawing. One was mechanical drawing. The other was sort of an analytical geometry course taught by a professor named Steven Coons who turned out to be one of the graphics founders. Ivan Sutherland credits Coons with being the important pioneer and inspiration for sketchpad.

Hendrie: Oh, isn't that amazing?

Bell: He was a great graphics person.

Hendrie: Did they have any logic courses?

Bell: Yeah. There was a two-semester switching theory course and a couple on logic design courses.

Hendrie: Switching theory had Boolean algebra in those days?

Bell: Yeah, and then there was a logic design course or kind of a digital systems design course and a couple of programming courses. At the time, we ran on Whirlwind, which was the MIT built machine that they ran certain jobs through. I also programmed the 650 and the 704 as batch jobs.

Hendrie: Okay. The 650 and 704 were there by the time...

Bell: Yes.

Hendrie: Before you left?

Bell: Yes, right, right at the end. The 650 was more of an open shop machine. Well, they were both kind of programmed or used the same way. You actually signed up for the 650 as a personal computer and the 704 was batch at the time.

Hendrie: Yeah. So you brought your punch cards to the 650.

Bell: These were only little student jobs, so you didn't really get a feeling for programming. My thesis advisor was Ken Stevens who, incidentally, is still there. He's about 10 years older than I am and he's still at MIT doing speech research. He's one of the premiere speech researchers. He has taught more people about speech than any other person. For my master's thesis, I built a sound level meter that you'd take out in the field and make histograms of noise levels. So he was an acoustician, among other things. But he took me in as an advisor after Richard Bolt had left. Bolt of course is part of Bolt, Beranek, and Newman or BBN.

Hendrie: Okay. Noise level. Was that at various frequencies?

Bell: Yeah. It didn't do frequency and level, but you got a histogram of the noise levels in various places and you set what bands you wanted to measure. Incidentally, all that training came back to me a week or so ago. I was getting annoyed with restaurant noise and decided a criteria for restaurants is now their noise levels.

Hendrie: Ambient noise.

Bell: Yeah. And so I went over to Radio Shack near us and bought a sound level meter. The guy said they hardly ever sell those and the boss finally pulled one from a high shelf.

Hendrie: Very good.

Bell: I haven't really used it yet, but I'm threatening to for these restaurants.

Hendrie: I think that's a wonderful idea. What you ought to do is start taking measurements and send them to Zagats and they can put them in their book.

Bell: Yeah. That's my basic idea, that, in fact, you look at it there. Now, my theory on this is probably the popularity of the restaurant is inversely proportional to the noise level. So people like high ambient noise and they really are there because everyone else is.

Hendrie: I see. It feels stimulating, it does something emotionally to you.

Bell: You're at this happening place because of all this noise that's going on. Even though you can't talk, you can't communicate with anybody.

Hendrie: That's a good theory. I like that.

Bell: Yeah, so, if I can get enough data, it'll be an interesting one to check.

Hendrie: Well, let's see. Did you get involved in any of the MIT clubs? I know lots of people who went on and did work like Alan Kotok who were at the Tech Model Railroad Club, that there were all these people that got interested in the switching equipment to run the model railroads.

Bell: No. Alan and all those guys were model railroaders or members of Temerc. They were sort of the beginning, the foundation of a lot of the computing that ultimately got done at MIT and DEC. They included Peter Sampson, Slug [Steve Russell], and others who ended up at DEC and certainly influenced us. I had a congenital heart ailment and so I really was not an athlete.

Hendrie: MIT is not known as an athletic school, either.

Bell: Right. And so I was a vice-president or president of the athletic association at one point. And I was a manager and managed the lacrosse team and I think the soccer team.

Hendrie: Oh, all right.

Bell: And I started out in the MIT band, but I quickly dropped out of that because of time commitments. Just the difficulty of practicing . . .

Hendrie: What instrument?

Bell: I played the trumpet. I loved that and I played it in high school band.

Hendrie: So you were in the band in high school?

Bell: Yeah, band and the orchestra. And I enjoyed that, but unfortunately, there are choices you have to give up, and I was trying to think of other things, but I'd say that mainly they soaked up time. I think I was an officer, did stuff in the fraternity but I think probably . . . Oh, and it was across the river from the clubs. There was probably some other club I was a member of that I don't remember, but the athletic association and the fraternity took time.

Hendrie: Okay. So when you're approaching graduation, you must have been thinking about where you were going to go and what you were going to do. Did you ever think you wanted to continue an academic career or did you want to go out and get a job?

Bell: The problem was the co-op thing had convinced me that I didn't really know that I wanted to go get a job, and so this is where serendipity kicks in. A friend of mine -- a really good friend and graduate year roommate, Bob Brigham, for whom my son is named -- and I walked into the department head's (Gordon Brown, an Australian) office. We walked in and he said, "Well, what are you going to do with your lives at this point" or something like that. I don't remember how we got there, but he basically said, "Why don't you guys go to Australia and help my friend, Rex Vowells, who's starting a computer program in their EE department. They've just got a computer at the University of New South Wales and the university is just starting up. It's an eight year old university and wants to pattern itself after MIT. Go there and teach and do some research, teach some courses." We said, "Well, that doesn't sound bad."

Hendrie: Yeah, that sounds like a great adventure.

Bell: Yeah. And so we applied.

Hendrie: Both of you?

Bell: And they accepted us and it was a wonderful, wonderful experience. I'll tell you a little bit about that, but I've got to put the other book end on. I was at the University of New South Wales in May of that year and gave a lecture on technology futures, I believe. I walked into the department head's office and outside of his office they had a key punch and a big reproducing card reader, card punch. I said, "Gee, did that come from the DEUCE [Digital Electronic Universal Computing Engine]?" And they said "What's the DEUCE?" And I said, "That was the English electric DEUCE called UTECOM, which was the University of Technology Computer, and it was about the second or third computer brought into Oz. It was a machine that Turing designed." I said, "I programmed on that. I think I used that key punch and we used that to reproduce cards." By the way, 2007 will be my 50th anniversary with Australia.

Hendrie: Oh, my goodness.

Bell: And they said, "Oh, my god." And so anyway, they invited me back to be a visitor any time I wanted to spend more time in Australia.

Hendrie: Oh, excellent.

Bell: But anyway, I spent a year there programming this machine that, again, Turing had worked on at the National Physical Laboratory.

Hendrie: Yes, I'm, of course, familiar with DEUCE, yes.

Bell: It came out of the ACE [Automatic Computing Engine]. Raj Reddy was apparently at the University of New South Wales at the same time I was there or near then. He had come from India and was a graduate student.

Hendrie: Oh, but you did not know him then?

Bell: I didn't. No, I had no knowledge of him.

Hendrie: So what sorts of things did you do? You taught. What kind of courses did you teach?

Bell: Yeah, we taught. Bob taught the first course and that was all on sort of switching theory, and then I taught the second one which was on logic design and digital systems design, computer design. And, in fact, I met somebody recently in Oz whose father had taken my course.

Hendrie: Oh, my goodness. Alright.

Bell: But, anyway, the main thing we did was we worked very hard. That was the first paper I had ever published. Bob and I wrote a program which was we called Symbolic Optimum DEUCE Assembly program or SODA DEUCE, which was a very difficult machine to program. It was the philosophy of Turing -- don't waste any hardware on what people can do. Make the people work, not the hardware. And so the various coding of the instructions were essentially opening gates for delay lines to go into different other delay lines. It was like you wrote this little traffic program to move a word at a given point in time from one delay line in through a gated adder and into a register.

Hendrie: The register was another delay line?

Bell: Yeah.

Hendrie: Everything was mercury . . .

Bell: Delay lines.

Hendrie: All mercury?

Bell: All mercury delay lines.

Hendrie: Okay. I didn't know whether they were mercury or whether they were acoustic.

Bell: No.

Hendrie: No, they were mercury.

Bell: No, we have them here in the museum.

Hendrie: Okay.

Bell: And we have the drum that Murray Allen, whom I met then, had given us. He was a professor at the University of New South Wales. He had built a computer, and I met him when I was there and he had retired. There were ten or 12 32 word delay lines, and then there was an 8k word drum that was accessed in 32 word chunks, bands, and then there were 256 tracks. There was a 16-head drum, I mean, a 16-head read/write drum or reader and a 16-head writer and those two head assemblies moved independently. . .

Hendrie: Oh, they actually moved?

Bell: . . . into 16 positions so that's how you got 256 positions.

Hendrie: So this was not a fixed head drum, this was a moving . . .

Bell: Right. Moving head drum.

Bell: Right. And it turned out that they always ran our program as a test program. Otherwise, it wasn't worth using the machine.

Hendrie: It was like a systems level diagnostic program?

Bell: Yeah, it beat the heck out of the drum head positioned.

Hendrie: It exercised so much stuff that . . .

Bell: But, what it did was to convert the DEUCE into a virtual machine with a one level store. I gave a keynote at Manchester at the 50th anniversary, and I said now I've got a perspective, now I understand how the one level store came about because we invented a one level store while we were in Australia and using the DEUCE. It turns out Manchester did it first and we didn't call it that at all. But we basically took the 8K memory and then made it a homogenous memory. We totally got rid of the notion that there was any hierarchy, and so you just wrote instructions and they sat in an 8K environment, not the working environments of the few 32 word delay lines out of which programs had to run.

Hendrie: It was one level storage.

Hendrie: But some of the memory was a little faster than the other memory.

Bell: And we took care of moving the program into the program store automatically...

Hendrie: Into the fast memory.

Bell: or the 10 delay lines.

Hendrie: Okay.

Bell: And then we also made symbolic assignments, so everything was assigned so you could talk about addresses symbolically. It was a three address machine so you needed three addresses to get control sometimes when you wanted to get into a loop but, other than that, we tried to optimize where you place the instructions. So you placed the instructions in the right place and then you had to place the data in the right place so that this thing ran as fast as possible.

Hendrie: But that was an optimization as opposed to...

Hendrie: The machine wasn't designed so that the next instruction must be underneath...

Hendrie: ...the head at the time you finished the previous one?

Bell: No, you had to say where the data was and where the next instruction would be.

Hendrie: Yeah. Classical three address for serial memory machines.

Bell: Yeah. Ultimately, English Electric used our program as a backend for a Fortran compiler. So they basically compiled Fortran into SODA instructions.

END OF TAPE 1, INTERVIEW 1

Oral History of Gordon Bell

Interviewed by:
Gardner Hendrie
Corrected by CGB May 9, 2008

Recorded: June 23, 2005
San Francisco, California

Interview 1, Transcript 2

CHM Reference number: X3202.2006

Bell: Sort of continuing in Australia. . . when I was introduced last month at New South Wales, they had done their homework and the guy introducing me said, “He wrote his first paper on that, and so he wrote a paper that was published in the British Computing Journal.” That was the first article I (actually Bob Brigham and I) had ever written. And I am still proud of that work.

Hendrie: So can you tell me more about the program, a little bit more? There wasn’t a symbolic assembler for it.

Bell: No, I mean, you know, there was a bare machine.

Hendrie: Did you literally with the used coding sheets in binary?

Bell: Oh, yeah. The coding sheets were in a form. There were 32 lines on a coding sheet and you were filling those slots on each of those and then those were brought into memory. And there was kind of a rudimentary thing that you loaded. I can’t remember exactly how programs were loaded, but users signed up and used the computer just like a personal computer.

Hendrie: Yeah, I’m just thinking of trying to get some understanding of how hard it was to actually write the program.

Bell: Yeah, because, you know, you started out with a bare machine and the loader in that format. I think there were a couple of binary loaders and that you loaded the drum, you loaded the delay lines, and specified where the instructions were going to be. But your coding was on these 32-slot sheets with fields for the operation, operands, and next location, and the paper had enough of a machine description that I could go back and recall it. Basically there was a part that I worked on, the parts I remember that were the hardest, were in a sense the resident run time. Our virtual machine had floating point, so I wrote all the floating point and then also the runtime system, which, in fact, brought down instructions from the drum when needed and into one of the delay lines. And I remember, we would constantly need a little more functionality in what it did. And we’d go into this 32 words run time and then find another way to code the function.

Hendrie: And struggling to keep it in 32 words.

Bell: It was the worst kind of dirty programming you could ever imagine!

Hendrie: Exactly. Oh, this instruction, and I add this to that, yeah. The instruction itself, I’ll use that as a constant. Yeah, all the tricks that you play. That’s fun but clearly violate rules of good design.

Bell: And so I do remember that as an experience. And the story Gwen likes to tell is that’s where I met my wife, Gwen Bell, and that I proposed to her on the New South Wales DEUCE. I wrote a program that was essentially just a little flowchart. The way the display worked was that you could bring down stuff into one of the memories. Essentially you had a 32 by 32 display grid, and you could write messages into it and it would sort of come down and you could bring them in and it would sift through this display store and that was delay line 10 I believe. So you brought them in and then you’d slip it by one and it would go off the bottom and then you’d bring another line into that. And that was the display, one of the output devices. Thus the memory was also the output or user interface.

Hendrie: It had a display on it.

Bell: Yeah. You basically could look at any of the delay lines on a scan basis. You looked at the 32 bits and the 32 words, but you could turn and look at any of the delay lines that you wanted. And so delay line 10 was where stuff was coming down from the drum and you could watch the messages flow. And so that was our loaded program where you could propose to people.

Hendrie: You got her in front of the machine?

Bell: Yeah. I said, "Here, run this program." And, you know, they had the switches that you could input. Say you input 32 bits at a time.

Hendrie: I see.

Bell: But it was purely a binary machine coding and you programmed and the initial programming was in row binary with either 12 or 24 32 bit instructions per card. Yeah, that gets you 24 instructions so you take 64 columns with 2 instructions. We may have had a preliminary assembler so you could put 1 32 bit binary instruction per card, otherwise you punched pure, row binary. But then the thing was such that we actually filled bits in when we had to repunch a row. I mean here you're keypunching this damn thing by hand and so you get pretty good about not making mistakes because you're punching bit by bit.

Hendrie: Yes, exactly.

Bell: And then so you moved from the top row to the zero row and so on. And so you make a mistake, it's oh, shit, what do you do? You pick up a punched one or chad and put the one back in to make it a zero.

Hendrie: You put the chad back in?

Bell: You put the chad back in and you do it kind of the right way and then you quickly go over to the reproducing punch and reproduce that card before that bit falls out. And I remember a few cases of well, this program ran yesterday. And so some bit fell out.

Hendrie: Because you'd forgotten to reproduce it.

Bell: Somebody forgot to do the reproducing. But you really should please reproduce those decks before anything happens.

Hendrie: Oh, that is funny.

Bell: But that was it, and it was a great experience. I mean Australia was just a wonderful experience. In fact so wonderful that I'm probably going to spend maybe up to half my time there in the next few years.

Hendrie: Oh, very good. Okay. So just a little bit not really computer history, but tell me, how did you meet Gwen?

Bell: Well, she was also a Fulbright scholar.

Hendrie: Now were you a Fulbright?

Bell: Yeah.

Hendrie: You went there on a Fulbright?

Bell: I went there on a Fulbright. That was part of the…

Hendrie: Part of the deal.

Bell: Part of the deal. Yeah, in fact…

Hendrie: You hadn't mentioned that. I didn't realize that.

Bell: Yeah, and so there were, I don't know, about 20 Fulbrighters, there was a nice collection of Fulbrighters. And I got there on my birthday and Gwen had already flown in. Virtually all of us had taken the boat and it was like a 20-day boat sea voyage.

Hendrie: To get there?

Bell: To get there. So I've had enough boat riding to probably last me. But anyway, she got there and she had a birthday cake. She was sent out and they got a birthday cake, a solid fruit cake and it was nice but different. And then over time, she was in the University of Sydney and she lived with a lady who was an art historian. And then I think at Christmas time, everybody came back from vacation. Three of us bought a Land Rover, traveled up through the center of Australia together, and then we got back and five of us, including Gwen, rented a house together. So we were roommates there when that happened. And so anyway, that was for a term.

Hendrie: Now what city were you in?

Bell: We were in Sydney. New South Wales is the state and we were in Sydney.

Hendrie: You were in Sydney, okay.

Bell: Yeah. I remember when I got there and had to ride the bus for a couple of days and I said, "This is for the birds." I went off and said, "I think I'll get a motorcycle." So I went to the motorcycle store and they had these beautiful BMWs and all that. And I said, "Well, what do you have for 25 pounds?" And they sort of started laughing and then this smile broke out and these two guys looked at each other and they said, "Do you think it runs?" And so they went off in the back corner of the lot where they had all these old motorcycles and they picked out this very large motorcycle. And it was the same age as I was. It was a 1934 Ariel. And so it was 22 years old. And so they brought this thing out, cranked it up, and they said, "Here, 25 pounds." And so I bought this motorcycle and used this motorcycle. I even ended up paying a 25 pound fine for speeding with it. A cop on a BMW bike picked me up on the one superhighway that went from my apartment to the University.

Hendrie: And that was your transportation.

Bell: Yeah, until we bought the Land Rover at Christmas time, and then three of us bought that and brought it back to Sydney and that was our transportation.

Hendrie: Oh, that's great. That's a great story.

Bell: Yeah, and the interesting thing is the people that I was there with, for example, Barry Thornton, who was on the math faculty, and he's still teaching. He ended up being the president of Honeywell Australia for some period of time. He's back and he's basically an aerodynamicist mathematician and he has medals from NASA to prove it. I met him again when I was back there last time recently and he was still doing his thing. You know, a great mind.

Hendrie: That's wonderful.

Bell: So you see all these people that you met at various times.

Hendrie: And then to go back and so many of them are still there.

Bell: Yeah, the other part of Australia. So Australia has been this long-standing not commitment but relationship. When I was at Digital, the University of Perth wanted to buy a PDP-6. And so we ended up selling them one, and in one of the processes, I went there and I hired Ron Smart. Ron became the head of the DEC Australia. Ron was the person who ran the comp center that housed the DEUCE at Unicom and then he ran the DEC office in Australia. Also Max Burnett had been head of the Australian office when I had met him at Digital. The last time I was in Australia, why they had an old-timers get-together at his house. And so Barry Thornton was there and that's where I saw Barry again. And so I met those guys again. And Max has got a wonderful collection of computing artifacts and is a good friend of the museum. Have you met Max?

Hendrie: I don't think so.

Bell: Max was the head of DEC Australia for some period of time and has a wonderful collection of hardware and manuals. He's probably got the greatest collection of DEC manuals outside of the museum. He's also got a huge collection of DEC gear, you know, everything from PDP, and I think there may be a 5 there, he had an 8, and various other machines. And it's taken over his house. I mean the museum is in his house, and he's trying to get this all moved to a site. I mean there's a great collection of gear in Australia. The Australian computer folks have amassed a nice collection.

Hendrie: They're trying to find a home for it.

Bell: Yeah. The Powerhouse Museum is one place they have, but this is another thing. But anyway, you know, it's going. It was nice to see him. His house is just taken up with artifacts -- hardware, and there's a wall of manuals.

Hendrie: That's wonderful.

Bell: He can convert any kind of media to any other kind with his collection there. He's got some 9-track tape drives, paper tape, DECtape and more.

Hendrie: That's good. Well, I think we probably ought to call a halt right now with you still in Australia and we'll take up...

Bell: Yeah, we're up to basically 1959 or so.

Hendrie: Oh, very good. All right, thank you very much.

END OF TAPE 2, INTERVIEW 1

Oral History of Gordon Bell

Interviewed by:
Gardner Hendrie
Cgb correction 5/11/08

Recorded: July 19, 2005
San Francisco, California

Interview 2, Transcript 1

CHM Reference number: X3202.2006

Hendrie: Continuing his oral history for the Computer History Museum and I think we left off, if I read the end of your last transcript, you're still in Australia and you thought it was about 1959. I'm not sure about the date.

Bell: Yeah, right. Well, actually I left Australia in August of '58.

Hendrie: Oh, so it was not '59, okay.

Bell: No. But anyway--

Hendrie: We'll get the dates right.

Bell: The tail end of Australia. One of the last things I did in Australia, and I don't know when this was, was I wrote a program for the DEUCE that proposed to Gwen Bell. You ran that program from the toggle switches and you entered yes or no and there was a little flowchart program with a loop in it. And there was a display on the Deuce which was actually part of the memory that you could put messages in essentially line by line so that 32-bit lines bits would come down in a scrolling fashion. And so, it was a 32 by 32 points.

Hendrie: Display, so you made the characters?

Bell: Yeah, so it was a 1,000 point display and then you sort of made these little dot matrix characters on there. I don't know how many lines, not very many lines, that you do 32 lines.of probably 5 x 7 or maybe smaller like 3 or 4 by 5 or 6.

Hendrie: And it was basically showing what the memory is?

Bell: Yeah.

Hendrie: This was a bit memory map.

Bell: It was a memory map.

Hendrie: A bit memory map.

Bell: Yeah, word by word, and so words would shift or scroll down from the drum and you could sort of slide these words down through there. So I wrote this program and then Gwen and I left Australia somewhere mid-August, arrived back in the U.S. in November. But anyway, let's see, I guess there were a couple of memorable or one memorable incident coming back. The first stop was Singapore and Gwen had a classmate there, a Chinese architect classmate, who we spent a few days with. His name was Bill Lim. And then we went to Thailand and spent a few days in Bangkok and then to Burma. Very few people had ever gone to Burma so we had a few days in Burma, then Turkey, Istanbul, and then Athens and then through Europe. And we spent until probably I think November, mid-November doing all of that, sort of crisscrossing Europe, being in London several times. I don't know, I think we were in most of the countries. We were in Holland, Germany, maybe Austria, Italy. We weren't in Yugoslavia because we took the boat from Athens back to Italy and then up. The memorable incident from a computing standpoint was we had written the SODA program to make it easier to program the DEUCE and to allocate software to storage and have a single level store. Essentially, it was the equivalent of what Manchester did to make a two-level store into a single level as and I think I mentioned that earlier.

Hendrie: Yes, you did describe that.

Bell: And then we went to visit the National Physical Laboratory in Teddington, outside of London. I don't remember whether I mentioned that.

Hendrie: No, you did not.

Bell: And, god, the name will come back, but anyway we -- ah *as the name returned to my memory* -- went there and visited Wilkinson, James H. Wilkinson, who was the famous numerical analyst. I went in there and talked about the program that we had just written. By the way, the paper ultimately got published in the British Computer Journal.

Hendrie: Oh, okay.

Bell: That was I think actually my – our -- first program and then our first publication. We were sitting around having tea and then Wilkinson said "What can you do with this program?" And I said, "Well, you can really write a matrix routine in like ten or 15 minutes. It was just basically a few lines of code. And he said, "I can do that with the DEUCE code" and so . . .

Hendrie: He could write it just in . . .

Bell: Anything, yeah.

Hendrie: In straight assembler, okay.

Bell: Right. And then I told him about this when he came to the museum in Boston in '79 I think and gave a talk. He said, "Yeah, I was pretty cheeky then." But he spent a lot of his life working at Stanford as a visiting scholar.

Hendrie: Oh, really?

Bell: Yeah.

Hendrie: Okay, so he eventually emigrated.

Bell: Yeah, but he's a great numerical analyst.

Hendrie: Oh, that's a good story.

Bell: Yeah.

Hendrie: And then eventually you came back to the United States.

Bell: Right.

Hendrie: Now, did you know what you were going to do when you came back to the United States?

Bell: Let's see, at that point I don't think I knew exactly what I was going to do. I'm not sure whether I had an offer from MIT to come back as a staff engineer for research. The Division of Sponsored Research had essentially staff engineers, and I think I may have had an offer from Ken Stevens. By the way, in fact, I sort of woke up in the middle of the night thinking maybe I ought to bring my computer over to this interview because I recall having seen the offer letter from MIT or Ken in my scan pile, and so I would like to query my memory to see, in fact, exactly what the offer was and when it was made and all of that as a test of my own MyLifeBits system that holds my entire live.

Hendrie: Yes, all right, yes.

Bell: And then I know the original DEC offer is in there too. But anyway, I interviewed there at MIT, and I know I interviewed at EPSCO. 2008 Addenda: At some point, I interviewed at BBN with Ed Fredkin and JCR Licklider (Lick) and they didn't hire me. The reason I remember this is that Ed has quoted it to me several times. My point in bringing this up is that I have been extremely lucky in the choice of jobs and BBN was clearly not a good move for me. So I have been lucky in not getting a job.

