Oral history of Edward S. Davidson (1939-)

Interviewed by Prof. Paul N. Edwards, University of Michigan School of Information, at Davidson's home in Ann Arbor, Michigan

June 17-18 and September 29, 2009

Edward S. Davidson is a computer engineer who pioneered pipelining techniques for improving processor throughput in both hardware and software. Techniques to which he contributed seminal ideas include simulated annealing, wave pipelining, multiple instruction stream pipelines, decoupled access-execute architecture, and polycyclic scheduling (aka software pipelining).

Born in 1939, Davidson grew up in the greater Boston area. He saw his first computer on a high school math class field trip and "knew then and there" that he wanted to enter the field. Davidson received his bachelor's degree in mathematics from Harvard University in 1961, having taken virtually all of the university's computer courses. He next attended the University of Michigan (1961-62). There he obtained an MS in the Communication Sciences program, which featured a broad intellectual range running from cybernetics and neuroscience to computing. Davidson worked for Honeywell as a logic design engineer from 1962-1965, acquiring considerable experience with circuit optimization. From 1965-1968 he attended the PhD program in Electrical Engineering at the University of Illinois Urbana-Champaign (UIUC). He then moved to Stanford University, where he served as an assistant professor from 1968-1973 and started a logic design laboratory.

In 1973 he returned to UIUC, where he remained until 1987. During that period he worked in the Coordinated Science Laboratory (1973-1980), served as President of Illinois Computer Research, Inc. (1981-87), and co-led the development of the Cedar, a highly parallelized supercomputer. In 1987 he was recruited to the University of Michigan's College of Engineering, where he remained until he retired in 2000. During that period he led research for Ford Motor Company on computer simulation of crash tests, among many other things.Throughout his career Davidson made numerous contributions to logic minimization theory and technique. Davidson trained numerous PhD students, most of whom who went on to become important computer engineers in their own right; he feels that graduate education was among his major contributions to the field.

This oral history generally follows the chronology of Davidson's life and career (see table of contents), with occasional digressions. While most of his students are mentioned, the interviewer asked Davidson to discuss a few of his papers and PhD students in more depth. Finally, the interviewer would like to note that Davidson is a natural storyteller, with a wonderful, light sense of humor and a gift for extracting lessons from his experiences, and recommend that users of this transcript also consult the audio files (indexed by timestamp in the transcript) to get a sense of his delightful personality.

Some of Edward S. Davidson's major awards and achievements:

- Eckert-Mauchly Award, 2000, IEEE Computer Society and ACM, “For his seminal contributions to the design, implementation, and performance evaluation of high performance pipelines and multiprocessor systems”
- Taylor L. Booth Education Award, 1996, IEEE Computer Society, "For contributions to the establishment of computer engineering as an academic discipline and for nurturing many leaders of this field during their formative years in the profession. "
- Harry Goode Memorial Award, 1992, IEEE Computer Society, "For pivotal seminal contributions to the design, implementation, and performance evaluation of high performance computer systems.
- Fellow IEEE, 1984, "For contributions to the use of pipeline structures in computer architecture."
- Designed and implemented an eight-node symmetric multiprocessor (SMP) system (1976)
- Developed the reservation table approach to optimum design and cyclic scheduling of pipelines (1970s)

Table of Contents

Interview recording part 1

Overview of major contributions

00:00:08 Paul Edwards: This is Paul Edwards. I am doing an oral history of Edward Davidson at his home in Ann Arbor and it’s June 17th 2009. And Ed has provided me with a lot of material and some of the best preparation I've ever had for an interview. I decided to set a context by asking you to talk about what you consider to be your major achievements and your major research areas. We will go into the chronology of your life and work and after that, but it would be nice to hear a bit about how you think about your own career and how you consider major contributions have been.

00:00:57 Ed Davidson: When people who don't know me ask me what I do for living or what I did for a living, I would usually say I teach Computer Engineering. And for my whole academic career that's how I thought of myself. My love of the work and the part I always liked the best was the work that I did one on one with my graduates students. it’s the human relationship there and the process of what we produced. Perhaps you more than the content. As far as research areas, from my PhD onwards I began in a computer-aided design, Logic Minimization area using branch and bound techniques. And branch and bound were optimizing in combinatorial spaces, one theme that has popped up again and again, I had a habit of going after large combinatorial problems trying to optimize them like an idiot. [chuckle]

And early on I got into pipelining and had a... Well a decade and then I later returned to pipelining as a major research topic. And that's perhaps the work that I am best known for. Yeah, and that and its extensions really set the stage for the follow on phases, which got into various problems. Like once you once you have a pipeline that is theoretically optimized, it’s running at a fairly fast rate and trying to overlap and how do you keep the beast fed, which leads off into memory problems and caches and branch prediction -- which I didn't particularly work in -- but just trying to predict, trying to get the information that you need close to the pipeline before you need it or better yet, before you know you are going to need it. Which is a difficult problem, pieces of that are getting solved. And that leads to interconnect problems and then pushing some of applications you get into parallelism and I got into big iron kind of by accident the Cedar project with David Cook.

00:04:13 PE: the parallel supercomputer?

00:04:15 ED: Yeah. And thereafter... Well you could argue that my role in Cedar was administrative, except it never quite became completely that. [chuckle] And then more administration at Michigan but then a chance late in my career to do a major project with Ford which involved supercomputers and application tuning, for architecture specific application tuning, which was a great way to cap a career. Got me into another, new domain which was a lot of fun. And something that was directly useful, or so we thought at the time. Seems to have been.

00:05:05 PE: Okay. You said in our communication earlier you didn't have much to say about this right now but maybe it all sparked something. I'd just like to read something you said in a talk. Can I do this? [chuckle]

Interview recording part 1

00:05:19 ED: Sure.

00:05:20 PE: (quoting Ed Davidson) "Because I myself, I'm still mostly happy to let others worry about device density and speed. Well, I look primarily at concurrency; how much can be done simultaneously during one clock cycle, how can I avoid stalling because something is not done yet." So I am curious about that, because it sounds like your own articulation of the unifying theme in your work, in some of what you just said.

00:05:45 ED: I am not exactly sure when I said that... It must have been one of the later talks. When you get to a certain age people force you to look back on your life as though you're already dead. And so I've done a bit of that. And then you try to find the themes living through it, of course it’s total chaos and you know afterwards there are patterns that emerge.

I always thought of devices as bricks. And I thought of myself as an architect. I had been calling myself an architect, and many people in my field do, for a long time before I really learned this is not what IBM meant when they coined the phrase architecture, to refer to computers. Where they meant instruction set architecture, I mean architecture like a building; style of construction. And that was always more interesting to me. When I worked in computer-aided design, I was minimizing gates and so on, but I wasn't designing electronics technology. It just didn't appeal to me. And it was very late in life that I learned and I talked about this a little bit at the Mac van Valkenburg Memorial where I was looking back at my years in Illinois, and making a tribute to Mac who is in Systems Theory who is also isn't involved with the bricks. But you know in a nod to physical electronics people, I mentioned that I had recently learned that to the people who work in the physical electronics, the devices are not bricks; they are systems.

00:07:55 PE: It's systems all the way down...

00:07:56 ED: It's materials. [chuckle] It's always systems because... So what we have is a hierarchy and Yale Patt has a very nice breakdown of that, of levels of design. And you start with physical levels and then you move on up through the logic design and structure the computer, I spent a lot of time there. And then you have the, what I would call the computer architecture, the higher level structures of pipelining parallelism and that sort of thing. And then you move on into system software and into application software and so on. Everybody focuses at one or two of those levels in their career, 'cause there's a limit to how much you can know. There are few generalists, but primarily that. And my advise to students and others advise to students usually has been your career would work better if you have a strong working familiarity with at least one level below your own and one level above your own. Because its in those joints where the real problems lie. And that's where the trouble comes. You can live narrowly in your own field of expertise, the world is a relatively simple place, but the problems may not be relevant.

00:09:31 ED: So at least one level up, one level down. I don't know, they always... Transistors always looked like bricks to me. And I had this problem when... If you don't mind me free-associating a little bit...

00:09:46 PE: Go on.

Interview recording part 1

00:09:47 ED: Okay. At Illinois we were running a Computer Engineering degree program, which was an undergraduate program, that I was brought to Illinois to help develop, and bring to fruition, the curriculum was very cluttered. And that's always been a problem with engineering, that we make a lot of requirements. And I got into a... We were looking to lighten it up and allow students more electives space, which is a fashion today. And there was a debate. You know the traditionalists were saying that the students don't know what they need and faculty do, and we know what makes an education and we should not have any electives and everything should be required. You give it to them and that's what they take, and then they can wear the label.

So one of the courses we were trying to shed for a requirement of computer engineers was a course in physical electronics -- back to your question. And that particular course wasn't just about transistor circuits and things like that, it was Fermi levels of atoms and devices and diodes and... So that's just so much I know about it. So I was... And we had area committees which ran different degree specialties and I was asked, as a representative of the Computer Engineering Area Committee (which ran our degree program), to talk to the Solid State Area Committee about eliminating a key requirement in their area. And Nick Holonyak was on that committee. And Holonyak had been the prize student of John Bardeen, and is now a very eminent person in his own right, holding key patents on LED's and so on, he was furious.

Which was a kind of his normal state. He wore red fireman's suspenders and he was often furious and quite vivid in his speech patterns. But he waved his finger in my face and said "What would graduates of your program do, if the world suddenly changed from silicon to gallium arsenide? How would they survive? And the purpose of education is not to train people in the technology of the day." With which I heartily agree. "It's to give them survivability in these changes". As usual I didn't think of an answer at the moment, but as soon as the meeting got over the answer came to me in a flash. Which is: it would make no earthly difference. When we deal with device technology at that level we have a set of rules, and when the technology changes there is a different set of rules and you adapt to that set of rules. You don't have to know the underlying physics; we are not working at that level. It's good if somebody does. But I didn't have an answer at that time. So Illinois students had to go on learning about Fermi levels. [laughter]. Then Mac van Valkenburg did a beautiful thing from a circuits and systems point of view. You know, a series of essays on un-cluttering the curriculum, which was a serious problem in Engineering.

00:13:36 PE: I'm gonna ask you to talk more on that later, because that one piece I was not really able to find, inside a couple of papers, you wrote about that, the undergraduate curriculum that...

00:13:47 ED: Oh! I am not surprised. They were obscurely published.

Early Life: Family and Education

00:13:54 PE: Okay. Let's now move into the chronological mode and kind of just through your working life. And actually even before that, let's talk about where you came from. Where you were born? Where you went to school? What your parents did?

Interview recording part 1

00:14:12 ED: Well, what colors a lot of my political view I think is that my great grandfather -- I'll do this briefly and move on -- but he grew up Russia and he came to... His brother served in the Czar's army and he was in ill health when he came home and died shortly thereafter. I think a common condition for many people in the Czar's army, but particularly also — so Jewish people say — for Jews in the Czar's army. And the family gave my great grandfather the name of his dead brother.

00:15:02 PE: How old was he at the time?

00:15:05 ED: He was a young married person...

00:15:06 PE: Still a boy...

00:15:07 ED: no. I'm sorry. At the time I don't know. But he went by his brother's name from then on so that he was exempt from service.

00:15:18 PE: Ah! Because they would think that he...

00:15:21 ED: Had been killed. Yes. And his name was on his brother’s tomb stone. When he was a young married person with several children... He was kind of a large strapping and intimidating guy, and he got into a fight with somebody and his friends came and told him that the guy he was fighting with was going to the authorities to report what had happened. So he left suddenly for America. Left one child at the border with relatives because she had some kind of disease that US immigration would not allow into the country. Came here, worked at a pants presser in the New York garment district, earned enough money to buy a ladies' coat store. His wife at that time told him that they should leave New York and move to Boston because there would be more eligible Jewish men for their daughters in Boston. Why she thought that was true, of Boston versus New York, I don't know. But they moved the coat store to the Italian north end.

00:16:33 PE: This would have been in the early 1900's?

00:16:37 ED: This is early 1900's, yes. And the immigration, I don't know if that's late 19th century or early 20th. And his particular skill was that he could look at a coat in a store window like Filene's or Jordan's, because somebody told him they wanted that coat, and he could go back to the store and without a pattern he could cut the coat.

00:17:04 PE: Wow!

00:17:06 ED: So that was his forte. He outlived his first three wives and the fourth one outlived him. He calculated how long he expected to live -- which is the problem with retirement, if you know when you're going to die you can plan perfectly. So he said I'll probably die at 84 and he budgeted whatever he had to run out at age 84. And he died at 83 and 1/2. [chuckle], leaving me $500 for my first year’s college education. Which looked like it might be enough at that time. By the time I went to college it cost two thousand dollars. By the time the kids went to college it cost twenty thousand dollars. I was the only great grandchild he left anything to, my mother assured me that's because I was his favorite great grandchild. But I think it was probably because my mother was one of his favorite granddaughters. My father had died when I was six and he might have felt a little bit of an attachment.

00:18:19 PE: So you were raised mainly by your mother?

00:18:21 ED: Yeah.

00:18:21 PE: And did she remarry or was that...

00:18:23 ED: She remarried, yes. She was a very strong woman and a huge influence over my life. Her values and so on.

00:18:31 PE: What was her name?

00:18:32 ED: Marion. Spelt with an 'O' for some reason, M A R I O N which is the male spelling. She... Her mother loved opera, but didn't have any formal education. And a rich relative advised my grandmother that my mother should go to a business high school curriculum because girls don't need college. And this was a rich relative who sent their own daughter to Radcliffe. My mother decided that she wanted more than that out of life and after graduating from the commercial high school program and seeing her younger brother getting ready to go to college, she decided she wanted to go. So she studied for the college boards on her own, found where they were being given, passed everything but math and flunked the math, found out that the math exam was being given in another city on another date in two weeks and passed it. Got into BU, she wanted to be a music major, she'd been teaching piano from high school age until age 65 or so, died at 72. Two years older than I am now. But they didn't have music majors. So she majored in biology and took music.

[laughter]

00:20:18 ED: A lot of my values come from her and a lot of my feeling of being an insider and an outsider comes from that history I would say.

00:20:31 PE: Say a bit more about that. What were the values that she communicated to you, and the insider and the outsider?

00:20:39 ED: Well I was always kind of rebellious, I was a bit of a free spirit. I didn't want the attention that I got as an only child and particularly the only child of a widow whose husband that she loved was dead. She used to say, "You don't have to be the best. It's not important to be the best, top 10 percent is good enough." [laughter]. And gradually I knew that I didn't have to be the best and in many cases I couldn't be the best, but being the top 10 percent was not too bad in high school. It was more difficult in college and it got more difficult all along. And top 10 percent of what for goodness sakes.

When she died I kind of went over my feelings about my life and the pressures that I was still feeling and was saying to myself, who am I trying to prove anything to anymore. She's gone. I don't have to do this. But the pressures were all internalized by that point, there is a limit to what you can do. So I spend a good bit of my professional life in that time, which was kind of mid Illinois time, kind of trying to get control of that one way or another and succeeding and failing in various ways. Her husband, Herm Steinberg (I carry the S as my middle initial), grew up in the north end of... I'm sorry.

00:22:28 PE: Just... Your father... I believe you were born in 1939?

