Oral History of William Daniel (Danny) Hillis II

Interviewed by:
Gardner Hendrie

Edited by:
Dag Spicer

Recorded: September 5, 2008
Glendale, California

CHM Reference number: X4654.2009

Gardner Hendrie: Well, we have with us today Danny Hillis. He's gracefully given us the opportunity to do an oral history of him. Thank you very much, Danny.

Danny Hillis: It's a pleasure.

Hendrie: Okay. What I'd like to do maybe is start with a little bit of family background, things like where you grew up, what your mother and father did, siblings, get a feel for the environment in which you -- your early years.

Hillis: Okay. Well, my father was an epidemiologist, so we basically spent my childhood going off to places where there were interesting epidemics going on, so I lived overseas a lot of my life in places like the Congo or Rwanda. I lived in Calcutta for a long time.

Hendrie: Oh, my goodness.

Hillis: So it was a fun environment to grow up in. By the time I was in sixth grade, I'd been to 12 different schools, so it was a pretty irregular education, and sometimes there really weren't good schools around where we were. I'd kind of get home-schooled, although we didn't call it that at the time. And so it was an odd way to grow up. I would say I had a scientific education, but it was biology, not engineering or anything like that.

Hendrie: All right. Okay. Now, did your mother work, or did she just <inaudible>?

Hillis: Well, my mother was always the one that was teaching me math, and she had actually dropped out of school to support my father going through college. But then when I was in high school, she went back to college and got her Ph.D. in mathematics.

Hendrie: Oh, my goodness, she got a Ph.D.!

Hillis: Yes, so she's now a biostatistician.

Hendrie: Okay. Oh, wow! All right. That's great. So when the kids grew up, she went back and pursued a scientific c--

Hillis: So I definitely got my mathematical bent from her and probably my experimental bent from my father. Always had a lab, and I would do things. I'd go into the lab and do things like tissue culture frogs' hearts and stuff like that in the lab.

Hendrie: Oh, wow!

Hillis: So, actually, when I went off to MIT, I thought I probably wanted to be a neurobiologist.

Hendrie: Okay. All right. Now, did you have any siblings?

Hillis: A brother and sister <inaudible>.

Hendrie: A brother and sister. Younger, older?

Hillis: Both younger. I have a younger brother who was-- nobody knew what he was going to do because he was always out in the back catching a frog or a snake, and he became a herpetologist and really an evolutionary biologist. And my sister became an M.D., and actually, she is the one that -- she's a neurologist.

Hendrie: Oh.

Hillis: So she is the one that really studies how neurons work and the brain works.

Hendrie: Oh, wow!

Hillis: So somebody in the family ended up doing what I thought I was going to do.

Hendrie: Okay. All right. Very good. Now, what was your father associated with? What institution was he --

Hillis: Mostly Johns Hopkins University.

Hendrie: Okay, because I know you were born in Baltimore. Is that right?

Hillis: I was born in Baltimore, although I didn't live there for very long. Then high school, I went back there.

Hendrie: And then in high school, by then, I knew I was interested in kind of technological things because I'd mostly gotten interested through denial of them. So I remember, for instance, getting really interested in computers when we were living in Calcutta but that you really couldn't get access to a computer at all in any way or even a calculator or anything like that.

Hendrie: Okay, so you got interested without physically having one that you could actually play with?

Hillis: Yes. And, in fact, I ended up building for the All India Science Fair a computer, but I couldn't even get a transistor. I couldn't even get switches. So I built my own switches. I made nails with wires going to them, and the nails went into pieces of screen to make switches.

Hendrie: Oh.

Hillis: And then I wired up-- I made systems of switches so then I made a computer to play tic-tac-toe that was all driven by the switches. But it was all -- I got inspired because there was a library in Calcutta. The British had a good library. So the British library wasn't a lending library, but you could go and you could look at things. And there was a copy, and they just -- I looked for computers. They didn't really have anything on computers yet in the library. It was kind of old stuff, but they did have Boole's *Laws of Thought*, which I thought was a really interesting title.

Hendrie: Yes. Now, did this have -- this gave you some--

Hillis: So that was my first thing that I read --

Hendrie: -- about logic?

Hillis: -- about logic, that's right. And, of course, it was much too advanced for me, but I sort of got the idea of And and OR [functions].. So that was my very start, and then building that machine out of switches was some embodiment of those ideas <inaudible>.

Hendrie: Do you remember what grade or how old you might have been there?

Hillis: Oh, I was probably -- I'd say I was fifth grade, probably.

Hendrie: Okay.

Hillis: So I had sort of exposure to a computer but only one that I'd built myself out of wooden nails.

Hendrie: Yes, yes, exactly.

Hillis: Turns out *Laws of Thought* was the perfect thing to have read.

Hendrie: Yes. Oh, wow. All right. So when did you come back to the United States and start really <inaudible> American schools?

Hillis: Well, so I came back, and I guess that I went and did more sort of chemistry/biology kinds of things for a while, but then I got a job actually at the chemistry department -- in high school, I got a job--

Hendrie: When did you come back to high school, ____ eighth or ninth grade?

Hillis: Well, I came back in probably seventh grade.

Hendrie: Seventh grade, okay. Your father--

Hillis: <Inaudible> into Baltimore.

Hendrie: -- stopped the wayward life and came back.

Hillis: Yes. And so I showed up on Baltimore in the '60s, having been out of the country.

Hendrie: Oh, my goodness!

Hillis: And the whole world was changed. Everybody was wearing flowered shirts and bell-bottom jeans, and the '60s was in full swing.

Hendrie: Yes, okay.

Hillis: And, of course, I was this nerdy kid who had built a computer out of nails.

Hendrie: Yes, exactly.

Hillis: I had a crew cut.

Hendrie: But you figured out how to fit in probably really fast, I'll bet.

Hillis: Yes, but it was a bit of a shock.

Hendrie: Yes, exactly.

Hillis: And, of course, all the kids knew much more about computers than I did even just because it was -- even though there weren't so many computers around, it was sort of a buzz, and --

Hendrie: So this would've been about what year?

Hillis: So this was probably sixty -- well, let's say, maybe in '68, maybe '69 is when I came back.

Hendrie: Okay. All right. Thank you for the context.

Hillis: So I got back, and I got a job at the chemistry department at Hopkins, and they started asking me what I was interested in, and I said, "Computers," and they're like, "Oh, we need a computer." They were building a mass spectrometer, which--

Hendrie: They were building a mass spectrometer, okay.

Hillis: So I was helping build-- I got a job to help build a mass spectrometer, and they said, "Oh, you like computers? Well, we thought we'd hook a computer to this." That was a pretty novel concept then, to hook a computer to an instrument, but they thought that was a good idea, and Dr. Kalsky [ph?], who is the guy that ran the project, he had done a bunch of government things, and so he had access to the government surplus list.

Hendrie: Wow.

Hillis: So he got a computer off the government surplus list, which was a Minuteman Missile computer.

Hendrie: I was going to say, now, that is a Minuteman Missile--

Hillis: So it was--

Hendrie: It was <inaudible>.

Hillis: <Inaudible>.

Hendrie: I think the [Computer History] Museum has one.

Hillis: And it was designed to run for about 15 minutes, but it was incredibly well built. It was a, I believe, magnesium shell on the outside, something very light material.

Hendrie: Yes.

Hillis: And it used current-mode logic.

Hendrie: Oh, my goodness! Okay.

Hillis: And it didn't come with a manual.

Hendrie: Oh, wonderful!

Hillis: So my job was to get this thing going.

Hendrie: Without a manual or --

Hillis: Yes.

Hendrie: Okay.

Hillis: So, eventually, I did find some place where I managed to get somebody to give me a Xerox of a manual that had the instruction set and things like that. But, remember, I didn't-- I mean I didn't even--

Hendrie: You hadn't studied computers. You didn't know anything particular, yes.

Hillis: <Inaudible>, exactly. So I didn't know anything about an instruction set, and we had that one, and then we also had a Titan Missile computer, which was actually even higher tech. It was in a little box. It was quite amazing for its time, but it was on the air-conditioning unit, the size of the room. So when I say these computers were only this big, yes, but all the racks of equipment necessary to -- that was the CPU core, but each of these was literally a roomful of equipment.

Hendrie: Okay, wow.

Hillis: So I went through these manuals and sort of got the idea of what an instruction set was and sort of how that worked, but there wasn't -- it wasn't really clear how to program them. The only way into them, they did have front panels on some of these things.

Hendrie: Okay, with toggle switches, yes.

Hillis: But I quickly figured out that just working out the binary codes for everything was too much trouble, so I actually wrote an assembler, although I'd never really heard the idea of an assembler.

Hendrie: You didn't know what you were writing, but you said, "I need something."

Hillis: I didn't know what I was writing. And, unfortunately, I missed the key idea of having symbolic jump addresses because e--

Hendrie: Yes, so the addressing--

Hillis: So the addressing was so absolute.

Hendrie: -- was so absolute, yes, okay.

Hillis: But you could in mnemonics for-- and you actually had a teletype, so you could type in and teletype it but read it on-- actually, it was a [Friden] Flexowriter.

Hendrie: Yes? Okay.

Hillis: You would read it on the Flexowriter and read it on the tape, and then I built an interface so that this Flexowriter could read the tape into the memory, and then you could type in mnemonics, and it would automatically compile it. But the big problem since I didn't have symbolic jump addresses is that if you ever forgot and you needed to add an instruction, then you had to go rechange every--

Hendrie: Exactly!

Hillis: -- every jump address after that.

Hendrie: Right, right.

Hillis: So I got tired of doing that, so I made my thing automatically would -- every 10 instructions or so, it would leave a little gap, which it would jump over so that you would have a little room to debug.

Hendrie: You designed spacers. Oh, that's funny!

Hillis: All of that got working, and in fact, I got it all up, and I learned a huge amount about computing.

Hendrie: Yes, but just learned it by-- because you had to learn it, _________ worked.