Hendrie: Oh, with Bernie Gordon?

Bell: Bernie Gordon's company and so I basically decided . . .

Hendrie: So you basically came back with some ideas of where you might go and where you would be going?

Bell: Yeah, and I had some offers before that. I think Philco in Philly was a potential. I had interviewed there. I don't know if I had an offer or anything like that, but certainly GE because I was a co-op. As a co-op student I would have an offer there. I also had an offer from NCR that included a written intelligence test.

Hendrie: Now, it was Stevens at MIT; what was his position?

Bell: Okay, Ken Stevens, who is ten years older than I am, was head of the Speech Research Lab and he was my thesis advisor when I made the sound level recording meter for recording histograms. And actually it was Dick Bolt, of Bolt, Beranek and Newman [BBN], who had started me out. He had stimulated me to write the thing, and then Dick left MIT and Ken took over as my advisor and I finished the project. And then, when I came back, Ken hired me to work on speech, so my first real job was a staff engineer at MIT working for Stevens and then working on speech and eventually on instrumentation at MIT's Instrumentation Lab. In fact, what we started with was a filter bank and, I don't remember, there are maybe 18 or 24 filters. These are tuned filters and they were at 200 or 300 Hz wide or less depending on where they were in the band and that covered the speech spectrum. And then you looked at how much energy was in each of those bands and so you got essentially a time series of what the sound was. And basically that's the fundamental technique that almost all speech I think still uses today, which is frequency.

Hendrie: That's where the data is.

Bell: Work in the frequency domain, frequency versus time analysis. In fact, the thing that people normally see is a display called a sonogram, which is a map of the intensity of each frequency in the Y axis. So you see the darkness of the frequencies versus time. And so, you look at this plot and people can actually read sonograms and determine what someone is saying. I mean not a lot of people can do that. 2008 Note; I saw this used in Australia to single out rare birds in a rain forest.

Hendrie: But it is possible to do.

Bell: It is possible, and so given that, I thought, okay, well this will be great because, I mean, people can read this stuff. I'll work on a program that can read speech.

Hendrie: Oh, so you're saying, yeah, so . . .

Bell: So, I would do speech analysis, and we had the filter bank and then I worked on an instrumentation where you basically took a loop of tape with a saying on it. Oh, I think these were five second loops or something like that, and so we went around this tape recorder and the thing was you glued its end together or taped it together for a loop.

Hendrie: Yeah, right.

Bell: And then on the stereo track put a timing mark, which was the sampling time, and then used that sampling time to then….so you made 24 trips around the tape and in doing that you….then I had a stepping relay that would go and move to the next.

Hendrie: Move from filter to filter to filter.

Bell: So essentially you get a sonogram, a plot of frequency shown by intensity or blackness versus time. That was my first task and then we built and started working on speech analysis. And I don't know where exactly the idea came from but we invented a process which we called "analysis by synthesis", which is fundamentally a technique that people still use to recognize a phrase. Fundamentally it's still in use on things like you're going to make an ideal synthesizer for a piano and you want to get what the waveform is. So you basically take the waveform of a piano striking a string and then you take that and you synthesize what you think are the components of that, the perfect tone and then there's all the harmonics and any kind of distortions. And so, you basically start throwing synthetic terms at it until you get the same sound as the input.

Hendrie: So fundamentally it's the same spectral band with all the same energy levels.

Bell: Right. And so fundamentally you "analyze" the content by synthesizing content and then you know the parameters of what that synthetic term is. And then you can sort of apply that back to the vocal tract where you move the vocal tract parameters around, and then you see if that's what gave you those terms, those signals.

Hendrie: Oh, all right, that's fascinating.

Bell: So in a sense the technique recurses or . . .

Hendrie: It's a trial by error recursive just getting the errors smaller and smaller each time.

Bell: And so different ways of getting which errors, what error do you want to look at, and so that was the basic method of getting the parameters of what's going on. And then we wrote a classic paper on that because it was also being applied at the same time for some handwriting analysis.

Hendrie: Okay, oh interesting, all right. Do you remember what year or at what time you were doing this work?

Bell: Now that was. . . So I started at MIT in January '59, so part of the constraint here was Gwen had to finish her Master's degree at Harvard, so I didn't want to go too far from Harvard .

Hendrie: Yeah in different cities.

Bell: Right. And so, I figured MIT was just the right thing because there was a project orientation. I liked the people and got that program working. Then a little later I went over and worked in another part of RLE at the Instrumentation Lab and worked on a thing called pulse mode computation, or pulse hybrid computation using analog components for simple inner loop like multiply and add. This was totally a diversion that shouldn't have been because of the analog nature. It just happened to work on the TX-0 because it had no multiplier.

Hendrie: RLE?

Bell: Research Laboratory for Electronics.

Hendrie: Got it, okay, got it.

Bell: I think it's still named that or may or may not still be named that. But it was located along Vassar Avenue in Cambridge called Building 20 and these were World War II constructions, plywood constructions, were noisy and it was filthy. I mean lots of truck traffic and you'd go away for a few days and your desk would be covered with sand when you came back. But, anyway it was a great place.

Hendrie: So, what did you do there now?

Bell: So, then I went over . . .Well, I guess the important thing about the speech stuff research was that I thought, "Oh, my god, this is not for me. This is basic research and science." I like to build stuff. I want to be an engineer.

Hendrie: Yes, okay.

Bell: And I've done the easy part, which is to get a program that, in fact, could get to this point of recognizing, of giving you what the components are coming out of the program.

Hendrie: I can give you the data.

Bell: Yeah.

Hendrie: Now, we need a little science.

Bell: Now, what are you going to do with it, how are you going to look at that, and I did it to understand speech. I think maybe we had a few trial recognition programs and several components to segment, or extract pitch of or something because you know where the energy is, and then did some segmentation and stuff like that. And then at this point in late 1959, I said “this is just too hard.”

Hendrie: And it did prove to be too hard didn't it?

Bell: Yeah.

Hendrie: 20 years and decades went by before people figured it out.

Bell: My estimate was 20 years, but I had no basis for that estimate, and then it's been about 40 actually to get it to a useable general recognizer, although after 20 recognition was in use.

Hendrie: Exactly.

Bell: People were making products in 1980-ish or so. So, along about mid-'59 I then switched to the Instrumentation Lab headed by Frank Reinjes, a principal engineer who worked on radar at MIT’s Radiation Laboratory. I was working on pulse computation because it turns out the TX-0 didn't have a fast multiplier. In fact in the speech program, because it had multiplication involved in it, I did multiplication by table lookup of logs.

Hendrie: Oh, really?

Bell: Yeah, to get the speed and so . . .

Hendrie: And this program was written, ran on TX-0?

Bell: Yeah, right.

Hendrie: I didn't ask you what computer this ran on.

Bell: Oh, yeah, this all ran on TX-0, so I was a proficient programmer on TX-0 and wrote this fairly good-sized program for the day to do speech analysis. Others used it for several years.

Hendrie: And the instrumentation was tied to TX-0?

Bell: Right, and then that was all part of the speech lab. And then I went over to this other lab, did this project -- again I think it was the Instrumentation lab -- and built a multiplier that would allow you to basically do sample and hold, basically put values on a multiplier and then pick those off and then feed the results back in. And so it was called hybrid computation and it was for a particular program. It was auto correlation or cross correlation of radar data, so I had a lot of data and needed to do it.

Hendrie: Okay, so fundamentally the multiplier, it was a digital computer but you used an analog multiplier?

Bell: Yeah and I think it had an integrator too, so you could use . . .

Hendrie: Oh, my goodness, yes.

Bell: So, anyway, we were able to do radar signal processing in real time. There was a paper report published on it in August, 1960. And then at the time the TX-0 was coming up and they needed a director to run it and I interviewed for the job.

Hendrie: Oh, my goodness.

Bell: With Wes Clark. I think Wes was kind of "overseeing" the TX-0 from Lincoln. Jack Dennis ended up doing that.

Hendrie: Oh, he ended up getting to be chosen to be the guy.

Bell: Right, to run it and sort of be the faculty advisor and whatever.

Hendrie: Yeah, right.

Bell: In fact, it probably undoubtedly had to be a faculty, or should have been a faculty member 2008 Addenda: In retrospect I was really lucky to not get the job. I have been lucky to have been rejected by two jobs (see my addenda re BBN), neither of which would have been good for me. If I had accepted either one, it would have been a dead end.

Hendrie: The PDP-1 had not been built yet had it?

Bell: No, no.

Hendrie: So, they didn't have a PDP, Jack Dennis didn't have his PDP-1 yet?

Bell: Right, and that's how I got to DEC. In that process, Jack wanted a tape unit on it so I designed a tape controller for the TX-0.

Hendrie: Oh, all right.

Bell: And I needed modules for it.

Hendrie: Ah, yes, exactly. You're not going to do it from scratch. You just did logic design.

Bell: Right, so I ended up getting modules from DEC, doing a design, and that's how I met the people at DEC. There was one particular circuit that you "needed" that turns out to be very useful and it's hard to do any other way, and DEC was the only one who had it. It was called an integrating single shot so that every time you pulse it, it would start and take a fixed amount of time from the pulse input. So, as opposed to just a normal multi vibrator where you hit it and it is unclear how long before it stops.

Hendrie: It gives you a fixed amount and you can change the capacitor and it will do that.

Bell: This thing actually integrated and so from the last time you hit it it would generate a pulse.

Hendrie: Ah, so it integrated and depending on how long it was between characters on the tape.

Bell: Yeah, right. It was the restart of all of this that . . .

Hendrie: Ah, okay.

Bell: That was critical for a tape because you didn't have the sort of nice properties.

Hendrie: I see and 3C didn't have it in its module either?

Bell: It was a very tricky circuit and I think they had invented it for the tape, the same thing for the tape controllers at Lincoln Lab for the TX-2. It is also called a boxcar circuit.

Hendrie: Okay.

Bell: So, anyway, I did that and it turned out that just in that whole process I ended up getting a job offer in May of '60 from DEC.

Hendrie: Now who did you interview with?

Bell: I interviewed with Ben Gurley. Ben was the guy, my primary interface there because it was about building this tape controller. In essence, I asked him, “What do you think of my design?”

Hendrie: Right.

Bell: Dick Best was there as chief engineer from doing circuits, and then in that process I met Ken and Andy. I met the whole team and got a job offer and I was number two as a computer engineer working for Ben.

Hendrie: And you decided to take it.

Bell: I took it, yeah. It was exactly what I wanted to do.

Hendrie: You were going to build stuff.

Bell: I was badge number 80. It didn't have the problems that I had seen as a co-op student at GE where there was a sea of desks. Everybody had their own little office, which was uncommon at the time because I remember my friend Bob Brigham from Bell Labs was visiting and he said "Oh, I'm impressed". The offices were made of hollow core doors. You know DEC was in a civil war woolen mill building. One company was milling doors and we bought rejects for partitions.

Hendrie: Yeah, of course, with the lanolin floors.

Bell: And so a company that made doors was in the Mill Building and they'd just go down and buy a bunch of hollow core doors and put these doors up for partitions.

Hendrie: Oh, okay.

Bell: There weren't doors on the door partitions but they were sort of all cubicles. Anyway, so that's how I ended up getting to DEC and one of my first jobs I think was at . . .

Hendrie: But did you ever finish the Mag tape?

Bell: No, I didn't finish it.

Hendrie: Oh.

Bell: It turns out that Bob Spinrad who was at Xerox PARC still cusses me for having to debug my controller, and in fact he said he ended up as my technician to finish the debug. So, it got wired and ultimately worked.

Hendrie: It ultimately did work but you--

Bell: It ultimately worked but I . . .

Hendrie: You had bailed.

Bell: I bailed out, yeah, just in time. There's nothing worse than having to debug a tape controller.

Hendrie: Debug something that interfaces an electromechanical device?

Bell: Yes, and tape is by far the most difficult. In fact, of course, it's always the job you give to the youngest and least experienced engineer.

Hendrie: Right.

Bell: When a new person comes in, you say okay, what do you want him to do? A tape controller.

Hendrie: Right, it usually isn't critical about exactly when it gets done. It's never on the critical path. You just have to have one.

Bell: But, anyway, I think I designed one of the units there. Another friend, Jack Brown who recently died, I think ultimately debugged that one too. We had two tape controllers. One was a programmed controller. In fact, the TX-0 one was I think a highly programmed one and so it was a good experience. All these things were good experience in terms of the I/O. In fact, that's how I got such a fondness for the I/O and also doing as much as possible in software.

Hendrie: Yeah.

Bell: Because you could trade off program for work in hardware control.

Hendrie: Okay, so it was, yeah, so TX-0 did a lot of the work.

Bell: Yeah, and the PDP-1 had good interrupts so you did buffer assembly and then you sampled the data and then brought it in with a program.

Hendrie: Okay.

Bell: And then we made one like that for PDP-1. It ran through the direct memory, DMA channel. And then when so I got there in August of '60, I wrote some programs and worked on various logical design problems.

Hendrie: Now, had Ben finished the PDP-1 yet?

Bell: Ben had made the PDP-1 number one and it was sort of a brown speckled one with a remote console and it had, of course, a huge number of wires coming to the separate console. It was a separate console just like larger machines including the 7090 and the CDC 3600.

Hendrie: Oh, my goodness.

Bell: With a table and then almost very much like the PDP-1 console today but yet just remote, so you had drivers and switch filters and stuff like that to deal with the remoteness. The worst problem was the connectors were unreliable after you used them a few dozen times.

Hendrie: All the stuff you have to do.

Bell: And I think I was the one who said, "God, let's get rid of this console, get this thing back out of there and put it on the computer" because it had this big bundle of cables for lights and switches. Nothing was doing much and there was a particular connector on it. I learned to hate connectors too at that point. You'd open it and close it a few times and it was a leaf, one of these kind of leaf connectors and multiple pins and lots of problems.

Hendrie: Oh, wow, okay. So now were you told what you were going to do when you were hired?

Bell: Just different things. I know I wrote a floating point package for the PDP-1 and I also was sort of instrumental in starting DECUS. I said first off we've got to share programs among the users. There is so much to write.

Hendrie: Yes.

Bell: And so we got DECUS going, probably my guess is in '61 or so. There weren't very many users at that point until after PDP-1. I guess we made a couple of these remote console ones, and then we streamlined it and put the console on the end like the PDP-1s we have kept... BBN was delivered the second one.

Hendrie: And that became more the production model.

Bell: And I think it was probably number three that was that way and I doubt there's anything left except the pictures of the early one.

Hendrie: Of the early one. Now, at the time that the PDP-1 was originally built there was supposed to be a PDP-2 and a PDP-3. Now was anything going on with that at the time?

Bell: The PDP-2 and 3 or the three's . . .

Hendrie: I actually have a brochure for the 1, 2 and 3.

Bell: Right.

Hendrie: That's how I knew that from my . . .

Bell: The PDP-2? I don't have the brochure on the 2.

Hendrie: Yeah, well this is just a combo brochure for the line.

Bell: Okay.

Hendrie: 1, 2, 3.

Bell: Okay. The two was left to be a 24-bit machine. The three was a 36-bit machine, and we had a proposal out and there was an architecture for it and it basically was kind of a big overgrown PDP-1 with index registers. And anyway what happened was we had this proposal and we sold one of these.

Hendrie: A PDP-3.

Bell: A PDP-3 to the Air Force Cambridge Research Lab, U.S. Hanscom field.

Hendrie: Oh, my goodness.

Bell: And I don't remember what year that was. It might have been . . . it was probably in 1962 or 1963, something like that.

Hendrie: So you did actively promote this.

Bell: Yeah.

Hendrie: But you didn't build one on spec.

Bell: Right, there was none built. It was all clearly vaporware.

Hendrie: Yeah, well that was the way people did it in those days and still do.

Bell: Yeah, here's this computer we have a dream about. Do you want to buy it?

Hendrie: So, did one get built?

Bell: No, but I have a wonderful story about how we dealt with the order we had for the 36-bit PDP-3.

Hendrie: Yeah, I want to hear the story.

Bell: Harlan Anderson and I went over to talk to the guy buying it, Charlton Walters, who was doing speech work research, and on the way over -- I remember the spot on Route 2, just as you make that corner that turns going back to Boston and you keep going to Bedford -- we had an epiphany. Harlan and I were over there and I was going to be the project engineer on it. I, or maybe Andy said, “Hey, the company is taking off. We have no business building another ~!@#$%^&*()_+ computer. We can't do this. This is a lot more work than we thought. It's this stuff called software that takes so much work.” We were just beginning to understand what a computer company looked like and what you had to do to be successful.

Hendrie: Yes, and the other stuff you have to do.

Bell: I can build this thing but, oh shit, what are we going to do?

Hendrie: All the other, we got to support and . . .

Bell: So, I don't know. I think Andy or I, one of us, reinvented. Well, we'll give them the 36-bit machine, but it will be in the form of two 18-bit machines.

Hendrie: Okay, oh wow.

Bell: So, we go over and tell him, "Charlton, here's the deal. We want to give you two machines. It will give you a lot more power and it will let you break up your analysis thing. It will provide your redundancy. We'll make the connection between them so they can be used together."

Hendrie: So they can talk to each other.

Bell: Yeah, and "We'll give you all of this capability." So, we made up a little nice story about the benefits and were able to convert the 36 bit PDP-3 order into two 18-bit PDP-1s.

Hendrie: With some special hardware so they could . . .

Bell: Yeah.

Hendrie: So they could be programmed to work together.

Bell: Much less hardware than we would have had to build otherwise.

Hendrie: Oh, my goodness.

Bell: That was the PDP-3.

Hendrie: That's as close as the PDP-3 ever got.

Bell: Well, not exactly, because some company, and I don't know remember the company's name, actually took the spec of the PDP-3 and they bought DEC modules and they built a PDP-3.

Hendrie: Oh, my goodness.

Bell: They didn't understand software either, or the curse of having a one-of-a-kind computer.

Hendrie: Oh, wow.

Bell: So, there was one PDP-3.

Hendrie: It existed.

Bell: One existed. I went over to see it.

Hendrie: Do you remember where it was?

Bell: Yeah, it was in Waltham along one of those streets parallel to 128th on the east.

Hendrie: Oh, okay.

Bell: I don't remember the company's name but I remember seeing it.

Hendrie: Okay, that's pretty interesting.

Bell: Right. But fortunately, we got a big order from IT&T that allowed us to continue and grow. And then, I don't recall when we made the deal, but it was sort of halfway through oh maybe the summer of . . . it would have been '61 or so because I know it was the new blue machine. We made a deal with ITT which was to build them a switching system called the ADX7300 and that was to replace a torn teletype tape switching system.

Hendrie: Oh, a torn tape teletype switch, ah.

Bell: Right. And interestingly enough I always wondered about if you ever ran two torn tape systems together, that is whether it totally spoke, whether just one torn tape system in the whole world or whether you actually had Europe and the U.S. in which case packet switching would have been invented. Thus it was invented many decades earlier than the fathers of the Internet ARPAnet claim.

Hendrie: Maybe you could explain what a torn tape system is. I happen to know but I don't think people reading this or watching this tape do.

Bell: Yeah, anyway, back in the days when people sent messages by teletype, you basically either talked on the teletype directly or you made a tape with the teletype. And the code had all evolved from

Morse code, but anyway there was a code on the tape and the code was in that case a five-bit thing called a Baudot code. Those five bits were used to encode 32 characters, and you had the upper case alphabet and some control characters and stuff like that. And they were read in parallel and then each character was what you had typed. And so you had a tape reader to enable you to send the messages stored on the tape. Instead of calling everyone directly, you put a message address on the tape that you first sent to New York or another switching center. The sent tapes were repunched at the tape reader. So I want to send a message from one place to another so I basically send it to a torn tape center. A new tape is created there. At this center someone read the destination address, took the tape to a reader at the center, and sent it off to the final address. This is similar to the way FEDEX works -- all messages are collected in trucks and flown to Memphis, are sorted in Memphis, and then flown back to their destination and delivered.

Hendrie: Now, this is a switching, this is really a switching system.

Bell: This is a switching system.

Hendrie: But manually the switching process is very manual.

Bell: Right, and basically on the header of the message you say, "Well, I want to go to somebody" and the way you did that was that a tape would go to the center and the tape would be labeled in some way where it's going and that that tape is for a type M machine and so on. So you'd basically just take the stack of tapes to a particular destination reader and you'd cue queue them on up for output.

Hendrie: And you'd send them out on a line going . . .

Bell: On another line.

Hendrie: On a reader. They come in and get punched.

Bell: They come in and are punched.

Hendrie: Are punched. Now you have the tapes and then you move it to somewhere.

Bell: And then you got the tape and then you move that to a station that’s going to read and put it out on a telegraph some other place.

Hendrie: So, it's a one stage packet switch by definition?

Bell: Right, exactly. It is a packet because . . .

Hendrie: The packet is the tape.

Bell: That's right and I don't think there were any practical limits on how much tape you want in your message but these messages were short. These were telegraph like messages or Western Union kinds of messages that you paid by the word. And so, basically ITT -- International Telephone and Telegraph is what it stood for at the time -- and John Ackley, who was an MIT classmate of mine, was the project engineer and was pushing this. He was in New York and we made a contract to basically build this switch so basically a tape could come in or the stuff would come in to the PDP-1, was stored, queued, and then sent out to the appropriate line when it was free. It is, of course, nothing now as that's what all of our hubs and routers do on the internet. We got rid of all the paper tapes and operators on roller skates moving tapes about in New York and other centers.

END OF TAPE 1, INTERVIEW 2

Oral History of Gordon Bell

Interviewed by:
Gardner Hendrie
Corrected 5/11/08 cgb
Recorded: July 19, 2005
San Francisco, California

Interview 2, Transcript 2

CHM Reference number: X3202.2006

Bell: I was the project engineer for the ADX-7300 for IT&T. And what that entailed was basically handling the telegraph. Well, two things -- one, handling the telegraph lines, which was taking in the 5-bit characters and then which were transmitted serially and then assembled into a parallel, 5-bit parallel character, and then brought into the computer and stacked placed in a word. And so that involved two things: One is the receiver and the transmitter on one of these lines; and then the other part was once it got into the computer, then multiplexing those in separately and putting them into the machine. So you essentially have all of these 5-bit characters arriving, and that was just sort of a standard multiplexing to bring those in and there was a kind of a multiplexer there.

Hendrie: Did you assemble them into 18-bit word and then...

Bell: No.

Hendrie: So, they came in one 5-bit character per word initially?

Bell: And they were then assembled, program assembled.

Hendrie: Program assembled, yeah.

Bell: And because they were coming in at 75 or 100 bits per second, so you've got a lot of time.

Hendrie: Now, the PDP-1, what was its performance?

Bell: It was five microseconds cycle time, so instructions took either five or ten microseconds.

Hendrie: I just wanted to get the perspective there.

Bell: Right, that's the spec on the PDP-1. And then it had memory size, the basic size was 4K words, the starter memory was 4K and then it ultimately got up to I think 64K words. And what we had just developed had been, well, we called it a sequence break system which was actually a program that was an interrupt. So, at the time, the basic machine had only one level interrupt. So anything would happen on the outside. You'd go and take that, whatever it was, and then interrupt and put it into. . . and take care of it in some fashion. In the case of this, we had 256 lines coming in, so we had a bit of multiplexing to do to get the 256 lines to all come in there and put them in some kind of order. So one of the options was a 16-line interrupt. And so I extended it to be a 256-line interrupt, so there was essentially interrupts on interrupts.

Hendrie: Okay. Then you had to do its software. You had to deal with the...

Bell: Yeah. So an interrupt would occur and then you'd get a number and say, well, I think we went to 16 places and then within those 16 places...

Hendrie: Go 16?

****Bell:** We had four places that you could go. So we had a unique place to go once an interrupt occurred. And so there was that mod to extend that to a greater, bigger interrupt system. And then, the interesting part was...

Hendrie: And you did this mod in hardware?

Bell: Yeah, it was all hardware.

Hendrie: It was all hardware mod.