00:22:33 ED: Yes.

00:22:34 PE: How did your father die? Did he die in the war?

00:22:37 ED: No. He had rheumatic fever as a child and they didn't have drugs like penicillin and so he was left with an enlarged heart, which my mother knew only after she had fallen in love with him. And one of the ground rules on getting married, the agreement was that they would not have children. Because he was not expected to live for very long. And years went by and she took very good care of him and he took very good care of himself, and she's getting older and she says "Why?". And so they decided to have me for which I am quite grateful. [chuckle] So then there is a lot of pressure on that. Well... To do it a little bit chronologically my father Herm grew up in the north end, went to... Well, west end, went to the west end settlement house which was for children with immigrant families. They used to give two scholarships a year, one to Harvard and one to MIT. And he was the second pick in his year. And the first pick chose Harvard, so he went to MIT and became an engineer and worked for the Boston Edison company. So that's where my engineering blood comes from; it's quite an accident.

00:24:02 PE: So he was an electrical engineer?

00:24:03 ED: He was an electrical engineer. When people asked him what he did he used to say he was a kilowatts salesman. He'd go out and try to convince people to switch from gas to electricity in their plants and stuff like that, ovens for bakeries and things like that. He taught at Wentworth Institute at one time. I don't know how many of these stories you want. There are many.

00:24:28 PE: I'm sure. You were about to say something before I interrupted you. Him, Herm Steinberg. So he is your father?

00:24:37 ED: Yes

00:24:39 PE: For a minute there you were starting to talk about him in relation to her.

00:24:42 ED: Yeah okay. So I loved him very much. But my memory of him is pretty much based on what my mother kept refreshing in my mind. And it's an interesting thing, when your father dies when you're six, because he's pretty much a mythical beast. And my mother attributed all forms of perfection to him. And that was the model that I was supposed to live up to. I mean I wasn't his replacement in bed but I was his replacement in her heart. And I was a little version of him. And that's a horrible thing to do to a kid; that's a heavy weight. She was appalled when I told her it felt that way; she never intended it.

She remarried when I was eight to a lovely gentle man Sid Davidson, who adopted me a couple years later. So he is my stepfather and he was an accountant. And a damned good public accountant. Not a CPA, he never did pass the exam, but he did his own system and he had a very close relationship with his clients. And whenever we'd go on a family vacation he would disappear and come back home for dinner late as usual with another client, who would be some small business operator in the area and so on. I learned from him about relationships with people and what it means that you work with somebody but you develop a personal relationship with them and that colored a lot of my thinking. But my mother was very strong and Sid was very gentle and they never had any kids and in many ways he wasn't a real father in the controlling sense of the word. He was certainly a father in the loving sense of the word.

It's an interesting way to grow up. I always felt sensitive about relationships with people and that has been my core. My mother's sense of life was everything in moderation. So I would tweak her by saying, "As my old mom used to say, gluttony is next to godliness". And she was not fond of my little excesses. Playing in the football band on Saturday mornings instead of going to physics class was not what she was sending me to college for. Interesting way to grow up.

00:27:59 PE: Okay. So where did you go to school? You stayed in Boston the whole time as you grew up?

00:28:06 ED: Yeah. I began in Roxbury at the William Lloyd Garrison School, named for the liberator and abolitionist. It was anything but a liberated school. A book was written about that school by Jonathan Kozol called *Death at an Early Age.* That was several decades later, but I recognized the place. The first day in kindergarten the teacher said you have to line up now because you have to go to the bathroom. And I raised my hand and said "I don't have to go to the bathroom". And she said "Yes, you do, hand up". And I said "No, I don't". The problem was that I thought go to the bathroom was talking about a physical biological need, and she wanted everybody to go to the bathroom because she wasn't going to be in the classroom and everybody had to be attended and marched in single file to the bathroom at the appointed hour. We weren't communicating, she thought I was being insolent, which I really wasn't.

And she walked over next to my desk which was screwed to the floor she reached in between the buttons of my shirt and lifted me to a standing position, ripping a couple buttons off, which is how my mother found out about it. It's an interesting school. Somewhere in the middle we were supposed to learn geography and we started with Norway and teacher had one book and she wrote in chalk on the blackboard and we had scratchy pens and wrote on paper with wood chips in it, and copied down what she wrote on the blackboard that was geography. I never knew what Norway was. We had release time for religious education, which was something I'd done around third grade. My mother was one of the few renegade parents that didn't allow her child to go to religious time because she didn't believe in a religious education.

00:30:16 PE: Christian religious education or...

00:30:18 ED: Anything. You know, you're a Christian you go to a church, you go to a temple, you had buses, they got you there, they got you back, you know, and teachers had a few hours off. Teachers were not fond of the kids who didn't go to release, during the school day, public school. So, the five of us, whose parents didn't allow us to go, were assigned long multiplication problems, which was good for me because I always got D's in arithmetic and penmanship and I eventually learned to do the stuff correctly. Got D's in arithmetic because I refused to memorize the multiplication table. It's funny. So there's always sort of crosscurrents.

We moved from there, in fourth grade. By that time my mother was remarried and I was adopted. I actually proposed to her second husband, asked him to marry my mother, 'cause I decided, life wasn't too good with her teaching music all the time and me coming home and rattling around the neighborhood until she got home at seven o'clock at night. [chuckle]. And he said we'll see what we can do about it and eventually did. And we moved to Newton. And out of Roxbury, smack into not only the most progressive school district in the area but the most progressive experimental school in that district, the Ward School. We had movable desks, teachers wrote assignments on the board, for what we were supposed to do that day, and you know, you do them in any order you feel like. Most of the kids didn't feel like doing them in any order. So they just walked around, talked to their friends, and I always did a... You know, the first one first and the second one next and you know, kept going till were done and I was done usually by noon. 'Cause school was a terror for me.

By that time I had learned to conform. But the discipline was good and the school was good. We were signed up for a theme to write. I had no idea what they were talking about, well write something about something, I don't know. So, my parents had bought a Compton's Encyclopedia. So I looked up a topic that I thought was what they were talking about and I copied two paragraphs on a piece of paper and turned it in. And they said "No, no it has to be in your own words", nobody ever talked about my own words before, it was not something that I had learnt. I didn't take to reading very naturally and I'm still a very slow reader. I read at one speed. I read at technical speed. So you know, although I read a lot of politics and so on I specialize in two page articles. I don't read books. I've read few books in my lifetime. I just grind, I can't skim, I even took speed reading courses, they had no effect on me.

00:33:25 PE: [chuckle] Interesting.

00:33:26 ED: But math and particularly, not the math I hated, the calculation, but the abstract structure kind of math, had such a beauty to it. And it's made for obsessive compulsives who only do detail work. And so...

00:33:48 PE: Do you remember how you got into that? Whether teachers that inspired you or...

00:33:52 ED: Yeah. The romance with math began with algebra and geometry in 8th grade. It was a junior high school teacher, Joe Nathanson.

00:34:09 PE: Sounds about right.

00:34:12 ED: And he used to do weekly tests and then if you finish the test first you could help grade the tests when the other students turned them in. So my friends and I used to race to finish them first. And what I learned after the first term was that if I finished first I didn't get very good grades, I'd make a lot of mistakes. And after that I never finished first. And I learned how to do arithmetic correctly because if I wasn't doing arithmetic correctly I was destroying the problem. And I became very compulsive about that and it's very natural to me now. But what happened was, it opened up a space. It was the geometry, and the beginnings of algebra.

I could see the space and it was just a wonderful feeling to move around in this abstraction and see what games could be played, and what happens when you change this and when you change that and where is it going. And the idea of proving things was just wonderful to me. We had some kind of an aptitude test, a multi faceted aptitude test. (Newton, being progressive, did those things constantly.) And I had a mix of aptitudes but spatial perception was off the top. That was the one, you know, folding up things, where does that end up on them on the paper. If this is to this, then what, you know. So from then, you know, my mother was very pleased of course because I did very well in something. But from then on through teenage years, we talked about values. whenever I would bump into something she'd say "Uh, space perception".

[laughter]

00:36:08 PE: So much for that.

00:36:15 ED: And then we had a preference test which was supposed to begin, this is the same age, you know, to begin to orient us toward what we might want to do with our lives.

00:36:23 PE: And what age was this?

00:36:24 ED: Junior high, eighth grade. And I knew that I wanted to be... I don't know if I knew the word “engineer” but that was the direction, you know, the math, engineering, building stuff, making things direction. And I knew this was a preference test that say, you know, what I really wanted to be and I knew what I wanted to be so I tried to answer every question the way I thought an Engineer would answer. So it came out I should either be an accountant or a street car conductor.

[laughter]

00:37:02 PE: Uninspiring choices!

00:37:04 ED: Yes, yes. You know, my father Sid, being an accountant, thought this was wonderful! And he never understood why I never wanted to do accounting. I thought that was just the most boring thing anybody could ever do with their lives. Which began one time when I helped him in the office and he gave me a column of numbers to add. And I added up all the numbers on a little Victor ten-digit calculator, you'd pull the crank on and I got an answer. And he said "Great". And I'd add a piece and he'd give me another set of things. And I did that whole set of numbers and I showed it to him and he said "Good, you got the same answer, it checks". I said "You made me do that twice!" [laughter]. I was furious [laughter]. "I'm never going into that place, I suffered through it once and I got the answer, I'm not going to do it again" [laughter].

00:38:09 PE: No wonder computers inspire you. They solve the problem once and they get it right the first time.

00:38:14 ED: Well... Or they don't.

00:38:16 PE: Or they don't.

00:38:16 ED: But if they don't get it right the first time, they're not getting it right the second time either. Computers is a field for instant gratification. Or it was when I was in it. It was early, and you get some idea and it's new. And you're building blocks, you're just sitting right there, I mean you could put things together and you do it. Computers is not a field for patient people, obsessive, compulsive, but people who want instant gratification.

00:38:48 PE: That's interesting.

00:38:49 ED: It's a great field. It was perfect for me. I just loved it.

00:38:49 PE: Maybe we can start to talk about that now. And...

00:39:00 ED: Let me tell you a high school story.

00:39:01 PE: Okay. Please.

00:39:03 ED: When I got to high school, because you know, they put kids in advanced classes and other kids, you know, didn't. So you had these accelerated classes. So my English and verbal skills were not that great. And they didn't like to put kids in one advanced class, you've attended the you know, gifted program, you're supposed to be gifted across the board. So I didn't get into anything. And all of a sudden these smart math kids are gone and I'm in this class that's going backwards, and I'm going out of my mind. So I go to see the head of the math department, and I explained my predicament. And he says "I'll put you in there". This is Joe Mervindale, who wrote Mervindale and Walters, one of the key high school algebra books. Think his first name was Joe. Anyway, so he puts me in the class. I go into the class, the term is 4 weeks over, they're speaking a foreign language. I don't know what the hell they are talking about, I am totally lost. I go to his office and I'm crying 'cause I know it's death to go back to the other class and I don't know what's going on, I'm on the moon. So he says "Why don't you come to see me after school". So I go to see him after school for, I don't know three days, for an hour and he's saying questions like, you know, reading off word problems in algebra and it'll end you know, how old is Susie? So he says "So how old is Susie?". I said "I don't know". I had no idea. He says "I know.". "How old?". He says "X"

00:41:06 PE: [laughter]

00:41:07 ED: So I got the idea that X is, it's not the unknown. It's the answer! It's just that you don't know the number yet. And so now you're not doing algebra anymore, you're doing arithmetic, and equations. And I know that stuff. And inside of you know, a couple of hours I had the whole thing. It was funny. And then he said that he came in and gave a guest lecture in our class. He wasn't our teacher and he came and gave a guest when the teacher was out and then he came up to me afterwards and he said "When I was giving that lecture, I saw your eyes riveted on me. And I know this kid is going to amount to something". So you know, it's paying attention.

And throughout my teaching career I just loved it whenever a student would come to me to ask about the content of the course. I didn't care if they were bright and getting it, they were slow and confused. If they wanted to learn the course and wanted to talk about that I was very happy. I hated creating homework sets, I hated making exam questions and grading exams. And I particularly hated the students who would come with “the dog ate my homework, then I need two more points.” ... I never learned to deal with that. I met people who could deal with it very effectively, I never could. And that made me really hate certain years of teaching. Because it's a generational shift, back and forth, it comes and goes. The downtimes were really bad.

00:43:18 PE: Looks like that teacher really made a difference for you.

00:43:21 ED: Yeah, yeah.

00:43:26 PE: One of the things you said in your notes: you saw your first computer on a high school math class field trip?

00:43:32 ED: Yeah.

00:43:32 PE: You want to tell that story?

00:43:33 ED: Joe Ferguson, senior high school math class. We had a pecking order in that class. This was the advanced class senior year, we were doing beginning calculus. I thought it was great. And I was generally, we knew the pecking order. I was generally the fourth top in the class. It was H. Harris Funkenstein who always did perfectly in everything. He went to Princeton and became a psychiatrist. And he was very... And unfortunately he died in a swimming accident. There was Barry Sachs. Barry Sachs went to MIT and as a TA taught a calculus course in French [laughter]. They asked if he could and he said "Sure". Steve Weiner who went to MIT and then went to Lincoln Labs where his father had worked. And Steve and I were close friends. And then me, usually.

I remember one time we had a, one of these tests, you know where we joust with each other, regional thing, questions you know, who gets the top score. And as it happened, Harris Funkenstein was out of town, and Barry Sachs was sick and Steve Weiner won. And In the high school assembly when they handed out the recognitions for various tests, math being just one of them, they called Steve up to get his thing and he leaned over to the microphone, he leaned over to the microphone, they handed it to him, he said, "I would like to thank Harris Frankenstein and Barry Sachs for making this all possible". [laughter] We had fun. It was a great time. And when we went to, on a trip, to Harvard. We didn't see the Whirlwind, but we saw Harvard and MIT. We saw the Harvard Mark IV.

00:45:52 PE: Were you aware of these machines already? As you...

00:45:57 ED: No.

00:45:56 PE: Computers were just kind of in the air. Or you just first encountered them.

Interview recording part 1

00:45:59 ED: Yeah.

00:46:01 PE: And this would have been around 1954, 1955... 1956? Senior year?

00:46:06 ED: 1956, well it would have been fall of 1956 or spring of 1957. So the Mark 4 was an old computer. Harvard did have the Univac 1, as its major service computer, that was also an old computer. But the Mark IV is a historical piece.

00:46:26 PE: Was Mark IV electronic or was it still electro-mechanical?

00:46:29 ED: It was electro-mechanical.

00:46:31 PE: I didn't realize.

00:46:34 ED: That was my recollection. I'm confusing Mark 1 and Mark 4. I think..

00:46:44 PE: That’s why I'm asking because Mark 1 is the one that everyone remembers.

00:46:47 ED: I think one of them was programmed by turning dials in the front. It ended up useful for doing basically endless loops and it was used for calculating, I know, gunnery tables or something in world war II.

00:47:03 PE: That’s probably Mark 1.

00:47:03 ED: So it was probably Mark 1. And that's when Grace Hopper coined the word "bug" because it was not behaving properly. She said "she wants to take a look there must be a bug in it". There was a moth in one of the relay context. So I just saw this thing...