Hillis: But, of course, I was inventing everything very badly, like one of the things I did is at some point I needed a random number generator, which is a whole other story, but I didn't know anything about it. So I just tried a few things. So I got something that sort of looked like it was generating random numbers.

Hendrie: Right. Yes, but probably, and it was too much trouble to look at the 10,032nd words over and over again.

Hillis: Exactly. So I did things like that, but I actually got it, and it was hooked to this mass spectrometer that I built, and it worked.

Hendrie: Really?

Hillis: It was just collecting the data.

Hendrie: Now, was this a summer job, or did you do this sort of in the --

Hillis: Well, it started as a summer job, and when school started, it became a part-time job, and then I started skipping days at school.

Hendrie: Because it was more fun.

Hillis: It was more fun.

Hendrie: It was more fun. Now, do you remember what grade you were in when you were doing this?

Hillis: Well, so by then, I was kind of -- I was in high school at this point doing that.

Hendrie: Would this be like tenth or --?

Hillis: And it was before I went to college, but it happened more and more. So I did it over a couple of years. So by the time I was in my senior year, I was spending a lot of time over there.

Hendrie: Yes, okay.

Hillis: But, in fact, I was also going through all the young kid/adolescence things of like trying to find how to make contact with girls and things like that --

Hendrie: Yes, of course.

Hillis: -- which, of course, is difficult to do if you're immersing yourself in computer manuals.

Hendrie: Yes, those are hard, I guess. There's a problem here.

Hillis: But I would try, and so they would say -- there was a female graduate student in the chemistry department that I really kind of liked, and she just wouldn't pay attention to me no matter what I did.

Hendrie: Oh.

Hillis: And so I felt like I was building this machine for her, and she should really appreciate that -- she was going to do her thesis on this.

Hendrie: Oh, she was-- yes, okay, she was going to use the mass spectrometer, okay.

Hillis: She was, but so I made it, and I tried to show her like all these great things, and of course, she was just like, "Well, just tell me how to turn it on."

Hendrie: Yes, you were just this sort of geeky kid. She wasn't interested at all.

Hillis: So that sort of made me mad, so I made this thing so that I-- I wrote her out a long-- she said, "Give me instructions on how to turn it on." So I remember there was this roomful of racks and equipment that are all connected to it, so I wrote out this long list of instructions, which basically made her go over to that end of the room and flip that switch and then go over there and turn that knob and then go over there, so it was a silly list of instructions just to make her --

Hendrie: Just to make her irritated, get back at her.

Hillis: Well, it wasn't exactly . Also, I liked hanging out with her. This was her <inaudible>.

Hendrie: <Inaudible> she beated [ph?] it a long time.

Hillis: So, anyway, it was a very sophomoric thing to do, but I was a sophomore. What can I say?

Hendrie: Yes, exactly. Oh, that's great.

Hillis: And then the other thing that I did is I had it-- with my random number generator that I had written, it would occasionally, after going through all of this, it would type out on the teletype and say, "Sorry, honey, I have a headache," and power itself off. And it would only do that one in 10 times, but I thought this was the height of humor in my little high school way of looking at things.

Hendrie: That's great. That is really great.

Hillis: So I always thought that was funny except then years later, when I was at MIT and sort of really knew more about things, I get this call from the chemistry department in Johns Hopkins.

Hendrie: Oh.

Hillis: And they say, "We've got a -- are you the guy that wrote the computer program for the mass spectrometer?" I'm like, "You're still using that?" And they say, "Oh, yes. That's the computer that runs the mass spectrometer. It works great except that it just started-- every time we turned it on, it says, "Sorry, honey, I have a headache," and it turns itself off." And what had happened is that I had not known how to write a random number generator, so in fact, it had gotten into a loop, and instead of doing it one in 10 times, it was now doing it every time. So I realized what had happened, and I sort of felt morally obligated to go back down there and fix it because I had <inaudible>--

Hendrie: Now, that's a great story.

Hillis: But then, of course, by then, I'd learned all about computers and assemblers, and I went down there and couldn't even get myself to remember sort of how to toggle things in in binary. And all I had to do was just find this one call to the random number generator, but there were no listings of the program. There was nothing. There was just the binary core image because in those days, it didn't go away when you turned the power off.

Hendrie: Yes, okay.

Hillis: So it was all in core the whole time. So all I had to do was just find this one instruction call and stop it. So I figured, okay, well, I'll just do-- I'll just put a halt instruction in the middle and see if it does it before it gets to that, and then I'll sort of--

Hendrie: You'll do a binary search.

Hillis: Binary search. So I did do that, but of course, every step of the binary search, I had to turn the thing on and restart it. And, of course, I had forgotten how to turn it on now, but all that was left was that list that I had written there that was, "Go over to one side of the room, turn this knob, flip that," and that was the instructions, and I knew that half the things on the list had nothing to do with it, but I couldn't remember which ones--

Hendrie: But you couldn't remember which ones they were?

Hillis: Exactly. So I was kind of hoisted on my own petard.

Hendrie: Oh, that's a funny story.

Hillis: So I had never written a program like that before. So that's my--

Hendrie: That's your high school computer experience.

Hillis: That's my high school computer experience.

Hendrie: Now, when it was time to go to college, where did you think of going? Did you think of staying at Johns Hopkins? I mean your father was -- was he a professor?

Hillis: You know, it's funny. I never really thought about it much. When people start asking you, "Where are you going to go to college?" and I read science fiction as a kid, and there's a science fiction story by Heinlein, *Have Space Suit--Will Travel*. And in that, the kid saves the world, gets the girl, and they give him a surprise, a scholarship to MIT. So I thought, okay, well, that's what I want. So when they started asking me, "Where do you want to go?" I said MIT, but I didn't really know.

Hendrie: You really-- this was not a thoughtful procedure that you would use to figure this out. Okay. All right.

Hillis: But as it turns out, it was a very good choice for me. So I went off to MIT.

Hendrie: Now, what did you think you were going to do?

Hillis: I thought I was going to be a neurobiologist.

Hendrie: Okay, so you still had neurobiologist In your head?

Hillis: I did, and I was interested-- I'd sort of imagined, because I was still sort of in the *Laws of Thought* thing, you know, thinking about humans think and artificial intelligence and so on. And I had read about the great Marvin Minsky at MIT. That was another reason I was pretty sure I wanted to go to MIT.

Hendrie: Yes.

Hillis: And so I showed up at MIT [in 1974], and the first night, there was a big party, and there was a professor that was asking students what they were interested in, and whatever they were interested in, he would tear them apart and say, "You shouldn't study that."

Hendrie: Oh, my goodness. Okay.

Hillis: And he got to me, and he's like, "What are you interested in?" And I said, "Neurobiology." And he said, "I defy you to tell me one good paper that has ever been written in neurobiology." Well, it just so happened that I had just read a great paper on neurobiology called, "What a Frog's Eye Tells the Frog's Brain," and it was just fresh in my mind.

Hendrie: "What a Frog's Eye Tells a Frog's Brain." <Inaudible>

Hillis: I decided I was going to show off, and this was going to be my big moment in the sun. So I started talking to the professor about this paper, and then he starts asking me questions. He said, "Well, how did they measure that, and doesn't this conclusion contradict that conclusion?" And he starts asking a series of questions and after a while just tears the paper to shreds and eventually gets me to the point of saying-- "So I ask you, should that paper have ever been published?" And it's like, "Well, I have to admit, now that you've pointed out all these flaws in it, probably not." And he said, "What about the guy that wrote it? Should he be allowed to publish it?" I was like, "Well, it does sound like he was sort of sloppy and not very careful with his experiment." "Okay, so thank you." And he goes on and starts talking to the next person. And then people come up to me and pat me on the back and say, "You were just talking to Jerry Lettvin, the author of "What a Frog's Eye Tells the Frog's Brain."

Hendrie: Oh, no! Oh!

Hillis: So I became friends with Jerry Lettvin.

Hendrie: Yes, of course!

Hillis: And then Jerry Lettvin-- but Larry Watman [ph?] did say, he said, "This is ridiculous, that neurobiology is just too complicated. You'll never figure it out. The stuff you're interested in is what Marvin Minsky is doing, and you should go over and find him." And I was like, "Well, you mean I could talk to Marvin Minsky?" "Oh, yeah, he's over there. Just go over to the AI lab." So I went over to the AI lab, and the door was locked, but there was a library-- library for computer science was down below. So I thought, okay, I'll figure out some way of getting in the lab. And so I started reading their proposals, their NSF proposals and things like that. And one proposal came, which was something that said, "We'd like to make a terminal for children that can't read, but we haven't figured out how to do that yet." So I thought, oh, here's my chance. So I went off and thought about how to make a terminal for children who couldn't read. Then I go and make an appointment, knock on the door, and I say, "I have an idea for how to make a terminal for children that can't read." And they're like, "Oh, really? We're looking for that idea." And they started talking to me, and it was Seymour Papert's lab. So I got hired over at Seymour Papert's lab, which was close. I mean it was at least – I was in the door.

Hendrie: Yes, you were <inaudible>, same area.

Hillis: So started working on that. Then I started looking around for Marvin Minsky, and I found his office, but he was never in his office. So I started asking around. Then they say, "Well, Marvin actually comes in late, and he's working down in the basement. He's trying to build a computer for only one person to use." He was trying to build a personal computer, but that was in the days before microprocessors.

Hendrie: Yes, right, right.

Hillis: And, actually, the microprocessors were jus t-- I think the Intel 4004 was out, but he wasn't building around microprocessors.

Hendrie: So this is early '70s, you're a freshman at MIT…

Hillis: Early '70s, that's right. And so I go down to the basement, I walk in, and there is Marvin Minsky himself, sitting there working on diagrams, and there's a big computer that they're building, and all these people around working and wire wrapping, and Marvin would draw a diagram--

Hendrie: Oh, my goodness!