Bell: Yeah, and then you wrote the routines to handle whatever is going to be handlable then. And then the other part was dealing with the telegraph line. And that was when, quote, I invented the UART.

Hendrie: Oh!

Bell: And the UART, people had been using single shots to that to deal with . . . I mean, that was the way it had been done. When we finally found out about them, people said, "Well, these don't work very well because you've got these analog potentiometers to set."

Hendrie: When you found out about the single-shot approach?

Bell: Yeah, right. And so I basically did a sample clock approach. And that was the method that is, in fact, in use. So you basically break the line up into a multiple of either 7 or 8 samples and then you take a sample from the time, you take the transition, and you go through a little state system to sample this serial telegraph line. And then you convert bit by bit, put that into a parallel buffer and then take it in. And in a telegraph line, as I recall, there were various codes used. But for whatever reason, there was a 7.42 baud, bit -- bauds were the actual bits on the line of which you got five good ones. And so one was the start character and then you had . . .

Hendrie: So there was a starting, a pulse of energy that was the start?

Bell: Six. Right. So you had 2.42 or others. And so we ended up with rounding that up to 7.5 as the signal. So basically you had a start 5 and then 1 and 1-1/2 baud at the end as a stop signal and then you started it again. And so in a sense it would kind of resync based on the next start bit.

Hendrie: Whenever the start came.

Bell: Whenever the start came.

Hendrie: And then it would just clock along.

Bell: Right.

Hendrie: You knew where the bit times were based on when the start was, and you'd sample through this and find out what the bits were.

Bell: Right.

Hendrie: Now, did you design a special module to do this? Or did you just combine a bunch of logic?

Bell: Initially, that was done with just logic.

Hendrie: That's a lot.

Bell: We had 50 modules which got us eight lines, 50-deck, 5000 series, that's a 500 kilohertz clock.

Hendrie: Modules.

Bell: Modules got us eight lines.

Hendrie: All right.

Bell: And then when the PDP-1 or PDP-5 came out, at some point I'd been arguing for a long time that this ought to be a module. In fact, of all the things I've kept as parts, mostly everything I've really given to the museum is because I just don't have enough space to put the stuff. And I do have some PDP-6 modules, and I also have a receiver and a transmitter. And these were extended module. So we got the idea of making these long modules.

Hendrie: Well, so they would be long in the sense they had multiple connectors on them or they stick way out?

Bell: No, they stick way out, because you didn't have a connector. You didn't have a pin problem, you had an area problem because they had 8 bits. They had 8 flip-flops and clocking and the counter and this other was maybe 15 bits or so of...

Hendrie: A lot of transistors of flip-flops on it?

Bell: Flip-flop. Yeah. And so that was put on an extended module. Ed DeCastro stuck that in the PDP-5. I think we made it for that just because we wanted to keep the size down, although I think by now it was like 6 or 8 DEC modules, a few standard modules. But it was still much more elegant just to plug these in. And people loved them. You know, people who were building their own systems loved them. And that's sort of how I fell in love with the serial communication, because in the process of building and testing the lines and stuff like that, I wrote a test program which had basically simulated this message switching system. And we had a bunch of old telegraphs that IT&T had set up and so we were playing with these things all the time. And so I got to love these things because they were so reliable. This is in contrast to the Soroban, which was an IBM modified typewriter that was used on the PDP-1. But it was a full typewriter, beautiful typewriter. But it was constantly breaking down because it was being driven, you know, the computer was driving it faster than any human could ever drive this thing.

Hendrie: And it wasn't really designed for that?

Bell: It wasn't designed for that.

Hendrie: So you did not use Flexowriters, which was the other choice in that era?

Bell: We had never connected a Flexowriter -- we used Flexowriters for offline preparation. But Flexowriter, you know, again, was not a preferred...

Hendrie: I know. Well, your competitors used Flexowriters.

Bell: Yeah. And they were both...

Hendrie: They were both bad choices.

Bell: They were both bad. In fact, I think we connected the first, used the model 28, which was a 5-bit Teletype, probably the most homely thing you could imagine. We painted it blue and that was what I used on the PDP-4, because I wanted to get rid of the typewriter on the PDP-1.

Hendrie: With something more reliable.

Bell: And I think -- oh, geez, I can't remember -- I think actually we had lucked out at that. We lucked out on the PDP-5 just on the 5 transition that the teletype ASR-33 had come out. I'm quite sure that's what we used. And it was right at the cusp of when it was coming out, and that's what Ed put on the PDP-5.

Hendrie: And this is what you used?

Bell: And so it had the virtue of making the thing low cost. You got a tape reader. You got a tape punch. You got an I/O device. And so from a computer standpoint it's ideal. You've got the whole works right there in one little package. And it was pretty reliable. It wasn't as good as the older Teletypes, which were really meant for heavy duty use. But, in fact, it worked quite well. And that's the device that was used for a long time until actually that drove them out of business with the LA, with the Decwriter, which is a matrix printer.

Hendrie: All right. Let's roll back. Yeah, let's rollback to this project that you're working on.

Bell: Yeah. So fundamentally, the project was completed.

Hendrie: Now, how long did this take you to?

Bell: I don't know maybe six months or something like that. And meanwhile, you know, the company was developing. Half of the PDP-1s were sold to IT&T under the ADX label. And, in fact, it was one of the key things that I'd say "saved the company" and made it a computer company, because just getting enough orders there was the key thing. Up until that time, the PDP-1 brochure -- which, you know, we have copies of -- had every possible device that you could imagine in there. And we ended up selling one of each one. And Lawrence Livermore bought this for an offline device, essentially kind of a 1401 pre-processor for 7090. And it had the tape units, of course. And it had a Remington Rand tape unit, because nobody used Remington Rand. And so we had made a special controller for Remington Rand. It had a high precision scope and a camera and this was a 4000 x 4000 point plotter. Absolutely beautiful scope for film -- actually, we had a film reader on there -- and a scope would be used for reading and writing. It had a connection to -- which we didn't do -- but a connection to a high-speed card reader. And I can't remember who made that card reader. George Michels of Livermore arranged to buy it...

Hendrie: Being George Michael, yes?

Bell: I remembered seeing it on the top floor of where all the debugging was taking place. And this poor guy was there from the card reader company and debugging logic on the PDP-1, and things would go awry. So occasionally you'd hear this screech and you know the cards were all jammed into this one little space. And so we kind of renamed it the high-speed card folder.

Hendrie: So, basically, you had all the data processing?

Bell: Yeah.

Hendrie: You had to do all the standard data processing peripherals sooner or later?

Bell: Yeah, so Teletype was an important company because that interface there got us used to Teletype. And another thing that we removed was we had initially a thing called a Tally punch, which was the first punch of the PDP-1. And then again, it was an errant device. I mean, it was like, not a nice device. You know, mechanically it would get out of line. It was asynchronous. And so we were delighted when we could buy the Teletype BRPE punch, which is on the PDP-1. Probably still in op. I haven't talked to the PDP-1 restoring guys, but I think they got that working. They needed belts of course, but it was a beautiful punch.

Hendrie: Teletype made remarkable electromechanical equipment that was really reliable and...

Bell: Yeah, it absolutely worked.

Hendrie: They had great mechanical engineers. So, you've gone and you've finished this switch for ITT. Ironically, DEC's matrix printer, the LA36, wiped them out though.

Bell: And then that's about the time we met when I started working on the PDP-4. And I think you triggered the PDP-4, which was to say, "We want a control device." You were working for Foxboro at the time, and you wanted a control device for a process control computer. And we said, "Yeah, we can make one of those." And I became the project engineer of the PDP-4.

Hendrie: All right. So that was the next thing you did?

Bell: Yeah, that was exactly the next thing I did.

Hendrie: And who decided what the PDP-4 was going to be like? Is this Andy?

Bell: No, I did. I don't know how the decision was made, but among other things that went into it was that we had gone from 5 Megahertz to 500 kilohertz modules. And 5 kilohertz were good enough to make the computers, so we didn't need these faster modules. And then also they were lower cost instead of transistor logic, it was transistor capacitor logic. And so, you operated at a lower speed but at considerably lower cost for the modules given the cost of the transistors at the time. And for whatever reason I decided that it wasn't going to be PDP-1 compatible. This was before I understood about programming and programming cost and investment. Which, by the way, there are still people who don't understand that. That is probably the most misunderstood concept in computing and in digital computing today. People just don't understand the value of software, the integrating effect of software, and why you, whenever possible, use somebody else's interfaces where there's an installed, or where there's a collection of software that you have access to or that you can get. Changing the instruction, changing the architecture is always tempting for whatever reason, and it's virtually always a bad idea. And in this case it was a bad idea. And there are lots of reasons why it was, "Oh, well, I hate ones complement arithmetic." It turns out that in PDP-4, it was going to be twos complement arithmetic. So floating point works a lot better and stuff like that.

Hendrie: Okay. So you made that change.

Bell: Yeah, I made that change.

Hendrie: From ones complement to twos complement.

Bell: I had made these this idea for very inexpensive index registers, called auto index registers, which you could indirect to memory and bump up the count. And so you basically can index through something in not too bad a shape. And so you save an index register. So I didn't need an index register and all the attendant logic.

Hendrie: Was there an index register in the PDP-1?

Bell: No. No.

Hendrie: No. There was not an index register, so this got over the marketing problem.

Bell: Yeah.

Hendrie: Your competitors had index registers.

Bell: Now we've got to put index registers in. And then there were a few other things that were "better" in terms of how it worked. I think one of the things was I gained a bit with all this fooling around and so it could directly address more memory.

Hendrie: Okay. Because you had shorter instructions, fewer instructions, plus you could encode them into -- because you were not compatible -- so you could fit them into a smaller number of bits and gives you another factor of two more memory.

Bell: But you still had to use bank switching to go beyond the 8K. And, we thought, oh, well, nobody would want more than 8K of memory at this point. And so, the first application, we can put in a bank switching on it. But anyway, I don't know how long it took me to realize this one point about the value of an installed base of software.

Hendrie: I don't think anybody understood it at that period.

Bell: Well, one of the quotes that Wilkes made, and I don't know when I heard him make it but I thought it was very early, he said “It wasn't very long before we began to realize the value of the software that we had created would vastly outweigh the cost of the machine”. And, you know, it's one of those things that's absolutely true. I was with a group a couple of days ago, and Chuck Thacker and I were there advising them about stuff. And then it was like, okay, we're going to use one of the CE versions for font. And they said, "Well, we just want to make this little change." And Chuck said, "Do not touch that. You cannot deviate from what Microsoft expects as a platform, because you cannot afford to do that." We've got, you know, there's one other very large PC vendor. He said, "I will not recommend their stuff because they go off and they make something that's 'better' and it turns out they can't track the software and can't support it and all, and they never get it right. And this match between hardware and software is one that you . . . there's just no tolerance for ambiguity there." You know, I think, probably kind of at the root of maybe the Jobs decision to use Intel. Well, they used plain old PCs.

Hendrie: I want to go back to one other. I didn't quite understand the difference of the upgrade of the modules and the memory or what trend changed there between the PDP-1.

Bell: The thing that made the sort of the rationale for the PDP-4 was that we could, in fact, make a machine that was half the cost of the PDP-1. So that was the rationale.

Hendrie: And so what modules did you use?

Bell: And the 500 kilohertz modules had just come out, which increased the density and reduced cost.

Hendrie: Oh, they were denser also?

Bell: They were somewhat more dense. And I don't remember where all of it came from. But in fact, the bottom line was that I was able to get the PDP-1, PDP-4, I think in something like one bay of seven or eight "crates of 25." I believe they were 25 across, versus three bays that I think were like 12 high.

Hendrie: And 25 across?

Bell: Yeah.

Hendrie: Still the same?

Bell: So it was a huge . . .

Hendrie: Reduction in the card count.

Bell: Reduction in the card count. And I think...

Hendrie: Were they 1 Megahertz on the PDP-1?

Bell: Five Megahertz.

Hendrie: Five Megahertz and you went down to 500 kilohertz modules?

Bell: Right, and then at a clock speed increase of -- again, something you don't do -- the cycle time went from 5 microseconds to 8 microseconds. Not a good idea. Kind of every time DEC came out with something that was worse and cheaper, it didn't really work out very well. Cheaper and faster's always good, but not . . .

Hendrie: But cheaper and the same is the furthest you dare go. You never want to go with cheaper and slower.

Bell: But anyway, the PDP-4 turned out to be the resulting architecture that went into the 4, the 7, the 9, and the 15.

Hendrie: Yeah. It had a long life.

Bell: It had a long life. But nevertheless, it probably should never have been. So what you have in this interview is a confession of my screw up. And if I'd just implemented the PDP-1 in these 500 kilohertz modules and probably stuck in some more instructions, you know, we probably would have done just as well.

Hendrie: But that's what you did.

Bell: That's what I did. And there were a bunch of ideas in there. The auto index registers and then the way subroutine calling was done. There was a trap that you could define, user defined instructions. And that idea basically came from Atlas, called Extra Codes, so that you could go in and trap some place and create another op code. This came from Atlas and we went on to use it in the PDP-6.

Hendrie: And you had in mind that maybe this would be useful in doing some of these special applications like building things like the torn tape system or something you didn't even know what it was.

Bell: Right. And that those were ways of call. You know, I'm having trouble remembering exactly how that manifests itself in the software. But that was a key way that we ended up doing "defining" more op codes.

Hendrie: Did you make any improvements in the interrupt system from what you learned?

Bell: I'd say I simplified it quite a bit. And then the other thing that was put in there was the ability to go in and index register, index memory so that there was certain process control, you would have an interrupt and it would go and you could dispatch to a certain place and then auto index that for doing histograms. And that ended up to be very valuable for the physicists when they were making counters.

Hendrie: Or pulse-height analyzers?

Bell: And the other thing was that direct memory access control logic was added. I'm pretty sure it was done there, but I don't remember when. There's the 4, 5, and 6, and there were a collection of ideas in the I/O system that all came out of that.

Hendrie: Exactly when you put DMA into the...

Bell: Well, DMA was in the PDP-1, but that was the device specified the register, specified where it wanted to go and the data. And in the 4, I can't remember whether on the 4 and the 5 exactly how those were done. But the 6 I know was done so that all the device state was in memory, so that basically a device would say, I've got a word to go in memory and it would go to a particular place and it would increment -- it would decrement the count and increment the address. And it would basically put it into, with two cycles, you could go into memory. And so I think there may have been three cycles. You do the same thing in a word count and a pointer in PDP-4. So the purpose of the control logic was to eliminate the need for block control in the i/o controllers.

Hendrie: Okay. And so you've got an interrupt.

Bell: I think maybe the 5 worked that way.

Hendrie: And then the computer set up where to go?

Bell: Set it up so you could make very low cost, fast block transfer access….

Hendrie: Devices.

Bell: Devices. Yeah. Because they were built to have lots of devices and to be able to handle them nicely. So those were all I/O refinements, interrupt refinements that made DEC a very desirable machine for connecting stuff. And then I think the key thing on the 5 was realizing that these should have been busses, and making the 5 be a bus architecture for I/O. So that made it still easier to get stuff. And that was kind of an I/O bus and then there was a memory bus, because otherwise on the 4, where you had the commit space multiplexers, in fact, we had a very elegant way of putting stuff in and patch paneling it in. But it was still fanning in devices rather than having a bus where you basically paid nothing for the I/O until you connected to it

Hendrie: For the multiplex. It was a distributed bus, a distributed multiplexing scheme is what it fundamentally is.

Bell: Right. And so that idea was used in the 5, it was also used in the 6. The 1 and 4 had been radial.

Hendrie: Okay. Good.

Bell: And then, in fact, I extended that notion. I think I had two big ideas -- the other one was the Unibus. And the Unibus was extending the bus even further, so that you have everything on a single bus. And that was, by the way, invented at Carnegie.

Hendrie: Oh, really?

Bell: Right. The interesting thing is, when you invent something, you can remember exactly the day and the circumstances of . . .

Hendrie: Of where you were.

Bell: You know, I was talking to Ted Codd of IBM who invented the database. And he was our contract monitor. He was monitoring an IBM research contract at Carnegie. And he was asking me what I was doing. And I was writing the book with Allen Newell that we called Computer Structures.

Hendrie: Yeah.

Bell: And so I described sort of this general structure of, okay, here's the selected set of switches that are linking memory to processor to I/O to this periphery. And so you had this sort of fanning out, and then I said, "Well, gee, a more general structure is, I just have switch and everything connects into that, and then you can have this bus that…" you know, and I was sort of explaining it to him.

Hendrie: And it just came to you by your explaining it to him?

Bell: It just came. Yeah.

Hendrie: I think when you explain to people like that, you try to figure out, you're trying to explain the fundamental of what's going on and in trying to do it, you suddenly get the insight.

Bell: Exactly. And the general registers for PDP-11 came out of this kind of explanation at Carnegie. I was talking to Alan Perlis and talking about the use of registers and the next registers and accumulators. Everybody was oohing and ahhing about the B5500, B5000, and I said, "Oh, you know in the 6 we were able to create a stack as part of the general register structures." I was trying to remember just how far we had gone there. We had not really gone to a stack or rather put the registers in a stack. We'd stopped just short of that. And in the case of describing that to Alan Perlis at lunch, I said, you know, "They should do everything. You should have the program counters, you should have the accumulators, you should have the index registers, the stack pointer, everything, and its how you control them."

Hendrie: Determines their use.

Bell: Determines their use, not having special ones that you can't make fungible. And again, then I worked with a student, Harold McFarland, who ended up doing . . . You know, he's one of these students that were basically there and sat in class and was designing computers all the time. And so, I worked with him, and then he took a job at DEC one summer. But the PDP-11 is another story. But I think the important next story probably is the PDP-5 or maybe finish the 4.

Hendrie: Yeah, why don't you finish the 4.

Bell: The 4 was designed for process control, and I think mainly it was with you and one was for a large Nabisco bakery in Chicago.

Hendrie: Yeah. I don't think I was yet at Foxboro, because I remember I got assigned to decide what computer we were going to use. Foxboro had not acquired the RCA Computer Group yet when the 4 was started. So the 4 was in the works and you had maybe one customer. And well, I think we should probably change the tape before we get into that one.

Bell: Okay, sure.

END OF TAPE 2, INTERVIEW 2

Oral History of Gordon Bell

Interviewed by:
Gardner Hendrie

Recorded: July 19, 2005
San Francisco, California
Edited by cgb 6/14/08

Interview 2, Transcript 3

CHM Reference number: X3202.2006

Bell: Right. During the interim we had talked about another thing, the Foxboro decision to use the PDP-4. And Ken's reluctance to put a drum on it probably stemmed from the fact that we had put a drum on the PDP-1 for BBN. Anything that had a drum on it was probably suspect. But we had made and, in fact, I did the original design of this magnificent drum for a PDP-1 that was a swapper so that in one cycle you could read a track and write the track into the memory and write memory onto another track simultaneously. This was by no means a trivial feat just because you had to synchronize the core with the slots on the drum and the slots or words had to be swapped.

Hendrie: Did you actually do it with one memory access for each word?

Bell: Yes, right. You would read out one word, put it on the drum, and then write back another word from another track on the drum.

Hendrie: A different word.

Bell: Yes.

Hendrie: So it was really efficient in terms of memory.

Bell: Oh, yeah. You could do a swap in one rotation.

Hendrie: It was literally a swap. It wasn't like you'll read the whole memory out and then you'll turn around and write it back.

Bell: Right, because this was for timesharing. You basically wanted to be able to swap an entire user's 4KW working memory in 1/30th of a second and have another user in place. You could move somebody out and somebody else in and . . .

Bell: And you did that for BBN.

Bell: Yeah.

Hendrie: Who at BBN was writing a timesharing system?

Bell: This was right at the beginning and in fact it was one of the . . . there's a whole story, the whole collection of stuff around the PDP-1 and around timesharing at that point because timesharing was just hitting in those early '60s. MITs' CTSS that Corbato designed for the IBM 7090 had started to work. We had the swapping drum that we had created for BBN and then it was also delivered to MIT.

Hendrie: Jack Dennis, did he buy one too?

Bell: He bought one.

Hendrie: And he did timesharing.

Bell: Yeah, and that was also . . .

Hendrie: The same drum.

Bell: And so at that point we sort of said this is hard to do because you're trying to read and write, you've got a lot of heavy currents going on everywhere for reading and writing, and so it was a high noise environment.

Hendrie: So it was hard to make the equipment work.

Bell: Right, and so we said okay, if you want just a plain, old drum we can do plain, old drum, and it was refreshing just to get a drum that you read or write to but didn't have to do swapping because swapping added a lot more registers because you had two parallel channels going -- just read or wrote, but not simultaneously.

Hendrie: Exactly. Simultaneous, synchronized, yeah.

Bell: So the drum really was part of the whole timesharing experience to make memory swaps fast. In fact, those drums were the device of choice when you're making timesharing systems.

Hendrie: So that was a . . .

Bell: The SDS 940. You used a drum and . . .

Hendrie: I think everybody did because this was prior to widespread disks.

Bell: Yeah, and that was . . .

Hendrie: That's what there was.

Bell: Yeah, and I think the original CTSS [compatible time-sharing system] used a drum and they swapped 32 k words on a 7090.

Hendrie: So that accounted for Ken's drum reluctance to put a drum . . .

Bell: "I don't want another drum," yeah.

Hendrie: There were a lot of headaches with that. I didn't realize you'd done the swapping drum, that that ever came up as one of your projects.

Bell: Right. Yeah, I'd forgotten about it too. It was part of . . .

Hendrie: The stuff you did during the PDP-1, time you were working on the PDP-1.

Bell: Right. And the other thing was that then BBN bought a large PDP-1 for doing hospital control. Jordan Baruch was making a real time hospital control system, medical control system, and in that case we had a drum swapper on it, and we also had disks because it needed a huge data base, he needed a data base there. This was in 1963, '64 period I believe because the PDP-6 had just started and the memory system was built to test the PDP-6 memory modules.

Hendrie: But he was doing it on a PDP-1.

Bell: He was doing it on a PDP-1, and in a way we debugged some of the memory bus stuff on this machine because it was doing high speed I/O into the memory while we were also running a processor in the other memory bank. And that was part of the PDP-6 -- it had that multiplex parallel memory structure.

Hendrie: So you had independent memory banks and you could be DMA or doing some controlled I/O while you were running code out of the other one.

Bell: And that was kind of where we first used that before it came out on the 6. Let's see, back to the 4 . . . But, I think we may have finished the 4, so ultimately it got out to Foxboro. Jay Forrester was on the board, and board concern at the time was about whether we are liable if this computer stops and dumps flour into the Chicago River. It was at the Nabisco baking plant.

Hendrie: Yes. That was one of our customers at Foxboro.

Bell: There were two that I think were kind of targets, that we wanted to make sure we could do. It may have been a power company. Now I recall the other one was Corning Glass.

Hendrie: Yes. We did some power companies and I remember Nabisco also.

Bell: And then . . .

Hendrie: The board did not realize that in some sense none of these were direct digital control. These were always just setting set points on analog controllers so there was almost always, if there could be, big trouble in River City. There was some backup control system in the analog domain -- classically it's the catalytic cracker which can explode under certain conditions, but there's a slide valve that comes down and breaks up two parts of the reaction, keeps the catalyst away from the petrochemicals and the most known number of refinery mega explosions have occurred when the slide valve stuck. You would never have a computer control the slide valve.

Bell: I was trying to think of any more on the 4's. And then I think the interesting part of the 4 was when we were invited to go to Chalk River Canada to bid on a reactor control computer. Ed DeCastro was in special systems at the time, and I don't think anybody else went to Chalk River and talked to them about the control, about what they needed in this system. Chalk River was a system that we sold. We sold a PDP-1 there for pulse height analyzers and counters and stuff like that. Ed was there to say okay, what is it you need, how many scalers, what, how many x to do the special front end system design.

Hendrie: Yeah. They built a whole bunch of special front end equipment.

Bell: So there was a fair amount of special hardware, and the spec was something like, well, we're assuming the PDP-4 was going to fail on this I/O and so this other part can't and it's got to be maintaining a certain state. So on the way back I said let's build a tiny, tiny, tiny computer, a 9-bit computer, that's going to just maintain the state, just have the program that's going to do all of this stuff and then feed it over to the 4 and vice versa for control.