00:47:21 PE: Just a footnote there. The word "bug" had been in use for a long time before that.

00:47:26 ED: Really?

00:47:26 PE: It was always told that she had coined it, but what actually happened was that they found this bug in the machine and they pasted it into a lab notebook and it says "first actual bug found."

Undergraduate years at Harvard University (1957-61)

00:47:37 ED: Ah! Thank you, I like that. So, I don't recall what they showed us about the machine. I remember it being big and doing math. And you know I just said "wow," you know. “I wanna build those things for the rest of my life. I wanna make those. Because they do my work for me and I don't have to do the boring part. I can live on a higher plane.” And then making them, my God, that would be a thrill. So then I got to Harvard, they don't allow undergraduates to take computer courses. Actually I was, perversely, I was admitted to both Harvard and MIT.

Interview recording part 1

00:48:31 PE: And you went there directly from high school. So 1957?

00:48:32 ED: Yeah. And I said "Okay". Well, at MIT I think I know what the curriculum looks like. This is my strength. Whether I could have competed against the typical MIT undergrads in that area, I don't know. They are an awesome bunch. Harvard had more diversity in the curriculum, and I wanted to learn some of the rest of it. And we had a next-door neighbor who had gone to Harvard and he said, "The nice thing about Harvard is, no matter how weird you are, every oddball can find his own oddball group". So I said "Okay, this is for me.". And I went there. ...

00:49:30 PE: So you were a math major?

00:49:32 ED: Yeah, well, couldn't major in computers. You could major in a thing they called Engineering and Applied Physics, which included the computer faculty, although computer courses were only for the grad students. And I looked at the requirements and you could major in Math by talking four math courses, that's four year-long math courses. So one year, one term basically. And then you could do whatever you want. You know the General Ed requirements and you are free. So I could take the other stuff. I could take the physics and the engineering, which I did. But if you majored in Engineering and Applied Physics, you had no freedom at all. And why would I want to do it? Somebody else might want to do it. If I could choose to do whatever I wanted, you know I could choose to do what ever I wanted to do or something else. So I went that way.

Unfortunately my undergrad academic advisor was Gary Birkhoff, who wrote Birkhoff and MacLane's *Survey of Modern Algebra*, and was the son of George Birkhoff, a very very famous mathematician. And Gary looked at my record and steered me into the Differential Equations course. When I probably should have gone to the math course one before that. So I had a gap and again I had trouble catching up. Worse than that, he simultaneously steered me into the physics course that required differential equations as a prerequisite.

00:51:18 PE: Right. I had the same experience.

00:51:18 ED: Did you?

00:51:19 PE: Yes I did.

00:51:21 ED: It drove me crazy. It took me two years to recover from that. So my undergrad was basically going back and forth between probation and thesis.

[laughter]

00:51:33 ED: Partly because of that and partly because of my being a waterboy for the football team and never having any kind of... I went to two games in all of high school and suddenly a friend of mine said “let's do this.” And I did that for two years. And then I dropped out of that, but my roommate was a student conductor at the band and he got me in as a bass drummer because I could play the piano and the guitar, which was a great way to do junior senior year. But it was not good for grades with Saturday courses. I digress a lot.

Interview recording part 1

00:52:08 PE: That’s okay.

00:52:11 ED: Okay.

00:52:11 PE: Actually Dan Atkins, I was talking to him a couple of days ago and asked him about you and he mentioned the piano and the guitar which I assume comes from your mother in a way. She was a music teacher or something at that time?

00:52:24 ED: My mother was a piano teacher. And she didn't believe in teaching kids an instrument until third grade. Because she said. you know, it makes no difference. What she would teach before that is rhythm appreciation and feeling of the music and you know have an act out music and do rhythm games, and which note is higher and learn to sing. She couldn't do that with me because I knew the way you spend time with your mother is you take piano lessons. So she had to teach me in kindergarten and so she tried. And then she tried to teach me the discipline of trying to play in a more strict rhythm and introduced me to the piano, and I couldn't quite do it right. This is when I was practicing by myself with the metronome going and I grabbed the horn and threw it against the wall. Well, it made a little hole in the wall and damaged the wall. So we stopped music lessons for about another year and I got back to it. The guitar was college, high school and college. Everybody was doing that. That's the early 60s, late 50s. Acoustic. Yeah. It's all folk. It's the folk era.

00:53:45 PE: Okay. So you were in Harvard, the waterboy for the football team. Anything that led you into engineering from that point besides the math? Had you pretty much fixed on going further then, or what was your thinking about your career while you were an undergrad?

00:54:08 ED: Oh, I wanted to design computers. So I was biding my time, getting whatever education I could and playing around in the meantime until senior year. And senior year I took every computer course that they offered because they let undergrads in. And that was when I hit my stride. I was minimizing grids and pentode circuits and triode circuits, back into stuff which was by the Harvard Chart method which is, you know, which is not as good form as the Quine McCluskey Method. Well, Quine was a Harvard philosopher. He was deeply upset that people would use this logic to design circuits. And optimize them. He was after truth. He was doing truth-functional calculus. He was not doing pentode grid minimization. And McCluskey picked up these methods, somebody at Harvard picked up his methods and invented the Harvard Chart method. McCluskey picked it up and went to prime implicants, and well you know, and the mechanization of the Quine McCluskey method, I think which is McCluskey master's thesis at MIT, not PhD. So I took switching theory from Warren Siemen, who later went to Sperry's Univac Research Lab in Sudbury, Massachusetts, took numerical analysis from Robin Ash, which I didn't like, that was you know, again the messy calculus, numerical programming and it wasn't fun for some reason. But I did okay in it. Logic Minimization I just aced. Half the room full of graduate students and I just had a ball at that. It suited my personality. And then I took programming.

00:56:17 PE: I'm sorry. I was just gonna ask you that everything that you took by then was on digital, or oriented towards digital computers? Was anything left of the analog world at that point in your education?

Interview recording part 1

00:56:31 ED: That's interesting. There was, but not in the computer courses. Somewhere along the line I took a course in analog computing. I think it was taught by the Applied Physics and Engineering faculty. And I remember we had a textbook called "A Palimpsest in the Analog Computing Art". [PNE: this was probably H.M. Paynter's *A Palimpsest on the Electronic Analog Art* (1955), a collection of reprints.] And it was taught, basically you know, here are amplifiers, here is a feedback amplifier and here's how you hook them together. It wasn't computing. It was, what are the circuits of an analog computer and how can you do integration and differentiation. And addition was tough, integration and differentiation are natural. So I learned that. When I got to Illinois years later, I actually had to teach two weeks of analog computing and then in the logic course, which lasted until the year after I taught the course. And I said “we are not doing that any more.” [laughter]

00:57:43 PE: That's interesting in itself. When you guys experienced that, they were still using these machines and making them well into the 1960s, even probably after around 1970 ... small ones for scientific work of different sorts.

00:57:58 ED: Well, if you wanna do integration and differentiation, you can do it that way. I remember a trip to United Technologies. Probably mid 60s. I am not sure if we went to United Technologies or we went to some place where somebody from United Technologies was. It might have been a DOD conference center. No, he held up a hunk of metal. And he said, let me show you a flight computer. That was a block of metal with holes drilled through at different angles and the air would be streaming through it and it would make calculations and you have instruments. That's an analog computer. I was always fascinated with the idea that yeah... I have tried this on various people to see if anybody would pick it up. It was suggested to me as a problem but I don't know how to attack it. Hill climbing is a difficult problem in a digital computer. Or let's take it the inverse way. Hill descent; steepest descent down a hill. And it’s a complex calculation, difficult problem. But if you build a surface and put a marble at the top, pour water on it, it immediately knows the path of steepest descent. Why is that and why can't we do it? I don't have a clue. I don't know how to approach the problem and I am willing to accept it. And other people don't know how to approach the problem. But it infuriates me that I've never met other people in the field that thought it was an interesting problem. You've to at least grant that it's interesting.

01:00:13 PE: Sure.

01:00:13 ED: I mean, it's something very fundamental we should be able to do if a stupid marble can do it. And does it, presumably it does it in a vast parallel array. [laughter]

01:00:30 PE: The marbles?

01:00:30 ED: Well, as molecules subject to gravity. God is doing it. [chuckle]

01:00:43 PE: Let's see... just talk a little bit more about Harvard. A couple of things you mentioned in your notes were, you were emulating the Electrodata E101 on the Univac I.

01:00:56 ED: Yes. I was hoping you would go there. Programming course taught by Peter Calingaert. First machine we had was the Electrodata E101. Electrodata was later bought by Burroughs and the machine continued to be marketed as a Burroughs E101, they kept the E for Electrodata. I could tell you a little bit about the machine, which is quirky and interesting, to me [laughter]. It was a desktop computer. No, I'm sorry, it was a desk computer.

01:01:33 PE: Desk sized.

01:01:33 ED: Desk sized. Which made it small in those days. And the... It had a memory, it did not have core memory, it had a drum memory. So there's this rotating drum, we had an expanded version which had four sectors in it. Each sector had a 100 locations but 40 of them were common. So you had these 40 common ones plus 4 times 60, 240, so 280, did I get that right?

01:02:14 PE: Yeah.

01:02:14 ED: 280 addressable memory locations. Actually you can't address them, you can only address a 100. But there's an instruction that says switch sector, [laughter] from then on you're addressing a different sector which is why they had a common locations, which pass information from the sector to sector. Okay, to program, let me start with an instruction. An instruction was three pins in a plug board, in a row of a plugboard. One pin was an op code, you put it in the column of the op code that you want, the other two pins were normally were the two decimal digits of the 100 location memory addresses that you wanted to get to, which location in a 100. Or it could be a intermediate parameter I think, I don't recall. Some instructions didn't have the other two pins like halt [laughter]. So there are 8 plugboards and there were more on the shelf, you could unplug one and plug another one in and was somehow, whatever. So that's how you program the thing and you know, you could write software on it. Then we were in the Univac I and that had, that was before mylar tapes, that had steel magnetic tape...

01:03:41 PE: Nickel-based, I think.

01:03:42 ED: Was it? Okay. And we didn't have card punchers, we had a writer that you type on and it wrote on the tape. So you'd sit there typing away and the tape was written in 60 character blocks. So you sit there typing away and every time you finish 60 characters the tape, which was mounted on the top of the typewriter would go PHUNK and a block would be written. Now you can't correct that block, you could say delete previous block or something, I don't know. There's ways of making correction but they were painful. Typing mistakes was not a good thing, if you finish the block. Then you could carry this tape across the room and stick it on a tape drive and you...

01:04:31 PE: You could carry it across the room?

01:04:33 ED: Yeah you could carry it [laughter]. So its memory was mercury delay line, which mercury delay could hold an acoustic wave. So it's like taping a tube out of a roll of paper towels and yelling in one end and going down the other end and listening and hearing what you said and run back to the other end and yell the same thing back, recirculating there because if you let it dribble out the end it's gone. So the [delay line] memory is constantly doing that. And it had 1000 locations of storage. Well it was way more than 240, that was. And that machine with 1000 locations of storage was used for the, by all the TV networks to report the results of the 1960 election when Kennedy was elected. That you know, you could do things with 1000 locations. I don't even know what the backup memory was. That was the tape and you could go to the tape.

So you had all this world in tape drives and stuff. The arithmetic was in two copies, they were always checking each other, what they'd do when they had an error I don't know, maybe retry or something. But they did double redundancy, no voting because you need three to vote. So our final assignment was to write an emulator for the Electrodata E101 on the Univac I. And my lab partner was Bobby Friedman who was a mathematician, who got to take the junior math courses freshman year, 'cause he'd gone to the Walden school in New York. Brilliant, brilliant guy, unfortunately he flunked freshman English and some other course because he didn't like them. So he was asked to take a year off. We ended up in the same class at the end. So he really wanted to emulate the machine. And one of the things that we did was that when the Electrodata E101 hit a halt instruction, it always displayed the plugboard and instruction number on its lights, the E101 did. And when it halted it displayed the location of the halt instruction.

01:07:07 PE: Set of bulbs, single bulb for each, each hole?

01:07:11 ED: Yeah, yeah. The Univac I when it halted displayed the program counter. So Bobby, being more obsessive than I, demanded that we figure out a way, that when the E101 program halted, the program counter would have the value of the plugboard and step number [laughter] of the E101 displayed as a program counter in the Univac I. So to debug this thing we had 3 hands-on sessions on the Univac I. We got to sit at the operator's console.

01:07:56 PE: Was that the first time?

01:07:57 ED: Yeah. And you could run for half an hour, twice, and the third time was the demo, to faculty, to Peter. So we got it working we thought and in the demo you've to run a program you've never seen before. So, Calingaert does not know how long people are gonna take, so you got your nominal time slot, and he says tell me where you're gonna be and I'll give you a phone call when it's time to come over, call you half an hour ahead. And so he calls Bobby and says "Hello this is Peter and it's time to come over" and Bobby says "Yeah Davidson cut it out" [laughter] He's so ignorant, he didn't know if it's really, it took a while to convince him that we really should go over there. But it worked and you know that's a great feeling. It's just a great feeling and it leaves a mark for life. That's victory. Much much better than what I had done the previous couple of summers, which was sitting at Edgerton, Germeshausen, and Grier, Inc., doing inverse convolution on a freaking desk calculator for Constance Franklin, who was the wife of Philip Franklin, head of the math department, and the sister of Norbert Wiener. I spent two summers with Connie, she asked me to call her, she was ancient at the time, well over 50. And I was a kid, so it's hard to call her Connie. But I sat in her office, in her cubicle. I was her slave and she would think great thoughts and I would be doing whatever calculations she handed me. In the parlance of a slightly earlier era I was her computer, because computers were people.

01:10:02 PE: Weiner himself was a computer earlier in his life. Did he ever come by?

01:10:08 ED: No I never actually saw him. Joe Jacobson, my Michigan roommate, when we worked at Honeywell together later, actually saw... We used to, when we were at Honeywell we were allowed to take courses at MIT which Honeywell would pay double tuition for. And so we used to do that, it was great. Three years at Honeywell, I only totaled one car getting to class [laughter]. But one day a true Norbert Wiener story. Jacobson was walking across the campus and actually saw Wiener and in a burst of enthusiasm walked up to him, realizing as usual that, you know, he said "Hello Dr. Wiener" and Wiener looked at him, you know, like “who the hell are you?” [laughter]. And Joe realized he didn't have anything to say and he just stood there with his mouth open and Wiener said "What do you do?". He had pretty, you know, sweet guy. Jo said "Oh, I work at Honeywell designing computers" and Wiener looked at him and said "The future is analog." And he walked away.

01:11:29 PE: Really?

01:11:29 ED: Yeah. All Honeywell stopped for two weeks, debating about what he might have meant.

01:11:34 PE: [laughter] seriously?

01:11:35 ED: Yeah, yeah. I don't know what he meant.

01:11:40 PE: The future is analog.

01:11:41 ED: But as you, as you push digital to its limits, it becomes analog. If you don't deal with the analog effects you can't make the progress. Was that what he meant? Heck if I know.

01:12:01 PE: Fascinating guy, Wiener. One more thing you were gonna talk about, at Harvard, was a story about Ken Iverson, that I think I saw you give in the talks, but tell it again.