Hillis: And I'm too shy to introduce myself, so I just sort of wander in, and I start just kind of looking at the diagrams. And sure enough, after a while, I found an error in one of the diagrams, some piece of logic that doesn't make sense. So I'm like, well, this is good conversation starter, so I pick up the diagram, and I go over to Marvin Minsky, and I said, "Look, I think there's an error here. This doesn' t--" And so he says, "Well, fix it." Oh! And I'm like, "Well, you mean on the diagram?" He says, "Well, do it on the diagram, and fix it in the machine." So I changed the diagram, and I go over, and I didn't know how to use a wire-wrap gun, but I'm kind of watching these guys, so I changed the wiring--

Hendrie: Oh, my goodness!

Hillis: -- and then I look, and I find another one, and so I started doing this, and then after a while, Marvin assumed that I worked for him, and he started treating me like I worked for him. So that's how I started working for Marvin Minsky.

Hendrie: Oh, that's great!

Hillis: So then we became great friends --

Hendrie: Good.

Hillis: -- and continue to be so. And Marvin really became my mentor, and I really got very -- he was great because he -- like Jerry Lettvin, he would always argue with the students. So you'd come into him with a proposition, and he'd try to argue you out of it. That was just his technique of getting you to think it through.

Hendrie: Yes, to make you really think.

Hillis: That's right.

Hendrie: Think hard.

Hillis: Right. And then occasionally, you'd run into some piece of-- you'd run into some question you didn't see the answer to, but he knew the answer to, and then he'd get up on the blackboard. Like I remember once him showing me the derivation of E to the I theta equals cosine plus I sign and doing it by Taylor series and just being amazed. But he was kind of explaining something, but it would always come out of some question that you were asking.

Hendrie: Ah!

Hillis: It was never he decided -- it was just like, "Well, here's the piece of knowledge that you need to understand why you thing wouldn't work or why --"

Hendrie: Okay.

Hillis: So he would--

Hendrie: Was sort of a reactive teacher, as opposed to a--

Hillis: Oh, yes.

Hendrie: -- "Okay, now, here are the three things I want to tell you today."

Hillis: Definitely interactive.

Hendrie: Yes, interactive.

Hillis: Very interactive. He didn't make the lesson plan. He looked for sort of when you would be interested in knowing something.

Hendrie: Yes.

Hillis: So that was pretty fantastic. And then in talking with him, I sort of hung around and I --

Hendrie: Now, were you still a freshman when this--

Hillis: Oh, yes, I was--

Hendrie: Yes, yes, this is all freshmen, okay.

Hillis: Yes. And, in fact, I got so interested in all of that, I kind of forgot to sign up for any classes when I was a freshman, so that was -- I got a little behind on the classes.

Hendrie: You got a little bit behind on the credits <inaudible>.

Hillis: I had this job at the Logo Lab with Seymour Papert, and then I had this --

Hendrie: Then you had this thing that you were doing with Minsky. Okay, cool.

Hillis: It was pretty fantastic. So I ended up --

Hendrie: Well, that's a great experience, though.

Hillis: Yes. No, it was a great experience. And, of course, they were teaching me a lot of the things that I would be learning in freshman -- and, in fact, I wasn't taking very many classes, but I was hanging out with just amazing people like Gerry Sussman, a fantastic teacher, and then I found out that -- and somehow, somebody mentioned Claude Shannon, and I was like, "Oh, yes, I wish I had met Claude Shannon." And they're like, "Well, go out and meet him. He's out in Winchester."

Hendrie: Oh, wow!

Hillis: So that's how I got to know Claude Shannon.

Hendrie: Oh, my goodness!

Hillis: I went out, and he was really not coming into MIT much in those days. He was more into juggling.

Hendrie: Really? Okay.

Hillis: And so I learned to juggle and went out and -- but then we had a lot of fun together. And he ended up being on my thesis committee, too. I was his last Ph.D. student.

Hendrie: Really?

Hillis: Yes.

Hendrie: Very good. Wow!

Hillis: But I mean that was, for me, just like--

Hendrie: Oh, yes, I mean to be able to hang out with somebody like that.

Hillis: Do you know what his master's thesis was? The application of Boolean logic to switching circuits.

Hendrie: Really?

Hillis: That was Claude Shannon's master's thesis.

Hendrie: Oh, my--

Hillis: And, of course, that was exactly what I had been thinking about when I read Boole's *Laws of Thought*, building a thing -- But, of course, Shannon had just worked it out beautifully.

Hendrie: Yes, of course.

Hillis: So, yes, that was before he even did information theory.

Hendrie: Oh, my goodness. All right.

Hillis: So, anyway, I got to hang out with really fantastic people. I met with --

Hendrie: Great mentors.

Hillis: --<inaudible> and the physics, I hung out a lot with Philip Morrison, so there was really an amazing set of teachers who would really just love it when you would go in to them with a question or an idea, and it would be a conversation. So I pretty much did my first two years of education by sort of going around and one way or another getting into the offices of great people that I admired and getting into conversations --

Hendrie: With them, and yes--

Hillis: -- <inaudible>.

Hendrie: And hopefully working with them.

Hillis: Yes, yes.

Hendrie: Okay. All right. Now, at some point, did you-- when did you decide officially that neurobiology wasn't where you were going?

Hillis: Oh, I think Jerry Lettvin officially talked me out of that pretty easily. I mean he really started talking to me about what you could really measure, what you couldn't measure, and so on. And I realized very quickly he was right. The things I was interested in, the technology at that time, you just -- was not up to studying them in neurobiology.

Hendrie: Right. So you weren't going to make much progress?

Hillis: Right. And AI -- the artificial intelligence lab --everything looked very promising at that moment. I mean that was the moment where--

Hendrie: That was the peak in promise.

Hillis: Everything looked perfect. We sort of assigned one graduate student to work on vision, another one to work on natural language--

Hendrie: And each of them would succeed.

Hillis: Exactly. So there was a huge optimism that -- and they were succeeding. That was the amazing thing. I mean they've all since just gotten these things that looked at line drawings and figured out blocks, and Terry Winograd had gotten this thing that you could say, "Pick up the big red block and stack it under the yellow tower."

Hendrie: Yes.

Hillis: And so Shank [ph?] was doing his sentence understanding, you know, "Joe killed Mary," or whatever his canonical -- and so it was really-- it was a very exciting moment. I mean people were -- in the vision lab, you had people like David Marr and Tommy Poggio that were doing very basic stuff like difference of Gaussian line elimination and stuff that is the bedrock kind of information, line finders, the basis of stereovision, things like that.

Hendrie: Okay.

Hillis: So it really all seemed like -- oh, and chess, right? The hardest things humans could do was play chess, and we had -- there were still members of the faculty that were saying, "Well, computers will never be able to play chess."

Hendrie: Right, exactly.

Hillis: And at the same time, Richard Greenblatt was making his chess machine, and Tom Knight was designing his Lisp machine, and we were getting better and better at chess. And it was clear to those of us inside that these people who were making their philosophical statements about computers not being able to play chess were just full of nonsense and that it was just a matter of time before computers beat people.

Hendrie: Yes, okay.

Hillis: So, anyway -- yes, so there was a time when it all looked very easy, and then as we got into it, what we discovered was actually it was just the hard things that were easy, like chess. The easy things, like recognizing a face, were, in fact, were very, very hard.

Hendrie: Yes, okay.

Hillis: And things like that first algebraic manipulator for ____ design, the Mathematica -- or the predecessor of Mathematica was just called Macsyma so that-- but that was hard stuff. That was like calculus, but we made a lot of progress on that. The easy things turned out to be much harder.

Hendrie: Yes, okay.

Hillis: So that wa s-- so I definitely with still basically interested in how does the brain work but approaching it from an eye standpoint.

Hendrie: All right.

Hillis: And one thing that was very obvious was that the computers that we had available to us then were just not anything like the horsepower necessary to do it. So, for example, the AI lab at the time, the entire laboratory, all of these vision projects and analogies and language understanding and things like that were being done on a single shared PDP-10, which had -- we were so proud of ourselves. We had entire moby[?] of memory, which was 256K of memory. That was shared for everybody. And we probably had 10 megabytes of disk space.

Hendrie: Yes? And so--

Hendrie: And you thought that was a great computing facility.

Hillis: Yes. Well, compared to whatever, we were just way ahead of everybody else.

Hendrie: Yes, that's right.

Hillis: But it was obvious to me that that wasn't going to cut it for doing the kinds of things I was interested in. So I got very interested in how could you make a computer that was many orders of magnitude faster and more powerful.

Hendrie: Now, do you remember when you started to get in? Were you a freshman, or were you a sophomore by now?

Hillis: Almost as soon as I went down there, I mean I remember thinking about it when I was a freshman because I remember starting to talk to Marvin about that in his first days working down in the basement. And it was also at a time when microprocessors were just becoming available. And that machine wasn't being done with a microprocessor, but they were discussing, "Well, maybe we should use a microprocessor," but it wasn't quite ready. But that got me thinking about microprocessors and realizing that --

Hendrie: There was this technology.

Hillis: -- there was a technology, and they were going to be cheap, and I could start <inaudible>--

Hendrie: They were going to be really small.

Hillis: -- component and what if you just hooked lots and lots of these together. So that-- I started thinking then. It was interesting. But then as I got to know more about it, it was another one of these things like the chess thing, which there were a lot of experts around who said it was impossible.

Hendrie: Right.

Hillis: And this is hard to believe, and this is the hard part about telling this story because today's perspective, it's hard to see how could people not just want to hook together a bunch of--

Hendrie: Right.

Hillis: -- because it seems like the absolute obvious thing to do.

Hendrie: Yes, but it's like many things, I think, when it hasn't [been] done before, ever before, it's hard to realize.

Hillis: Well, I think, also, there was Gene Amdahl's work that confused everybody. So what had happened, I don't think Gene Amdahl was so confused, but I think that his reporting of his work had confused people. So what he had done is he had tried to make -- hook together a few big computers like -- this was mainframes the size of a room, but he was trying to make multiple processors with them. And what he had noticed was that the more that he hooked, the less efficient they got. And he -- and in analyzing why that was happening, he came up with a rule that sort of showed that if you divided up a task into several different pieces and tried to do them all at once, then things would be limited by the slowest task.