Hendrie: Rather than doing it with hard wired logic which was the . . .

Bell: The normal way you did the job.

Hendrie: Yeah, special systems had done memory testers and testing systems like that.

Bell: Oh, yeah, and we had very complex systems designed . . .

Hendrie: . . . with hard wired logic. DEC had a lot of experience in doing complex hard wired logic.

Bell: That was the way you designed digital systems before we realized the general purpose computer can do everything. The computer industry got started that way simply because the computer companies would go off and make the integration before there were sort of system integrators that would do that same function. So we were driving back that evening to the airport and it was, I remember, very, very cold and we were sort of sitting in the car talking about this, and I said this ought to be a computer and we got back and started the design. And then as it got evolving, it took on this character and went from 9 to 12 bits and in the process even stopped at 10 bits.

Hendrie: Do you remember why it did that?

Bell: No. It was like, well, it's only 30% more cost and we get so much more out of it than just having it be barely able to do it the job. We feared we might have run out of memory or something.

Hendrie: It had a built in A to D, didn't it?

Bell: That's right. This may have been the reason we went from the 9 to 12 bits. The way we got the cost down was it had a built-in A to D using a combination of hardware and software. The accumulator there had a D to A attached to it and a comparator to incoming analog signals. So with the program you basically went off and successfully used that to generate the signal and did a compare and, using various forms of trial and error, determined the input signal voltage. So it was very inexpensive and in fact we did everything to take cost out of the design. The program counter was in memory and so it had just one accumulator. That was the whole state of the machine. We used some of the ideas that were in the 4 in there, like auto index registers in memory. Also, it used a bus for easy I/O connection, unlike its predecessors. And it used 2's compliment arithmetic that made multiple precision work better.

Hendrie: You'd already figured that out on the 4, the arithmetic. Did you know anything about the CDC 160 at this point?

Bell: Yes. The CDC 160 was a 1960 machine so we had that to look at. I'm pretty sure we had the LINC to look at. So we had a bunch of computers.

Hendrie: Lots of places to steal . . .

Bell: Yeah, exactly, and then in the case of the 160 the big decision really was rather than having an address that was relative off of the program counter for short addresses or to have them in banks, and so we went to a bank structure. I think there were 128 word pages. So in a funny way they were very small, little pages that you flipped in and out, and I still don't know whether that was a good decision or not overall. I think overall it probably was. It gave you these places because later on there was a huge amount of swapping that went on in these small pages, and so they were used in a funny way as overlays which would have had to be done in the program and the boundaries may have been a good thing to save, but that was the key on the PDP-5. It was a nice way to think about programs.

Hendrie: That became the PDP-5. Wasn't it called the Program . . .

Bell: Data controller.

Hendrie: Something else, 12, the 4, before it became an official computer. I think I have a brochure for it being called a . . .

Bell: A controller.

Hendrie: A controller 12.

Bell: Right. We knew it was a computer. And then it was officially named the PDP- 5.

Hendrie: It came into the series of DEC computers.

Bell: Yeah, but it wasn't very long because in fact Ed had designed it as a computer and with all of that capability and then, I don't think he did it, but somebody else did the logic that had to do with the work at Chalk River. But essentially these are sort of two instances. The 4 and the 5 were really good instances of this. You're designing for a target customer, you know what it is you want to do, and they were good, important exercises to be able to do.

Hendrie: But then you tried to generalize it so you don't . . .

Bell: It's good to have them demanding in some regard so that you're able to cover the whole space. That gets up to the 4 and 5 and then now the 6. I'm not sure exactly when the 6 started. I've got my notebooks and I ought to go back and look at when the first . . .

Hendrie: When the first scribblings of 6 or what it looked like or . . .

Bell: Right. When or why that was done. I'm sure Andy triggered it in terms of "Gordon, go off and look at building a serious machine," and then it was a bigger machine, a machine designed for timesharing. So we'd been playing with timesharing and then the MIT CTSS system was up.

Hendrie: And people had been taking about them.

Bell: Yeah, and our early machines were in a sense way too small to be worth time . . .

Hendrie: Worth timesharing.

Bell: There wasn't much to share. They didn't have floating point and they had very little computation ability and then limited IL because of the memory structure. So the idea was to improve on all of those dimensions, so a lot more bandwidth, a lot more expandability, a lot bigger machine, a real computer that can handle numbers. That would have been probably in '63. I can probably pinpoint it to some extent because I remember working on the instruction set architecture and op codes at a conference on nuclear instrumentation at the Naval Postgraduate School in Monterey where we described the work that had been done in Chalk River. John Leng, who was ultimately the division head of PDP-10 and had started to work for DEC, and I coauthored -- or rather he wrote it and I was a coauthor of -- a pulse-height analyzer for a PDP-1 that had been put in at Chalk River. It was one of the first papers on building pulse-height analyzer using computers. So I can find out when it was. The conference was in Monterey and at the naval postgraduate school and so I can find out what that date was and then I remember working on the architecture at that point in time and working through the op codes. In a way it's the final stages of the op codes at that point in terms of what was going to be done.

END OF TAPE 3, INTERVIEW 2

Oral History of Gordon Bell

Interviewed by:
Gardner Hendrie

Recorded: February 2006
San Francisco, California
Edited: CGB 6/15/08

Interview 3, Transcript 1

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.

END OF TAPE 1, INTERVIEW 3

Oral History of Gordon Bell

Interviewed by:
Gardner Hendrie

Recorded: February 2006
San Francisco, California
CGB 6/15/08

Interview 3, Transcript 2

CHM Reference number: X3202.2006

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: Yeah. This is going to be...

END OF TAPE 2, INTERVIEW 3

Oral History of Gordon Bell

Interviewed by:
Gardner Hendrie

Recorded: February 2006
San Francisco, California
Corrected: CGB 6/15/08

Interview 3, Transcript 3

CHM Reference number: X3202.2006

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.

END OF TAPE 3, INTERVIEW 3

Oral History of Gordon Bell

Interviewed by:
Gardner Hendrie

Recorded: February 2006
San Francisco, California
Edited by cgb 6/15/08

Interview 3, Transcript 4

CHM Reference number: X3202.2006

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.

END OF TAPE 4, INTERVIEW 3

Oral History of Gordon Bell

Interviewed by:
Gardner Hendrie

Recorded: February 2006
San Francisco, California
Edited cgb 6/16/08

Interview 3, Transcript 5

CHM Reference number: X3202.2006

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.

END OF TAPE 5, INTERVIEW 3

Oral History of Gordon Bell

Interviewed by:
Gardner Hendrie

Recorded: October 19, 2006
San Francisco, California

Interview 4, Transcript 1
Edited cgb 6/15/08

CHM Reference number: X3202.2006

Hendrie: We have with us today Gordon Bell in a continuing series of interviews covering his long and very eventful career in the digital world. I think we left off last time with the beginnings of the VAX project and the collecting of a team. Could you start there and talk about once a decision has been made to proceed to design this system, how the team was assembled and who were the leaders and what were the first things that needed to be done?

Bell: Okay. The first meeting now looks to be April 1st, 1975. I have on my website a report by what was called the VAX A group. So the VAX A group was the small team that was with the lead architect, Bill Strecker, and we talked about what the machine is going to be and all of the issues and made various decisions, and that was the team that then spread out and actually implemented it afterward. There were six people. I led the group. It included Bill Strecker who I claim is the architect of VAX per se. He had been studying this kind of an architecture while part of the research group. Steve Rothman who was a hardware implementer and was part of the 11/70 group, and he communicated with the hardware group to actually implement the first machine. Dave Cutler who was responsible for VMS and was going to be the implementer of that. Richie Lary who was I'd say probably the most versatile generalist in all of DEC and still remains that way. And then Tom Hastings who was a very meticulous scribe and actually took all the notes of everything we did. And this was described in a memo. I wrote the various design decisions down and why we ended up making the ultimate in the CISC architectures, that is it had everything in it. It had a PDP-11. It had a COBOL instruction set so that you basically could do virtually all of the things that COBOL wanted kind of in one instruction -- it turned out to be a micro instruction or a subroutine, queuing so that you could add to queues. It had stacks and things that were remnant of the PDP-11. So that was the basis. We met for about three months, worked through all of the issues e.g. pages sizes and what was going to be in the architecture, and every couple of weeks, we allowed I think one or two people from marketing to come and talk to us and then we sent them out for questions. I don't remember any of the questions that we had for them. I don't think there were very many. And then we had one customer input and that was Ken Thompson from Bell Labs, and we talked about what VAX was going to be and is that going to be fine for UNIX. So that was the process. Meanwhile, Steve Rothman had communicated with most likely Bob Stewart who worked for Bill Demmer, and they actually implemented a very general machine that is one of the prototypes sort of hanging around. By the time the architecture was done I think we had a working machine in September. It was a microcoded machine -- it was basically a programming task and the people could then start thinking about using it or whatever for benchmarks or testing codes.

Hendrie: They had built a general purpose machine, microcode sort of machine that could act as a simulator, knowing that it might serve this purpose or was it for some completely other reason?

Bell: I don't remember exactly how or why we had ended up with it, but we had this machine basically that was done. We had a bunch of technologies around that had been done for PDP-11's, and so a lot of those were put on steroids to do the 11/780. There was a bus that ended up being the bus that we used for the 780 and it was a very general, or well, it had been certainly modified and extended and everything when we finally built it. It was the first machine we ever had where we basically loaded the microcode. IBM had been doing that for mainframes but we had a floppy I think in the base of it to load the microcode for us. Then we proceeded on to implementation and I remember going out and, quote, selling a few. The first couple, I had made a presentation at Carnegie. A professor by the name of [John A.] Pople who ultimately won the Nobel Prize for computational chemistry -- he was sort of the inventor of it -- was going to get one of the first ones and he actually ended up getting number one or number two. A couple of the early machines went there and I think that was one of my sales calls when it was introduced in late'77. I don't remember when they actually delivered it -- I think it was later, in early '78. And then I went to Japan and gave talks about it at the various universities and then to Australia in the summer of '78, and by now it was clear that the machine was going to be successful. And by the way, in the process one of the other things that I did was to write the ads for it since I had always complained bitterly about advertising. What we ended up doing was a very simple idea. It was a testimonial ad program and it got to be quite a thing to get to be asked to give a testimony -- users loved to be selected for our "ask any user" ads.

Hendrie: Pushing you to buy a machine to do it. Right?

Bell: Yeah, the campaign was called "Ask Any User" and the users basically featured various aspects of the machine. It had new arithmetic and it was a nice virtual memory and a good sized memory and so at that point in time it was a very hot machine and it sort of became the standard for computation. In fact, I still to some extent live with the legacy as all the supercomputing guys always accused me of killing supercomputing, of being the one person that really fouled up and destroyed supercomputing because then everybody had their own computers and now they didn't need the centralized Crays and things like the large comp centers.

Hendrie: Those were relegated to only the most difficult, largest jobs.

Bell: Anyway, so that was really kind of a byproduct of all of the VAX, and when it was clear it was going to be a success and it was worth betting everything, I had gone to this trip to the Far East including Japan which was my first trip to Japan. Actually, that was in the summer of '78. I came back from that and spent 3 weeks diving in Tahiti. So we had enough experience with it, the machine was going well and so basically at that point I proposed and wrote the VAX Strategy which essentially bet it all on VAX and that we would build a number of machines and that we would stop PDP-11 stuff. I didn't say anything about the PDP-10 at that point. We stopped building PDP-10's later and everybody accused me of killing the PDP-10, but in fact in reality what happened is no one there was able to build PDP-10's. Alan Kotok had stopped working on PDP-10's and no one knew how to build a PDP-10, so we had had an engineering project and we had to abandon it because the guy doing it had built 360's and when he found out the PDP-10 was nothing like a 360 then all the pipelining techniques wouldn't work at all and so basically we stopped the project.

Hendrie: What was that machine called internally?

Bell: It was called Jupiter.

Hendrie: Who was the project engineer?

Bell: Bill McBride was head of the whole project, and I don't remember the engineer that was doing it but he was hired because he had done such a good job on the 360. But the team hadn't really built a 10, so basically all that learning had gone.

Hendrie: And it didn't help, required a new set of rules and . . .

Bell: Right. All of the idiosyncrasies and what you had to do because the 10 was great in that everybody loved the byte pointer in there but that means virtually all the instructions are sequential. It had a huge number of bytes pointed at things and that meant that fundamentally every instruction required an indirect address and so we had to do something to that which was not built into the hardware.

As an afterthought, 08/08/24, a company called Systems Concepts in San Francisco had built a great PDP-6 using TTL. Shottky and I tried to get it licensed in late 1982. Stu Nelson was the designer and Mike Levitt the president. But when I had my heart attack, communication stopped with them because I was the only one they trusted. The deal would have been worth 10 million or so to them with absolutely no work except to sign the license. According to Wikipedia their name was changed to the SC Group and they sold a few computers.

Hendrie: I'd like to go back a little bit and maybe talk about some of the issues that came up from the time in '75 when you sort of got this project rolling that you had to deal with.

Bell: Right. Well, I think the big thing was getting to the point in '75 because there was a long period of gnashing of many teeth, of do we extend the PDP-6 or the PDP-10 or do we extend the PDP-11? And by now the PDP-11 was going like gangbusters, the 11/70 was out there, the rule was 8-bit and the 10 was doing well too. It was clear we could build smaller machines and that in a sense it didn't make very much difference. But on the other hand we had so much tied up in the inertia, velocity, momentum. Everything was going to the . . .

Hendrie: Of the 16-bit line.

Bell: Of the 16-bit line.

Hendrie: And 36-bits, was it going to look good on your image even though it did the job.

Bell: Right, and so that was part of the big decision and that was done prior to deciding that we were going to go that way when we started up in April '75. In some sense, I don't think there was much of a decision about the approach we took. In retrospect several years ago Cutler asked me, he said, "Well, wait a minute, why did we go that way versus a RISC?" And I said it was unfortunate because at that time it turned out all the technology was just at the wrong point on the curve. We'd write fast, we'd read write memory and couldn't build a large cache that was the basis of what RISC needed, would have been predicated on so there was a big thing about that. The microcode memory worked very well. There was an issue of compatibility that came up and so those were a lot of the concerns about the decision to go the way we went. Then once we made that decision there was the VAX A Bluebook which has the instruction set in it. In fact, I was rereading it the other day because I gave a talk about the history of minicomputers through DEC to a group at Berkeley, UW, and UCSD as a course. I reread that part in the VAX A Bluebook to decide how we had made those decisions. But at that point once you went down the microprogramming path, you basically put everything in, anything that was going to be used in a sense more than once you sort of put in. We had a polynomial instruction in the architecture too.

Hendrie: I didn't realize that.

Bell: Right. No big deal. You just put the tables there. We had a very elaborate "call" instruction and I don't remember what the ultimate impact of the effect of a call was. We didn't use it, we weren't using it that much or it was too specific, but certainly we had everything that would. . . You sort of put everything on the stack and called the procedure.

Hendrie: Was there any provision for the customers doing any of the microprogramming?

Bell: No, we didn't want to go down that path at all, absolutely. We wanted to stay away from user microprogramming because of the difficulty of maintenance and . . .

Hendrie: And support.

Bell: Support and compatibility going forward. The biggest thing we had was insistence that we were going to maintain compatibility because there were many different models for the PDP-11 but they were all fundamentally incompatible, and it was I'd say a tribute to programming that in fact we could run across all of these machines. There was an article that Ron Brender wrote in the software group called "Turning Cousins into Brothers and Sisters" or something like that, but basically all the early floating point or the arithmetic had different options and there were a whole bunch of different options and different levels of memory management and it evolved over time. Everybody was sure "we can't stand doing this again".

Hendrie: You can't do that . . .

Bell: Never can do that again. So at that point we really worked on compatibility. That was a key part of it, that the models had to be compatible as we went forward.

Hendrie: What was next after the original VAX? What was the next machine that was started?

Bell: I'd say . . .

Hendrie: Taking the next step . . .

Bell: The next step was this big event in the summer of '78 when I was on a diving sailboat for three weeks and it was really kind of an "ah ha" and saying look, we've got to build a range of machines but unlike the 360. And so I got that idea from the 360. I had written about that at Carnegie when I wrote "Computer Structures" with Allen Newell. There were two things that were different. In the case of the 360, all the 360's were fundamentally mainframes. They were all operated the same way. They had different operating systems. In the case of VAX, they all had the same operating system, they had the same environment, and the thing that differentiated them was in fact where they lived in the hierarchy, that is, we had predicated clusters so our mainframe was in fact what has now become the way you build mainframes, scalable clusters. So we pioneered clusters by aggregating 780's and then bigger machines together saying we would never be able to afford building the big machines that IBM was building and the way to solve that problem was simply the cluster things or MPs or cluster, whatever made sense at that level as opposed to continuing to push as much into technology. Now that lesson got forgotten later on and that was one of the bad parts of what happened to DEC. The 9000 was clearly one reason for failure because pouring $3-6 billion into a product that didn't make it was quite a bit of money. I don't know how much it cost, but it was . . .

Hendrie: Very costly.

Bell: Very costly and could have been something else and it also established a kind of psychology of "Gee, we're going to use that machine to go off and head into IBM", which was crazy. So anyway there was the hierarchy which was at the top, it was a cluster, and then the sort of distributed many kind of thing that we all knew and loved, and then at the lowest level then we said everything's going to be work stations and PCs and that those things are going to be the desktop and those things had to all run compatibly and so you could choose anywhere you wanted to compute and that was the basic idea. When I came back from the summer trip to Tahiti, I started that crusade to go install that and that meant getting rid of a bunch of 11's and other projects and focusing on VAX and then starting the VLSI programs to get MicroVAX.

Hendrie: You had some concept of how you could execute a really small machine.

Bell: Yeah, and those things all took off. Initially, there was a huge amount of uproar about that. We stewed over that.

Hendrie: You mean in terms of killing off the 11?

Bell: Yeah, and deciding that we were going to bet it all. Initially Ken was fairly non decisive and aloof and observant. One day he'd ridicule it, then he would listen and then ridicule, and eventually I think he actually bought in to it. And the culmination of that was I went to the board, I think in November or December of 78 or early 79, and I described what we were going to do and that became known as the VAX strategy. It was a diagram in the shape of an 'E'. At the top part of the 'E' was a cluster with a cluster interconnect and then all of these levels and things were connected together by an 'I' which was a network interconnect. And fortunately in the '78 time frame we were able to make the connection with Xerox, and Bob Metcalfe for the Ethernet.

Hendrie: Tell us a little bit about that story. What happened there?

Bell: Well, first off, the strategy was in a sense predicated on an 'I for Interconnect,' of having some way of doing that, and we had two or three different ideas about connecting everything using a local area network.

Hendrie: You had some idea of a way to do it but you hadn't thought of . . .

Bell: And at that time our friend Apollo wasn't started so we had two or three different network interconnect structures there that we were looking at. We had some rings and we had other stuff. We didn't know a lot about the Ethernet at that point although that had come out I guess in '76 or so and that's what PARC was using to connect their workstations. What happened was Bob Metcalfe joined us as a consultant. He had left PARC and said, "Gee, I'm here, what I want to do is propose that there be a standard here and I want to try to broker a deal here with Intel." Or I mean with Xerox. And so that was kind of the beginning of the DEC, Intel, Xerox cabal that in fact ended up proposing Ethernet. And I can probably find it, but I don't recall exactly the first meeting of that but it was with Phil Kaufman, who was later the president of . . .

Hendrie: Of Silicon Compilers.

Bell: Silicon Compilers. He was at Intel and he and I got together. We met using the ATT's PMS picture phone meeting service in San Francisco and Boston so we got together virtually. I think we signed up for two hours of time at a couple hundred bucks an hour and we had a meeting and basically agreed to be partners in terms of developing the Ethernet standard. And I think it happened kind of concurrently that we had approached Xerox for the patent and to be part of that and they were the group brought in and David Liddle headed it. I recall the meeting and it was in the Parker Street building and our guys had been meeting with their guys. I don't remember exactly who they were but I remember coming to the meeting and at the end of the meeting I said, "Okay, where are we?" And they said, "Well, we've got to get your management to do something and then we got to get our. . ." And I said, "Okay, let's make it very simple." Here's our management asking for this patent so I said, "What do you want the letter to say?" And I sat down, wrote the letter on a random word processor right there, signed it, and they walked away with the letter of understanding. They went to the . . .

Hendrie: Went to the Xerox managers . . .

Bell: . . . and they said okay, these guys want it and so it was a patent.

Hendrie: You wanted to take out a royalty free license . . .

Bell: They basically agreed that this be a standard and that was the beginning of Ethernet.

Hendrie: Had Metcalfe left?

Bell: Yeah, he had left.

Hendrie: He had started 3Com?

Bell: No, he hadn't started 3Com yet. That actually happened after I believe. . . He might have started it, but he was there as kind of a free agent advising us. He was a consultant so we actually I think paid him for consulting on what a network would look like. Anyway, that all got put together, and the culmination of that was the 1982 -- I think February '82 -- announcement and that was Liddle and I and Bob Noyce announcing the Ethernet at the World Trade Center in New York and we talked about this being the standard, this is how we see computers being connected together. And then I remember one of the people from Business Week was there saying, "Well, Wang's got their Wangnet and why aren't people going to use that network because it uses the existing cable TV?" And I said, "Well, first off there's not much cable TV in corporations, and even though people know how to install cable, we don't really see these things as being compatible at the time or necessary." And then I said, "Oh, and besides, the biggest problem with cable TV is this is sort of a direct coupled coding as opposed to analog coding, and even though you've got multiple channels on cable and people were starting to fool with cable TV for networking, the biggest problem is think of it like your house or your city distribution. You've got water, sewage and gas coming into your house and in principle you can put them all on one pipe but sorting them out is the difficult thing." Anyway, so I sort of blew her off. But anyway, that was the announcement and then we made the announcement in London and we also went to Holland.

Hendrie: What were the responsibilitiesof the partners in this?

Bell: Initially, the way the thing was structured there was a working group of I think eight that were designing it, and in fact one of the things that was contentious was I'd get this call from Phil Kaufman periodically and he'd say Olivetti wants to be a partner and if you don't want them to be a partner they won't play. And I said look, we've got eight people, they know more about networking and I don't know that Olivetti knows anything about networking and no. That was basically a nice way of telling them to pound sand. There'll be plenty of time to comment on it after we get it proposed. I said I don't even go to the meetings. And then one Saturday night I remember I had gotten a call from Rob Wilmot who was the chairman and the CEO at ICL so he was the head of the U.K. company, and he said, "Hey, this is exactly what's needed, we really need it, we believe that the industry needs it. I will bring you all the Europe ECMA. I'll bring all the European computer manufacturers into it." And I said great. And he said I want to take a look at it and I said come on over. And so he brought over a bunch of his designers and we ended up at a meeting at dinner. The final dinner was at Colonial Inn in Concord, and so that brought them in. And then ultimately after it was announced and the products were out there, IBM sort of came out with their wiring plan for rings, but their LAN was years after. Basically, they didn't have a product at all. The token ring was many years after Ethernet was announced and what they did announce was their usual way -- that IBM will take care of you here, just install these wires, and it was sort of this bundle of wires and it had a couple of cables, a fiber and this and that and everything.

Hendrie: It was a hodgepodge trying to appear to be part of this.

Bell: Yeah, and so anyway install now and pay later or play later.

Hendrie: You said you didn't go to the meetings. Who were the technical representatives from DEC?

Bell: Dave Rogers I think was the technical lead, and I don't remember who all of them were but they were the best people that the world had at that point in local networking because we had the Xerox guys who had had experience with their earlier net, the 3-megabit net, and so we had that working experience and then we had some good circuits people. It might have been actually Tony Lauk who was one of the DECnet architects, so we had that kind of architectural experience.

Hendrie: And the Intel people were going to implement . . .