01:12:15 ED: Well, I really wanted to begin to work in computers. And I think this was the summer after my senior year. Yeah, must have been because I'd had some computer courses. So during the senior year, I heard that Ken Iverson was developing this interesting programming language called APL, a highly symbolic language. The joke about APL, it had symbols for matrix multiplication those are operators and things like that. The joke about APL is that it was so constructed that it was virtually impossible to make a syntax error, anything you wrote would compute something [laughter]. So you didn't get your normal bug indication, you know, this is wrong [laughter] hey, it's got something, you've to figure it out. But you know, then he had the APL keyboard with all the characters he needed to program in APL and it was mathematical which I liked and there was programming which I liked and so on. And he was in a research project developing this stuff, so I figured what the hell. So I just went to his office unannounced, didn't take an appointment, I didn't know you were supposed to, so knocked on the door, he was sitting at his desk. He looked up, he said "Yes?"

And I introduced myself and it was a big desk, and you know and I said "I've been studying computers and I really like them and I wonder if I might work for you this summer. Is there a possibility?". I didn't know if he had people working for him or not. He doesn't invite me to sit down, he doesn't tell me to go away. He looks at me for awhile and says "What makes you think you're interested in computers?". And nobody had ever asked me that before and I didn't have a clue. I mean, the kind of questions you ask are all very hard to deal with, it's not... I'm not, I'm introspective in some ways but not in that way, you know, question what I like, I know what I like. I just don't know why [laughter]. So nobody had asked me and I didn't have an answer and I thought and I thought and I said "Well, to me there's something really beautiful about having a few basic building blocks and being able to construct something arbitrarily complex out of them". And I paused and took a breath and he sat there staring at me for what seemed like an eternity, and I didn't have anything more to say. And so I stared back at him and finally I said "Well thank you for your consideration" and I left. And that was the end of the conversation.

01:15:32 PE: He didn't say anything?

01:15:33 ED: Never said a word, never said a word. He just stared at me. I don't know whether, you know, whether he didn't have funding or he found my answer uninteresting or he is a shy mathematician and doesn't know what to say, or whether he is thinking "God, how can I get this kid out of my office?"

01:15:53 PE: Perhaps he was thinking "That was quite a profound answer and I'm not sure how to digest it."

01:15:58 ED: I hope he thought that. [laughter] So I never did find out. I never met him again, and that was the end of that. Strange experiences. So I ended up... What did I do? Yeah, I guess that's when I went back for a second year at EG&G. Working with Connie and getting to know her was great and just a view. She told a Norbert Wiener story, which was basically that growing up as a sister of Norbert Wiener had its downside and she didn’t get attention from anybody. She's a lovely, lovely person. But she said the beginning of when the family realized that something was unusual was when Norbert sat on the floor at the age of, at two or three, this is well before nursery school age, and he is sitting there with a book up in front of him and he's turning the pages. Maybe not two, but it's well before any formal education whatsoever. So he's turning the pages, he looks at the left page and the right page and he turns the page and he looks at the left page and the right page and he turns the page and his father turns to his mother and says "Isn't it cute? Norbert's pretending he's reading a book. Isn't that sweet?" And then somebody asked him a question about what was on the page, and he answered it.

[laughter]

01:17:45 ED: Oh my God! So, he was privately schooled from then on. Connie raised a family. She had been working as a professional, raised a family and her husband suggested that she might enjoy going back to work. She'd been away from it for decades. And she didn't know if she could do professional quality mathematics anymore. He said "Well, you're never gonna know until you try". And she went on the Atomic Clock Project and to this day I've no idea what an atomic clock really is but she worked on it and did some of the fundamental work on the atomic clock and then went to EG&G. And she said, couple of things she learned in her professional life is that first of all you never admit, if you're a woman and you work in the field, you never admit that you can type. Because you'll be a typist. And it’s good to say that you know about computers, because you might get an office near the computer room and the computer room is air conditioned. [chuckle]. And the draftsmen try to get in a room where they're not sweating all over the drawings, these were the old days. But that’s part of why she... I mean she could have done inverse convolution, but she had better things to do with her time. But it’s part of why she said no to doing some of these calculations. Because you had a fight to affirm your professional status, if you're a woman in one of these companies. You need a proof of it. Did you know technically what you were doing?

01:19:48 PE: Sure.

01:19:48 ED: I didn't have a clearance at that time because it takes time to get a clearance. And most people in EG&G who were working...

01:19:54 PE: What’s EG&G?

01:19:55 ED: Edgerton, Germeshausen, and Grier, Inc.

01:19:59 PE: You said that earlier. I just needed the acronym.

01:20:01 ED: They were in Boston and Las Vegas and Utah. My understanding was Grier was in addition to his mathematical skills, he was also a gambler. So he had a Las Vegas office. But he was also near some of the...

01:20:14 PE: Office.

01:20:14 ED: Yeah. The Atomic Energy Commission sites and we talked, it would collect bomb data and analyze it. And they made xenon flash tubes which was used for, I don't know, pre LED kind of high intensity LEDs. And Edgerton of course was the speed photographer with the strobes. Doc, you called him. I think Germeshausen was the business guy. Made it all run. So everybody who knew anything had Q clearance, which was the Atomic Energy clearance, and it would take more than a summer for me to get it, so I didn't have any clearance. So I wasn't allowed to know what I was working on. But I knew if you do convolution you have an input and transfer function and an output, and he was there measuring the output, and the input was called the source. And that'd you have a few nanoseconds of data before there isn't any data anymore. And that meant somebody would say something like “shock wave.” [laughter]. That was a pretty good idea.

01:21:27 PE: Did that work?

01:21:27 ED: Yeah. He got a detector and there's a shock wave going by that destroys the detector and you got some wild idea what the transfer function of the detector might be and you're measuring the output and you're trying to inverse convolute to recover the input. And the input looks something like a very sharp rise. [laughter]. So the data to say the least was a little bit, imprecise. And the transfer function was imprecise and the quality control on these devices were less perfectly in range. And if you test them, you lose them. So you don't test them again. That was like a cartoon I saw, Charles de Gaulle carrying the atomic bomb on his back into the Sahara desert with the caption that "If we test it, then we won’t have it anymore." Computer problem in many cases. So we used to have a joke: the straight line is a curve that you can pass through any four data planes. Unless they happen to be at the vertices of a rectangle, in which case you don't know which way to slant the straight line. [chuckle]. But we did our jobs.

01:23:02 PE: Is there anything else you want to say about the Harvard period? Your undergraduate years.

01:23:09 ED: Well. I had an English professor one that taught freshman English, assigned a theme for us to write. And I wrote it in class, he assigned in class, I wrote it in class, it was about Memorial Church. And I talked about feeling isolated especially with the Church in the campus. But it was pronounced to be a Christian church which is not the campus. And felt really sad every time I heard church bells ringing and so on. The next class he read my paper to the class as an example of how not to write a theme. And the class had a great time I'm sure, deep red on my face, they all knew where it came from. It was low-level political muckraking, very very amateurish propaganda. And unsuccessful. I stayed in my seat as long as I could, didn't leave the room with anybody. And as I was straggling out of the room he looked at me and said, "I think you have something. And I would really like you to take my short story course." I said "Wow! Where did that come from?" I had just been totally humiliated. But I took his short story class. I didn't do very well. Halfway through his class, he said what's going on with you, what do you like"? I never read literature. So he said "Who's your favorite short story writer?". Oh my God, I never read any short stories. So I said Guy de Maupassant Which was as close, I had read two of his when I was sick one time because they were short. He said "Oh!". [laughter]. He said you should read John Dunn who is, you know poetry, not short stories, but he chooses his words very carefully. And you'll learn choosing words, why he did it. And he gave me some other suggestions which I forgot. And then he said "What are you majoring in?". And I said math. His face just fell. "Oh!". And it was dead, it was over.

01:25:49 PE: So he lost you...

01:25:51 ED: Lost me. I'm gonna waste my life on something horrible and I'm being evil, gone, dead. So that was the end of that. But I like words and know how to work with them, I just spent a lot of time with graduate students on writing. And expressing ideas and how do you get that across and bang it home. It's particularly important because over half of my PhD students were foreign born.

01:26:25 PE: Really.

01:26:25 ED: Yeah. I counted them last night. Out of 49, 26 were foreign born. And a number of them came back afterwards and said that the writing put them in a unique position. They would get into some design group in Intel or one of the other big companies. And they were able to write stuff for the group. Same thing happened with the kind of modeling. He would be asked to figure a model and try to construct an argument about whether the proposal was any good. So there was a niche thing there. That my students tended to share.

Interview recording part 3

University of Michigan, MS in Communication Sciences (1961-62)

00:00:04 PE: Okay this is part 2 of the interview with Ed Davidson. And we are about to talk about Michigan where you went right after Harvard from 1961 to 1962, and got an MS in Communication Sciences. How did you come to choose that program and when did you choose it?

00:00:29 ED: Well my senior year at Harvard I knew one year of computer courses wasn't enough to [chuckle] go be a professional and I really wanted to take more and there weren't very many places with computer degrees. Michigan was a leading university at that time in its computing center and MAD, Michigan Algorithm Decompiler, I think they called it, and a big step up from [chuckle] that UNIVAC I campus computer to the IBM, I think it was a 709 at that point. And had a bunch of interesting people. And I honestly don't recall how I found out about the Michigan program but there weren't a lot of programs to choose from. And..

00:01:36 PE: What else you mentioned is that you never got a degree from a computer science department. Because they were still in formation in the first part of the 60's, mostly didn't form until the mid 60's.

00:01:50 ED: Right. So Harvard had a nucleus of computing faculty because they were building the Mark 1 or Mark 4's and they had gathered some people to do that. And Michigan really had one right from the start, it's gone in spades now, but you know continued that tradition, but right from the start a really strong emphasis in the computing services on campus. Academically you know I gradually learned over years how that program formed, but they really were the vanguard. It really was. I don't think I realized that when I came. I don't remember what were the places I applied to. But I know one big advantage of Michigan was that they accepted me. [laughter] And since I had a kind of a checkered career environment, that was nice.

00:02:46 PE: I see. Yeah.

00:02:48 ED: Now I came to get a PhD. It was a PhD program. And I came intending to get a PhD. But decided to bail at the end of masters, for various reasons due to the nature of the program.

00:03:11 PE: And they had decided to call it Communication Sciences, after a debate about whether to include the word Computer in the title of program.

00:03:20 ED: Yeah, I didn't know that at the time. It was called the Program in Communication Sciences; it was not yet a department. And we graduate students in the program had this story which basically has been substantiated. One day Art Burks and Gordon Peterson met in a supermarket someplace, both doing family shopping, and started talking and were commiserating that each of them had a student that didn't seem to satisfy the degree requirements in their home department. Art Burks' student was John Holland, who was into parallel computing and interesting models of parallel computing at that time. And I don't know who Gordon Peterson's student was but I guess he didn't fit with Linguistics in a classical way. So they said, why don't we form a new program and you know, we'll get PhD's for the students who are doing legitimate and creative work, who just weren't satisfying the classical degree requirements. So they formed Communication Sciences. They brought in so many disciplines that had to with the transfer of information. So that included Electrical Engineering, it included Philosophy, which was...

00:05:05 PE: Burks was there...

00:05:06 ED: Burks' home department because philosophers could do logic. And he did logic and he was a principal designer of the ENIAC computer with von Neumann and Goldstine. But he was in the Philosophy Department. Peterson was in linguistics. He had come from Bell Labs were he was, at least the co-inventor, if not the inventor of the Vocoder, which was a device that would speak for people who couldn't talk, touch with the throaty voice. I think that's what it was. So he brought in some classical linguistics. So he went into phonemes and international phonetic alphabets. They brought in Manus from Biology or Genetics. I don't know, we had Biology and Genetics. Biology but mostly Genetics. It was the genetic code.

00:06:14 PE: It was kind of early days for the genetic code as an object of study.

00:06:19 ED: It was really a pioneering program. They had pharmacology in there, because we studied biological systems and had drugs cheat receptors, by getting something to be accepted as something that is not, and you know, interacting with disease that way. Hank Swain from Pharmacology. We did atomic theory. I'm trying think if there was anything else in there. Yeah, you name it, it was in there. We did McCulloch and Pitts theory, we did the sodium pump and the stuff like that. We read von Neumann papers on the computer and the brain. It didn't impress me at that time, but it has impressed me since, now that I've grown up a little more. Yeah, it was interesting things. The brain thing was that if you look at the mass of the brain, almost all of it is structural stuff that keeps the brain from being floppy. So talk about it as a computer, that's like cabinetry and physical structure for stuff to move through. And if you take that away, most of what's left is the axons that carry the signals from one place to another. So in a computer, that's like the wire in a electrical interconnect. And a very tiny percentage of brain is the neuron cell body that is what we might think of as doing the computation or storing the memory of the brain.

And so why do we think we can build computers with less wire? If God can't do it, how can we do it? Communication, which I didn't appreciate at that time, but communication is the essence. The logical operations, that was our obsession in all our computer courses and what we thought we were doing professionally, is not the big problem. The big problem is how do you move the information to where you need it, and get it in time so that all the stuff that you paid for doesn't sit around waiting and you can do some useful work.

00:09:01 PE: That's one thing Cray got into when he began building supercomputers, his machine was reducing the distance between the processor and the signal that travels as far...

00:09:08 ED: Oh boy, did he ever! And tuning the distances. So he built the Cray I... What? [chuckle] He built the Cray I with three different chips, the packages. Small scale integrations. One package was the memory chip. I think it might have had 1K bits of memory, but I am not sure. Another chip was, I think it was a 16 bit register. The third chip was two NAND gates. And because the number of pins in the package is usually even, somehow, and because of the power pins that he needed was something he ended up with the... And because he wouldn't waste the pins, because he's a very efficient guy, he ended up with a 5 input NAND gate and a 4 input NAND gate in the package. Now the problem is that a 5 input NAND gate is marginally slower than a 4 input NAND gate. So on the 4 input NAND gate on the way from the output of the gate to the output pin of the package, the wire takes a circuitous path to make up for the difference in time.

Because if you're designing a very fast computer, it’s not as important as you think how fast the components are. What matters is the predictability of their speeds whether they finish with simultaneity, so that you can get a wavefront of information moving through. And if there is no skew in the wavefront so that it moves at a steady rate in a narrow window of physical components, you can spot another wave from behind it very quickly. But if it’s dispersed all over the place you’ve got to wait for turnaround time, and that kills you, the performance is dead. That's amazing. And he did this stuff, he did these designs pretty much by hand. The big joke was that Apple Computer they bought a Cray computer to do their computer-aided design. And at Cray they finally convinced Cray not to write stuff on pads of paper, and he bought an Apple Computer to write down the logic they wanted in his Cray. [laughter]. But it’s all, you know, it’s space and distance and communication. It’s something that I learned in from Michigan without realizing it; without absorbing, without capitalizing on it. And came back to much later in life. Not that I contributed to it, well, I did in some ways. But I was in some ways very much in the logic design thing. And to me all these other things, you know, learning there are no clicks in Japanese (which was surprising for them to say because the fact that there are no clicks in English could have been a much simpler example), but I can do a click, I don't know how you are going to type that but...