Hendrie: Yes.

Hillis: And because people were thinking of computers as sort of giant room-sized things, when they were thinking of multiple computers, they were thinking like three or five or six.

Hendrie: Yes, right, yes, small numbers, right.

Hillis: And the way that they were thinking about dividing up a task was to basically divide up the program and have this part of the program run on this computer and this part of the program run on that computer. And, sure enough, that does get more and more inefficient. Unless you can just exactly get parts of the program that take exactly the same rate, one of the parts of the program is going to be the bottleneck.

Hendrie: Right, and, yes--

Hillis: And the rest of the computers are going to sit there and wait for it. So that's what Amdahl discovered.

Hendrie: Yes.

Hillis: And so -- and that discovery got turned into what I'd call Amdahl's Law, and it came in various variations, but basically, it's dominated by the slowest part. But it got translated into sort of a false conclusion, which was that the number -- that the efficiency of computers would go down as you hooked more and more of them, and that was such a strong prejudice at that point. For many years after I started working on parallel computers, I would give a presentation, and then somebody would inevitably hold up their hand and say, "Well, this can't be true because haven't you heard of Amdahl's Law? Don't you realize that these computers are getting less and less efficient?" So it was a mental block.

Hendrie: Yes, okay.

Hillis: It seems silly in retrospect, but it was -- it was sort of given knowledge that everybody knew in computers. And I have to say if I came at it from the direction of computers, I probably would've been convinced by it because that's what you were being taught in school when that was --

Hendrie: Right. And on the surface, if you think about it, it makes some sense.

Hillis: It sort of makes sense, yes. It has a good argument to it. It almost seems like a mathematical necessity.

Hendrie: Yes, okay. Could we take a pause now?

Hillis: Yes.

Hendrie: And we're going to change tapes.

Hillis: Okay.

END OF TAPE 1

START OF TAPE 2

Hendrie: Think about the earliest sort of thoughts you had about how to solve this computation problem that you saw as limiting what you wanted to do.

Hillis: Yeah. Well, in fact I think -- so I was lucky because I was interested in a particular problem. I wasn't just trying to make faster computers in the abstract. So I kind of believed the Amdahl's Law argument seemed right for computers in the abstract, but somehow I knew it didn't apply to my particular problem because I knew that the brain was doing all these computations using switching components that took many milliseconds to switch.

Hendrie: But there were an awful lot of them in there.

Hillis: But there were an awful lot of them, that's right. So it was obvious that the brain does things by using a lot of slow components in parallel. So I didn't, at the time, see what was wrong with Amdahl's argument, and sort of assumed that the general point was true, but said, "Okay, well I won't make a computer used for general computation, I'll just make something that you use for these AI kinds of computations like vision and so on ________ selective." And so I actually wanted to build a computer and I had had a little bit of experience then playing with Marvin's computer. And Mead and Conway wrote a book around then explaining how you could design a chip. So I started designing some chips and sending them out to be fabricated. MIT had a great...

Hendrie: Somebody had to sponsor you here. You're in MIT and you're starting to spend money, so tell me how you worked around that issue.

Hillis: So first of all that was in the good old days of DARPA, which we understood at the time. They were doing crazy ideas like trying to get networks of computers to talk to each other and they were...

Hendrie: Exactly, oh yeah, just absurd things.

Hillis: And so DARPA sort of gave a general grant to the AI lab and was basically a grant to Marvin Minsky and Seymour Pappert, and Pat Winston and people like that to kind of spend on what they were interested in working on, so in those days DARPA kind of backed people rather than specific projects.

Hendrie: Projects.

Hillis: That's right. So the AI lab was the environment where that was happening. So we had some things that we were supposed to be working on like natural language, vision processing and then we had a lot of stuff that everybody was kind of doing in the background as sort of -- that were doing in the background, let's say after hours kind of playing around. So for example, one of the things was that somebody has basically made a, what's effectively a video game, Space War!, and it was running on the [DEC] PDP-6, the first video game most of us had ever seen. There was also the students sort of hacked it so you could write your papers on it. We had this program called Million Dollar Typewriter which we were using this million dollar machine to actually edit a paper. There were people playing with ways of drawing on the computer.

Hendrie: The computer was their...

Hillis: Email, for instance, we could end little messages to each other. We could even send messages to our friends over at Stanford that were working on things there. Xerox Park, so e-mail systems, networking. So the interesting thing was, of course all this stuff was the stuff that we were kind of sneaking around, weren't supposed to be working on. But so it's actually funny to look back because if you look at all the stuff we're supposed to be working on like speech recognition and understanding, well that stuff still barely works. That's still research topics. But all the stuff we weren't supposed to be working on like video games, and email, and word processors, those have become multibillion dollar industries. So, in a funny sense we had it backwards. We had no idea what was really important about what we were doing. But there was a real community of people. And of course, in those days too, everybody who used a computer could program the computer. There were a few professors that didn't, but. -- So we assumed that -- we believed, actually, computers were going to become common and everybody would have them, but we assumed that -- the implication is that everybody would become programmers.

Hendrie: That's how you used a computer.

Hillis: That's right. That was it.

Hendrie: If you couldn't program you couldn't use it. There wasn't much packaged software.

Hillis: Everything was completely open. Anybody on the ARPANET could connect to our computer and start changing lines of codes. Like I was changing the instructions in the core and nothing stopped it. And I remember how shocked we were when it came time to upgrade the computer and Digital Equipment Corporation wanted to charge us for the operating system. And we were like, "You mean you're going to charge us for the software? Didn't we write this software and give it to you?" Well then they were like, "Well, okay. We won't charge you, but we'll charge everybody else." And that made people mad. In particular, there was this one guy that was hanging around MIT at that point. He had come over from Harvard. He was one of the better programmers around. He had been working on the word processor, and it made him furious -- Richard Stallman, he was like, "Software? Software should be free. You shouldn't charge for software," and he absolutely mad. It's like and so one of the things we decided we weren't going to use the operating system. We'd just write our own.

Hendrie: Yeah, that's a way to get around that.

Hillis: So we ran our Incompatible Time Sharing System [ITSS], which it was literally called. And Richard Stallman wrote Emacs, which was his free text editor, which was much better.

Hendrie: Exactly, used by many programmers 20 years later.

Hillis: Yeah, and again you have no sense of, I mean, at the time this sort of seemed like just a silly argument. You have no sense at the time realizing, "Oh, this is sort of an important moment in history of which way does it...?

Hendrie: Yeah, you have no perspective when you're immersed at this time.

Hillis: So we kept on working on that. So I kept on working on the chip design. One of the things DARPA did was it ran a facility called MOSIS [Metal Oxide Semiconductor Implementation Service], actually it still does, where university students can design a chip and then they would fabricate it for you. So I started designing chips, first just for practice, but one of the first chips I designed was a dynamic RAM, which seems pretty silly. And I decided this was a little hard for me to sort of do in my spare time so I got an undergraduate to help me design the dynamic RAM who turned out to be a pretty bright guy. So I had this undergraduate working on it who was Brewster Kahle.

Hendrie: Oh my goodness! All right, okay. There were just too many smart people there.

Hillis: So we got this dynamic RAM working and we were going to make the computer, I mean, so when I say build a computer I mean like down to your building the dynamic RAM.

Hendrie: Yes, you're going to build it all.

Hillis: I mean this is down to the transistor. And of course it never even occurred to us not to build the operating system, the compliers, the...

Hendrie: Oh yeah, of course you're going to build the whole thing, okay. Who was sort of your advisor at the time that you were working?

Hillis: Marvin was my mentor.

Hendrie: Marvin was your mentor and he was supporting you going and doing this. He said, "This is a good idea."

Hillis: He was, but then also, but actually also Gerry Sussman was incredibly important. Marvin didn't really design chips, but Gerry knew all about the chips and so on. So Marvin was crucial and Claude Shannon was very important to me, and so I had a few different people that were, but Marvin was definitely my primary mentor.

Hendrie: How many years had you been at MIT when you're starting to do these chips, really starting down this road?

Hillis: I don't think the chips started for a few years. I mean I think I was-- I'm sorry, I don't really remember it step by step. Another person that was very important was Tom Knight. Tom was the one that really built things around there so I knew I needed to hang out with him. So I literally just moved into his office. I think he was a little bit annoyed at first, but it was like there was an extra desk, so.

Hendrie: Yes, exactly and then you could ask a lot of questions.

Hillis: I knew I would learn from him. So I would say that I was probably later in my undergraduate curriculum, an early graduate student because I kept on getting involved in projects and things like that. And it was a real community of people, but you kind of had to prove yourself. There was a hierarchy because of course we only had one computer, so during the day you sort of had to be a faculty member or maybe a graduate student with seniority that logged on the computer in the day. So the low level people like me who had no priority...

Hendrie: You became night owls.

Hillis: We became the night owls. That's right. So we'd have to be there overnight because that was the only time we could actually get any time on the computer. So there was a community of people like me and Richard Stallman was in that category, and Richard Greenblatt that were coming in and working later, and those were the people that were -- and there were a few other people like Paul Mockapetris was over there making what became the Domain Name System, although it wasn't called anything like that. These were all just kind of hacks that were pulled together. There was a group over at Xerox. There were small numbers of people that were working on this. So I knew of Alan Kay. In fact Alan Kay was somebody I really wanted -- He was one of the few people I didn't succeed in working with that I wanted to. And he came-- there were these distinguished lecture series where we would have like Hamming come and talk about the Hamming Code, or Alan Kay come and talk about his idea of the personal computer, the Dynabook. And I remember when-- most of these guys would show up in suits and ties and so on, but when Alan came he showed up wearing blue jeans and sneakers. It was sort of a shocking concept in those...

Hendrie: In those days.