Bell: Yeah, they came in. I think maybe there were a couple people from Intel and it wasn't fully balanced on three, three, three or something like that. But in fact they produced the Ethernet Bluebook basically that then was commented on. And I think HP had probably made the biggest contribution having to do with some grounding and stuff that made it work better. But anyway, that was the origin of the Ethernet. My original slides for the product introduction at the World Trade Center are on my website. The slides had a couple of great quotes: "Ethernet is the UART of the 80s" and "the network becomes the computer" that SUN adopted. Several folks later told me that they thought I was crazy making the claim, but now after the fact they acknowledge that I was totally correct.

END OF TAPE 1, INTERVIEW 4

Oral History of Gordon Bell

Interviewed by:
Gardner Hendrie

Recorded: October 19, 2006
San Francisco, California
Corrected June 25 ,2008

Interview 4, Transcript 2

CHM Reference number: X3202.2006

Hendrie: Going back to the original VAX development, were there any things that came up in the development that proved to be particularly difficult hurdles to overcome or represented challenges that you hadn't anticipated were going to occur during building this quite large machine?

Bell: When we first built the 780 I would say the answer is absolutely not. We'd kind of done a lot of it before. It was the first really serious microprogrammed machine that we had built. It was bigger but I don't believe there were any real crisis. There probably were some in terms of, oh, we don't have any space to put in these instructions. I recall Richie Lary worked on it from time to time to get things to fit so there might have been a crisis there but that was probably it. We had simulated and understood the whole architect, the cache structure that we had, and it was nice -- it really worked well. The decision to use the PDP-11 and have that be our build in an RSX-11 compatible machine was absolutely great because in fact it allowed us to run a huge amount of the software. All of the software was running on day one on PDP-11's or the compilers and things like that when we announced so we could come in with the software running and that was a big deal. So in a way it all went very smoothly. The crisis really on VAX occurred when we implemented the ECL machine, or the using of ECL gate arrays machines. We had two projects, the 750 and then the 8600 which is Venus, and in the case of the 8600 the project just totally got out of control and was an absolute mess that cost DEC dearly in many ways.

Hendrie: This was the ECL project?

Bell: This was the ECL project and in a sense it came out in, oh, I don't know, '85 or so. Vinod Khosla, the founder of Sun, says that if it hadn't been late Sun would never have been able to get started. What that did was because everybody was waiting for a faster 780. They couldn't get it and so Sun could now take all that money that was allocated for 780's, the 8600s, and buy workstations with them and that's what they did. And so that was kind of a byproduct of a project getting really screwed up. In fact, the 8600 was just an unmitigated disaster in terms of a project and I went out there to Marlboro and was out there for six months working on it.

Hendrie: It was done out in . . .

Bell: In Marlboro, and in fact it was under the same guy managing the ECL PDP-10. Bill McBride wasn't the project engineer but he had another counterpart who was leading the 10. Ulf Fagerquist of course was the boss for both of them. Then there was George Hoff who was the engineering manager for the 8600 or for Venus. And fundamentally it was kind of a classic story of what can happen to an organization. When I went out there it was sort of clear what happened because I sat in on one of Ulf's staff meetings and there were all these people around who had nothing to do with the project. Here's George and I think McBride and then everybody else was sort of the finance guy, the human resources guy, and this guy, the facilities, all the 10 or 11 people in a room, and so Ulf was spending his time on all of this crap and had no knowledge of the projects. From this experience I made a rule: "If I ever know more about any project than anybody in the management chain then let's get rid of the intermediate managers." This was a pretty hard thing to test because I generally knew more about every project in the company than somewhere in the chain. So you'd go up and down the chain and occasionally people would know something about it but in general the managers didn't know what was going on.

Hendrie: At what stage in this project did it start to unravel? Was this in the paper design stage?

Bell: Yes, it was in paper design. It was a little bit past the paper design. The team hadn't quite finished all of the design. It was manifesting itself as, "Gee, we've got to take a risk on getting Motorola, buying chips from Motorola for the gate array, so we've got a bunch of essentially manufacturing gate slots that are lined up and now we're having trouble and do we take a risk on those things building the chips?"

So that was kind of one of the warning signs that popped up, and the design isn't quite done so there were a bunch of things like that. We had a huge design review -- I know it was a Friday the 13th, I don't remember which Friday. All I remember was that it was that bad, it was in a very large room, a cafeteria, at Marlboro and everybody was sitting there, and Bill Demmer was there and he was from Littleton and George Hoff talked about the design. George said, "Well, we've got these problems and we're not quite done with this or that" and was asking whether we should get these chips fabricated or not, and then lots of things like, "I'm not sure about what Motorola is doing and I don't trust those guys" and all this stuff. And so there were all these bad signs from Motorola and so it's their problem. "No, I don't know what the gates are going to look like or the power", some damn thing like that, and so it was the end, 4 or 5 o'clock, and I said does anybody have a feeling about this? Well, this is going to be tight to make the schedule. All the people in the room thought it was going to be tight, but doable. And I said, "This is the worst project I have ever seen, this is a disaster, a f***ing disaster, this machine will never in a million years work, there is no way that this thing will ever work, you have no verification of it, you're not simulating it at all, there's no way, you're not even doing any code reviews, there's no good debugging philosophy to it where you can do that, you've got some of these little things, you're not even done with the thing!" And then I went in on Monday morning to the operations committee and I said, "I got news and this is really, really bad." And Ken said, "Well, you'd better leave right now." and I said, "I think you're right." And so basically Mary Jane and I went to Marlboro and spent six months and started working on the project. We had to start a simulator from scratch. We had some simulation efforts but we couldn't buy one. We had that. There were a whole bunch of things that were started to get that. Then I remember Ulf and I went to Motorola and looked at where they were and I came back and I said it's not Motorola's problem, they're going to be just fine, don't worry about them. Shortly thereafter, I said to Alan Kotok, "Alan, you've got to come in here and look at the project".

Hendrie: What was Alan doing?

Bell: Oh, Alan was doing telephones probably. He loved to work on the phone system or whatever.

Hendrie: He'd gone off to do something more interesting. He was tired of doing 10's. You weren't going to build any more 10's so . . .

Bell: Yeah, we weren't doing any 10's and so he was having fun with a phone system or some damn thing like that, and it was always something to consult on and so I enlisted him to come back and said you've got to play the referee architect. Venus was one of those projects that in the first week individuals or groups were telling me they're really the architect driving this whole design thing, and then the second week everybody was telling me no, they had nothing to do with this.

Hendrie: That is funny.

Bell: Yeah, it was pathetic. In fact I called it the Soul of the New Machine 2, or Soul 2. I think it was called after the book about DG. Not even the DG team did this as poorly. It was of course late and was really a financial disaster. I can't remember the number, but DEC was losing 2 or 3 million a day on not having it, so you knew what the effect was going to be. And I said if we had had it we would be delivering this amount.

Hendrie: How late did it end up being by the time it got out?

Bell: I think it was probably three years late. It was a factor of 2, yeah, at least a factor of 2 schedule slip.

Hendrie: Huge amounts of . . .

Bell: Yeah, I said we're not going to build, we cannot build this machine until you can show that it's going to work and because you've got these gate arrays and they're not probe-able or fixable.

Hendrie: You can't stick a probe in them.

Bell: Yeah. Well, they had some fake gate arrays that they could probe and look and see what was happening but you couldn't build a whole machine out of them. Or if you did, you had to plan that's how you're going to build it and we didn't have a plan that said we're going to build an emulating kind of machine and that wasn't it or whatever. And so there was that, and I'd say that was probably a big effect on DEC, and then in a way to survive we developed a simulator.

Hendrie: It did eventually work, right?

Bell: Yeah, it eventually worked. DEC sold a lot of them and it was successful. Meanwhile, I had gone to Littleton and got them to start the 8800. I sort of pissed them off because I went out there and they were farting around with 8800 with a TI technology called I2L [Integrated Injection Logic] that never hit the market. I had them switch to ECL, really as a backup to the 8600. In retrospect, they might have better jumped to CMOS. That way we would have converted some great designers earlier and changed their careers.

Hendrie: What were they theoretically doing?

Bell: This was the 780 team so they knew how to build computers. The other guys, well, really they probably didn't even know how to build a computer but they were building something out of I2L [Integrated Injection Logic]. So I was watching all that and this is TI and they had a similar kind of thing and this I2L is getting slower all the time, it's not clear TI's going to make this or not, whether they're going to manufacture it or not, and they were off working on that. I said, "Forget it, you guys, don't build that, this is not going to work, it's just too hard, you're not going to get the speed with the TTL clock that you have to get to have a 780 replacement." And so they switched. The two came out just about the same time as the 8600, maybe a little bit later, but that was kind of a backup because I wasn't sure that the 8600 was ever going to work. In fact, I don't know at the time whether I would have said we don't need both of them, but the 8800 was a superior machine. It was a little bit slower but it was a dual processor and so it was just a nice, clean machine and I trusted the guys who had done it before. And then meanwhile over in Hudson people were saying, "Yeah, we're moving along, we'll eventually get there with CMOS." But they were years away, three or four years away from getting enough high speed, so we were stuck for awhile. But anyway those were some of the things.

Hendrie: Some of the things that proved to be . . .

Bell: Right, it was that transition. In a funny way, making the transition to VLSI, going through gate arrays. because Dave Cain had made the gate array transition -- and you probably knew Dave as the founder of a company that used the 68K to make a lab computer. He had made the 750, a beautiful machine, a small machine almost as fast as a 780 but very reliable and it was a gate array machine, and then there was a 730 which is an easy machine but a lower speed and very small.

Hendrie: Talk about the issues with building the MicroVAX portion of the line.

Bell: Well, that was . . .

Hendrie: You had one start that didn't go quite as far as you thought.

Bell: There were a couple of machines. One was the first MicroVAX. It was actually done in Seattle. Dave Cutler did that. Dave of course is a whole story, many tapes in himself, but don't think I said anything at all about Dave Cutler.

Hendrie: No. You didn't say anything at all about Cutler. I was going to roll back and get the software story in a little bit.

Bell: Right. Well, certainly Dave had finished the operating system and drove that and basically wrote a large part of it and had a small team that did that. And then after that he went over and wanted to do a PL/1 compiler because he said, "Gee, I've never done a compiler."

Hendrie: I was going to say . . .

Bell: But I needed a compiler. He was in New Hampshire at the time and so he had half a dozen guys and they built the PL/I compiler because they wanted to start with something really of a challenge. He had a very nice group, a very tiny, little group. The part that was kind of interesting about it is that he had architected it such that the backend used a common run time and so ultimately that became the backend for all of DEC's software, so it was languages.

Hendrie: All of DEC's compilers.

Bell: All of the compilers, and so he worked on the run time and the backend and that was for Pascal and I think C and PL/I. I think Fortran ultimately moved over to it but that was another machine, another language that it had initially had. So Dave had worked on that and then that came out, and I don't remember when it came out. It was probably '80, '81, something like that. He then basically came to me in 81 I believe and said, "Look, I've had it, this is a totally bureaucratic place, can't get anything done, all this bullshit, and so I'm leaving." I asked what he was doing. And he said, "Well, there's a guy at Livermore that wants me to build a language and we're going to build some compilers," or something like that. And I said, "Okay, well, look. We'll fund you, we'll invest in your company. Why don't we set it up and we'll meet the guy." Whoever it was was a typical Livermore physicist who knows everything. So they came out and we had dinner someplace in New Hampshire -- Dave and I and probably Larry. I don't know whether Portner was part of that. There were probably four or five of us at dinner. This guy had such a tremendous ego and he sort of said when we asked what his company was going to be doing, "Well, my programmer here will do this," and so he just kind of sloughed Dave off as oh yeah, he's . . .

Hendrie: As his programmer.

Bell: His programmer that's going to implement his vision of this great piece of code. So after this Dave said, "Well, I've decided I'm not going to do that." I said okay, that's all right, I can understand your frustration because we had just moved the disk guys to Colorado and I said I think it's not good to not have everybody in Maynard anyway. So take anybody you want, go anywhere you want and do anything you want to do. And so he and Roger Heinen and a couple of other guys that he had worked with went around the country looking at sites. They looked at the valley and ended up going to Seattle, and then he said, "Okay, I'm going to build a MicroVAX. I've always wanted to do some hardware, I'll build that." Because the MicroVAX wasn't going to be ready in time because it was a VLSI chip and the chip had a long gestation time and it wasn't ready. And so he said he'll take a more conservative route, we'll do whatever, and it turned out that he ended up buying the first chip from Silicon Compilers which was a custom data path, a VAX data path. So it was a 32-bits, in essence a fancy ALU, but in fact it had the registers and everything else with it and the control and so you just add the microprogramming and you've got a VAX. So Dave pulled some people that could manufacture stuff and that team actually ended up putting it into manufacturing and I don't know how many were sold.

Hendrie: In Seattle.

Bell: Yeah, in Seattle and that was the first MicroVAX. It was small. It was a workstation sized machine and he also said, "Gee, by the way, I'll throw in another operating system just for free." So basically he put a run time, a real time VAXElan, which had a lot of stuff in it too. It was probably the only operating system that was written totally in a provable code. It was kind of a very fancy typed check Pascal that was used for real time control. You downloaded the VMS images to it and so it was totally compatible with everything, with all the other machines.

Hendrie: He was going back to his roots to a real time operating system.

Bell: Yeah, and so it was probably the world's best real time operating system bar none. So anyway, that was Dave.

Hendrie: He kept wanting to do something he hadn't done before.

Bell: Yeah, absolutely. And the irony is he's still doing it. He of course did NT and that was the basis of all Microsoft operating systems. He was the guy who actually did the 64-bit NT. And Microsoft getting that out, which is no menial feat. Now he is working on a new project that I think will, quote, save us again. And there's never anyone who's been as productive or prolific and really just produces this sort of beautiful work as Dave. There's just nobody like him. We've got all these guys that have done really one nice thing but no one who just continues to put out really solid, beautiful software.

Hendrie: Elegant products.

Bell: Yeah, and so he's doing it again which I'm delighted about.

Hendrie: That'll be very interesting.

Bell: Yeah, and it is. I'm excited about it.

Hendrie: We did the other day Dave Cutler's story. Talk a little bit about if there were any issues in the software development of the original VAX. We talked about the hardware and some of those things and yet it was either the compilers or the operating environment.

Bell: I think it went very smoothly. It wasn't your kind of typical thing where you're starting something new. The labs [Bell Labs] had gotten an early VAX and I think that they had ported a UNIX System 3 or 5, whatever, AT&T. Then we had a bunch of VAXs that went to Berkeley that they built the Berkeley Unix on. Coming back I think about these folks that keep reappearing. Dit Morse who is the only guy I'd ever fired when he was heading the PDP-6 timesharing system in 1963.

Hendrie: You told us quite a few tapes ago about that.

Bell: Yeah. Anyway, Larry Portner came to me in the late 70s and said "Hey, I'm going to hire Dit." and I said "Dit, I haven't seen him in years, since 1965 or so!" And he said, "Yeah, we need somebody who's really been working on file systems." And so he basically came in and did consulting on the file system. I think ultimately Larry also fired him, but he did spec the file system. There were others, and it was like these different people, Bill McBride. Whether these things get seen or not, it doesn't matter but it's kind of interesting that there's a little management philosophy that shows incredible tolerance. So he in turn really screwed up the PDP-10, and then I think Ulf got rid of Bill McBride because the PDP-10 was equally screwed up. And Bill Demmer came to me and he said "Hey, I need a project engineer and do you mind if I hire Bill McBride?" And I said, "Well, I wanted him fired but if you're going to do it and hire him, that's your problem."

Hendrie: It's on your watch.

Bell: And then three months later well, Bill left or he fired him. I remember Gordon Moore telling me how they had interacted with people at IBM. He said the problem with IBM was they're just absolutely brutal in the way they manage. He said at Intel if we have a guy and he screws up then we give him another job and about the third time we say well, maybe it's the guy. But at IBM the guy screws up he's out of there, no second chance. So we give him about three chances. And so I always remember the story of that -- oh, yeah, we don't want to fire the guy, we want to give him another chance. But anyway so Dit had done the file system design and then was let go.

And people come to me from time to me and they'd say, "Hey, there's this great language on VMS." So many people! Anyone who has used VMS just says, "Oh, god, why isn't that out there?" But it was the DCL, the DEC command language. I was never a fan of command language so I didn't even know. I knew the 360 one from CMU and it was the world's ugliest language, just ugly. But Dave had a beautiful command language that would drive everything and that's what a lot of people wrote applications in, in this command language, and it was a very nice thing that he had done.

Hendrie: Can we go back to the MicroVAX and take the next step in the MicroVAX story?

Bell: Well, what was going on in MicroVAX was it was being designed in Hudson. DEC was on a treadmill to continual movement to I'd say master VLSI. I don't recall that there were really any disasters per se, but it was much more of just the sheer size of the jobs at the time and the state of the tools. This team is probably the most talented in the industry and those guys are spread throughout the industry at AMD (Rich Witek), Intel that took over, and companies they started like Dan Dobberpuhl.

Hendrie: It was just a horrendous amount of work.

Bell: Right. And I think Carol Peters was one of the people that had worked a lot on MicroVAX. The final MicroVAX II, which is actually the one- or two-chip MicroVAX, was subsequent. It came out after MicroVAX I, the Cutler machine using the Silicon Compiler.

Hendrie: At what level was the technology of this? Was this N-Channel Silicon gate or was this CMOS?

Bell: At that point it was CMOS.

Hendrie: It had gotten to the point where you were doing CMOS.

Bell: Yeah. I am not sure if there were any intermediate N channel stuff or not.

Hendrie: So it was a dedicated CMOS.

Bell: Yeah.

Hendrie: Full custom CMOS chips.

Bell: Yeah, absolutely. I had come back to DEC in 1972 to get into that whole business, and then we started out with the LSI-11 which was a deal that we had made with Western Digital and then gradually acquired the technology and the learning. Carver [Mead] had helped a lot on all of that and we got the people together to just understand it. Alpha was probably one of the best, hottest machines around. Also they developed the StrongArm for embedded computing that was ultimately sold to Intel. The basic problem was that the company spent like a semiconductor company but didn't have the sales and marketing to go along with the many great products it produced. Some of the people that were involved in that, and then also I guess they did the StrongArm chip, and so a lot of the people that are key in CMOS really came out of the Hudson environment. Dobberpuhl, for example, and the rest of them, so I think it was a super group. I'd say probably the only bad thing that happened -- and I can't really reconstruct the thoughts and the history of it -- was undoubtedly too much do it yourself. We can't really afford to be investing in those very large plants for our own stuff. So not making the right deals in manufacturing and capital equipment and all of those kinds of decisions. They just didn't feel right certainly.

Hendrie: Did you get involved at all in any of the early DEC workstation/PC ventures?

Bell: Yeah, those at that time were disasters in terms of looking at would I do anything different or not, and I think given the way that it all played out, probably the big error there happened around the time I left in '83. In a sense the whole story was written about that time and that was the standard had been established and that was all right when I left. I'd say the whole book on the history of PCs had been opened. DEC introduced three separate personal computers at a time when the world had clearly standardized on the IBM PC standard. This was really the root of the problem. The PDP-11 might have been okay, but it was held as proprietary.

Hendrie: If you were a good prognosticator, you could see what was going to come.

Bell: Yeah, and you didn't have to be very good either because everybody was out there. So I'm not going to fault us for doing the three because there was, for example, the DECmate, and that was a PDP-8 and that was a fairly simple thing because of the tiny, little engineering group. They were very productive. They had customers, they had legacy, they had software and so we weren't investing a lot. It was used both for small business and word processing.

Hendrie: A company project.

Bell: And in fact I tend to look at it as the DECmate was a wonderful typewriter and it competed with Wang and so I tend to think of DECmate as just our typewriter or our terminal or whatever. It was very nice and so I wouldn't have changed that. The PDP-11 was our bet on what was going to be the work station PC, and in fact as it came out it was absolutely superior to the PC, there's no question about that, but the whole issue of standards really overwrote everything else. Well, standards and then also the other thing was, geez, we didn't even allow people to connect to it. So DEC, who had been the sort of champion of having I/O and buses because they had a very nice, little bus structure -- I don't know whether it was hot pluggable or not -- and we had both RT-11 and a version of RSX running on it. But it didn't have the panache of what IBM had. The 11, called the PRO, was held proprietary so there was no way to compete with an open standard like the PC. The Intel-Microsoft PC was the Rainbow.

Hendrie: And all of the other vendors.

Bell: Right. I don't know how long it was before it was very clear that that's the way the industry's going to structure, but it wasn't very long after the PC. And DEC never got it. It took years. I know they didn't get it in by '86 or so. I was at NSF and Ken sent a PC down for me and it had everything but software and I said, "Where's the software?" He said, "Well, we got to have special software because it doesn't run IBM software, doesn't run Microsoft software." So they never got it or they didn't get it at that time. They ultimately got it but it was too late. DEC was not used to following standards it didn't create.

Hendrie: Far too late.

Bell: Yeah, and again it's one of those things. DEC could have been a player in the PC like HP or anybody else that was starting up because it was a volume kind of issue and DEC was okay in manufacturing. DEC could get itself together to do volume when it had to and it would have been a great challenge, but when they finally did do it they were always behind. And so I'd say certainly the PDP-11 was a waste and it didn't have the right stuff, but in this case it was really the PC coming in as proprietary when the world wanted standard.

END OF TAPE 2, INTERVIEW 4

Oral History of Gordon Bell

Interviewed by:
Gardner Hendrie

Recorded: October 19, 2006
San Francisco, California
Corrected June 25, 2008

Interview 4, Transcript 3

CHM Reference number: X3202.2006

Hendrie: You were just talking about Barry James Folsum, Mary Jane Forbes, and Don Folsum.

Bell: Right. So we had the three machines, the PDP-11 was the Pro. During the project, it had been called a KO for knockout or Ken Olsen machine. And then Barry James had been the advocate of bringing in initially, I think, an 8008 version, the predecessor to the PC. No, I guess it was the Z80 Zilog version based, the 8-bit machine and then making the transition to the 8088 Intel 16-bit architecture. So he was driving that standard.

Hendrie: Was this the Rainbow?

Bell: That was Rainbow. Yeah, that was the first machine. The compatibility issue I think, in terms of really not making that a fully compatible machine, was certainly the major and fatal flaw in terms of what DEC thought about. Because it was so clear once other people were manufacturing it. Certainly that that was standard and everybody was building to that standard. The paradox of that, by the way, is that everybody says, “Oh, if IBM hadn’t made that a standard, then they would have had the whole market and blah, blah, blah.” And I said, “You don’t get it. The fact that it was a standard, otherwise it would have been like the UNIX market, the fragmented market.” The guy who actually made it a standard was Wintel and that was the standard. I know at the time or shortly after the announcement, Bill sort of says, “Gee, we can’t afford to make an operating system for you, one that’s unique. You’re going to have to be brought in. It’s got to be compatible with what we’re doing.” That whole issue of platform standards and compatibility is what made the whole PC go, of course.

Hendrie: And that was so counter-cultural to a company that prided itself in creativity.

Bell: It was counter-culture and, in fact, almost so counter-culture to the industry. In spite of itself it ending up with a standard, to a certain extent with Linux or with Red Hat, so that there’s a company there and the companies seem to be maintaining some compatibility. My own belief is that they will ultimately fragment, like in a way that’s not unlike what happened to UNIX, but perhaps not as bad. But certainly to get the kind of volume you have to have a standard and that’s what didn’t happen before Linus.

Hendrie: Was that still going on when you were there? Let’s get back to your career.

Bell: Yeah. I left after my heart attack in the summer of ’83. I’d say at that point that summer or shortly thereafter it was pretty clear to me that the PC was the standard. Clearly DEC hadn’t gotten the message to that effect in ’86 or so, because when I was at NSF Ken sent me a fine DEC PC. In fact, it was nicer than a PC, but it had no software for it, so what the heck? It didn’t matter. I think we have all these different things that were important things to the change in the course of the company -- in ’83, not making that switch, certainly by ’84 someone saying we’re going to be the best PC manufacturer out there. If you’re going to do that or you could have gotten out of it presumably, too. But DEC’s aspiration was to certainly be kind of a full service company and so you basically have to be compatible at that point. You have to offer those things. And DEC had that knowledge, because it had owned the whole terminal market, printing and CRTs. We still see some of them. Those things are still around, the VT-100 and friends.