00:12:40 PE: Bang! [PNE: ED is talking about the “click” sounds that exist in some languages, notably in southern Africa.]

00:12:41 ED: And I can do all the tongue positions and stuff and I could do this sound today, it’s roughly "aargh". And so these things were a joke to me. And von Neumann's book on the computer and the brain was a joke to me, and I traded it for a record of songs and ballads that I played on my guitar.

[laughter]

00:13:08 ED: It was not what I wanted. The program wasn't going where I wanted. I wanted to design computers. I was being very limited and quite immature, intellectually. My roommate John Jacobson was... And he was just in his element, his program was wonderful. And people would... He started at Tulane and went to University of Cincinnati, or maybe the other direction, and Michigan. And so as an undergraduate he was in a number of universities and majored in almost anything you could name at one time or another. He was a genuine intellectual. He could read. But he majored in everything... Including poker at Tulane. His flaw was that he couldn't pass the foreign language exam, 'cause he found it so boring and couldn't bear to go to class. He didn't seem to have the facility... But his English was phenomenal; he just couldn't do it. So when people asked him what was his undergraduate major, he'd say "Well, I guess to have to say first year French, because I took that more often than anything else".

Interview recording part 3

[laughter]

00:14:37 ED: You know but you get into the biology and stuff, he worked in Swain's lab, and the Biology stuff and he just... I never did figure out what the heck it was about. And there was another guy in there who when they'd explain some calculation on a matrix of variables, he would raise his hand, because it was good to ask questions, the instructor would pay attention to you and his question always was "Well, can you do that in three dimensions?" and the answer always was "Yes". So, our nickname for that student, we called him "The Great Subscripter". So, he would always add one more subscript, that was his knowledge. We were all, I think most of us, we were a pretty bewildered bunch.

There were some geniuses, Holland got his PhD, was the only one at that time to have a PhD. There were a lot of students piled up trying to get a thesis topic approved. Because everybody on the Committee insisted that the thesis have something to do with their discipline. Who the heck can integrate all that stuff? The faculty didn't integrate it, the students were having to integrate it. So you had guys like Mike Harrison and ... Jim Facker, and a bunch of really, really brilliant people just jammed up, not getting approved. And I looked at that and I bailed. And when I said that I wanted to leave, Gordon Peterson asked me to come to his office to see him and he gave me another one of a series of lectures that I had heard about how I was running my life. And that, you know, it wasn't designed as a Master's program, it was designed as a PhD program and I didn't have an education worth having at this point and what the heck was I going to do with my life, if I didn't stay for the PhD. I was doing okay academically, but I was living like a bum, living kind of like a beatnik .... My roommate Joel's mother used to come up from Toledo and steal my dirty laundry and wash it. I didn't even realize she was doing it. Strange life.

But I did my homework and I got great grades and I watched every movie that played in town at the three movie theaters. An odd life. So, what was I going to do? You know, I don't know if he was genuinely concerned for me or if he was really concerned for the reputation of Michigan with me out on the street claiming to have a degree from this place in Communication Sciences. But my mind was made up. By the time Joel decided to leave a semester later, Peterson had a different speech for him. I am making Peterson sound like a bad guy, he wasn't, he wasn't. He was a very decent person. But Peterson's offer to Joel was that he could get a Master's degree on condition that he never return.

[laughter]

00:17:56 ED: Or, he could take a leave of absence. So, I think he picked a leave of absence, which is why he actually never did get a degree from Michigan, which is too bad. He could get his undergraduate degree by an act of the Regents, which conferred on him a degree without passing the foreign language department.

[laughter]

00:18:20 ED: Strange. But brilliant and inspiring. And the summer institutes, which I was too immature to take advantage of, were wonderful. Everybody, everybody who was anybody in computers, came to Michigan during those years to do summer institutes, teach them. And the smart students were also, who really wanted computers were getting jobs and were working at the computer center, which I didn't have the maturity or sense to do. Talk about the way my mother influenced me. She would say to me, the worst thing that a Jewish mother can say to her child, she would look me in the face and wave her finger in my face and say "You have no survival instinct". It was true. She'd be amazed to know that I am still alive now.

[laughter]

00:19:18 ED: She would also be amazed to know that Michigan eventually invited me back as head of the Department. My mortal fear was that somebody would recognize me from my student days.

00:19:27 PE: [laughter]. What about Burks? He was still around?

00:19:31 ED: Yeah, well, you know that's funny. Burks, I took Burks' logic course and one day... He didn't remember me at all, which was a relief. But he lectured about the ENIAC. We learned a lot about the ENIAC and we learned a lot about von Neumann. Because Burks edited the Burks, Goldstine, and von Neumann papers. He put that onto history. He was willed the unfinished papers of von Neumann and he edited them and he add his own contents on. A rather dry person to students at that time. I thought he was remote and sardonic. I have learned later that he was a bit shy. So, one student asked him one day, you know, what, since we kept learning about, NOR gates I think they were all the time. All the examples were NOR gates, whatever, it is either NOR or NAND, whatever they were using in the ENIAC. And so one student said "Yeah, what kind of gates do they use in computers these days?". And he didn't have a clue.

So we wrote him off. In fact, we shouldn't have. I mean, why should he know the parts catalog of the week? It is not what he was teaching. He was teaching fundamentals. Something that I clung to later in life. I refused, you know, not being a reader, I refused to memorize the parts catalog of the week. I could not tell you product for product what the hell was going on that week in the field, it was moving too fast. I tried to do it one level more abstract, and do the fundamentals, which don't change as rapidly. But when the technology changes, you just change the parameters and you are in business with the new technology. What the heck. You are just in a different place in the space, that's all.

00:21:26 ED: But Burks one day tossed out this problem, we were talking about different connectives as the logicians are wont to do: you had a implication, if A then B. I had AND, NOT, OR, NAND, NOR, you know, equivalence, not equivalent. You know, all these logical operators on boolean variables. And he tossed out that if you are given a random set of operators, it is very difficult to tell whether they are logically complete. But you can show that NAND is logically complete by itself, because you can make AND or NOT out of it and Boole long ago showed AND or NOT was complete. So, now you get that one, you know. Well, NOR is too, obviously is dual or you could read the proof that dual things are in it all over the place. And implication is an interesting one. Because it is the A B Bar function, I think that's right or A Bar... Yeah A and... If A implies B then if A then... Well, figure it out. I don't know. One of them is complemented and one of them isn't and it is either an OR or NAND. Basically you can make it complete with a, if you have the constant false, you need the zero. You don't need an inverter.

Interview recording part 3

So, if you have a zero, which you always do, because if one is, we used to have a saying at Honeywell: "What is truth?" and the answer is "Truth if five volts". So, if you have power supplies and electric circuit, obviously you have one and zero, one is high voltage and one is low voltage. But then in an arbitrary collection with arbitrarily many inputs into the function, how could you do it? [PNE - several sentences omitted here, where ED became briefly confused about the chronology]

00:24:22 ED: I didn't know about branch and bound methods. But I constructed one. And I obsessed about it and by the next morning, I had an exhaustive test for whether an arbitrary set of connectives, logical operators, was complete. It was exhaustive and it enumerated all the possibilities and I constructed it in a chart and you know, you apply them and you get down to one level functions that you can get and then you start combining those functions with the operator again and you get all the two-level functions, and as soon as you get AND or NAND to pop out, you are done. But at a given level, any function that you have already seen, is uninteresting. Because you have already got that one. So, very soon, you get to a level where there are no new functions, and that happens very quickly. Because this is a very simple task actually. And as soon as you don't get any new functions, nothing new is ever going to happen no matter how far you go. And so in a matter of minutes you can tell if a set of operators is logically complete. So, he had said it was a hard problem. So I was very happy and I wrote it up and I brought it to him and I said "I solved the problem that you mentioned in class yesterday" and he said "What problem?"

[laughter]

00:25:56 ED: And I described the problem and he didn't recall stating the problem exactly that way. Or not that he recalled how he might have stated it, or ever stating anything like it. And I said "Well, you have stated the problem in this write up" and the writeup was like three pages long, because that is all it took. I started holding it out to him and he sort of half reached for it, because I was reaching it toward him, and as he did that, he said "What would you want me to do with it". And I said "Well, would you be willing to read it and tell me what you think?" And he sort of slightly grimaced and slightly closed his fingers and slightly didn't, and I slipped my hand back, but the paper unfortunately stuck more to my hand than his and the staple went into his finger and he started bleeding. And I am actually not sure who ended up with the paper.

But eventually I submitted it to IEEE, or it might have been ASEE at the time, journal and they had absolutely no interest in it. So that was the end of that. You know, these little attempts, you know, it's life. I am still proud of having done that, you know. I didn't know it wasn't an important problem and I solved it, it was nice. I had a similar experience in calculus one time, where a professor asked us to solve an integral. I always hated integral calculus, for some reason I liked differential. I hated differential equations, these things where you memorize a million techniques. I have no memory. I have memory for storage, not for what formula you apply to what thing. I never liked trigonometry except for the unit circle, I liked that. Brute force solving equations; there has to be some logic to it, not cases. So he gives us this horrible integral homework and I thought "oh, you know, maybe you can do integral by parts." So I do integral by parts and one part works and the other part didn't. The part that didn't looks like you can maybe do it by integral parts. Pretty soon, it is two o'clock in the morning and I work for eight hours on this damn problem and I am not going to quit because it is not, it is making progress, it is transforming itself step by step, it is not repeating itself. So I am not in an infinite loop. And I have invested all this time. By seven o'clock in the morning, I had an answer. I was really proud of myself, because I had, you know, I was a little over my head in calculus at the time and I was struggling. So here is a problem I felt was hard and I solved it. So I turned it in. It was, I don't know, a twelve page proof of that integral. As usual in those days, you don't get your homework back very quickly, so it was months, it seemed like months, it was definitely weeks. Eventually, I got that homework back. And there was a short comment on that proof that said, "Wasn't it obvious to you that I had assigned the wrong problem by mistake?".

00:29:38 PE: And that is all it said?

00:29:39 ED: That is all it said.

00:29:42 PE: So he didn't even read the proof?

00:29:44 ED: He certainly did not read the proof. Did he look at the answer? I don't know if he looked at the answer. But the problem was messed up in the book. You know, it was not supposed to work that way, and a rational and intelligent person should figure that out. You learn these lessons, but they are painful.

00:30:06 PE: Yeah.

00:30:08 ED: Could have been kinder.

00:30:10 PE: Yeah. Yeah, clearly both of them, those..

00:30:14 ED: But it is memorable. You know, it is memorable. Herb Schorr, can I tell the Herb Schorr story? You know, these things stick..... He was a, you know, brilliant guy. Princeton PhD, had been a graduate student of Ed McCluskey's. Let's see, I was visiting IBM one day, he was director of the lab. And I had some friends there and we were walking down the corridor towards the lunch room and Schorr was twenty feet in front of us down the corridor and McCluskey used to joke that Schorr wrote the longest PhD thesis that's ever been written, which was like 500, 700 pages long. And I knew McCluskey was there at the time, since after I had been at Stanford and that McCluskey's crew needed a Lab Director. And his theory of PhD supervision is that you keep... You don't let your smartest students ever finish. You keep giving them more work to do, because that is how you maximize output and you get the best stuff done. And if you decide that your student is really not any good, you quickly give him a PhD and get him out of the door, so that you don't have to waste any more time with him. (I am exaggerating.)

So he used to say "Herb Schorr wrote the longest PhD piece in history that anybody wrote", so I am going down the hall, the person I am with says "There’s Herb Schorr. Do you want to meet Herb Schorr?" I said "Oh, you know, he is busy talking to somebody. He is going to lunch. He has got business to do. He is a Lab Director. He doesn't know me for that matter, well, you know. He doesn't need that interruption instead." And my host said "Oh, let me introduce you". So he goes up to Herb. "Here's Ed Davidson, he is visiting from Illinois".

Interview recording part 3

And Herb turns around and says "How do you do?" and he turns around and he is walking off with his friend. You know, "How do you do" and gone, you know.

However, I had already energized my mouth and so, to his back as he is walking down the hall, I say "I heard that you wrote the longest PhD piece that's ever been written". Herb stops, dead in his tracks -- this has got to be fifteen years after his PhD -- stops dead in his tracks, turns around, faces me, red in the face and says "Do you know what that son of a bitch McCluskey did to me?". All these years later, he is a Director of IBM Research Labs and he is still mad.

[laughter]

00:33:31 ED: I don't know.

Honeywell Computer, logic design engineering (1962-1965)

00:33:34 PE: So, let's go from Michigan to Honeywell. How did you get there? You were a Logic design engineer ?

00:33:42 ED: Yeah, well.

00:33:44 PE: EDT for 1962 to 1965?

00:33:47 ED: Yeah, three years. Three years, my goodness! It is a short three years. And a long three years. So, I left Michigan. Students wondered, I said I was going to leave and going to get a job and work in computers, since no one would teach me to design them and I would just go design them. And people looked at my physical state and said "What makes you think you can get a job?" and I said "Oh, it is not that difficult. You know, wash my clothes, take a shower, go apply for a job. It is not that I don't know this stuff." So I interviewed around. Honeywell was in Boston and I was going back home. It was my first... Michigan, you know, my first year away from home and that was hard. I've never been on my own before really, so.

00:34:32 PE: So you were living at home when you went to Harvard?

00:34:34 ED: I lived at home the first year. Actually, Harvard admitted me conditionally, which was doubly why I wanted to go there. I was admitted on the condition that I not apply to live in a freshman dorm. What was happening, was Harvard was renovating the freshman dorms, and so they didn't have enough freshman dorm space. So they let in more locals than they normally would have to fill the freshman class and made us live at home. So, I was one of those who was not allowed to soil the elite in the yard, another sense of isolation. But, that was okay. And then, sophomore year, my parents agreed that I could move in, but only for one year. Of course, I never moved back. But it was close enough to take my laundry home and get the car for the weekend.

00:35:26 PE: Right, so Michigan is your first year away from home and... We were talking about going to Honeywell. How did you end up there?

Interview recording part 3

00:35:36 ED: Oh, yeah. So I wanted to go back near home and Honeywell was a major computer company near home. So I applied there. And they were hiring. And the personnel guy says "Well, we are starting a new project and we need logic designers." And he sort of paused and said, "Are you willing to do logic design?" because a lot of people thought it was beneath them. And I said "I would like to do logic design for the rest of my life."

[laughter]

00:36:14 ED: And he said "You're hired". And I was in heaven. I really was. Man, this is my life objective, I did it right and it worked and there I am. And they were starting a new thing, which was the Honeywell 200. And I was one of the designers of the Honeywell 200. I could talk about that or I could do a little corporate history leading up to that.

00:36:42 PE: Let's hear the history first and then hear the rest.

00:36:45 ED: Okay. So, Honeywell was originally... Honeywell Computers -- Honeywell is a thermostat company, let's make no mistake about it -- founded by a man with a very appropriate name of Swett. I think spelled S W E T T, if I am not mistaken, but I don't really know. So they thought like a thermostat company, stodgy old company at the top. But they were getting into control systems and so they had Honeywell Controls. And control systems were getting into computers and they thought they ought to get into computers. But they didn't know anything about computers. So they went together with Raytheon and they formed Datamatic Corporation.