Hillis: And I remember sitting in his lecture and saying, "You know, I think that that's kind of -- that's what I would like to be. He seems to be doing really interesting things and he doesn't have to wear a tie. I like that." And I tried to contact him and actually work for him for a summer job, but it fell off his list. I always kept that until years and years later when Disney offered me to be a Disney Fellow. I said, "Okay, I'll do it, but only if also invite Alan Kay, because I've always wanted to work with Alan." So that's how Alan and I became the Disney Fellows.

Hendrie: Is that right? Oh, very good. So you succeeded.

Hillis: So eventually I succeeded in working with Alan. Of course Alan's now in his offices here at Applied Minds. So I did eventually succeed, but that one took me many years to work out.

Hendrie: Well, let's get back to the story of you designing this computer.

Hillis: Okay, so I started making chips and basically it worked out that if you made an incredibly simple computer -- remember, chips were very small in those days.

Hendrie: Yes, couldn't put any transistors on them.

Hillis: If you got it down to its bear bones you could actually put multiple computers onto one chip. And so now part of the trick was you had certain pieces of it that were shared and so if you said, "Well these are --" but in the way that I was going to program the computer they were all going to be running the same program, just on different data. So if you made the compromise in saying, "Okay, well since they're all running the same program if I could kind of share the instruction fetch units and things like that then I only had to put one of those on the chip and I can...

Hendrie: But you can put-- yes.

Hillis: That's right, so you ended up sharing a lot of the components. Now that meant that they were overly constrained, but it was a compromise necessary to get more than one onto a chip. And then once you could get more than one on the chip then you could start dreaming about building computers without t-- I mean, I remember telling everybody, "We could build a computer with a million processors," and that was in the days when people were saying it was impossible to get beyond three or six to program. So anyway, that's the kind of stuff that the DARPA people loved to hear.

Hendrie: Yes, that was good.

Hillis: Yes, but of course I was still probably an undergraduate at that point, and I didn't really have any budget for doing any of, anything like that. So I went down to Mike Dertouzos. He had some money. MIT got paid a certain amount of money for inventing the core memory and he got to sort of allocate that. So I think he gave me like $10,000 which was enough to buy parts to play around with. More than that, it kind of gave me my own little budget to spend.

Hendrie: Yes, very good. That's very exciting.

Hillis: I could get circuit boards made and things like that. And then at some point -- but Marvin and Patrick Winston and so on, they all got excited about it. So when it came time to write DARPA's proposal of the next stuff that they were doing, I think Gerry Sussman actually wrote a section saying, "Oh yeah, we're actually going to do this thing, and Danny Hillis has this idea, and that got into the -- now it was sort of official part of the lab project. And so, but what happened was so then as I became a graduate student I started working on this budget more and more, and it became clear that it was actually building this thing was going to be -- I mean I did a lot with undergraduates like Brewster Kahle. But it became pretty clear that this was more than a graduate student could, and a band of undergraduates could build this.

Hendrie: The amount of work and the size of the task became more and more obvious as you worked your way through it.

Hillis: And so it became sort of -- we started worrying about, "Well, we're taking up too much space with the whole team of people, and we're ordering things." We were sort of getting to the point where it would have been a noticeable part of the lab budget to do it. So at that point I said, "Okay, well I'm going to have to do this outside the lab. I'm going to have to make a little company to do it," which was considered a weird thing then. Graduate students didn't really do that then.

Hendrie: Of course they didn't.

Hillis: But I did it, and Marvin was very supportive about it. He was like, "Yeah sure, that seems like the right way to do it, and you can hire and you can get some venture capitalist to put in some money on it and things like that." And I knew somebody that knew Bill Paley, who was the head of CBS then, and so I didn't know who he was, but he had a lot of money.

Hendrie: Now, was this Cheryl that you knew?

Hillis: This was Cheryl.

Hendrie: Now how did you meet Cheryl? I have to get this into the...

Hillis: That's a long story, but through Marvin. She wanted to start a company in something.

Hendrie: Yeah, she was more interested just starting a company. She didn't know exactly what.

Hillis: Exactly, and she had just done something in genetics or something.

Hendrie: Was she a student at MIT?

Hillis: No, she wasn't. She just -- she was just an entrepreneur.

Hendrie: Just an entrepreneur out there. You don't know what she was doing before you approached her.

Hillis: Well, yeah she had started another company in genetics, or she had had some role in starting a company in genetics.

Hendrie: Okay, yeah, she was working for a young company in some role that was genetics-oriented.

Hillis: I don't know if she was working for them or she had just helped start them. I think it was more that she just helped start it.

Hendrie: Okay. And Marvin knew her

Hillis: Yeah, and I don't know-- I think she found Marvin, probably. So I didn't know that. But the big funder for it ended up being Bill Paley. So Cheryl brought me to see Bill Paley and I had no idea who Bill Paley was.

Hendrie: She had met Bill somehow.

Hillis: Yeah, she knew Bill.

Hendrie: She knew Bill before you knew Cheryl.

Hillis: Yeah.

Hendrie: Okay, I've got it.

Hillis: And I had no idea who Bill Paley was. I sort of showed up in his office and didn't know anything about raising money or anything like that, but this was like -- supposed to be somebody we could get some money from, and I knew we needed some money.

Hendrie: So fine you're just, "Tell me what I'm supposed to do here."

Hillis: I remember going in his 5th Avenue apartment and walking in and immediately seeing all of these Picassos that I recognized, hanging on the wall. And it was sort of like a college dorm except they weren't posters they were the real paintings, and started realizing, "Oh, I guess I'd never really been in a rich person's house before." You see paintings that you're recognizing.

Hendrie: That you recognize as the real thing.

Hillis: Oh yeah. Yeah, so there were Degas' and Matisse's just all over the place and I went, "Oh, okay." So anyway had dinner with Bill Paley and he sort of asked me what I'm doing. And I start trying to explain to him everything I was just explaining to you, but probably much less clearly having not had the benefit of sort of having explained it as a in time sense. So I talked about how we needed to build this computer and so on. So at the end of dinner Bill Paley says, he says, "You know," he says, "I didn't understand a word you said." He said, "But," he says, "I'm a pretty good judge of people." He says, "What exactly are you asking for?" And I said, "Well, we need $10 million dollars," or whatever it was. And Bill Paley says, "Okay, I'll give you half of it." And so that was my first money-raising experience. So I'm sure that Cheryl had set it up quite a lot before then I would say, but from standpoint, "Hey, that's great. You just walk into peoples' rooms and there was the money."

Hendrie: And you ask and they say, "Yes."

Hillis: And they give you money, right. So anyway, so with that money, I mean, I'm still a student, so this is done while being a graduate student. I started, basically took the group of undergraduates that had been working with me and gave them jobs. We got a little place out in Waltham. It was an old house, the Robert Treat Paine House which was built by one of the signers of the Declaration of Independence.

Hendrie: Oh my goodness, a historic house.

Hillis: Yeah, we were literally in this old historic house. And I was -- at that time my only vehicle that I had was a fire engine, so I would drive my fire engine out to the Paine House and get out in this old-- it literally a Richardsonian mansion, it was literally designed by Richardson.

Hendrie: Oh my goodness. Oh, wow! That's pretty cool.

Hillis: Yeah.

Hendrie: Okay, and that's where you started.

Hillis: That's right. It was designed by Richardson, yes. And so that's where we started the company and started building and designing the chips. And eventually we started getting more practical about it in the sense that, for instance, we decided that dynamic RAMs had come along. We could buy those off the shelf. We didn't have to design our own DRAMs. But actually, I have to step back a little bit and tell you one of the, maybe even before that, the first day of the company. In fact I'll step that back even before the first day because I had always been -- so sometime I'd gone to a conference that Ed Fredkin had had where he invited me even though I think I was probably an undergraduate or an early graduate student at the time, a sort of computation and physics [conference].

Hendrie: Yes, he's been interested in that for years.

Hillis: Right. And he had basically had the idea -- he felt like cellular automata were super important and might be the basis of physics. And so he brought together a conference, which was a lot of really interesting people and that's where I first met Freeman Dyson who's become a really good friend of mine. That's where, I mean, people like Dennett, Stephen Wolfram who I first met at that conference. It was really fantastic.

Hendrie: Yeah, some very interesting people.

Hillis: It was a really interesting conference. But I gave a little paper about how cellular automata might underlie physics in some way, because that was basically the whole thing of the conference. But this guy comes up afterwards and starts talking with me and starts asking really good questions, and a guy with a kind of a New York accent and I didn't really recognize, but really interesting questions and After a while I realized, [it was] Richard Feynman. So Richard Feynman and I got to be friends and he's like, "Oh yeah, my son is going to MIT next year. You should look him up." And so in fact he sent his son to me and his son became one of the undergraduates [working at the new company].

Hendrie: Oh, working?

Hillis: Yeah.

Hendrie: Oh wow, okay.

Hillis: But anyway, so Feynman and I got to know each other at that conference. And so whenever I would come out to California I would visit Richard Feynman. And when I was out here in Pasadena just before starting the company I said, "Hey, we could use some Cal Tech students at the company too. Do you know anybody that wants a summer job?" And he said, "Well, I mostly have physicists," he says. But he says, "I do know this one guy though. He is a physicist and he doesn't know anything about computers, but he's really very hard working and he's willing to learn. He wants to learn about computers, and I think he'd really be a good hard worker." And so I was like, "Well, if you recommend him then great, I'll hire him. What's his name?" He said, "Richard Feynman." So that's how Richard Feynman came to work summers at Thinking Machines [Corporation].

Hendrie: Because he loved working with you and wanted to learn about computers.

Hillis: He volunteered.

Hendrie: That's a great story.

Hillis: So he would come every summer pretty much to the end of his life to come out there and work a summer job. And it never would have occurred to me to have the chutzpa to ask him.

Hendrie: To just even help.

Hillis: Right, but he just invited himself. So anyway so he showed up for the very first day of Thinking Machines. Because remember we really don't know what we're doing, and everybody's been sort of focused on...