Hendrie: You mentioned your heart attack. Would it be appropriate to start off with what led up to that and what happened?

Bell: You never know what leads up to a heart attack. A cardiologist that I talked to within the last 10 years sort of said, "Ah, stress has nothing to do with it." But basically, my view is that your psyche and stress is a major contributor, as is your heredity and whether you smoke and stuff like that. Well I didn't have those kinds of problems. But I had a double bypass in March of '83. I had some sort of angina attacks before that. I had been logging those. I had two or three events that were kind of blackout kinds of things before I had the actual one that was fatal.

Hendrie: You knew there was trouble somewhere in River City.

Bell: Yeah, and then a full-blown heart attack. We were skiing in Snowmass [Colorado], it was early in the morning and I basically passed out. I was with the DEC ski group and one of the wives, Kay Marcus, heard me fall or something and then Bob Puffer, who worked for me at the time, basically gave me CPR and saved me. I was then in a coma for about a week in Grand Junction and then I was flown directly to the Denver hospital when I was ready, and then the next day I had a double bypass. I took a stress test about two weeks ago. I had a minimal heart attack when I was ill. I was biking in France and I blew out one of the bypasses. So I have been living with one pipe there. I took a stress test a couple of weeks ago and my cardiologist just said, "Gee, I think maybe you ought to go in and do something." So actually, I got a stint a week ago. So now this one bypass is working better. I will probably want to see if we can get a little more blood flowing there eventually. They're willing to take another pass at it in another year or so. GB: In July 2007, I had another double bypass to restore the old flows.

Hendrie: That was an era when not everybody knew CPR. You were lucky.

Bell: Oh yeah. Bob was certainly very simply my savior. I owe my life to him. Bob was a volunteer fireman, among other things.

Hendrie: So he had learned these things?

Bell: Learned all of that and gave me CPR. When I was at DEC, Ken and I used to have very strong hard arguments. Certainly, those were quite stressful to us both. After that, I basically said, "Well, I've done about as much as I can do. The VAX strategy is in place. It's clear that all the products are in the pipeline. We could make semiconductors, we had the clusters, they were just being announced. I want a less stressful life." It was that kind of a thing. I think for anybody who can or has a heart attack probably really has got to think about some change in their behavior. I certainly changed my diet a bit, but not drastically. It wasn't a hard problem. Now I use chemicals to get it down. My cholesterol at the time was 220 or something, which was normal for that period. Now its 120, but with Lipitor and its friends.

Hendrie: So you decide this was really the impetus to . . .

Bell: To change. Yeah, because I wrote the note when I left DEC. I came back from the hospital and I sat in on some very contentious meeting about something and I thought, "You know, this is just too hard." To go ahead and make that decision that I would have gotten it made in a particular way, but it would have been a lot of work and a lot of angst. I think it actually had to do with workstations. It was a workstation kind of meeting. What's DEC going to do in workstations? It was one of those things of "You know, it's going to be a fight just to get it done and to stay on path." One of the things that happened after I left was much more diversion of let's appeal to the market. We've got to listen to the market. We need a workstation. Okay, well the MicroVAX hadn't been there yet. On the other hand, everybody was selling 68000 workstations. Now we're going to make 68000 UNIX workstations. Well, why are we going to do that? We've got the MicroVAX. It was right at that threshold. What happened is a bunch of different machines got introduced, and it was at that point, I'd say the peak of DEC. So I'd say from '83 on, if you look at the revenue, it was virtually straight up until '88 when DEC had its cap. I don't remember what the market cap was. It was a very significant market cap. I guess it was number two when I actually left. But it maintained number two. At that point in time, at the peak, it really had introduced all of these platforms. So there were a ton of platforms. It had the MIPS as a UNIX workstation, but after Sun had kind of established itself as the UNIX workstation company. I think there might have been a couple of versions of UNIX for the MIPS and then there were a couple of versions for a PC. So DEC had a lot of different platforms. It had like seven platforms that had to be maintained as programming environments and support.

Hendrie: Almost like the PDP-11 days of all those programming environments.

Bell: Yeah. And in this case, you were buying the programming environment from an outside source. I think DEC had maintained a couple of them, but it then really just couldn't maintain those machines. That was a huge cost of DEC's demise.

Hendrie: That made it really very difficult and more and more costly to move forward without the commensurate volumes.

Bell: Yeah, right. You diffuse the market. You can't afford investments in all of them and also people don't know what to buy.

Hendrie: Exactly.

Bell: Because you have to make a choice and be ensured that there's going to be the software for it, which they couldn't say.

Hendrie: From your own personal point of view, you decided that you probably didn't want to stay around DEC and decided to leave. What did you do?

Bell: Henry Burkhardt came to me and said, "Hey, I'm starting this company with Ken Fisher. Do you want to join us?" Basically, I talked with the guys and it was an insane business plan. But my attitude was hey, I'm leaving, here's a train. I'll just get on this train and if I don't like the train, I'll get off. What the hell. No big deal. That's basically the attitude I had. I had joined the board of Silicon Compilers at the time, I think. So I said, "Well, there's that life. I can just kind of do that kind of thing as a free agent. But here's the start-up. Let me see what that's all about." So we formed Encore. It didn't take me very long to get absolutely educated about start-ups, having never been part of one since the founding of DEC.

Hendrie: What went on there? What did you do? Or was it really Henry's baby?

Bell: As I said, it was kind of an insane business plan.

Hendrie: Why was it insane? Talk about that.

Bell: The Encore story. I have a little bit of that in High Tech Ventures. John McNamara who wrote the book with me said, "We'll get sued. You've got to take it all out". So we took it all out. It didn't add that much to the book. But the plan was we were going to buy a computer company whose name I forget. I recall they were in Florida. I don't remember which one. It was kind of an old-in-the-tooth business and a 16-bit architecture, but they had a little bit of a niche. We were going to acquire them. In order for that to happen -- because you can't quite do it that way -- you needed another company in addition to what we had. So we had bought another software company that was a friend of Henry's called Formation, which was a DG software company.

Hendrie: I know about that.

Bell: In North Carolina, which we ultimately sold to Sperry.

Hendrie: Mr. Gilmore.

Bell: Gilmore, yes exactly. And it was a very nice product and it, in a sense, saved Sperry and made them viable for a little longer. They're still viable, but we met with the board member. The plan was, okay this team of three and a couple more. We had a CFO from Prime and Carl Wassman, who was a financial type.

I will always remember the day Encore "came out". Probably by far, Encore's "coming out" was the most embarrassing day of my whole life. It was pathetic. Here Ken Fisher was, this marketing guy, this sales genius that had made Prime and Prime had been straight up. Well Prime had gone up. We introduced VAX and Prime went straight down. We just totally cut them off at the knees. He was fired from Prime and so, of course, like any good sales guy, he's pissed. "How could they do this to me? I took the orders to make Prime, never mind that Apollo came from Prime. I'll get even." We formed the company and lo and behold, he then hired the whole damned top-level sales management team about six or seven sales district managers and VPs, all of that. He said, "Well, we've got to announce the company." We couldn't say, "Well, our plan is to buy this other company or to acquire this other company or whatever." So we did this in New York. This was at some damned financial channel -- it was put out on the the financial TV network. The most embarrassing day of my life was sitting up there with 12 salesmen -- Ken and Henry and 12 salesmen, a line of these guys who I had never met and talking about, "Oh yeah, we're starting this great new computer company." It's like, "Oh, shit. Is this thing going to work? What is going on? This is not the way you do things."

Hendrie: This isn't how you do it.

Bell: This isn't how you do it.

Hendrie: You build this product and then you tell everybody.

Bell: Right. That's why I say it was just absolutely terrible. Right after the announcement of our coming out party, we went to meet with one or two of the board members of our potential mergee. One of them was Morgenthaler from Cleveland, the father of Gary, the Silicon Valley VC, who was on the board of the company to be joined with. We talked about acquiring the company. We had a slight problem in difference. They had a book value of like $100 million or so. We wanted an equal book value. For five guys or ten guys, we said, "Our book value is like $100 million. We're going to do an equal merge. No, no, maybe we'll give you $5 million for that and then you're going to come in and run the company." It took about a minute or two into this conversation for me to get it. I don't think I understand these financial transactions, but oh, I think I do understand! It took a couple of minutes. Oh, this is a dream, it is not going to work at all.

Hendrie: These people have a balance sheet and they can add it up to $100 million. We just say $100 million.

Bell: Yeah, that's what our balance sheet is. Gee, our stockholders could sue us if we did this deal for half of the company. So we went home and that was Encore I. Ultimately, I call these Encore I, II, III, or IV or whatever. I think we ended up with Encore IV or something like that.

Hendrie: So this was Encore I.

Bell: Encore I. Then Ed Fredkin came in and said, "Hey, you guys. Let's make PCs and sell them." Okay, I guess we could do that. So we did and Ed was sort of beating us up about trying to do that. Nobody could get enthusiastic about making PCs and selling them. Henry and I were like, "Okay, well let's see. How do we do that?" We couldn't get with the program. Then I think there was another one in there we explored for a while. But what we ultimately ended up with was a series of startups because after all somewhere in the process we ended up with 50 million in the bank. We acquired a DEC engineering team to make a multiprocessor. We had Formation. There was a group in Littleton that we hired and we made the Multimax, which was one of the early multiple microprocessors I defined as a "multi". It was right after Sequent and the other company who was first was called Sequoia. Anyway, it was one of the first multiple processors.

Hendrie: We were making multiple processors, but fault-tolerance was the story.

Bell: For fault tolerance, you had a clear focus. Anyway, ours was a general purpose machine. We used the National chip, we were a competitor with Sequent. Charlie Rupp came with us and he built a very nice, absolutely gorgeous terminal which was a 19" terminal that had multiple windows that you could connect to. So we'd go to these sales guys and say, "Well, you have to establish a channel for these. You have to make a market." They said, "We're mini-computer salesmen. We don't know how to make a market. Or sell terminals." So anyway, we didn't get very far with that. But it was a very interesting and quite a nice terminal. So the terminal was just sold by ourselves with a few on the side. The company ultimately sold a number of Multimaxes.

Hendrie: You built them and they worked?

Bell: We built them and they worked and there was a little bit of flack on that, and again, I went back and started doing more engineering. Somebody said they were trying to help marketing and whatever. As the design progressed, it got behind and whether or when it would work is unclear. Somebody said, “You know, if this thing doesn’t work, it’s your fault.” I said, “What? Okay. You mean I’m the chief technical officer, so man, I’ve got to go work on this product.” In fact, I was talking to somebody the other day of how the Mac saved my life. I had gotten back in with the engineering of it and the project simply wasn’t clear. The project was out of control. These were young engineers and they were enthusiastic and bright and all that and lacked discipline and management.

Hendrie: And hardworking.

Bell: And hardworking. But “What are we going to do next?” was how they were approaching the engineering. So basically, on a Saturday I went to New Hampshire and bought a Mac, because they had just started shipping MacProject. I came back, sat down one Saturday morning and started putting the project into MacProject. Then I came back. I bought my Mac on Monday and then I said, “Okay, we’ve got to get serious.” When we finished, it had 500 blocks and we had posted it on a large whiteboard and that became the project management tool. But then we did the standard project management stuff of “Okay, every Monday morning we’re going to have an update of the project: what did you do last week and what are you going to do next week and are there any changes to the schedule?“

Hendrie: Where are we?

Bell: “What are you we going to do?” So that was getting Encore going. We got a DARPA contract, which was based on a design because I was interested in building scalable machines. We proposed to DARPA that we’d build a 1,000-node multiprocessor. We basically took the Multimax, which is a 20-processor. Then you could put 20 of those together and put them in a hierarchy and then use the hierarchy to manage the coherent memory across all of the machines. So that was basically the proposal for what became KSR.

Hendrie: I was going to say, I’m hearing KSR in the story a little bit here.

Bell: Exactly.

Hendrie: Okay, memory coherence.

Bell: That was at a time when I believed in a religion where memory had to be coherent. I gave up on that religion in 1994.

Hendrie: But this isn’t 1994.

Bell: This isn’t 1994 yet.

Hendrie: This is 1984 or 1985.

Bell: Yes. We made the deal, I think, in 1985. We got a DARPA contract. I made the DARPA proposal. DARPA got granted the proposal. It was accepted after I left. I left in this case very early, because what happened was Henry and Ken got into an argument about the company and how it was going to be run and things like that.

Hendrie: Oh, really?

Bell: So fundamentally, the board fired Henry and so I said, “If Henry goes, I go.”

Hendrie: And that was that.

Bell: And that was that. That was simple. So I left. It’s too bad. I think that the company could have done all right, but in fact, it was just poorly managed. By the way, Julius Marcus for marketing came over to help. And Bob Puffer for manufacturing so it could have worked, but Ken valued friends over competence. His friends were just incompetent.

Hendrie: You had really good people.

Bell: We had really good people. And the problem was Ken didn’t know what good people were. After I was there not very long I sort of said, “Well, he’s maybe five milli-Olson’s as a manager.” Even though I may or may not have agreed with the way Ken Olsen managed, he did manage versus leave them alone. “Ken Fisher, what’s your management philosophy?” “Hire good people.” “Then what?” Encore survived much longer than I ever expected.

Hendrie: Didn’t it merge with a minicomputer company?

Bell: SEL. It was a different one. It wasn’t the one we ended up with. It merged with them, and then I’d say a few years ago the assets were sold to Sun which actually ended up with things like the memory coherent protocols and things where some patents were involved. Those were sold, too, as the motivator.

Hendrie: I see. I didn’t realize that.

Bell: Sun paid $150 million a few years ago, I believe.

Hendrie: So Ken Fisher is probably doing fine. As soon as he bought the company in Florida, I said, “I know what’s going to happen. He’s going to move the company to Florida, so he can live in Florida.”

Bell: That’s right. Anyway, that was Encore.

Hendrie: That was that. So you quit?

Bell: Yes.

Hendrie: Now what are you going to do?

Bell: So in January 1986 I didn't have a job. I had been in California. I was on the Technology Advisory Board and I was a board member at Silicon Compilers. So I went out there to the valley periodically. One of the times I was out there, I had met Ben Wegbright and Steve Blank and maybe a couple other guys. By the way, one of the things we also did in Encore was encouraging John Hennessey. "Hey John, how about becoming part of Encore and we'll fund you to do the MIPS chip."

Hendrie: Oh, my goodness!

Bell: So John still credits me with getting him to start MIPS.

Hendrie: Really? Did that actually happen?

Bell: John got it going. He went and just did it.

Hendrie: He got funded.

Bell: Not me. Mashey [sp?] and I don't remember who actually funded MIPS.

Hendrie: It got funded by a couple of venture capitalists out in California.

Bell: Exactly. John Mashey and John Moussouris I believe were founders.

Hendrie: I was going to say John Moussouris was the other founder.

Bell: Who owes me $100 bucks, by the way, for a technology bet. But anyway, that was another story. John got the company going.

Hendrie: Okay. Let's take a break. I'm running out of tape.

END OF TAPE 3, INTERVIEW 4

Oral History of Gordon Bell

Interviewed by:
Gardner Hendrie

Recorded: October 19, 2006
San Francisco, California
Corrected 6/27.08

Interview 4, Transcript 4

CHM Reference number: X3202.2006

Hendrie: Why don't we take a pause in your career because I think you're going to move to the West Coast and talk a little bit about your involvement or what happened in the genesis of the Computer Museum. Maybe tell that story a little bit at least from your point of view.

Bell: Right. You saw the e-mail that I sent out in the last couple of days. Given Bill Gate is going to come into town, I just sort of wanted to find out when our first connection with Bill was and it was after I had left when the museum and Bill had connected. That was the main thing. Well, anyway, going back to the genesis of the Computer Museum itself was the first really written public item was a little brochure, a six page fold out brochure, of a thing called computer generations. I had developed a theory of how computers formed based on technology generation and that these generate different kinds of machines. I had worked that kind of as a hobby way back to 1620 or so with mechanical devices and probably you can even go back to ropes and things like that, stones and ropes for records. But anyway in various times I had gotten interested in the problem of collecting artifacts. Ken Olsen was also particularly interested in that too. Ken wanted it from a perspective of the machines that he had worked on, namely Whirlwind and TX-0. So in the early '70s he had acquired the Whirlwind and the TX-0. A guy by the name of Bill Wolf had acquired the Whirlwind in the late '60s or so principally for the software. But of course there's not a chance in hell you could ever make that machine work again or afford to power it.

Hendrie: He put it in a building in Concord.

Bell: Yes, but I don't think he ever got it working. There were just too many cables and just too expensive and the power bill would kill it. It was at the wrong time because transistors were coming in. Ken was able to buy the machine along with Bob Everett who headed Mitre and was responsible for Whirlwind. It was put in a storage in Concord in cabinets, in storage crates, and then in fact I think also it ended up in some trailer in one of the DEC parking lots by the pond at one point in time, but that was kind of one of the geneses. We had also acquired the TX-0 through a collection of circuitous transactions because it was government surplus and initially obtained by a junk collector who wanted the gold. That was another thing. That was Ken's kind of first transistorized machine.

Hendrie: I didn't realize that he had also . . .

Bell: Yeah. Wes Clark was the architect of the TX-0 and the TX-2 and then Ken was responsible for the circuitry and some other things too. So anyway, that's how we got those artifacts. When I came back to DEC in '72, I had started to acquire various artifacts and thought, yeah, it's a pretty interesting time. Let's really collect those logic and other items, and I started collecting things in my office. I think at that point in time I had a piece of the DEUCE and I don't remember how I'd gotten it. It was the early vacuum tube machine that I had worked on in Australia I discussed earlier. And then in '75 was the first kind of record of a museum and we called it the museum project. I don't think we can find those memos, but by the way we might be able to now that Ken. . . although it's not clear how Ken's files are or whether that's searchable, whether we can find them through his files or not. Surely Ken's archives have items on the formation of the museum at DEC. Unfortunately, mine don't.

Hendrie: At Gordon College.

Bell: Yeah, about the museum. I think this would be a good point to go and try to ask them for anything that had to do with the museum.

Another origin came about when I used to go to Washington periodically for the National Academy of Engineering meetings. The first time I went to the Smithsonian I was really pissed that they had an absolutely terrible exhibit. They had some old computer, a Burroughs that I had never heard of, in one corner and they had some junk lying around. I tried to find out who was responsible and a lady by the name of Uta Mertzbach was responsible. She was a mathematician who was a Scheutz (engine) Scholar. I never could get anywhere with them and so I said, "Gee, we got to collect all this stuff before it disappears." So I basically said to Ken, "Well, we've got to do this, we've got to have a museum to collect all those things", and I had given some talks on my generational theory and then made the whole brochure. A lot of that was kind of still riding on the inertia that came out of the book that I had written with Newell on computer structures. And so we ended up taking the closet in the main Mill building 12-1 lobby and putting a glass front on it and putting all of our artifacts from my office in there.

Bell: And so that was the beginning of the museum. It was called the museum project. Then it moved to the Marlboro lobby. I don't remember what the date was. We had the whole lobby. It was a beautiful building. The RCA computer company built this Madison Avenue building in Marlboro that at the time was in the middle of nowhere.

Bell: And so we took all of that space and it became the Digital Computer Museum. It was really strictly a play on words as the Digital Computer Museum and it operated there. Of course it backfired and we had to call it the Computer Museum. Gwen at that point decided she wanted to run the museum and took it over and took that as her project and did all the exhibits. Well, we used DEC industrial designers to do design stuff there and then we pulled everything together and meanwhile we'd been given a piece of the TI ASC. We had been given other machines and other artifacts and this was while it was still a part of DEC. And then Gwen was able to make it a public museum as part of DEC in sort of the '81, '82 time frame, which was no mean feat because of the IRS kinds of requirements of sort of hands off and all of the other stuff that you have to do and how you feed money to it and who's on the board and stuff like that. And then we got board members and what have you, renamed it to be The Computer Museum and then it moved to Boston.

Hendrie: That had happened . . .

Bell: In '83 or so when I was leaving DEC, it was moving to Boston. Anyway, that had been a deal that Gwen had cooked up with Mike Spock who was the head of the Children's Museum. He came to her and said, "Gee, the other tenant as part of the Children's Museum (which was an auto museum) is folding."

Hendrie: That's Larz Anderson.

Bell: The Larz Anderson Auto Museum. Yeah.

Hendrie: The museum that had come in from out in Brookline.

Bell: Anyway, so we got the space and that was the transition about the time I left, and then that was a potential threat too because I was no longer part of DEC. But Ken and everybody maintained their responsibility to the museum and it continued. DEC was the dominant provider in Boston and sponsor.

Hendrie: I wanted to make sure we covered that. Now you've left and you've finished Encore and what year are we? Do we know?

Bell: Yeah. This was '85, December of '85 or January '86. I decided at that point I was going to go live on the West Coast. I had been out there on a couple boards. So the MIPS had been happening, and one of the guys out there I said I saw was Ben Wegbeit with Steve Blank. And so we said, "Well, let's make work stations based on the MIPS or build a simple computer" and basically that was the genesis of that startup that was undoubtedly the most fun I've had as a startup.

Hendrie: That was how that . . .

Bell: . . . that started. And then Allen Michaels got involved because these guys were all the founders of Convergent. The company got together in January of '86 and it was called The Dana Group because it met on Dana Street in Palo Alto where Ben Wegbreit lived. It was called Dana Computer and it remained Dana for a while until the Dana Corporation . . .

Hendrie: Came in and said . . .

Bell: . . . came in and said, "Hey, you guys, get the hell off of our name!" So it became Ardent, and notice that Ardent starts with an 'A' and it's Allen looking through the dictionary for a name.

Hendrie: I see.

Bell: What are we going to be and it was literally we have to change the name tomorrow morning. We even had a contest. Steve Blank wanted to call it Defiant.

Hendrie: It was based on MIPS. When did you meet Hennessy?

Bell: I have known Hennessy going back to while he was doing his work at Stanford, so I had known him from DEC and Stanford Days. I knew what the project was and that's when I said the world needs a chip like that because we're stuck in Intel hell.

Hendrie: Intel hell and Motorola.

Bell: Yeah, right. Motorola was really doing that but at that point the RISC thing was just ripe. It was just right at the right time and you make the compiler and then you had the right caching structure and everything worked exactly right.

Hendrie: Exactly, and everybody followed and nobody succeeded.

Bell: Right. Yeah. In fact, the irony is I wrote a paper for some consulting company that sold position papers, and I think that was maybe '92 or so, but I think I'd given myself 10 or 12 years that there are six platform companies and at most there can only be three and so HP went down first.

Hendrie: Now you get to guess which ones.

Bell: Yeah, right. HP and DEC and then Sun hung on. The first two were clear. It was Intel and IBM and then the question was is somebody else going to be a survivor too.

Hendrie: Tell me a little bit. Did you recruit any people from your previous history at DEC into Dana?

Bell: Well, I didn't stay very long at Dana and I don't remember the exact chronology. I think at the same time Erich Bloch had contacted me to go to NSF. And the Dana guys were doing just fine. I was helping recruit there and doing architecture and I didn't want to manage the project and so Erich had recruited me to start up the National Science Foundation Computing Directorate. I did that starting in the spring of '86. So I went to NSF then in early 1986 and that was a startup, too, as I took all of the funding areas from other parts of the NSF and formed a directorate called CISE for Computing and Information Science and Engineering. The 2008 budget is maybe a half a billion dollars of computing research money, maybe more. It's one of the largest directorates, but I pulled together some of the hardware engineering, all the software engineering, and the supercomputing centers that were just forming that had network as part of them, pulled all of those together and formed a directorate, AI and I think Library Science. There's a bunch of different areas, and that was the beginning of the computing directorate. That was really a fun time. I enjoyed it. There were alternate days I felt like somebody could actually run the country, but it was not nontrivial but I got an idea of how things worked and what the executive branch did. And Reagan was the president then, and the problem with him and I'd say the problem with most of the presidents is that they don't have any understanding about science or engineering or where does all this money come from and certainly science figures in to all of it. There are a lot of hugely contentious issues and he's not willing to hear conflict. The last thing somebody wants to have working for them is to have this person who's got all these problems and certainly that's an area where you've got all of these cross agency problems which are hugely complex. Virtually all of the Presidents' science advisers have gotten fired, or neutered.