00:37:31 PE: Oh yes!

00:37:33 ED: And their first computer was the Datamatic D1000, which was progressive and not progressive. It was the last commercial vacuum tube machine ever built. And it was wired with printed circuits. So, it was vacuum tubes on printed circuits. Printed circuits were an advanced technology.

00:38:06 PE: Yeah. It is considered a rare combination, so.

00:38:07 ED: So, basically, from the experience, which I can go into a little bit, Honeywell learned two things. One was that vacuum tubes are obsolete and the transistor things really works. And the other was that printed circuits are not a reliable technology. Because they were ahead of the curve on that. So their next machine was a transistor machine discretely wired without printed circuits. Bad mistake again. D1000 filled three rooms. One room had mag tapes, one room had the operators console and I don't know what the third one was, CPU or memory or something else. I mean, it was a huge machine. The tapes were mylar tapes with a vacuum chamber tape drive, Honeywell invention. Brilliant invention, and a great technological advance. Because the vacuum chamber tape drive didn't allow you to do a floating head, so that the tape did not come into contact with the read head. And it allowed you to go rapidly back and forth. Whereas IBM, for example, had a contact read head, and I'm getting a little ahead of myself, but at IBM if you had read the same block of information forward and reverse a few times, it would scrape all the mylar off the tape. And Honeywell could do it forever. So, it's 3 inch wide mylar, huge, heavy tapes.

Interview recording part 3

00:38:07 PE: Really?

00:39:47 ED: Yeah. Big, big vacuum chambers. So, there were a couple of problems with the machine. One is the, one day, someone was sitting at the operators console and their tape wasn't reading. And so you read back and try to reread and it would crash again and read back and reread and crash again and say "What the hell is wrong with this thing?" And it finally occurred to him "why don't I go into the room where the tape drives are and look at the tape drive?" and in fact, with the vacuum tube circuitry, the tape drive had to be water cooled and the plumbing had sprung a leak. So the vacuum chambers on that particular tape drive were actually filled with water. So what it was doing, was washing the tape back and forth as it was dragging it through and trying to read it. [chuckle].

00:40:39 PE: Washing machines.

00:40:39 ED: Yeah. The other problem was that the major customer for the D1000 was the Treasury Department. They used it to read punch-card bonds, treasury bonds which were turned in. And unfortunately, a lot of people, for some reason, women, I presume women, when they would redeem their bonds, they would kiss them goodbye.

00:41:01 PE: Oh no!

00:41:02 ED: And so the bonds are getting lipstick on them and the lipstick was getting on the brushes on the card readers and that was a problem. Other than that, the machine worked pretty well. It was big and expensive and the technology wasn't the right technology for the era. And so, Honeywell and Raytheon sold a grand total of seven of those machines. Now, they are pretty far in the red. I think it was at that point, although I am not sure, that Raytheon bailed and became Honeywell EDP, Electronic Data Processing. Next machine they built was a Honeywell 800. And I used to know the name of the prime mover of the 800 and I have forgotten it.... He was brilliant. He figured out... Now bear in mind, the 800 is a big machine, it's transistor technology discretely wired; not printed circuits. Bad technology again. But more than that, architecturally, it was a mixture of brilliant and forward-looking and badly implemented. So the 800 could run eight programs simultaneously. Later, I am pretty sure it was later, the Control Data 6600 could run ten peripheral programs for I/O simultaneously, doing a round robin on their I/O processor and Control Data got...

You know, it was like a pipeline stage with ten programs rotating around, called a "barrel and slot arrangement". The slot was where the active logic was. So, one position is the barrel of ten programs rotated around did calculation and the other one would do input/output and so on. So of the ten slots, about three of them did useful work, and the rest were just waiting to take their turns. This was because I/O was slow and the logic of the Central Processing Unit was fast and it was economical to build the peripheral logic with the same technology as the central processor. But now it was too fast for the I/O devices. So build one set and have it shared by ten I/O devices; makes sense. That's Cray, I think, Cray and Thornton, before Cray Computer, at Control Data. Honeywell, I think was before that. So they ran eight programs in the central processor, not in the strict round robin, but the idea was that if one of the programs was in I/O wait, input/output wait, it was skipped over in its turn. And you run the ones that aren't waiting, that have something to do. Or if there are none of them waiting, you run all eight. So you can use the central processor logic more efficiently this way and compensate for the I/O wait. Problem is, two things hadn't been invented yet. One of them was memory protection. And the other was operating systems. So, it was up to the users of which ever programs happen to be running simultaneously to keep from interfering with each other.

00:44:42 PE: So they had to partition the memory themselves?

00:44:44 ED: Yeah. The application guys, for running with particular partners who would cooperate in the... Honeywell sold eventually 25 of those machines. They asked their customers a couple of years later, how many programs do you run? One of them said that they often ran three programs. Another one said that there was a day when they ran five. And then I do not think that they ever did it again. Everybody else, there was maybe a third. But everybody else just ran one program. Then you have lost it, you are paying for way more than what you are going to use. So the next machine was a Honeywell 400, scaled down from the 800, done with printed circuits this time and transistors and they sold about 25 of those.

Now we are up to where I joined the company. Not that I made the difference [laughter] as a grunt logic design engineer. Shortly before that J. Chuan Chu had been working at... at Sperry I believe. And had noticed that the IBM character-oriented machines, the 1400 series and 1420 and so on, business machines, character, 6 bits, not bytes — 8 bits, that comes later with the 360. The character-oriented machines were getting old in the tooth as far as technology was concerned. It had been years since IBM had upgraded. So they were not competitive in performance. So his idea was let's make a machine that is compatible with the IBM character-oriented machines and can run those programs, but in modern technology and just be very fast and cheaper. And we'll take IBM down. And Sperry looked at themselves, and I think it was Sperry, yeah Sperry UNIVAC, looked at themselves and looked at IBM, and said “no thank you.” [chuckle] And Chu went to Honeywell. And Honeywell said “we will do it.” That is what the Honeywell 200 was. And that was like a month before I happened to apply for a job. So they were hiring logic design engineers by the bucket. And I came into that.

00:47:52 ED: So one day I am a graduate student and the next day my boss Lou Oliari tells me "Okay, kid, why don't you design a three-quarter-inch tape drive controller?" Oliari, O L I A R I, Lou. So I had to learn about the three-quarter-inch tape drive controller. Honeywell tape drives are good, they are down to a half-inch tape by this time, mylar, vacuum chamber, great tape drives. And they have a lot of control in the drive itself to protect the drive and so on. And the interface is relatively clean. But the three-quarter-inch tape drive [chuckle], right... The industry was moving toward half-inch tape by that time. The three-quarter-inch tape drive was actually designed to work with the 800.

And somebody at that time had a thing called Kangaroo format. The 800 did its I/O in blocks of 48 bits, but the tape did not have 48 bits across in one row; it had I think 8 bits across in one row. I think with parity. And so they would do 6 rows across the tape of 8 bits and it would a 48 bit block. And so they would write block by block. The Honeywell 200 is a character-oriented machine, 6 bit words, characters. And so it wants to read... Yeah, that is right, 6 bits at a time and the tape drive is feeding it 8 bits at a time. Beyond that, Kangaroo format was designed for reliability purposes developed by somebody, who knows, he had an idea. So you know long check on the block and he tried to, and he had parity on a row and he tried not to put related bits on a row, he scrambled up the 48 bits in some horrible pattern to make a block of 48; it is a checkerboard thing which is why they call it Kangaroo. And theoretically the parity check and a long check was supposed to give greater reliability that way. But as far as I was concerned, my 6 bit characters were coming in all helter skelter all over the place. The tape drive runs at a pretty good clip, but variable speeds, so there is a speed tolerance in there and you do not exactly know when the next frame is coming in. And I have got to design a controller. I have got to pull stuff in, 8 bits at a time, unscramble it and feed it to the processor 6 bits at a time to do my records and there is a little bit of padding and stuff like that. So how do I do that?

00:51:02 ED: Well I know all about minimizing logic. I am going to minimize the logic. So I designed that controller using what I thought was the minimum number of flip-flops. Where I discovered that if I stored a 48 bit block plus another two frames, I could meet the speed tolerance and do the unscrambling and feed it to the processor 6 bits at a time and it would work.

Now I didn't absolutely minimize the lot, the number of flip-flops, but I convinced myself that I probably could not do it in fewer for that buffer. Any seasoned engineer would have buffered 248 bit blocks using 96 flip-flops instead of the 68 or 70 that I had used, and it would have been a very clean design. [chuckle] I have got communication pads all over the place, I mean that unscrambling logic was unreal and the combinational functions that I had to add to do bit sparing and unscrambling and speed tolerance and all that, was just unbelievable.

But I had never designed anything before. So how do we design? We write these things down on paper, and we type them up on forms, and they get entered into a computer-aided design, CAD system. Which is running on a Honeywell 800. Which is keeping records of this function and that function and this goes on this card and this slot and so on. So the physical packaging of the system is interesting. The tape controller was in one drawer, there were four drawers to a cabinet. The cabinet is a short thing, a little lower... it’s about desk height. So it is maybe three foot by three foot by four foot, a solid rectangle and it has these four rotate-out drawers that come out the front, hinged on a pin. In a drawer you could have 160 cards of logic. And what is on a card... I knew were some random cards like clock generators and one-shots and odd logic like that. But also you could have two flip-flops on a card, you could have a couple of gate-buffered inverters or gate-buffered drivers. Which is basically an AND OR structure followed by an amplifier or an inverter. And the gates were variously, like you might have like a 3-4-4 gate buffered inverter that would mean a 3 input AND gate a 4 input AND gate all OR'ed together and then inverted to a driver.

00:54:08 ED: So after you design your logic you have got to figure out which cards to package it up on. And the circuit design people looked at our preliminary logic and defined a population of cards that would match our preliminary logic. And I got in trouble because when I turned in my cards somebody in circuits got furious at me and said "You have never used any of this card". And I said "Well it did not fit". And he said "We designed that one for you!". And I said "Well that was preliminary. When I finished I did not need them" [chuckle]. He was really upset because they tried to minimize the number of distinct packages, because they had spares. Well my logic was huge, my combinational logic went through the roof, I had the flip-flops down, but that is not what should have been minimized, you know. I learned to minimize what should be minimized gradually, with experience, and do it clean.

Interview recording part 3

Debugging was unreal. So what they did was we had the CAD system, they had built six prototype machines and they gave three of them to the guys who were going to become the field service Engineers, and three of them to the design engineer s. And we worked 24-hour shifts, seven days a week on that shift, but we were debugging 24/7 in shifts. The field service Engineers and the design engineers were racing each other and they had two copies of the design database. When I say CAD, this is just record keeping; there is no simulation, there is no verification.

00:56:03 PE: I was going to ask you about that. So you just kept a list of the things you were designing?

00:56:09 ED: Exactly. The idea was we will make six of them and then we will debug it and then make it work. The back plane was wire wrapped. So you could re-wrap stuff and unwrap and entire card and put a different card in.

00:56:23 PE: And this was presumably all just text and numbers. No pictures or design structures like that?

00:56:30 ED: There were text and numbers in kind of a chart form, like character graphics. So that was one thing, there was a layout diagram, you know. So that was CAD, that's what passed as CAD. They made two copies of it...

00:56:46 PE: Did that CAD system have its own name? Do you remember?

00:56:49 ED: I have no idea.

00:56:50 PE: Because there are not many of those, but it is okay.

00:56:53 ED: I know we were very excited because an improved CAD system came in one time from SAAB which Honeywell had formed a partnership with, and we were all eagerly looking forward to it and discovered with horror that all the text was in Swedish. [chuckle] So that was the end of that. [laughter]. When I finished my pencil design and entered it, I turned it in to my boss. And several weeks went by and I didn't hear anything back. There was no design review, nobody critiqued my design or anything and finally I could not stand it anymore, I had nothing to do, it was dead calm. So I went to Lou and I said "You know that tape controller design I gave you a couple of weeks ago?", he said "Yeah." I said "How did you like it?", he said "What do you mean?". I said "Well, did you look at it?" and he said "No". So I said "Well, where is it?". And he said "It's in manufacturing." [chuckle] And I felt the ground open up beneath my feet. It had never occurred to me that these random scribbles that I had done would go into manufacturing without being verified or critiqued or anything whatsoever.

00:58:24 PE: And did it work?

00:58:25 ED: Well, the debugging was difficult. There were lots of problems. It was a very complicated design, it was a complicated task and I was a greenhorn. To be fair, Honeywell, when they ran 24/7 they put the engineers on what they call rotation. Which meant that we worked one week days and then the next week evenings and then the next week graveyard. At the time I had met Ellie, who became my wife, and we were seeing each other every night. So I was debugging and dating fairly intensely and debugging at any odd hour of the day. Plus I had been evicted from my apartment because I had a pet that the landlord didn't like and eventually settled for an apartment that I could afford where the realtor was showing me around all these apartments that I did not like, they were overpriced, said "Well I can find you a place in your price range but would you mind living near black people or nurses?". I said no, and as it happened I ended up in an apartment building where some nurses lived on the floor above me and they got off at midnight and partied until dawn. And so the combination of my debugging life and my social life and the nurses — after 4 weeks of that, I came down with mononucleosis.

01:00:11 PE: You must have never slept.

01:00:12 ED: I never slept. And it was interesting to watch the debugging style. Oh yes and also in the middle of that, you know bear in mind that this is a machine that the company, in its total history, had sold 57 computers. And now they've built six prototypes of the new computer. It's a cheap computer, it's a small one. They're making a big bet on this thing. So the CAD system ran on the Honeywell 800, but *which* Honeywell 800 was an interesting question.

01:00:57 PE: Ah, of the six prototypes.

01:00:59 ED: No, no, the six prototypes of the 200.

01:01:01 PE: Sorry. 200.

01:01:02 ED: The 800 is an older machine. They're still manufacturing it and selling them. Every time someone buys one, they make another one. [laughter]. Okay so what they do is they run the CAD for the 200 on the 800 that is coming off the assembly line before they ship it to a customer. Our computer-aided design is their burn-in test. One week, they ship the 800 that all the CAD is on to a customer. And for the next two weeks, we don't have a CAD system, because they don't have another machine coming off. So everything's on paper at that point.

The debugging of the design engineers and the field service guys is very interesting, very different. The... When something peculiar happens, the design engineer goes back to his personal motivation in the design, analyzes the logic, tries to figure out what's the mistake that could lead to this behavior, and pinpoints the problem and fixes it, probably introducing other problems. But you know, that's the fix that he has in mind, he tries it. Gradually and systematically working his way through. The field service guys had a cram course where he's going through all the logic in the design. Whenever anything is wrong, he tries to locate the area of the design where the problem probably is, rips out all the logic, and redesigns it, in that area. So, we design engineers make 10-15 wire changes a night. The field service guys are making 100.

01:03:03 PE: Wow!

Interview recording part 3

01:03:04 ED: Unbelievable.

01:03:06 PE: And are they reporting to you what they're doing?

01:03:08 ED: No.

01:03:09 PE: [laughter]

01:03:09 ED: Two different machines, two different CADs, it's a race. First one to get it working, that's the design. Who gets it working first?