Hendrie: Well, you know where you're going. You don't know how to get it, so there's a little bit of, "How do we get there?"

Hillis: That's right. But then there's also -- there's a whole bunch of things about, I don't know, I'd just gone through this thing at MIT of we had some basic patents about like putting more than one computer on a chip and things like that. We need these for -- we need to have these things patented, but I went to MIT and sad, "Look, I want to patent it. I'll give you the patents, but can you please patent them? You can own the patents." MIT wouldn't file them. So I went out and we took the money that we raised privately and paid for filing them ourselves and then tried to give them to MIT and MIT basically wouldn't take them.

Hendrie: Oh my goodness. Oh my goodness.

Hillis: So, I mean, but anyway we're worrying about a bunch of stuff like that, so in the days when universities really didn't like patents that much.

Hendrie: And that's really surprising for MIT considering the amount of money they made from the core memory patent over many, many years.

Hillis: Well, the thing was it was because I wasn't -- I was a student, so I wasn't working for MIT.

Hendrie: Got it. Okay, that's what the hitch was, okay, all right.

Hillis: So it really wasn't much argument. I mean, it kind of -- they were confused.

Hendrie: They were confused, yeah. You were just not in the normal pattern.

Hillis: Right. So we were doing all this startup stuff and I'd kind of forgotten to sort of plan for Richard Feynman coming. So we're sort of sitting around the first day, "Gee, we're here, what do we do now," and Dick Feynman walks in. He literally salutes and says, "Richard Feynman reporting for duty sir." And it's like, "What would you like me to do? What's my first assignment?" And it's like, "Oh, no. What am I going to do with it?" I mean, I haven't organized anything. I don't have any...

Hendrie: You've never worked in a company. I mean, you're a graduate student.

Hillis: And I've got Richard Feynman asking me what he's supposed to do.

Hendrie: Yeah, right.

Hillis: So I said, "Well, maybe you can see if there's any application in the machine to quantum chromodynamics." And he looks at me and says, "That sounds like a bunch of boloney. What do you really want done?" And I said, "Well, we don't have any supplies like pencils and paper," and he's like, "Okay, that's my job." And so he went off. So his first job was to go down to the stationary store and buy pencils and paper.

Hendrie: That's pretty good. That is pretty good. All right.

Hillis: But I think he was very helpful in sort of suggesting how I should organize things, and his model was the Manhattan Project. So he was like, "Okay, well you should have group leaders, and you should have seminars, and you should have..."

Hendrie: Because he had some experience from that from when he worked on it, yes, in at Los Alamos.

Hillis: Right, when he did that. So he kind of coached me in management. So I'm from the Richard Feynman School of Management. So that's how we sort of got organized and got going. And of course we quickly realized that we needed some people who knew more than we did about actually engineering things and so on and recruited in some more senior people that had actually built computers like Dick Clayton from Digital Equipment Corporation. He was crucial.

Hendrie: I was going to ask when you decided that you needed somebody like that and how you found Dick.

Hillis: Surprisingly late in the game and I don't really remember exactly how we found Dick. He was -- maybe Gordon Bell probably recommended him, because I was friends with Gordon.

Hendrie: Right. And Gordon invested.

Hillis: Gordon did invest.

Hendrie: At the same time I did.

Hillis: That's right. And that's right it was really an A list of investors. Claude Shannon was an investor.

Hendrie: Yes, it really was. Well, it was a really good idea.

Hillis: Yes, so it was fantastic. So I think Gordon probably found Dick.

Hendrie: Yes, because Gordon would know Dick and would know that Dick was -- because Gordon had left DEC by then, and Dick probably didn't want.

Hillis: That's right, and Dick had passed on the 11/45, the PDP-11/45. So anyway so I think Gordon said, "You need somebody who really," or we went to Gordon. I don't remember. But Gordon was definitely giving me good advice.

Hendrie: He was helpful, yes.

Hillis: And in fact it was Gordon, also the other piece of really good advice Gordon gave me was, not design my own DRAMs, which seemed so obvious in retrospect.

Hendrie: Yes, but at the time.

Hillis: But at the time it was just like, "Well hey, but I can. It's just silicon. I can design everything from scratch. Why shouldn't I build my own?"

Hendrie: Yes, if I can I should.

Hillis: So Gordon probably found Dick, and then Dick started bringing on -- Dick knew other engineers, marketing people, things like that, so people who were really the turning point I would say.

Hendrie: Yeah, people experienced at the commercial, the stuff you didn't know anything about, the commercial.

Hillis: That's right. Yeah, and so Dick was -- I don't think he ever really had the title, but he was basically president of the company in the sense that he was the one with the most experience about the computer business and how do you actually hire and organize marketing people, and sales people, and a purchasing department and things like that. So that was really Dick -- is the one that really knew how to do all of that.

Hendrie: When did the transition occur in your sort of thinking from this is a research project, we're going to build one of these and we're going to work on the AI problems with this and maybe we'll build two or three of these, but this isn't a commercial enterprise. I mean, there's some subtle transition that occurred at some point.

Hillis: So first of all you have to understand that I never thought of building The Connection Machine as the research project, because I still wanted to do the brain, the AI problem as the research project.

Hendrie: You just wanted this tool.

Hillis: This was the instrument I needed to -- so this is a side track.

Hendrie: So this was an instrument. This is a side task.

Hillis: This was a side track, exactly.

Hendrie: All right, pretty big side track.

Hillis: So it was a pretty big side track, yes. And in some sense, and the company, again -- at the beginning the company was for me a sort of a side track on a side track, because I never set out wanting to build a company. It was just in order to build this tool we couldn't do it in the AI lab. We had to have a company. So that was a side track on a side track. In retrospect I came to really care about building the company, and the company and so on, but going into it I didn't realize that.

Hendrie: That isn't how you got into it with that --

Hillis: Yeah, that's right. Probably I would have done a bunch of things differently if I had realized that that's what I was building, because there really wasn't a lot of thought put into how to build that other than getting great people, which we absolutely did.

Hendrie: Yeah, well that's one of the best things.

Hillis: Cheryl and I both agreed on that principle from the very beginning. And so we just had amazing people from the start. And of course having Dick Feynman from the beginning that made it really easy to get lots of other people to hangout there, so we had the Sydney Brenners and we had -- Jerry Wiesner used to come over there and work, and so it was a pretty amazing set of people. Eric Lander back in the days when he didn't know anything about biology, I think he did some of his very first gene stuff on The Connection Machine. So it was really neat because it would attract kind of brilliant people.

Hendrie: Yeah, it was just the environment that would attract.

Hillis: So that was, I would say, you don't realize having it at the time, but it just turned out to be a magnet for amazing people. That was the very best thing about it because in principle a faster computer or something everybody wanted to use it for something else. So the CM-1 [massively-parallel supercomputer] was all this little tiny bit serial processor that we could squeeze on a chip.

Hendrie: How many did you get out of a chip?

Hillis: I think we got 16 on a chip in that one and then it went up to 64, but the...

Hendrie: Sixty-four processors.

Hillis: Per chip, that's right. But the first ones had local memory. Later we moved the memory off-chip. But then it became clear that a lot of the things people wanted to do with them -- they wanted to do floating point calculations, and the bit serial processor...

Hendrie: That's serial floating point.

Hillis: Well, we could do it.

Hendrie: Yeah, of course you could.

Hillis: But multiplications were sort of in squared time. We could do it in variable precision. We could do 43- bit floating point and did sometimes. But it became clear that people really wanted a floating point unit that just did that. And by then you could buy...

Hendrie: This is as you discovered other problems that...

Hillis: As we discovered other problems. So the CM-1 basically only had the bit serial processor. By the time we got to the CM-2 it had floating point. They kept on going. By the time we got to the CM-5 they all had independent instruction units and they could -- so the trick of sharing the instruction fetch between them we were passing because chips had gotten bigger at that point so you could put -- you could give them each their own instruction fetch and so on.

Hendrie: Need a pause, change the tape, okay? And then we'll be back.

END OF TAPE 2

START OF TAPE 3

Hendrie: All right. We're in the middle of the development or The Connection Machine. We're sort of--

Hillis: Yes, so the basic thing that happens is although I started out just trying to build an instrument for doing something that I wanted, I end up with a computer company, which is something you have to sort of have to pay a lot of attention to.

Hendrie: Yes.

Hillis: -- which, at the time, seemed to me a little bit like an irritation of, hey, I just want to get this instrument done. But you can't really just sell one computer. You can't engineer one computer. We're taking this money from people; we have to sell some more to people.

Hendrie: Yes.

Hillis: And it turned out pretty quickly obvious that the AI market consisted of one machine to me, and that was not going to pay for the engineering.

Hendrie: Right.

Hillis: And that the place where people really wanted it were the floating-point people, and I remember saying, "Well, God doesn't use floating-point," which I later had to amend to, "Well, God might not, but the customers do."

Hendrie: Okay, very good, yes.

Hillis: So we quickly sort of got pulled in to the making of supercomputers, and then, of course, the super-computer market was all about Crays and vector machines.

Hendrie: But you had a really -- you had an idea that was --

Hillis: Well, yes.

Hendrie: -- was early in its -- was in its infancy, another way of doing it.

Hillis: And as it turns out, a really good idea was buried in that, and I don't claim to have thought of this, but we just sort of found it, which was that it turns out that the big flaw in the way that everybody was thinking about parallelism was this idea of dividing up the program and doing different parts of the program in parallel.

Hendrie: Right.

Hillis: And because we were thinking of very simple programs, running the same program on lots of machines <inaudible>--

Hendrie: On just tons of data.

Hillis: Yes. And as it turns out, that was a better way of doing all these other big parallel problems, too, which is do data parallelism rather than program parallelism so that you had essentially the same program. I mean it might be taking different branches on different machines, but the same program was running on every machine, but it was given different data to work on. And so that idea of data parallelism, I think, really turned out to be the key thing that made massive parallelism work.

Hendrie: Yes, okay.