Hendrie: Because . . .

Bell: Because there's something that has to be dealt with.

Hendrie: Science is not politically correct.

Bell: Yeah, and that they demote it or certainly they're way down in the pecking order of what to do because all the power resides at the cabinet level. It's like Rumsfeld having to argue with the Department of Energy. So here you've got these two very powerful kinds of guys going after each other and then you throw in these other little bureaucracies like NASA and the National Science Foundation. The wonderful thing about it is that NSF is a thing called an independent agency like the Veterans Administration so it just kind of sits over there. It doesn't report to the executive branch and so it's good and bad. It kind of gets money from the committees per se and so you've got a huge political problem there because it's the President's budget but yet in this funny way he's not quite . . .

Hendrie: . . . a powerful advocate.

Bell: Right. There's no line management going back to him. In fact, I just had a discussion with one of my friends today about that. I'm on the Department of Energy Committee to look at supercomputing. Well, it turns out another old friend from the Computer Museum is the Secretary of Energy, Sam Bodman.

Hendrie: Yes, Linda's former husband.

Bell: Well, anyway, Sam Bodman has obviously perhaps given money to the Republicans or to Bush, in addition to being competent and knowledgeable about energy as a former MIT Chemical Engineering Professor. So the Department of Energy is reeling in money and everything else for overall support. And so our little committee is looking at supercomputers and what are you going to do with all of those machines, and meanwhile they've just gotten their charter that basically looks to me to be the same as the National Science Foundation's in spite of the department's charter to focus on energy. So they've got so much money that they're going to have computing for whatever you want. It's no longer a mission program but it's, "Hey, if it's just support of good science we'll give you computing."

Hendrie: That's pretty interesting.

Bell: Very interesting and so there's competition now at the federal level for who funds what science, as if we didn't have enough intrigue before.

Hendrie: Has the NSF fundamentally taken the place of DARPA? DARPA really did this role of funding computer research.

Bell: Right, and in fact what has happened is that over the last few years NSF had to absorb that not because it wanted to but more because DARPA has really abdicated anything but mission funding. And in this regard I think there are all the people in computing. . .The universities are not getting money from DARPA now and principally because that's been a decision by the director of DARPA. It's a tragedy, a real tragedy for the country. This is bad, this is really bad. The current director of DARPA, a guy by the name of Tony Tether, fundamentally is giving money to the Beltway Bandits undoubtedly for PowerPoint slides. Over time DARPA was having to become more political vis-à-vis the universities and so it had gone through a period of a very small number of very large projects, small number of schools getting funding to being more egalitarian. It looked more like NSF in funding too many, in my view, too many projects, not enough . . .

Hendrie: Not enough in-depth projects.

Bell: And then now it's that there's none at all, so that's kind of where things are with NSF and how it's gone and I think we're in a pretty tragic sad state at this point in time with DARPA, and now we may be in an equally bad state with the Department of Energy taking over more science per se from NSF.

Hendrie: When you got to NSF you had a charter. Erich was the director to form a computing division. What did you do?

Bell: This was a sort of true tree graft of pulling together a bunch of programs from other divisions. So I basically rearranged an organization, and I'd say the part that was probably the most contentious was what are you going to work on? I outlined a program basically in parallelism, that in fact our biggest problem with computing was going to be how to exploit these machines that were going to be coming into existence over the next couple of decades. And so that was one of the cores of the thing if I go back and look at the statements I have made. The other thing that happened was what to do with the supercomputing thing because we had just. . .That goes back to the VAX because the computing guys had sort of said well, I had wiped all these centers out, now what we've got to do is re-create centers because we're not getting enough computers, we can't run big jobs on our VAX. By the way, it might have been totally different had the 8600 come out. My belief is it would have or maybe not, because in fact what happened was the Cray in '76 or so and so the Cray-1 and Cray XMP and stuff others like that were coming. Those were very powerful machines that are different animals.

Hendrie: They did have a big leg up on big . . .

Bell: Yeah. We simply couldn't do large . . .

Hendrie: . . . scientific problems.

Bell: You couldn't do those. So what happened was when I got there the division director was out opening centers and getting more solicitations for centers, and I looked at the demand for time and then I said, "Wait a minute, we don't need any more centers, let's get these running." So I ended up saying, "Well, I think we needed about four centers, not six centers."

Hendrie: At this time there were . . .

Bell: There were six.

Hendrie: And he wanted to open more.

Bell: He wanted more and then he also wanted centers because to get the program going in the very beginning he went out and bought computer time at some of the other centers and so these guys wanted to be centers too that would get NSF support. There were a few million going to Minnesota and other places and now we had real centers being formed, one at San Diego, one at Pittsburgh, one at Cornell, one in Illinois, and the von Neumann Center was opening up using the CDC Star. They weren't able to deliver it and they wanted to switch to Cray. And I said no, they got funding based on the fact that they had a different computer and now that the different computer folded I said well, let's fold the center. And so basically after I left the center ultimately was closed down and then Cornell was an IBM center. They were funded significantly by IBM and they had the 3090 vector. IBM had put in several of big vector 3090s, vector machines, there. And so one of the things I did was decide that well, we're going to run Unix on all of these machines so that we can now have some application compatibility.

Hendrie: Affordability of some of these applications.

Bell: Absolutely, the big problem here was apps. I remember writing a position paper to Larry Smarr at Illinois and to Sid Karin at San Diego asking them to switch to UNIX. They had Crays and they were running one of the Livermore time sharing systems or the DOE's, and I said, "Hey, you guys, you're going to run Unix, Cray- CTSS, the Cray timesharing, the Unix timesharing system." Ultimately they did but there was a lot of gnashing of teeth.

Hendrie: And screaming.

Bell: "You're telling us what system to run?" And I said, yeah.

Hendrie: You don't have to do it but the funding is likely to dry out.

Bell: Yeah, I invoked the "golden rule". However, that is small compared to the decision to take networking away from the centers and create a separate part of CISE to handle NSFnet. They had the networking. They were each building a network infrastructure for the U.S. to support just their own users. As you can imagine, we were creating 5 independent star networks. Why did we need networking? Well, it's for our supercomputers. That's the way it was sold more than any other way because we almost understood the application. Yeah, we got a network and here's the Illinois network and . . .

Hendrie: And it starts at the centers.

Bell: Yeah, and so I formed a network division and then that was the division that actually worked on the NREN Plan. NREN was the plan for internet. That plan was used to get funding and we funded regional networks from that group to tie in to the backbone. With a backbone network we got all of the other agencies involved in doing all that. This was in response to the Gore Bill [Supercomputer Network Study Act of 1986]. Gore, NSF, and his staff had gotten Gore to write a bill that sort of said the NSF should write a plan for . . .

Hendrie: National infrastructure.

Bell: Yes.

Hendrie: Or the information superhighway.

Bell: I led the cross-agency group from DOE, DOD, NIH, NASA, and NIST that put that plan together. We had a meeting in 1987 in San Diego. I remember getting up the last day of the meeting in the morning and saying here's the plan. There were about 300 people there. We had had sort of subcommittees and everybody was talking about different things and I had listened for several days, went in to listen to all the technology and other ideas for research. Well, what's happening, what's the state of art, what is it going to be, and it was clear that Optical was nowhere in the immediate timeframe. So I drew a diagram that had these five waves of bandwidth versus time going out to 2000 or 2005 and then I said this is the plan, this is what we're going to do.

Hendrie: This is what we're going to do.

Bell: Yeah, and note there's no research in networking per se until we can go to Optical networks. This is an engineering problem, we've got to now get this managed and we started building the network out and then ultimately there'll be fiber that will kick in around early 1998, when we started using higher speeds, but until then it's just network engineering. The first thing was phase zero stabilizing what we have because we don't got nothing now and then put in T1 and make that work and then we go to T3 and then we go into optical. So that set of curves and those dates turned out to be almost precisely what's ended up happening. I don't remember how we made the deal, but Bob Kahn was there and I said how about becoming the funding instrument for all of this because NSF can't deal with all of the funding and making this happen, just go out and get it done. And the first deal was that Michigan took over the management at the network center for the first wave, and I think they still do most of that and they work with IBM who supplied the initial routers and gateways and was responsible for the network.

Hendrie: For the basic backbone and this is the national backbone.

Bell: The backbone's been replaced but with higher speed links. Now NREN 2 is out and so we are starting down the path that seems to be back to switched optical links. I led the group that developed a three volume plan or rather the summary plus background of two volumes, well, the three volumes. The plan is of what we're going to do and why we're going to do it is about a 15-page report that went to the President's science advisor, Bill Graham. I tried to get him to write a really strong computing infrastructure charter, but he didn't want any part of that. He was a very weak guy because he had to interface with Reagan or someone on his staff. The way it worked was there was a cross agency group called FCCST, Federal Coordinating Committee for Science and Technology, for IT and so I led the networking part. There was somebody from DARPA leading the people infrastructure and their high performance work, Jim Decker led the supercomputing, and I put the networking part together and so we had a great plan. I walked in with my friends and said, "I want to respond to the Gore Bill. Here, the Gore Bill's due. We're going to send this off and have it done." And they said, "Like hell you are, you can't do that, it's too good, no, you'll make us look weak and disorganized" like they were. And so we waited another year to create the high performance initiative, a little blue book that came out. In fact, Steve Squires drove the production of it. He came in from the DARPA because they were working on high performance computers and also CS infrastructure.

END OF TAPE 4, INTERVIEW 4

Oral History of Gordon Bell

Interviewed by:
Gardner Hendrie

Recorded: October 19, 2006
San Francisco, California
Edited by cgb 6/28/2008

Interview 4, Transcript 5

CHM Reference number: X3202.2006

Hendrie: Was that the principle activity or most significant activity that you were involved with at NSF?

Bell: Yeah, I think the thing that we did was really establishing networking, getting the supercomputing program on the right foot, even though it meant there was some blood flowing, and switching, going to a more universal standardized environment. And certainly the NREN response to the Gore bill was really important and getting that program off and putting the parts together was really great. Also, the overall program and organization I set up, along with the goals for CISE that included a focus on parallelism were important.

As a by-product, I established the Gordon Bell Prize for parallelism in 1987 that is over 20 years old and currently funded at $10K/year. I have funded the ACM to run it for another 20 or 30 years, too. The prize is administered by the Supercomputing conference and papers are presented there. In 2008, the parallelism is about 200K to deliver almost one half petaflops. In 1987, that was one half a gigaflops with 1K-fold parallelism. So I consider the prize to be very useful that came from being there.

At that same time, I was commuting to the west coast because the Dana group was building its graphics computer. I would go out every month or so on a weekend or to a board meeting and interact with those guys. And then I don't remember when it happened but, at one point, Allen . . .

Hendrie: Allen Michaels?

Bell: Allen Michaels called me and said, "I fired Ben. You've got to come here and run this project."

Hendrie: Oh, my goodness.

Bell: I said, "Oh, wait a minute."

Hendrie: Why did you do that?

Bell: "Why did you do that?" And he said, "Well, he's not getting it done" and then he said, "What do I do?" And now let's just talk about the Ardent group, it was first Dana, then Ardent, and the best group of engineers I have ever met in one small . . .

Hendrie: In one spot.

Bell: In one spot. I mean, we built the supercomputer, a parallel vectorizing FORTRAN compiler, blazingly fast graphics, the whole thing with this team of about 50 engineers.

Hendrie: How many?

Bell: 50.

Hendrie: 50, okay.

Bell: Everyone had done spectacular things both before and afterward, you know? For example, Cleve Moore Moller was the founder of MATLAB. Okay, "What do I do?" And I said, "Okay, Allen, I'll come out in a couple weeks but here's what you do. You take all those guys and you put them in a room and you don't let them come out until they have a schedule that they all agree to." It's that simple. Once they have the schedule, have them meet weekly and ask everyone what they did the last week and what they would do next week. There were about a half dozen groups: the computer hardware with about four complex boards with very complex custom chips for the vector processor, memory, i/o, and graphics, graphics software, the O/S, and compiler.

Hendrie: Yes, okay.

Bell: And so fundamentally that's what he did. That's what happened. When I came out, I think the schedule was being formed at the time, and then I interacted a little bit as they were doing it and ultimately got a schedule and I introduced the term schedule fantasy factor and said, "We have to manage to the schedule and that's what's going to happen." There SFF was about 1.8, i.e. it was taking almost twice as long to do things as they predicted. What happened was classic with I'd say a super bright engineering manager. Ben had projected his schedule onto everybody somewhat by intimidation.

Hendrie: They project their own ability.

Bell: It was a way of extracting commitment. And so what Ben was doing was he had, like, ten guys reporting to him and he managed them one on one and he was the gateway and he clearly could do that. But he did it sort of by intimidation. He'd bring them individually into his office and said, "Now, you got to do this, this, and this" and, "We need this by this and that, will you do it?" and "No, I can't have it done," "You got to have it done," he asked, "Next?" There wasn't co-ordination of the parts except through Ben. Worse yet, with that kind of management, you don't have any coordination and then also you had no commitment.

Hendrie: You have no commitment. You have unrealistic schedules. So you got rid of that, and it worked.

Bell: Yeah. Back to basic engineering management, just like I did at Encore when I had to take that one over.

Hendrie: Now, tell me, what was the fantasy schedule thing?

Bell: I don't remember when I (or if it is something someone else came up with) had come up with the name schedule fantasy factor. You take any future milestone time, you look at what the commitment time is for that milestone, and then divide the actual by the committed and anything greater than one is a fantasy. I think their schedule fantasy factor, at that time was 1.8, so it was taking them 1.8 times longer to do something than when they said it was going to take. After they made their own schedule together they got it down to 1.0, a rarity for engineering.

Hendrie: Exactly.

Bell: And so . . .

Hendrie: There's every reason to predict that it's the same factor forever.

Bell: It's a constant for a group. Yeah, that's exactly how optimistic people are. You have to be realistic.

Hendrie: It's got to be realistic.

Bell: And so they got it right down to one. Actually, we were on time and, by the way, any time anybody comes in with less than one, I mean, to me they're heroes. The only time I ever saw that was the Network Appliances startup team got a product out in 0.8th the time they committed.

Hendrie: Wow.

Bell: And so at the time there was a question about funding. The archangel who was leading the group who was dealing with a lot of the funding wanted to hold them up because they hadn't achieved their funding milestones, they didn't have the VC yet and so some money was due and I caved. I said, "Hey, you know, let's not hold them up" and he wanted to use that to extract more flesh and so I said, "No, come on, let's get on with it." And so all the angels were really pissed at me because here's a guy that caves on funding.

Hendrie: Right. But they'd done just a spectacular job.

Bell: Yeah, exactly. They should be rewarded. I Said, "I'll never be a VC."

Hendrie: Okay.

Bell: So I stayed at Ardent and got it to ship.

Hendrie: About how long was that?

Bell: We shipped in '88 which is still amazing for the project, the three-year project, and then we merged with Bill Poduska's company.

Hendrie: Yeah, it was Stellar.

Bell: Stellar and Ardent to form Stardent.

Hendrie: Yes. And they both thought they were their own worst enemies, and if the merged they'd be okay, only to discover Silicon Graphics was actually the enemy.

Bell: Yeah. It was pretty bad but, anyway, I still get Christmas cards from Kubota, which is the Japanese company that funded both of them. I left and then did angel funding and consulting until '95.

Hendrie: So you left, maybe this is '88?

Bell: '88 or '89, yeah. I was enjoying life and just floating around the valley and was on Suhas Patil's board at Cirrus Logic and other things including consulting at SUN and Intel. I tried to get Intel to build a vector processor. The net result of this is during the period 1983 till 2007, I have invested time and/or money in over 100 companies or about four per year.

Hendrie: Wow.

Bell: Yes, from '83 'til now. And then one of the other things that happened during that period while I had worked with Heidi Mason was the creation of the Bell Mason Model and Diagnostic for startups. Heidi had been the editor of a book that I had written at DEC. She was out here doing marketing -- she had started a marketing company and then we started collaborating on consulting. She had me help look at some companies regarding technology. That's when we created the Bell Mason Diagnostic that we have licensed to Coopers and Lybrand and others. The idea for it came to me while going back and forth to Washington while I was an NSF in 1988. The idea is that there are 12 dimensions for measuring a startup company. You plot the state of the company on a 12 dimensional kiviat diagram. Excel calls this a radar chart or polar chart.

Hendrie: Well, tell us more about that. What is that diagram?

Bell: The whole idea of evaluating startups and projects had its genesis back at DEC and in the early '80s.The popular idea that was being pushed in the '80s was the idea of expert systems and heuristics. I wrote a little paper called "Heuristics for Building Great Products" and said management and business is just a set of rules. So I cooked up about 40 rules and segmented them into about seven dimensions. You had product, market, team. After I left DEC, I got involved in all the startups. I like startups because of their simplicity. When I left DEC, I basically said, "I'm not going to ever go back to a large company." You know, these startups are so simple because they're just projects. They are not companies! Think of it that way. So, by this time I was commuting, and Heidi was talking to me about all the marketing problems that were going on and the companies she was advising -- it's their plan, it's the team, it's the sales, etc. We were doing all this stuff, and then I had enough knowledge too, because I was on the Silicon Compilers Board. John Doerr had founded that company and was the chairman. And so I had a pretty good idea of how startups really worked. And then I started writing about it and I cooked up the idea of just 12 dimensions. I liked that number, and then the idea of heuristics was how you look at and measure them based on things that they should be doing. So we had a grading system, we had the dimensions, we had what I thought was the orthogonality. I haven't changed my mind at all from all that. And so I started getting more experience with it, started writing. I was fascinated by this the idea of stages. I found out the stages didn't mean anything. I read all the academic papers and the stages only meant we ran out of money. We need more money, that's what the stages were.

Hendrie: Yes, those are the stages. Stage 1 is how long the first money will last.

Bell: Yeah. And I then used or rather took the software engineering model to say these are really the stages of the product, these are events that you can measure. So Heidi and I cooked up the diagnostics, which has about 1,000 questions over the 12 dimensions. There are actually four stages but then sub-stages and thus specific times at which you can really measure companies. And then I wrote the book, it came out in '91, and, again my old friend John McNamara came to my rescue. I started writing the book and I had a publisher -- actually Addison-Wesley -- and they tried to get me in a co-author. But all the co-authors were doing was re-editing or rewriting stuff I had written, and John looked at it and said, "No, its written wrong. Just turn it 90 degrees. A chapter per dimension as opposed to a chapter per unit of time." And so I changed all of that and then I wrote the first draft and then John rewrote it and then we ended up with a really good editor at the end, too, and she did a lot.

Hendrie: Good.

Bell: I like the way it reads.

Hendrie: Good. Hadn't John written, way back when, a book on communication?

Bell: Yeah, John had written a book on communications at Digital and he liked to write. In 1978, Craig Mudge had convinced me to write a book about the DEC computers. And so I started the book with Craig and found out that Craig needed more help to get it done. He wasn't writing at the velocity that I write. And the book would never get done or would certainly take a long time.

Hendrie: He'd never get his parts done?

Bell: So Heidi and I called John and I said, "John, you got to help on this book" and so John came and helped get the book Computer Engineering done.

Hendrie: Very good. Excellent.

Bell: Yeah, so the key is knowing who to ask.

Hendrie: Yes, exactly.

Bell: When you get into a bind.

CGB 6/29/08: *Let me digress. I have been incredibly talented, also called lucky, in finding people to work with or who have adopted me. Career claim: lucky & befriended, starting with my parents.*

- *1934 Parents providing a great home; allowing me to be the oldest person at MSFT;*
- *High school math Mrs. Strong and science teacher Bill Heinberg;*
- *1952 MIT several extraordinary teachers & a few friends;*
- *1958 Fulbright Scholar, Australia. Gordon Brown, an Aussie, who headed MIT EE Dept and meeting Gwen Druyor;*
- *1959 MIT Speech Lab Ken Stevens;*
- *1960-83 DEC (#80) Ken Olsen, Ben Gurley, Harlan Anderson, Dick Best, Alan Kotok, Mary Jane Forbes, Dave Cutler & many others incl. Bill G;*
- *1966-72 CMU Ivan Sutherland introduction, Al Newell, Al Perlis, Rod Williams, Herb Simon, Dan Siewiorek, Sam Fuller, Raj Reddy, …'*
- *1976–Computer Museum, Computer history Museum, Gwen Bell;*
- *1983-2008 100+ startup entrepreneurs e.g. MIPS, SGI, NetApp, Cirrus Logic, John Sanguenetti, Steve Blank; Heidi Mason and I collaborated to create a methodology for evaluating startups.*
- *1986 National Science Foundation (CISE) Eric Bloch;*
- *1986-1989 Ardent Graphics Supercomputer, 35 great engineers;*

- *1991 Microsoft Research—Nathan Myhrvold, Rick Rashid…;*
- *1995 Microsoft Research—Jim Gray & Jim Gemmell & Vicki Rozycki; and Sheridan Forbes*.

Hendrie: Okay. We got off on that book but you're eventually going to get to Microsoft? Have you gotten there? Are there any other stories on the way?

Bell: Yeah. I think, you know, there are 90 stories on the startups, and what I did on those, by the way, I ought to do something with them. I have about four hours of some very pithy remarks on some of those because I basically took all of the shirts and cups and coats and sat there before a video camera and rambled on, "Here, see this is the story of this $185,000 coffee cup and what I learned."

Hendrie: Yes.

Bell: Yeah. I did the startup and consulting gig until '95 and then I went to Microsoft. But in that time, in '94, I ended up for whatever reason meeting Jim. I had never met Jim Gray before, and so Jim and I met in '94. He came to my house and we talked about what computers were going to be like and we decided we had exactly the same religion about the way computers were going to be built and he called them the smoking hairy golf ball or bricks and I had just written a paper on networks and nodes which we changed to SNAP, which is Scalables, Networks and Platforms, and the only thing we got wrong was that we thought ATM was going to be the interconnect.

Hendrie: The connecting fabric.

Bell: The connecting fabric so that we could deal with the scaling across the geography, but ATM didn't happen. But everything else was the same and that scalables were going to take over all these other classes because I had refined my computer class formation theory. All different little classes and niches are going to be wiped out and replaced by scalable PCs. They are going to wipe out workstations, PCs are going to go right up through to minis and mainframes and eventually supers. They'll be the dominant structure and a scalable network. Then Jim wrote that up and he gave a talk about it in Berkeley in three McKay lectures. That was kind of the beginning of our belief.

In '91, I had met Nathan Myhrvold -- I think he called me, probably Cutler gave him my name. I had convinced him to give a talk in Geneva when I was giving a talk on startups because I was giving a talk about my book. We had two days on where computing was going. So I got Nathan to come and speak. We dined in the Two Star Geneva restaurant. Nathan said, "Okay, now come and help me start research at Microsoft."

Hendrie: Ah, okay.

Bell: And so I went to Microsoft and Nathan and I noodled about how it was going to start. I don't remember that he ever asked me to run it or not but at least I know I said, "Let's find somebody really good to run Microsoft research." And I've still got my notebook that has the list of people, qualifications, and our grades as to their suitability.

Hendrie: Oh, very good.

Bell: And the pros and cons of who we were going to hire.

Hendrie: Yes. Very good.

Bell: And then Nathan and I then went off and went to Bell Labs, met some people there, we knew various other people and then we went to CMU and Rick Rashid was at the top of our list. And so we went to Pittsburgh and met Rick who was much better than what we had thought because he had kids and the kids were using computers and he had appreciation of all the subtleties of consumer electronics. The family came to Redmond and then said, "No." He met Bill and said, "I'm doing just fine. I'm doing the Mach operating system and I'll just keep doing this." And so I told Nathan, "First off, he can't say no. What is this?" And so I wrote a four-page letter to him and his family saying you can't say no, don't you get it? This is the best job in the world. You can always be a professor. You'll be an even more valuable professor if you do this. But here, come and do this. This will offer you an incredible amount of financial freedom and, if you ever want to do anything, you can.

Hendrie: You can do it, yes.