01:03:19 PE: Them.

01:03:20 ED: Absolutely. Absolutely. They're monkeys man. They just do it!

01:03:28 PE: How well did it work?

01:03:28 ED: It worked! It worked.

01:03:31 PE: Gosh.

01:03:32 ED: It probably didn't help that I was out with mono for a couple of weeks in the middle there. But we got it working. Okay, so Honeywell... Okay, what I didn't know is that... Why would a company be so crazy as to give someone as green as me this God awful complicated design to do? The answer is it wasn't the real product. It was the legacy tape drive from the 800 for customers who might want to buy 200 and had previously had an 800 but didn't want to buy a new tape drive. So they could use the old tape drives to at least read these old tapes on the new computer. That's what they were after. The real tape drive for the 200 was a half-inch tape. They gave that a much simpler controller, and they gave that to an experienced engineer to design. But I didn't know that. So they're not as crazy as they appear.

So after I finished that, they decided to have me take over the design of a paper tape controller. And that, in one controller you could control both the reader and the punch, but the punch was trivial. The reader was a little delicate. The punch you just output the information, and it goes bang, and you do the next one. The reader you could have errors, you could have you know other problems, you have speed problems and so on. Instead of going chunk, chunk, chunk at your speed. I look at that design, I look at the controller the tech had started, I said "This is the simplest thing in the world. There's nothing here". I finished that design in a week. And I pored over it. I wanted to be... The thing is, it was so simple. Honeywell had never done a design that turned out to be bug-free. I wanted that record. That was for me.

01:05:47 ED: Well, I only had, I think, two bugs. One of them was fairly spectacular. One of them was, that if you read a block forward, and you get an error, parity error, what you do is you read it backwards and see if you come out okay that time. And you get errors all the time because paper tape holes are not clean when punched and you get lint in the hole. And it's an optical detector. So, my problem was, if you read forward, and then read it backward, it has the spring roller arms that hold the tape. So when it suddenly reverses from forward to backward, the roller arms go bang against the top, and it stops, and the tape breaks. So that was my bug. Reason: in the paper tape drive, which we bought from National Cash Register, there is no logic. It's a raw drive; all the intelligence is in the controller. I was used to dealing with the more expensive magnetic tapes, where the protection logic was in the tape drive itself. And so time-locks, and so on, were built in. It wouldn't reverse before it was ready to reverse. It had all its vacuum chamber control logic in the tape drive. Turned out that was an easy bug to fix.

Now they're selling computers. Different phase. This company that had sold 57 computers in its lifetime. In the first two weeks after they announced the Honeywell 200, got 500 letters of intent to buy. All the IBM character companies, the people who were owners, were desperate for a better solution, and they were flocking. Honeywell didn't know what hit them. They never imagined it would go that way. So what do they do? They go out and they hire every IBM salesman that money can buy, and put them out on the street. Sell, sell, sell. They're building these machines as fast as they can crank up the factory line. They're flying out the door. One of the first big machines that was a takeover from IBM was at S&H Green Stamps in New York. Turns out... Am I going to have to describe S&H Green Stamps?

01:08:35 PE: [laughter]. I guess a lot of people will not know what those were. So you'd probably better say a little bit about them.

01:08:41 ED: It used to be in that era, in the early 60's, and we're talking '63 now, if you went to a supermarket, as a premium... If you left your garage, and paid your 23 cents a gallon for gas, they would give you a washcloth or a glass or soap or something. You go to a supermarket, you had this: they give you S&H Green Stamps. One stamp for each dime that you spent. So you start racking up stamps pretty quickly and then you have to lick them or wet them and paste them in a book. And when you get...

01:09:19 PE: I remember my mother doing that.

01:09:21 ED: [laughter] When you get 80,000 books, you can turn them in for a bridge table or something.

01:09:26 PE: Or a chaise lounge. That's what they did.

01:09:28 ED: A chaise lounge, exactly. Premiums.

01:09:29 PE: I think I got one that way.

01:09:32 ED: So what you do is you take your books down to the local S&H Redemption Center — which doesn't offer salvation, it offers premiums. And you turn in your books, and you pick out a premium that you want, and they give it to you. Now they have to do accounting for that. How do they do accounting? The cash registers at the redemption centers have paper tapes in them. They punch paper tape. Every night the paper tapes all go to New York City, where a room with 35 paper tape readers reads them from all over the country, and tabulate what the heck is going on and that is how S&H Green Stamps runs.

So they threw IBM out and bought Honeywell. And they've got a room of 35 paper tapes reading all night. And one day I am walking down the corridor and Lou points his finger at me and says "You are taking the 7 PM flight to New York, to S&H Green Stamps. They're having a problem with their paper tapes, and if it's not fixed by morning, don't come back. You don't have a job."

01:10:58 PE: [chuckle] Wow! Is this because of your controller?

01:11:00 ED: Well we don't know. So, I do a deep gulp. And what had happened was over a period of weeks, S&H complained to the local field service guy that every now and then when they have an error on the tape, they go read back and read forward (because I'd fixed the tape break problem). It didn't do that, but every now and then, in some pattern, the tape would lock up and read forward through the entire tape without transferring any data, without a command. And so then they have to recover, and go back, and you know, it may or may not work the next time. So they've been complaining without satisfaction for several weeks, until finally the president of S&H called the Minneapolis president of Honeywell and said "If you don't fix this problem by tomorrow morning, out you go, and back comes IBM". So it trickled down, as they say.

So he says "You are going to New York. Don't come back." I said "Gee Lou, what seems to be the problem?" [laughter]. And he explained the lock-up problem and I listened to him carefully, and I said "Well, that problem cannot possibly be in my controller because I know my design cold and there's no way that my controller can do a thing like that". A little bit of a bluff. I didn't think it could. But I couldn't have been that sure. And he said "You're going. You've got your flight and you're gone. And you better fix it, whatever the problem is". Thinking fast, I remembered that the paper tape reader that we bought from National Cash Register also had some electronics in them, minimal logic in the drive itself. It was designed by someone from the peripheral unit design group, and I knew who that was. So I said "It's probably on his side, and if I have to go, he has to go". Well they're fine with that. They'll buy two tickets. [That other guy] was ready to kill me. But he had to go too.

01:13:31 ED: So we take the 7 PM flight. You know, we were told this at 4:00 in the afternoon. We take the 7 PM flight to New York, and we try to make it happen. So we pick on the tape drive, and we start going back and forth, back and forth, and very soon it locks up and drives off the tape. No questions. So okay, now we can observe it. Let's look at this function, let's look at that function. We're working, we're sweating bullets. We have the field service guys with us. So there's about four of us poring over this thing. And we can't... We look at the signal that tells the drive to drive forward and that flip-flop locks up. We look at the clock going to the flip-flop, we look at the set and reset signals going to the flip-flop. They're not telling it to lock. There's no signal. Of course, it would just be a pulse on the set that would set it, you know a random pulse. But we're looking, we can't see any pulse. What the heck is going on? Ah, and I might add, the flip-flop is in the drive. The flip-flop in the controller in fact is not telling it to drive off. So it's not my problem — except it is my job that I'll lose if we don't fix it. So we are in the drive in the logic that he designed and we can't find any input to that flip-flop that is causing the output to lock up.

01:15:18 ED: Until finally, the guy that did the peripheral design says there are two inputs to the flip-flops we haven't looked at. The ones that you normally don't consider. One is called power and the other is ground. Hang the probe in the power signal. And the power signal, +5

volts you know, flat as a lake in a dead of the summer, it’s got hair on it like you wouldn't believe. And it's coming, fuzzy and goes away, fuzzy and goes away and then the flip-flop locks it. And then the drive goes forward. So, what the heck is that. And we are watching it, you know in disbelief, and eventually while we are watching the scope we can hear "click-click", "click-click", "click-click". And every time we hear the click, there's fuzz on the voltage. The click is the relay that runs the drive motor in the tape reader. So every time the tape reverses direction, the drive motor clicks in the other direction, there's a power surge, it’s getting coupled into the power to this flip-flop, it goes back and forth a few times and then some noise hits in the wrong place and everything locks up and in a set state and it just keeps going in the same direction. So we said "Okay, where is that coming from?"
Turns out it wasn't his fault either. The factory, when they built the tape drive, made a neat bundle with... The power that was driving the AC motor relay was cabled in a bundle with the voltage supply that went to this flip-flop. Because it looked pretty. And so there is cross coupling between the wires, and off you go. He was furious because in his instructions from the factory, he'd said, do not couple high voltage and low voltage signals. If you want to couple low power signals and logic signals together, you know, you separate harnesses. But they just bundled it all together. So we went in and we had these...

01:17:58 PE: But it was only happening on one drive?

01:18:00 ED: No no. We looked at one drive. It happened on all. Once we found the problem, that's the one we are going to fix. So, now it’s 3:30 in the morning. We go to the blue plastic harness ties, with these clippers and we cut all the ties. We spread the wire out in a rats nest all over the back of the tape drive, slam the door shut that whole rats nest mess is stuck inside, run the program again and then the power supply is clean and the drive won't lock up. We turned to the [chuckle] couple of field service guys that were there and we said, "Well there's a plane to Boston at 5:00 AM, and we are going to be on it, you guys do that to the other 31 drives there and get done by 7:00 AM and we are out of here". And we never heard another thing about it.

01:19:05 PE: Wow! That's a great story.

01:19:08 ED: But that's how life was. It was bizarre. It was really bizarre. They used to give out security badges. If you come to work, do you have your ID badge? We thought that was really funny. Why do we need an ID badge really? And if we didn't have one, you give us one. Funny when, I had to sign in for that one and turn it in at the end of the day. And they would say, well you know, someday we might get a defense contract and you have to show that there had been secure facilities. We want to everybody to obey security. And it was new to us, so we used to have a contest, to see who could collect, you know sneak out without turning in your non badge, temporary badge that they give you, see who could collect the most badges in his desk drawer. And eventually they would run out of badges and they, we were doing all kinds of crazy stuff. And one guy was doing debugging in the middle of the night. He would come in, he did his designs at night, he didn't like to work during day, because it was quieter at night. He was doing debugging one time and he had his head inside a chassis that he was wiring when he suddenly realized that there were a bunch of high voltage signals in the chassis. And his head was near it and he was afraid to back out, because he was afraid he would touch it. So he started screaming. And eventually another nightowl came in and saw him there and started... He said "Get me out of here", "get me out of here" and the guy started laughing. He said "Get me out of here" and he said, wait just a minute, I will have to bring in some witnesses. Nobody else will believe that this happened. So he went and got other people to witness it before they shut the power down and let him out.

01:21:05 ED: We had a printer... Just bizarre. We had a new printer that we were designing. We used to buy printers from IBM. And every printer we got, same with card equipment... They sold huge numbers of machines and we didn't sell much of any. So every one we got was a different model, slightly different models. We had separate documentation for every one in the field. We couldn't live with that, so we started doing our own and they had high-power vacuum tube logic in them. Somebody designed a new printer and they were doing demo to a customer. They were late doing the design and the customer wouldn't postpone the demo. And the customer comes in, and the instructions to all the engineers like me were “clean your finger nails and shine your shoes and wear decent shirt and tie and stay out of sight, because these are important people coming.” I ushered them into the demo room with this printer that had never been powered up. It never made it to debug. Why they thought this was going to work, I don't know. So there they are in front of these big guys from a company and they explain a little bit about the printer and they turned the power on and somebody had mis-wired the power supply and the thing goes bang in a puff of lightning. Paper catching on fire. Customer leaves, and says "Well, I will never buy anything from you guys". [laughter]. Craziness. Cool things.

01:22:51 PE: I was going to tell you — I worked with Honeywell machines. I think I told you this earlier. It was much later, I mean 1975, and that's back when the tape drives you are talking about were still there. Were still very very good. We had a funny problem, with them one day, which these stories remind me of. It was actually for a number of weeks that randomly, the tape drives would dismount. The windows would come down, the tape would stop and there was no obvious reason for it all. And we had field...

01:23:25 ED: Window comes down if you want to remove the tape drive?

01:23:30 PE: Right. And the field engineers came over and over again, they couldn't figure it out and they dismantled everything. The drives, the computer itself, looking for the problem. So weeks went by, this kept on happening, and finally they figured out that it always happened at the same time of day. And then they eventually understood that it was happening when the sun would pass at a certain angle coming through big windows and there were optical sensors inside the drive and when they hit these things, they just messed up the signal. And it only lasted for 20 minutes. But it was long enough to really mess up the whole thing.

01:24:13 ED: Right, because the optical sensors have to sense whether the tape is at the right level in the little chamber to give it slack. Indeed.

01:24:25 PE: Probably many thousands of dollars of FE time went into that particular problem. Alright, so Honeywell. You have a few notes here. We have got the paper tape story. “Don't go to the library, don't meet in the cafeteria” [reading ED’s notes]?

01:24:50 ED: Yeah, so in these heady days, IBM then made an announcement that they were going to announce something. And everything stopped.

Interview recording part 3

01:25:05 PE: This is around 1964.

01:25:06 ED: And then they announced the Systems Research. Honeywell got whatever IBM customers that were, that did not, for one reason or another, wish to convert from the character-oriented machine to the byte oriented 360. And they could run on them. That crumb of the market made Honeywell number 2 in the industry, ever so briefly.

01:25:45 PE: But that was going to be a dead end.

01:25:48 ED: They didn't know that. So they designed the Honeywell 1200. They designed the Honeywell 2400 or something and just following, following. We were coming up with new and different kinds of designs for different markets. Because we had nothing else to do. And they were running ideas up the management chain and management would ask marketing. And marketing would say, "No, we can't sell that machine. You can just design the 1200, which were selling like hotcakes, we don't need anything different". So management decided not to pursue the machine. You do that 3 or 4 times, and the environment just goes dead.

So the design engineers were meeting in the library. We read books, we talked to each other, we met in the cafeteria, and we discussed ideas, we were in a huge group and we moved out of the Sunshine Biscuit factory in Newton Highlands into the Clevite Transistor factory that Honeywell had bought in Waltham. It had a big cafeteria, we met in there. Then comes the edict that they don't want to see design engineers in the cafeteria talking, they want to see us at our desks working. Most of us had no work assignments. And some friends of mine and I wrote a memo and tucked it on the bulletin board as if it had come from Chu. We forged a signature on it, posted it in a 10 minute period all over every bulletin board in the plant, that announced the promotion of some bureaucrat to Director of Cafeteria Reservations and reminded everybody that he was the genius who also thought that you couldn't buy donuts before 10 o'clock in the morning, the coffee break time.