Hillis: And it was one of these things that once you started doing it that way, you would never go back to thinking the other way. It was jus t-- and, indeed, that's the thing, that all massively parallel machines work that way today. A graduate student programming a machine, it wouldn't even occur to them to do it the other way because it's so built into --

Hendrie: Exactly.

Hillis: -- the obvious way.

Hendrie: All right.

Hillis: And so we basically added and built in the hardware of the machine what's now sometimes called basically map/reduce, sort of the fundamental thing of the machine. It maps a function of a whole bunch of different pieces of data, and then it reduces it by kind of a logarithmic effect. In The Connection Machine, both of those things were sort of built into the hardware. You had a way of broadcasting out the function that you wanted to put over the data. You had lots of local pieces of data, and then you had a network that helped you reduce them, so you could logarithmically add them up or something like that.

Hendrie: What were the biggest problems that you had to solve in the first Connection Machine? Was it the communications between all of these?

Hillis: Yes, I'd say the engineering problems that we had to solve.

Hendrie: Yes, engineering problems, yes.

Hillis: Yes, so certainly building the communications network was, by far, the most difficult, and those were the days -- so packet switching was pretty novel then, and I was familiar with it because the Internet used it, although there was no Internet. It was the ARPANET used it.

Hendrie: Yes, and you knew -- you go through ARPANET. That's right.

Hillis: So I said, okay, well, that's the way to make the communications system. We'll do it like that, and we'll do packet switching. But packet switching wasn't quite fast enough because we didn't have enough wires to make parallel wires, so these were a bit-serial. And just the time it took to serialize something took a long time. So this was sort of pipeline packet switching, that for a long packet, it would sort of be like circuit switching.

Hendrie: Yes.

Hillis: And so the router was the switch that was basically moving these packets through the network. We had tens of thousands of things sending packets all at the same time. And there was a real question with the switch. So I'd basically designed the switch from the circui t-- from the silicon up but almost did it as a switch that would lay out very well in silicon because that's kind of the way --

Hendrie: Yes, because you had the experience in that part. Think about it that way.

Hillis: That's right. And you could look under the microscope and sort of trace how things went, so it was laid out very geometrically. But then the whole system was connected into a hyper-dimensional cube or kind of the pattern of Fast Fourier Transform. And so you had wires one hop away, two, four hops away, eight hops away, and so on. So then the question was, well, was there any pattern? Were all these packets guaranteed to be delivered? Because we didn't have retry in it; we didn't have time to kind of retry something like an--

Hendrie: Okay.

Hillis: So you would send a packet, and if there was another packet using a wire, you'd sort of wait, and so there were little cues that would wait until that wire went by, and --

Hendrie: Okay, yes, so you've got to have buffer registers in there.

Hillis: That's right. So then the question was would this all guarantee it'd always deliver the packet, or would be there some situation in which the buffers would overflow.

Hendrie: Right.

Hillis: I was pretty sure that I designed in such a way that the buffers could never overflow, but I wasn't absolutely positive. I didn't really have a proof. But I knew all these smart people like Dick Feynman and Bell Labs' Andrew Odlyzko [sp?], and they all kind of got interested in this problem of does this algorithm really convert?

Hendrie: Oh, okay.

Hillis: And so they all started working on it, but Dick Feynman is actually the one that solved it, and he solved it with the craziest proof I've ever seen. The proof started out, "Let X equal the number of ones in the addresses of all of the packets." And then he wrote a differential equation in X.

Hendrie: You're kidding?

Hillis: And so he just formulates the whole thing. I've never seen a computer science problem --

Hendrie: Oh, my goodness!

Hillis: -- approached by this. Basically, he proved that X -- the limit of X is zero, so therefore, all the packets who were addressed relatively, they all got delivered.

Hendrie: Oh, my goodness! All right.

Hillis: It was a crazy -- and when we started, of course, we didn't have any computers that we worked on. We ordered some from my friends over at Symbolics, so we were their biggest customer.

Hendrie: Oh, my goodness, okay.

Hillis: But they had to hand-build them, so in the beginning, when we were doing those calculations, our only computer was -- we bought it from Radio Shack. We went down and bought a TRS-80, and so that was what Richard Feynman solved his differential equations on, was the TRS-80 to prove th t--

Hendrie: The design you had --

Hillis: -- the whole design would actuall y--

Hendrie: -- would actually work with the number of buffers you'd chosen beforehand and all of that?

Hillis: Yes.

Hendrie: Oh, wow. Okay.

Hillis: So, anyway, that was obviously the primary engineering. I mean there were a lot of sort of normal engineering things of power and cooling and packaging, and we had really interesting -- some package designers that were great. On sort of the aesthetic side, a woman named Tamiko Thiel, who really worked very closely with me to figure out --

Hendrie: It's a beautiful machine.

Hillis: -- and she got that basic shape of the cube of cubes. I knew that I basically wanted that translucent look with lights flashing in it, but if you just did it as a cube, it was kind of boring. She's the one that figured out -- well, she understood a hypercube enough to sort of get the idea of make the cube of cubes so that it--

Hendrie: Okay.

Hillis: -- and so we did put a lot into making it look good. In the end, people like Ted Belladel [ph?], who's somebody who did a lot of very complicated cooling stuff, that was a significant problem for it, but we really wanted to go with the air-cooled. That was a big advantage that we had over Cray because Cray basically had to put in more water pump -- the cost of the water pumps cost more than our machine. But the people that had millions of dollars to spend on computers were all people who were Cray customers, basically, and so we were, whether we liked it or not, selling against Cray because those are the people that had the budgets even though this machine really wasn't --

Hendrie: Wasn't ever designed to sell against Cray or --

Hillis: It wasn't designed to--

Hendrie: -- had a completely different objective. Who did you sell the first one to? Where did the first one go?

Hillis: So the first couple of them, one of them went to Perkin Elmer Corporation, where it got-- I mean it was all hush-hush and secret at the time, but it got used to control one of the very first adaptive optics systems for a space telescope.

Hendrie: Ah, a space telescope that might be looking down.

Hillis: Yes.

Hendrie: Yes, okay.

Hillis: And the National Security Agency was also an early customer.

Hendrie: Yes, all right.

Hillis: But we had a few -- Lockheed did something for designing airplanes very early on, doing fluid flow. And, actually, I'm sorry, they first got it to design what -- this was, again, hush-hush at the time, but the first stealth fighters.

Hendrie: Oh, wow. Okay.

Hillis: And so that was--

Hendrie: Yes, and so simulate the --

Hillis: They were trying to <inaudible>.

Hendrie: -- to understand these very complex reflective patterns from those radars. Okay.

Hillis: So there were a bunch of people that had budgets around for various things. And then, but over time, who became really important customers for us were the oil companies, so Schlumberger and ExxonMobil, those became really important lead customers for us because they were using it for seismic processing, and that was just -- could use huge amounts of data. So I would say that that ended -- and reservoir modeling. So there was a whole bunch of stuff that was basically simulating using finite element analysis or finite difference analysis of various sorts, a lot of simulation kinds of things.

Hendrie: Yes, where there are massive amounts of data.

Hillis: Right, and that worked beautifully in the parallel space because you basically divided up some piece of this simulation problem and put it on different processors, almost like an analog computer. In fact, when I first went to Schlumberger, what they were using to do -- they literally were soldering resistors. They were taking big sheets of tens of thousands of resistors and making analog models of the resistivity of [tectonic] plates and things like that --

Hendrie: Oh, my goodness.

Hillis: -- and then applying voltages to them and seeing what voltages they got off.

Hendrie: Oh, wow.

Hillis: And, of course, that all went away really quickly as soon as you could compute.

Hendrie: As soon as you could compute, yes, exactly. Oh, that's pretty interesting.

Hillis: So there was a whole class of modeling things. There was a class of kind of cryptographic things.

Hendrie: Yes. Now, all of those seemed to be applicable to the first Connection Machine?

Hillis: Yes.

Hendrie: What stimulated you to move to --

Hillis: There were more floating-point calculations because everybody was lobbying for a standardized floating-point at that point, so that simulated the move to the CM-2.

Hendrie: Okay, and the CM-2 had floating-point at each?

Hillis: That had floating-point units.

Hendrie: For each of the processes?

Hillis: It was a little bit funny. What it did is it had a pipelined floating-point unit so that you had a -- so instead of having 16 floating-point units that you all used on one cycle, because this was bit-serial, you put your bits out in bit-serial and sort of turned them sideways and pipelined them through the system.

Hendrie: Ah, okay.

Hillis: So it was kind of a kluge, but it was an add-on.

Hendrie: Oh, it was an add --

Hillis: It was an add-on to the CM-1 chip. So you still had the basic CM-1 processor chip, and then you had this floating-point unit added.

Hendrie: And another chip that was the float -- was it another chip or--

Hillis: Yes, there was another chip that was the floating-point.

Hendrie: Floating point, yes. So you would pair up a 16-processor chip and a floating-point chip?

Hillis: That's right.

Hendrie: That's sort of --

Hillis: And then, of course, we quickly realized, and everybody wanted more memory, so the CM-2 had a lot more memory on it. That was the off-chipped. I think we went -- CM-1 had static memory that was off chip, so we went from on-chip memory to static memory to dynamic memory --

Hendrie: So big dynamic range.

Hillis: Big dynamic memory. And then what happened quickly is the constraint of- - because the way that the instruction units were shared, basically -- well, different processors should take different branches.

Hendrie: Yes.

Hillis: They sort of had to wait for each other till they got back together to move forward. And so that put a lot of constraint on the programming when you had a highly branched code and so on, so --

Hendrie: Yes, it was -- when the processors weren't literally -- when the program was not very data-dependent --

Hillis: Right, it was just--

Hendrie: -- and could just roar through giant amounts of data with the same code, so as soon as- - there's data dependency--

Hillis: -- <inaudible> level of synchronization.

Hendrie: Yes.

Hillis: So we had to synchronize it at a finer level than we really wanted to.