Bell: You can do it. I started the technology advisory board for MSR and we got the same guys who are on it now -- Hennessey was on it, Andy Van Dam, Raj Reddy, Ed Lazowska. Hennessey resigned ultimately, and then Richard Newton who is now deceased was a member.

I've been happy going to the TAB meetings and the like, and then Jim and I got together in '94, worked for awhile and Jim had been consulting and then he called me and said, "I can't deal with this. I just need a place to work and a project. This is just not any good for me." So I said, "Well, okay," so I emailed Microsoft and I said, "Hey, we got to hire Jim and set up a lab." And so they took him immediately. It turned out he had been talking with them for a year or so, so probably all I did was to crystallize the charter.

Hendrie: It didn't take very long.

Bell: It took, yeah, about a nanosecond to hire Jim, and then Jim calls me, he says, "Okay, you've got to be part of the lab. You need a job, you know, your life is just floating around doing whatever. You need discipline". So he basically convinced me to go to Microsoft. In August '95, I joined.

Hendrie: So you've been there.

Bell: Yeah, I've been there ever since.

Hendrie: Very good.

Bell: And then his first question was, "What are you going to do?" You know, he hired me and then the next day he said, "Okay, what's your project?" And so I said, "Telepresence," because I'm going to stay here and I want to work at home. And I started out in that and then I, fortunately, hired Jim Gemmell, which was great, and Jim worked on various parts of telepresence and scalable multicast. One of the things was that I had us acquire a company doing telepresence. We gave up on it in '97/'98 because our protocol stack is too slow, the machines aren't right yet and it's going to be a long time before he gets there. There's just too many problems at this time. Interestingly, it has taken about 10 years and Skype is probably the most important catalyst.

And then I fell into the current project in '99 of deciding to scan all my bits and that got me onto this path of cyberization of the world. I think I've almost got everything that I've been working on, capturing all the personal bits I can find since 99, and building Bush's Memex that we call MyLifeBits..

Hendrie: Excellent. Do you have any comments to make about all the things you've done, you've accomplished, you're sort of the proudest of?

Bell: At one level, I go to the museum affair that we did like Fellows the other night and I thought, "Gee, that was really good. It's an institution." And, in fact, it's beyond a project, it may live forever and it's a great thing. So the museum is very high up in that list in spite of all the contortions that it went through to get here. It's something I feel proud about.

Hendrie: Yes, exactly.

Bell (6/28/08): It is critical to add here that if Gwen hadn't decided to do the museum at DEC, none of this would have happened the way it has. The Computer Museum in Boston was critical for collection and to get some passionate people involved that have remained as such including yourself who has been on the board from almost the beginning, Dave House, Grant, and John Shoch, and undoubtedly others I forgot. Ironically none of the Boston trustees have remained engaged.

Bell: Because, in fact, sometimes you kind of look at this thing and say, "Wow, that doesn't seem like that long." But on the other hand, it's over 30 years ago that was rooted in 1975. At one point I thought, well, if we could have a permanent building by '95 or so, something like that, that would be fabulous. I guess I'd say it really exceeded my wildest expectations. I think it's a tribute to just a few like you and Len Shustek and then the way we actually were able to get the money and the building at the right time, that timing was just spectacular because that gave at least five years of boost and probably ten. Having Bill contribute is a big deal, too, to get his support.

Hendrie: It certainly is.

Bell: Because that's all time, I mean, that's all trading money for time.

Hendrie: Exactly.

Bell: So the museum's got to be way up there. I always feel bittersweet about DEC because of wanting it to live forever. And that it should have lived.

Hendrie: It should still be here?

Bell: Yeah. And it sort of makes me very angry. Actually I've gotten over it mostly for example, you know, Dave Cutler had a similar feeling. So you get angry at these egos or incompetence that had it happen that way.

Hendrie: Yes.

Bell: But you certainly see DEC as a great time, that it was a great experience and you have to live with yourself, you have to take a philosophy away from that of, well, okay, at least a lot of good people were trained. I guess I tend to look at everything as kind of projects in a way and some are more important and lasting than others. That's basically how I conduct myself -- as an itinerant project engineer.

Hendrie: Yeah. Which ones . . .

Bell: I like writing. I'm in a quandary now trying to decide what to do next. When I was doing speech, research I quit it because it was a 20-year project. Well, it's a 40, 50-year project. Memex or MyLifeBits is sort a of brain augmentation or memory augmentation device, recording and augmentation and storing and all of that at a personal level, a person's life events and knowledge so I can't even see an end. There's no end to the potential for it.

Hendrie: Yeah.

Bell: So in a way I feel good about finding a project that has no end. On the other hand, it's a little frustrating as an engineer.

Hendrie: It isn't something you've done before.

Bell: Right.

Hendrie: All my projects have had ends.

Bell: We can do this in two years, three years, six years, let's go do it. I've loved all the projects. I mean, it has been frustrating at times but that's the way all projects are or should be, at least. Otherwise, it's not a project.

Hendrie: You probably didn't do something, would take a big enough step.

Bell: Right. Exactly.

Hendrie: It wasn't hard. Alright.

Bell: But, anyway.

Hendrie: Good. Well, thank you very much, Gordon, for your patience.

Bell: I agree. This is amazing, that we got to convergence.

Hendrie: We converged. Good. Thank you very much.

Bell: I was worried there about an hour ago.

END OF TAPE 5, INTERVIEW 4

I would like to add the following interview that I just gave for a PBS Videotaped Production about Ken Olsen on 8/11/08 that was being produced by Gordon College by Dan Tymann. Questions were sent to me and I wrote these answers. The actual interview was of course slightly different.

Ken Olsen PBS Production: Questions for Gordon Bell Interview at the Museum on 8/11/08:

1. Discuss your personal history prior to joining DEC. Grew up in a college-farming town, Kirksville MO, worked in my father's electrical contracting and appliance biz; went to MIT got a master's as co-op student in 57, followed by a Fulbright to Oz. Came back worked for a year at MIT Speech Research Lab using the TX-0 that Ken, Andy, Ben Gurley, and Wes Clark had designed at Lincoln Lab. I was designing a tape unit controller for it, needed modules, and went to DEC to buy them.
    a. When did you join DEC? Got an offer May 24, 1960, from Harlan Anderson for $9600 x 639.24% =61K. I joined in August 1960.
    b. How were you recruited? I was a customer, met everyone… Ben, Ken, Andy and they liked me and made me an offer. I met Dick Best, the Chief Engineer who was a great circuit engineer and really was responsible for getting us working components. Ben was a giant -- he was interesting and worked at all levels from devices to Operating Systems. Designed the analog displays including a high res scope.

       In retrospect since these guys were great engineers, so how could I say no?
    c. Why did you join the company? It was what I thought engineering should be vs being part of the very large sea of desks environments I had seen as a coop student. I was the second engineer. It looked like I could really help and be the engineer I had dreamed of.
2. Describe your first interactions with Ken Olsen. Perhaps when I went to DEC to buy modules. I met Ken on maybe the 2^{nd} or 3^{rd} time. My interaction was with Ben Gurley and then Harlan Anderson who wrote the offer letter.
    a. What were your impressions? DEC had an environment that matched what I thought engineering should be: lots of responsibility to just design and build stuff. Interaction with customers to help get it right, etc. Smart people to interact with.
    b. Did you share the same vision about computing? I don't believe Ken and I have any shared vision of computing. Ken's view was as interesting artifact, like a car, to engineer. Ken may or may not have ever cared much about the actual use and programming.

My own view is more like children that you are helping grow. You have aspirations for them and work with each new generation to do something they have never done before. You feel they are the most important things in the world and the challenge is to help them reach their potential.

c. Was there already a strong DEC culture in place? Yes. Much was: bi-weekly progress reports, open communication, The Engineering Committee and Engineering Notes that resulted in specifications, etc. It was non-hierarchical and politics free… you could talk to anyone, express opinions, consensus was good versus having to lobby a hierarchy to run it up the chain with formal decision making steps. This difference only becomes clear if you live in other environments. I like being first so having unique computers for unique apps was appealing. Given my speech experience, real time was important.

d. What was missing and how did you shape the culture? I really can't say I shaped the culture on arrival… over time, I did. Later I made some observations: "He who proposes, Does." Make what you sell, not what you can buy. Or if you can't sell it, don't make it." The engineering culture wants to design and make everything! I abhor this in engineers where you are just copying and hopefully improving.

3. What was your sense of DEC and Olsen's commitment to technical innovation? I am not sure we thought in terms of technical innovation. We were just building interesting computers that hadn't been built before. Real time and human interaction was the basis that came from MIT.

   a. Describe the intellectual environment at the Mill. Physically, it was quite Spartan yet spacious. We all had offices that were made by hollow core doors from a mill works that made doors located in the mill complex. Eating together in the cafeteria was important for communication.

      Intellectually it was data-driven consensus. Give responsibility to the project engineer. We wrote quite a bit about taking responsibility.

4. Describe the first products you designed. Note below from my vitae. Basically extending the PDP-1, UARTs, and then PDP-4 and 6, while shepherding the PDP-5 that led to the PDP-8. So I count these among the first mini and first timesharing computers.

   a. Describe the process of getting them into production. My first real project was to extend the PDP-1 for use as a Message Switch by ITT—in some sense a forerunner to the Internet. The upshot was the UART and becoming enamored with serial communication links.

      DEC designs were in terms of logic diagrams. On the west coast, designers used logic equations. You make these diagrams on D size vellums that were 8 x a standard page and then made copies with a blueprint machine. These diagrams represented logic elements that you assigned to the DEC modules e.g. couple of flip flops, a half dozen AND gates. Then you made a wiring diagram of the back panel that women wired up. The wiring was done in another building, Bldg 5. Engineering along with Ken's office was in Bldg 12. Every day, you could check on the progress, just waiting to put the modules in so you could check out the machine.

      <u>Note from my 1966 vitae here's what I worked on</u>

i. 8/60-1/61 Collaborated on general design of the DEC PDP-1 computer with responsibility for the direct memory access unit, extended memory control and single channel interrupt system. Specified input-output devices operation including displays, magnetic tape systems, card equipment, and general interface. Designed system and logic of large high-performance, one revolution, swapping drum for the first timed shared computer system using a PDP-1 computer.

ii. 1/61-11/61 Project leader holding the systems responsibility for the DEC PDP 1 computer system marketed by ITT as the ADX-7300. Specific systems designs included: communications interface equipment for telegraph lines; a multiple channel interrupt for processing a large number (256) of lines; and duplex computer inter - connection control equipment.

iii. 12/61-12/62 Project leader for the DEC PDP-4 computer system with responsibility for the systems, hardware and software development. This entailed managing and developing the computer and input output units, preparing computer manuals including a system of drafting control for use in recording the machine's status in design, production, and the field. Designed special features which facilitated the interface of discrete peripheral devices for markets including direct histogram taking to memory (used by physicists for pulse height analyzers) and multiple channel clocks in memory for handling multiple counting devices, e.g, flow meters. Participated in market planning for the computer and in detailed sales, especially multiple unit sales to process control and experimental users. Managed software development including a one-pass assembler, FORTRAN compiler, editing program, debugging program, maintenance routines, and a simple business oriented interpreter for DEC internal data processing.

iv. 1/63-4/63 Specified design of PDP-1 based timed-shared computer system used by Stanford (teaching machines) and Bolt, Beranek and Newman (hospital control). This design was conceived to parallel the PDP-6 system. Each system included simultaneous operation of multiple memories from up to 4 processing units. The communication lines were interfaced via special purpose communication line units which were scanned via a peripheral input-output device. Both a high performance program swapping drum, and a mass storage file are attached to the system.

v. 2/63-64 System design of the DEC PDP-5 including system diagrams, internal flows and logical operations. Managed systems programming which developed a FORTRAN compiler, MACRO assembler, editor, and debugging programs for the PDP-5.

vi. 3/63-64 Project leader for the DEC PDP-6 computer system (approximate size is equivalent to an IBM 7094 or 360144-50 capacity or is priced at $500,000) with responsibility for the systems block diagram design, market testing, and arithmetic processor specifications including simulator for processor logic checking. Managed hardware development including detailed logic of processor and some other units. Designed a series of logic circuit modules for use at intermediate repetition rates (l0 MHz).

vii. 4/64-4/65 Managed software development including design of time sharing multi-programming monitor, FORTRAN 11, a MACRO assembler, editor, debugging and maintainence routines.

viii. 4/65 - present Manager of Large Computer Engineering. This includes the following: Supervision of the development of improved memories, processors, and input-output control units; Design of a multiple PDP-6 arithmetic processor system for Lawrence Radiation Laboratory (Livermore). Designing memory page segmentation hardware to improve the performance of the PDP-6 as a time-shared system. Designing a computer-aided design system for converting logical design input to hardware.

b. Why did you leave in 1966? I saw that DEC had a full plate of products and that the last thing it needed was another computer design or system. I felt I knew how to design computers and I was challenged by CAD. The 8 was just introduced and that industry was taking off. We could build them almost in an assembly line. The PDP-10 had been specified. I was a bit burned out. I wanted to reflect. As much as anything, I wasn't needed.

c. Did you continue a relationship with DEC during this period? I remained as a consultant while I was at CMU and instrumental in making the transition from a 6-bit character world to the 8-bit byte. This included the transition when the guys left to form DG, and the transition to the PDP-11 that I had worked on with a student, Harold McFarland. Andy Knowles headed the 11 marketing and engineering and made it a great success.

5. Why did you come back in 1972? I had a sabbatical to Australia planned and Win said why don't you come back here and run engineering. Also, I thought DEC had to get into LSI and they weren't moving. Also, the 11 was limited and would have to be extended… So I felt needed. Plus, I was getting tired of just advising grad students and writing proposals. My book was out. And I enjoyed the engineering of things more. I remember Ken flying me around NE in his plane and we looked over the DEC factories.

    a. Describe your new role and responsibilities. I came back in 1972 June and was put in charge of Power Supplies and Core Memory, plus a staff role of running the Engineering Committee and looking at all the software that was proliferating as a result of the Product Line structure that demanded unique products. In Feb 1974 I made a proposal to run all of engineering.

    b. Describe the products you and your organization developed. We were responsible for all the products except product line specific products including disks, semiconductors, computers, and software. This lasted till I left… in spite of the fact that argued for divisionalizing the terminals and printers.

6. Describe the process of implementing Ethernet. We needed a LAN as it was a critical part of the VAX strategy. Sam Fuller had worked with Bob Metcalfe to get us one using the Xerox patents. In the end, we made a deal with Intel and Xerox as a consortium to design it based on evolving the PARC Ethernet design.

7. Describe your intellectual process in conceptualizing VAX. There was really three phases: 1. deciding to extend the 11 (hence VAX-11 for virtual address extension), 2. building and marketing the 780; and 3. the VAX Strategy that I posited in the fall of 1978 after it was clear that VAX was accepted in the market. I might add that the VAX was involved in winning at least two Nobel prizes. John Pople of CMU in computational chemistry, and another one in Astro physics and plasmas.

The design was done by a half dozen of us starting in April 1975 I think. Bill Strecker is really the architect of the Instruction set. We held meetings on the 3rd floor of building 12 and at the same time a team was making a breadboard to test the architecture. Bill Demmer put it into production and the software was carried out under Roger Gourd. The team included Cutler, Strecker, Rothman, Richy Lary, myself, and Tom Hastings.

   a. Describe the process of getting it approved and into production. Demmer who had taken over the large 11 engineering including the 11/70 was responsible for the 780.

8. What were the strengths of the DEC culture that allowed it to thrive? A general honesty of letting the data decide. I think the Product Line structure that served the company well. This came about in 1966 that was co-incidental with my leaving whereby the profitability was distributed among a number of individuals who had responsibility for the development, market, distribution of products.

   I think DEC's demise in the late 80s came about when this responsibility was diffused among 3 dimensions: products, sales, and the Product, actually they were Market Product Lines.

9. Discuss the graph you created in the early 1980's about the future of microprocessor performance.

   a. What was the reaction? I assume you are referring to how CMOS would cross over bipolar and ECL. This was the basis of the ill-fated 9000 that was one cause of demise. Other companies made similarly bad decisions e.g. CDC was wiped out, IBM, Fujitsu, and Hitachi all waited too long.

      Alternatively, in 1975 when we started VAX I had a frightening diagram that showed memory costs would be so low and the implied systems would be so cheap that I didn't know where the profit could come from except massive volumes… turned out to be true. So the PC was a dominant reason of demise.

10. Describe the circumstances surrounding your heart attack. I had had a couple of "events" that I hadn't paid attention to. My cholesterol at 220 was normal for that time when high was normal. The issue, I believe, was clearly stress. I can recall a meeting in Portner's office with Ken when we were both shouting at each other and Larry said you guys can't do this because we need you both…and I thought it isn't going to me. The heart attack itself occurred on a Snowmass ski trip with a bunch of my DEC colleagues. Bob Puffer saved me with CPR. The surgery was performed in Denver.

    a. Why did you decide to leave the company? I was one of the few who would directly challenge Ken. I believe Ken was always competing with me. So I blame letting personal stress get to me. I thought it would take a decade before DEC got into trouble. I came back from the bypass and sat in on a contentious meeting about workstations that I would have normally have dominated to get a consensus… and said this is just too hard.

       I felt I wasn't needed. Here's what I wrote.

Dear friends,

I rejoined Digital in 1972 to be more relevant to real versus academic computer engineering and to sponsor the establishment of LSI and VLSI engineering; to enhance the 11 o that it would be competitive in the 80's; and to encourage longer term, higher quality products.

Over the last 11 years, I have enjoyed working with what I believe is now one of the world's best engineering organizations as measured by technical leaders, managers and products. Leading and helping build this organization has been the most exciting project I've ever worked on. It is this group who has built the great products to feed the company. It is also why I think Digital is in a leadership position in computing.

As a major bonus, we have also established a computing environment which I think is both impenetrable competitively, and yet flexible enough to build on for the succeeding 5th and 6th computer generations. **This structure should last for at least a decade**!

The only work remaining to attain the environment is significant VAX—based workstations and special function servers, both of which are well under way. Digital is well positioned to maintain its leadership product position and it has an incredible base of engineers and an architecture that no one can touch.

Now, I need a change. I am going to try something entrepreneurial, at a different scale and in a different fashion. I intend to maintain my positive relationships with Digital and not to enter into a competitive product arrangement, instead I would hope due to Digital's leadership product position to become a major DEC customer in my future venture.
I have enjoyed being here and want to remain friends with all of you. I hope to see you in the future...perhaps at some ecumenical place such as the Computer Museum.

Sincerely,
Gordon
18 July 1983

b. Did you continue to have a relationship with DEC and/or Ken Olsen? There was minimal interaction; some was through the museum that DEC continued to support. I had breakfast in the mill in 1985 with the Executive Committee, got my 25-year clock. There was a bit more interaction when I went to NSF in 1986 and I encouraged DEC to get the contract for the Internet that IBM won with Michigan.

11. Discuss your work post-DEC.

a. 1983-1986. Started Encore with Henry Burkhardt and Ken Fisher. We built quite an interesting machine. My Science article in 1985 on "multi's" has turned out to be correct. Left because we couldn't agree on how to run a company.

b. 1986-1988 Started the National Science Foundation CISE Directorate. This included leading a cross agency group that proposed the Internet; NSF ran it.

c. 1988-1989 Ardent Graphics Supercomputer at a time when 40 other minisupers started.

d. 1989-1995 Angel investor. Wrote a book on startups. I helped start MS Research. The eventual outcome was spending time and/or money on over 100 companies by 2008.

e. 1995- present MS Research. Started working on Telepresence and did it until it was clear that it was going to take a longer time. Then I started on my current project that can best described as building Memex, a system that Vannevar Bush described in 1945. Basically the religion is to put everything in Cyberspace to aid memory for work, learn, health, and immortality.

12. What were the failures of DEC leadership, culture and strategy that led to its demise? It just has to be the leadership that includes Ken, and his top level team that were arrogant based on the 80s success of VAX and failing to deal with the changes surrounding PC and UNIX. The crowning flaw was Bob Palmer that could have saved in given DEC's lead in the Internet… and like the big failures in the financial industry of 2008, he was the only winner.

    Clayton Christensen blames the technology—the micro and DEC's inability to understand and exploit. This is true that it posed a problem, but IBM was able to survive and thrive despite being in the same boat. HP thrived and SUN got started… so it is not technology but rather understanding it that they failed at.

    Basic flaw: really didn't understand at a visceral level the real way how the industry worked. No real understanding of software at the top.

    A series of bad decisions to do everything that Ken championed. Not understanding what a computer platform was and the cost of developing, offering, and supporting. You have to choose! This was how the VAX won! Ken hated killing projects and loved having multiple eggs.

    Finally, the other half of the problem was the destruction of the marketing organization that was getting OEMs and Software suppliers to adopt the platform.

    We contemplated a divisional structure and this should have been used to peel off the edge products like terminals and printers. E.g. we led HP in printers and could have had that business.

    I am told that the Product Line measurement and control structure was abandoned in favor of measuring PL, Field Offices, and Products…in essence having no control mechanism. Despite its flaws, I think the Product Line (should be called the Market Lines) is and was the best for DEC and would have carried it through.

    I was lucky enough to be able to state this as clearly as I can in the last words of Ed Schein's book, DEC Is Dead: Long Live DEC.

13. Assess DEC's place in the history of the computer industry. We achieved the status of being number 2. Our products and many of our people are remembered fondly. If you are over 40 there's a good chance you've used VMS and remember it as the best system ever. VAX was the platform that at least two Nobel Prize Winners use. I take the most pride in the people who left engineering and did spectacular work after they left.

    Dave Cutler is clearly number one on my list above all others. Dave's work on RSX, VMS, the languages and backends for VAX, and MicroVAX 1 was spectacular. Then he did it again and again with NT as the basis of XP, etc. And recently he created another system to be revealed.

14. Assess Ken Olsen's place in that history. In 1977, he infamously quipped, *"there is no reason for any individual to have a computer in his home."* However, this quote is often taken out of context, since the personal computer he was referring to was not the same as the modern PC, but of science fiction, environment controlling. Olsen on many occasions mentioned having computers in his home. In 1987 he gave the first of his infamous snake oil speeches' referring to the *Unix Conspiracy*. Both statements are true… but you have to really interpret them.

    I think Ken was a really good CEO until 1980 and then he changed as he got too close and involved to challenge his team. I think I know when this happened—when he was interviewed by Business Week and made claims that the CEO needed to be more involved. Until then I think he spent his time sensing where things were going and then demanding answers and solving conflicts. The team lacked the understanding of the computer software, the whole hierarchy through apps and content and how the industry worked.

    With the success of VAX in the mid 80s the company may have been blinded to the fact that computing is such a fast moving and changing landscape driven by exponential change.

    Ken sent me a PC while at NSF and the cabling was great. Unfortunately, it didn't run the industry software.

15. Why do so many former employees still speak with such passion about their years at DEC? We are all proud of what we produced and the way customers thought of the products. And then the other part is the spirit of doing work. An open environment. Smart people. I would like to think that it was relatively politics free. I recall when I was an NSF, and IBM came into my office and said that the VAX was giving them problems… I said it was designed to. Unfortunately, IBM and DEC traded places. IBM starting developing the outside software industry, and DEC tried to take it in and do everything themselves.

    I recall going back around 1990 I believe and going to a Field Service facility and needing to print a document. There was a room with dozens of printers and computers and some little clerk said I couldn't use it because I wasn't in that organization. Then, I knew DEC was doomed.

16. Do you have a personal message for Ken Olsen at the opening of the Ken Olsen Science Center at Gordon College? I hope that those who spend time at the Ken Olsen Science Center become as inspired and as enthusiastic about science as Ken. This is a wonderful tribute and inspiration for all of us.

17. What do you remember about Ken, a question at the interview? Ken was pretty complex with many modes. I do remember a nice sense of humor in particular a poster: called Marketing or Engineering in response to the cabling on our equipment that has only gotten worse. It was a photo, with many explanations e.g. we didn't have the right cables, we didn't have time. Also he wrote a series of parables to illustrate organizational issues… I think I egged him on. Will always remember tooling around Bermuda on motor bikes with them barely able to carry Ken and the helmet barely fit. We went there for Woods Meetings a couple of times… and sat in a cold dark hotel room.