Hendrie: Yes, I got it.

Hillis: So, and by then, microprocessors were becoming cheap enough, so we just-- we should just build this out of standard microprocessors. They had developed to that point. So by the time of CM-5, we were buildingout of a lot of standard microprocessors.

Hendrie: Was there a three and a four or -- in somebody's head?

Hillis: Yes, in somebody's head, there were sort of hybrid things that were going in between it, but there was one where we were designing our own. We considered the idea of should we design our own microprocessor because the microprocessors weren't quite there, but we could see them coming, and we kind of figured out by the time we designed it, there would be ones that were available. So it's always kind of a guess where is technology going to be. You're always trying to kind of --

Hendrie: A little bit of predictive, technology predict, that's right.

Hillis: Yes. So by the time CM-5, we were using off-the-shelf microprocessors --

Hendrie: What did you end up using?

Hillis: -- but we had our custom-- those had SPARCs in them.

Hendrie: Okay.

Hillis: And we had our custom chips, but they were mostly used for the communication and so on.

Hendrie: Yes, you obviously still had the communication, right.

Hillis: Yes, and we still had the ability to work them in that very synchronized, closely synchronized mode, which was really useful for debugging things --

Hendrie: Yes.

Hillis: -- because you got rid of all the race conditions and things. So you can synchronize them, and that, you kind of synchronized them as much as you wanted to. And, typically, you only would end up synchronizing them at least kind of communication cycles, where you bring all the communication together.

Hendrie: Okay, where everybody shuffles the results to the correct next process.

Hillis: But by then, we were also running such high speeds that I mean a wire in that machine could have five or 10 bits along it.

Hendrie: Yes, okay.

Hillis: And we still had a synchronous model of the machine. That's the wild thing. So every one of those wasn't like -- don't do this these days. You pay a cost of a little bit of possible synchronization error or something like that.

Hendrie: Right.

Hillis: Seems to work. But what we did is we actually kept all those processors in a kind of general relativist -- or special relativistic synchronization, which was that each one has a phase locked loop on it, and it phase lock looped to its ______.

Hendrie: Ah, okay.

Hillis: And then the wires had a variable delay which was phase lock looped in such a way that the wire was always some number of cycles long. So it was either three cycles long or two cycles long or four cycles long. So you just treated that like a pipeline. So you put a bit in. So the whole system was like a synchronous system. It was absolutely deterministic. And even the refreshes and things like that were determinist or you could synchronize. And so the whole thing, because we were all scared at that point of what happens if you get all these thousands of processors and it does something a little bit different? How do you tell?

Hendrie: Yes. <Inaudible>.

Hillis: Everything is always sitting -- these days, I think that people just kind of ignore that probably, and you just assume that it all works pretty well, but we were very paranoid about it because we'd just never done that. And then the other thing that we did in that is that we did the disks the same way. We said, "Oh, well, let's have parallel desks." But then we realized you've got a reliability problem if the disks are less reliable, so you have to do error correction across the disk. So that was how the first RAIDs got put on those machines. That's right. So Thinking Machines ended up with all the basic patents on RAID disk arrays, too.

Hendrie: Wow. Now, when was this? Was this done on the first-- on the CM-1, or when did you--

Hillis: The RAID disk arrays were probably done on CM-2.

Hendrie: Okay.

Hillis: And I remember us saying, "Well, you know, mechanical disk units, how long could those last? Should we even bothering filing these patents?"

Hendrie: Little did you know!

Hillis: Because surely we're not going to have things spinning around!

Hendrie: Oh, my goodness, yes, exactly. Surely not! It's such a ridiculous system! That's funny. That's funny.

Hillis: So more and more, the design in the machines became a general purpose thing, and in fact, by then, some of the things that I had wanted to try in AI actually got more enough known about them so that they got realized to be harder problems than we all thought that they were. In some sense, the things I wanted to do had had sort of become more like what's now called Semantic Web, and it's doing inference across those kinds of things. And that's, like all these things, turned out to be a much harder, more subtle problem than we ever imagined it was.

Hendrie: Yes, yes.

Hillis: But the place where it ended up, it ended up being applied a lot to AI because people started doing these connectionist models, where they started really modeling neurons, and we actually ended up selling some machines for that. So those were the only machines that we really sold for AI, that and vision research.

Hendrie: Okay. So you did end up selling a few machines?

Hillis: We ended up selling a few machines for AI, but it was never a significant part of the market for them.

Hendrie: Wow.

Hillis: A significant part of the market was the kind of main supercomputing market that Cray had established.

Hendrie: Yes, and it was just some of the problems with the way the data was essentially being processed worked with your machine in a way that -- better than Cray.

Hillis: Yes, and, well, way better except that in our machine, you generally had to express them in a different way than you expressed them for the Cray because the Cray, they had all vectorized them.

Hendrie: Right.

Hillis: And sort of optimize it for the vectors that were like 100 units long or something like that, so the way of expressing it. So it turned out the programs pretty much needed to be written differently for The Connection Machine, and so that was the toughest thing in the market was always-- if people would rewrite their programs, they could get a factor of 1,000 in speed, but usually, the person that wrote the program didn't work there anymore. And, also, the other thing, there was a huge -- I'd say that Cray was much more sophisticated than we were about selling it and marketing it and the politics of it and so on. And so they were much better in the marketplace than we were in a lot of ways.

Hendrie: Yes, they were much more effective in the marketplace.

Hillis: Yes, yes.

Hendrie: Yes. But I could see that the software, where all -- Cray just needed to make a vectorizing version of Fortran, and lots of codes would run pretty darn fast. You needed a -- the idea would have been take a Fortran program and somehow have it compile to your machine.

Hillis: But, of course, in the end, all of that was a sidetrack. Fortran programs were a sidetrack, and I kind of sensed that at the time, but it was really where the market was. But, of course, if you look today --

Hendrie: People don't do that.

Hillis: -- people never converted all those old Fortran programs. Just a new generation came along and--

Hendrie: And was willing to rewrite them so they really got fast in a distributed model.

Hillis: Yes, and it took a generation to do that.

Hendrie: Yes.

Hillis: But a lot of really interesting people sort of saw that and started programming and that, and it was really fun, so the amazing people that you wouldn't guess had anything to do with computing or parallel -- I mean like Neal Stephenson, the science fiction writer --

Hendrie: Let's pause. Let's see. Let's wrap up the -- you had thought about some of the issues of what you were doing. What sort of-- how did The Connection Machine story sort of wind down, or what's the end of that phase story?

Hillis: Right. So it had a very happy, prosperous 10 years of selling machines, followed by a couple of really bad years, where we basically ended up having to break up the company and sell the hardware of it. Part of the company got acquired by Sun, and eventually, the software part of the company got acquired by Oracle.

Hendrie: Okay. All right. But it was a great ride.

Hillis: It was really an awesome ride, and it seemed like a disaster at the time, but every time I run into somebody that worked there, they say, "You know, that was the best thing I ever did in my life."

Hendrie: And the most fun, and it felt productive.

Hillis: It was really fantastic, and it was really an amazing group of people not just working at the company, but also, our customers were just an incredible group of people, the people out there on the edge with hard problems that were just right on the edge of being solved in every possible area of science and engineering and also commerce and other things like that. So it really was an extraordinary group of people, and it's almost like a secret club that we're still all a member of.

Hendrie: Yes, exactly. That's cool. That's really cool. I know you've done lots of other things, but I don't think we have time to really get into all of those things, but I wanted to ask you a couple of general questions. What, at least in the period that we covered so far, what are you sort of most -- proudest of accomplishing?

Hillis: I think that I'm most proud of accomplishing is that I think it was something that -- it was that capturing the imagination of that set of people, and I think over the long run, that's something that's made the most difference is those people have gone off and done other things that had nothing to do with The Connection Machine, but I think a lot of them sort of either got inspired by that or learned ways of thinking about that, or so on. And I keep on finding more of them that are sort of different, and I found out the other day that Sergey Brin, one of the founders of Google, was one of these Connection Machine programmers as a student.

Hendrie: You're kidding!

Hillis: No.

Hendrie: Oh, my goodness. Oh, wow!

Hillis: Yes. So I think that the thing I'm proudest about is it inspired a sort of people that then went off and did much better things, and that's the -- having, in some sense, being able to crystallize a community of people like that, I think, was the best thing about it.

Hendrie: Do you have any advice you would give a young person who seems to be interested in science or engineering?

Hillis: Absolutely-- best thing is find the best people. Find the people that really inspire you, and move in with them.

Hendrie: Go make a connection with them.

Hillis: Yes, make a connection with them because that's really the thing that has made the biggest difference in my life is that I've always just made sure that I hung out with people who were way smarter than I was and really learned from them.

Hendrie: Okay. All right. Do you have any thoughts about why it might be important to have a computer history museum and preserve the history? I know you were involved with it very much in Boston, though not so much now. What would you say to people about that?

Hillis: Well, so it's clear that when historians look back at this period in history, if they were going to have like a one-liner about what was going on right now, it would be computers. It would have something to do with computers and digital. I mean this is the changing thing in our moment. This is the catalyst of all change at our moment in history. There was a time when it would've been the railroads or an airplane or something, but right now, it's computers. And by right now, I mean these decades that we live in. And so the thing that everybody will want to know about now is what happened with the development of computers right now. So historians of the future, this is the information that they're going to be interested in about this period of time. So politics comes and goes, or lots of movie stars come and go. That will lose its interest over time, but humanity will never again go through this moment of transition in history. That will be something that is of permanent long-term interest to humanity.

Hendrie: All right. Well, thank you very much for taking the time, Danny, to do an oral history for the Computer History Museum. I'm sorry we didn't have time to talk about some of the other exciting things you've done in your life after being a computer architect and designer.

Hillis: Well, you can come back and interview me in another 20, 30 years about the 10,000-year clock.

Hendrie: Very good.

Hillis: It's a deal?

Hendrie: All right. It's a deal. Thanks.

END OF INTERVIEW