01:27:53 ED: And everybody thought it was [hilarious], except for the guy we were naming. He didn't think it was funny at all. And he ran around tearing it down, writing to his boss, and his boss cracked up laughing, fortunately — instead of killing us trying to find out who did it. We idled away the rest of our time holding tiddly-wink contests into the ashtrays on our desks, because we used to smoke. We all smoked. We would take giant size desk pads and fold them into paper airplanes and hook a bunch of elastic bands together, tie them between the posts of our cubicle and launch them across this 500-engineer bullpen. Some of them would stick into the foam surrounding air conditioner ducts that were exposed in the ceiling. I was a willing participant in all of that. But then Ramamoorthy, ...CV Ramamoorthy, who eventually went to Berkeley as a faculty member... He had just finished a PhD at Harvard and was running a little research group. And since I had nothing to do, he allowed me to sit in his research group when he talked about stuff and that was great. And I started learning things. And he was impressed with me. So he went to Lou and asked if I could be transferred to his group. And Lou refused and said no, I was too valuable as an engineer. I had no work assignment and was not doing anything, totally making dimes stand on edge and shooting paper airplanes. And one day, some blue-ribbon people were taken on a tour of the engineering bullpen, and I didn't know that and just as they walked by I launched a paper airplane. Which missed the guy's head by 2 inches and lodged in the wall opposite. And I decided then and there, it’s over. I've got to leave this place. There is no future.

01:30:25 ED: So after that, you know, Honeywell really didn't produce more machines, but they bought the three C's company in Framingham, CCC, which is... I don't know, Computer Control Company or something like that. [PNE - this is correct.] And they made a line of control computers which fed into the Honeywell product line and after Honeywell ruined that, I think they bought... Did they buy RCI [PNE - Renaissance Computing Incorporated]? I think they talked about buying RCI, the Specter Series. Maybe they didn't.

01:30:51 PE: I am not sure

01:30:51 ED: It was a conversation, because RCI was getting tired of making Specters' which are, I think, were 360 knockoffs. Then they bought GE [General Electric’s computing division, not the entire company], which had a very successful Multics project at MIT, and then they ruined GE. Then years later, I was brought in as an academic, nobody knew that Honeywell at that point was brought in as a review team for a computer idea from General Electric that management was worried about, as it seemed rather progressive. And they said you know, can you trust these people? I said well, it is rather progressive, it is pushing the technology. But if you stay in the computer business, that is the kind of thing you have to do. And so they immediately canceled that. You know, they ruined GE which had been a wonderful company in computers.

01:31:52 PE: Were they bought by Bull later? Or...

01:31:54 ED: Okay in desperation they emerged with Machines Bull in France, then became Honeywell Bull. You know like the partnership with Raytheon. It was to expand the market. But that you know that was one disaster company joining with another.

01:32:11 PE: They had some nice posters. Bulls made of circuits.

01:32:15 ED: Oh yes, that was the 200. Those were the advertisements for the 200. They took electronic circuits and they made... It wasn't... That was before the merger with Honeywell Bull but they had different animals. Some were bulls, some were kangaroos or whatever.

01:32:32 PE: Yeah. That was beautiful, that was a great campaign.

01:32:34 ED: The other interesting with the 200 was that, Honeywell didn't have any operating system. People didn't know what we were doing. And we had to run IBM code. Well, they designed a piece of software called the Liberator which took IBM code and translated it to Honeywell code, which wasn't difficult as they were virtually identical. We did a design with the IBM manual on one side and the data sheets on the other, you know. The instructions were the same instructions and so on. There were some lawsuits over that. I don't know how they were resolved. But they couldn't get the operating system to work. So one day Honeywell hired a salesman from IBM and announced that the operating system team could go to something else. They didn't need to fix the operating system. And this guy had come in with a copy of the IBM operating system which they ran through the Liberator program and whatever came out the other end became the Honeywell operating system for the 200. Amazing, those were the days.

Interview recording part 3

01:33:54 PE: Okay. So...

01:33:56 ED: So, Ramamoorthy told me, I was a smart young fellow and I should leave and go to graduate school. What I really wanted to do was, what I should do is go to Illinois and work with Dave Muller for a PhD, because Dave was a genius. And I really didn't want to leave home. So I took another pass by Harvard. And whoever interviewed... And I think it was Hao Wang. There were two Wangs: one was at Wang computers [PNE - An Wang] and the other was a professor at Harvard [PNE - Hao Wang]. But he interviewed me. He said, "Well, if you came here for a PhD , who would you want to work for?" And I said, "Well, I never thought about that, but I have been doing logic designs, so I guess I would like to work with Tom Bartee". And he said, "Tom Bartee? He is not somebody you would work for. He is not a real faculty member, he is just from... " 'Cause he had written a book in logic design, he said "He is just a guy, an engineer from Lincoln Labs. We keep him around because somebody’s got to teach the Logic Design course. You want to do something that’s worthy of a PhD". So I decided I didn't actually want to go to Harvard, never applied there. I applied to Stanford and Illinois and couple of other places. But Stanford turned me down, Illinois accepted me. Talked to Dave Muller and decided in 10 minutes that he and I were on different planets. I learned later... Dave was a lovely person, and a brilliant guy. He was working on speed-independent logic. And I had by this time taken David Huffman's course on hazards and race conditions at MIT. And... Great course. Huffman and...

01:35:44 PE: Hazards and, I am sorry...

01:35:45 ED: Race conditions and how to design asynchronous logic and minimize flip-flops and state tables and stuff. Hazard is due to timing problems; you don't end up in the state that you thought you were because you're doing multiple transitions and if the wrong one goes through when you are clocking it off, you'll just end up in some funny state. But you can design state tables so that it’s immune from hazards. So race conditions are bad things. Race conditions can cause hazards. Muller’s theory of speed-independent logic had two critical elements in it, logical elements. One of them, the definition of the element had an essential hazard in it. So in fact the speed independent logic, as he designed it, wouldn't work. I didn't know this at that time. It came out later. But when he was asked... Muller was in Math Department. When asked about it later, he was asked if he had ever studied Huffman’s hazard theory. He said "No, why would I do that?" And he said “ because your S element or N element, whatever it was, has an essential hazard in it. It can't be built.” And he said — he's a mathematician — he said "I never said it could be built". I said, if you had this element and that element you could... I proved that you could design speed-independent logic. And he is absolutely right. That his element can't be built. The ILLIAC II was designed with speed independent logic, using Muller's theory. So it wasn't going to work and they soon discovered that. So they...

01:37:40 PE: Money spent on a wrong theory.

01:37:42 ED: They fixed it by doing an incoming test on every transistor that they used in that machine, and put the slow ones over here and fast ones over there, so that they could control it. Unbelievable. So that's... You know, you learn.

Interview recording part 3

University of Illinois, PhD in Electrical Engineering (1965-1968)

01:38:05 PE: So who did you work with then?

01:38:08 ED: So I went over to the Digital Computer Lab which eventually became the Computer Science department, but not yet.

01:38:16 PE: This conversation with Muller happened after you got to Illinois? Or...

01:38:20 ED: No. When I was interviewing, deciding where to go to grad school. And I was looking for an advisor and hoping to get admission.

01:38:28 PE: Okay. We're still talking about the same point now. You left Honeywell...

01:38:33 ED: Yes. So with same trip, I walk across the street into the Digital Computer Lab. But that was a laboratory that built ILLIAC's research contracts, ILLIAC 1, 2, and 3 was just started. Four hadn't started yet. So I met Bruce McCormick, who was doing the ILLIAC 3, and he thought I was a smart fellow and was interested in taking me up. And I met Saburo Muroga, who did threshold logic and minimized threshold logic using linear programming. He was the world expert on threshold logic. I wasn't particularly enamored with threshold logic, but I liked linear programming and optimization. And I had had Little's course at MIT, taught in the business school, on Operations Research, and I thought I knew stuff about linear programming and combinatorial programming and optimization. Which was the area that I wanted to work in.

So I really wanted to work with Muroga. And he didn't have any money. And I needed support. As I told Ellie when I proposed to her that I had a great salary at Honeywell and she wouldn't have to worry about security if she married me. I was making I think 6500 dollars a year. Might have been 8000 by then. When I started at Stanford years later I was making 12,000 —16,000 if I had enough research money to pay myself over the summer. I thought that was a lot of money. But I didn't have enough to support myself in graduate school because the year after I married Ellie we bailed and moved to Illinois.

[laughter]

01:40:33 ED: And she started supporting me working in child guidance clinics. So Muroga didn't have money, and I needed financial support. McCormick had money because he was building the ILLIAC 3 and he said "Oh, kid, you like optimization? I got an optimization problem for you. Can you solve the traveling salesman problem?" A classic mathematical problem, which is very hard and geniuses have worked on that problem. Well, I didn't know that at that time, so I said "I don't know. Why are you interested?" Well, turns out his cards were plugged into the back of the ILLIAC 3, each pin came out on 2 holes. When you poke up a paper tab into the hole... ... Just 2 holes on every pin. So if you want an output to go on a bunch of pins, what you do is go to first load that you're gonna hit, shove the pin from the output to that load and then you go to the other hole, and shove another paper tab and a wire in there, and connect it to the next thing. So every output run to multiple loads has to be a linear run. And he could save wire on his back plane by solving the traveling salesman problem, to get the minimum amount of wire, with theoretically the fastest signal. Well it’s insane to solve a classical... I don't know if it’s NP hard, NP complete or whatever — it’s NP something. To ask a graduate student to solve this going up against professional mathematicians who haven't done very well with it for years...

01:42:26 PE: Plus how many holes are we talking about? Many many thousands. I assume that...

01:42:29 ED: Well each run is maybe of 8 or 10.

01:42:34 PE: Yeah. But that’s hard enough.

01:42:34 ED: But this is insane. And who would design a thing with serial runs? Everybody was using wire wrap pins where you could get wire wraps on, you can do a tree. And who cares about minimizing inches of wire to do a PhD. It didn't make sense. But he had money, so I went to work for him. For a year I worked on a traveling salesman problem. I actually designed an algorithm and published it. I have the only published article on the traveling salesman problem in which the word traveling is spelled with two L's.

[laughter]

01:43:09 PE: Well, that will make it easy for anyone to find on Google.

01:43:10 ED: You could find it. It is not a good algorithm. In the meantime I kept going to Muroga's seminar. And I introduced into the seminar I would take credit for, he asked me to, he was doing linear programming, I introduced combinatorial optimization, all kinds of discrete branch and bound methods and stuff. He asked me to talk about a survey paper on that stuff, two of them actually. And each one took me 10 weeks of seminar. Because they didn't let me move through it. So essentially I was a seminar for that semester.

01:43:48 PE: This is in your first year?

01:43:48 ED: First year. Yeah. Well, because I knew computers and I knew what I was doing and what I wanted to do as a thesis and I had worked already. So I was leading this schizophrenic life. I was taking undergraduate courses in Electrical Engineering plus the Physics requirements, required of Electrical Engineers, which were... I had basic physics but I also had quantum mechanics and atomic physics. And electromagnetism, and you know the miscellaneous CE that I had picked up, 'cause you can't get a graduate, or couldn't get a graduate degree in Electrical Engineering from Illinois without satisfying the previous levels’ degree requirements in that field. I didn't take any computer courses, I'd had them all. Well, I took Threshold Logic from Muroga. He taught a 3 semester sequences nothing but Threshold Logic, in which he read his book in class and we were usually about 10 pages ahead of him. Because he had to translate from Japanese. Well, anyway. And when you'd ask a question he’d look at you and say "Hi". And we thought he was very friendly, but it turn out, “hai” means yes in Japanese. And there was a woman in the class, and he turned to her in the second week of class and said "Why are you still here?". Didn't much like having women in his class.

Interview recording part 3

01:45:11 PE: Did she leave?

01:45:12 ED: I don't remember. You know, I don't remember things like that very much. In those days I wasn't sensitive to it. Muroga was a good guy, but I wasn't going anywhere and I was going crazy with McCormick and then I found out that McCormick had headed up several different projects, none of which had ever been completed successfully. Dan Atkins actually did some fine work with McCormick.

01:45:39 PE: He was there at the same time as you, is that it?

01:45:40 ED: Yes.

01:45:41 PE: That's right.

01:45:43 ED: We were in the same group.

01:45:44 PE: He told me he remembered you as being senior to him, I think, because you had ...

01:45:47 ED: Might have started a year ahead or something...

01:45:49 PE: That, and the fact that you were older, because he was coming straight through...

01:45:49 ED: He was straight from Bucknell. Yeah. I don't remember him as being junior to me. [laughter] He was another guy in the group. So then my neighbor Gene Slottow... We rented a house. But it was a real house, 'cause Ellie was earning a good living. And Gene Slottow, who invented the plasma display.

01:46:20 PE: S-L-O-T-T-O-W I think.

01:46:22 ED: Yes. He was the co-inventor there. He said "You know, if you got all these problems over in TCL why don't you come over to Coordinated Science Lab? It's a very different lab. And there's this young guy Metze on the faculty there", M E T Z E, Gernot, G E R N O T, he went by Gerry. And his real name was Metz. But people would keep calling him without the E so he decided he'd rather have it mispronounced than misspelled. He called himself Metze.

01:47:00 PE: What nationality is that? Is that...

01:47:05 ED: German ethnically, but his family, he grew up in the Sudetenland. And he could remember going into the bomb shelter when the Nazi troops were coming through and eating up all the canned peaches in one day with his sister. His father was the organist at a local church. Metze's job was to hang... Like Stuart Little, hang into the organ and hit the sticky peddle when it stuck. Sweet guy. He had designed the arithmetic for the ILLIAC 2. And Jim Robertson was his thesis advisor. He did some pioneering stuff in arithmetic. And then he came over to CSL work with Seshu on some photons.

Interview recording part 3

01:47:56 PE: Why is it called Coordinated Sciences Lab? It's a strange name, what are they...

01:48:03 ED: No. It makes sense. It started, it was a JSEP lab, Joint Services Electronics Program, which was Army, Navy, Air Force and it funded generic research on solid state. But other things. During World War II it was called the Control Systems Lab and it did radar stuff. At the end of World War II it broadened out but it was known as CSL, and it had systems people, and it had physical electronics, it required some computer people doing radar computations, had network circuits people. So what do you call it? They wanted to keep the CSL so they came up with the stupid name. [laughter]

01:48:54 PE: Okay I've heard other stories like that, but the acronym can really matter when.... Can be very persistent...

01:48:59 ED: Yeah. You don't have to... Well I guess it did have to refer to the stationery. So CSL was a free running place. It had the Joint Services Block Grant which gave certain amount of research freedom because you didn't have to write a proposal for anything that you wanted to do, which was wonderful. And Metze's style... Well he was working with Seshu on logic... [PNE - Sundaram Seshu].

01:49:24 PE: Spell that name.

01:49:24 ED: S-E-S-H-U. A real genius in reliability. Coming home from a consultant trip and driving late night from Chicago back to Champagne, he crashed into a bridge abutment and died. And nobody knows whether he was... I mean there's some speculation that he was depressed at the time, though we don't know that. It's not been confirmed. Metze being a young faculty member announced that he was taking over the group. And he would run the group. So he was open. His whole style was open. A style that I did everything I could to emulate in my career. If a graduate student came to him with an interesting idea, he'd find a way to do it. So I said "Look, here's what I want to do. McCormick is killing me with the traveling salesman problem [laughter] and Muroga has no money to support me and I think I know how to do a branch and bound optimization technique for minimizing combinational logic." Metze says "That sounds interesting. Do you want to work with me?" Done. Done deal.

01:50:56 PE: I'm gonna have to stop for today. That might be a good place to do it. So we pick up with Metze tomorrow...

[Interview recording part 3 begins here]

00:00:23 PE: So this is Paul Edwards again. We're here with Ed Davidson on the third section of the oral history. And today is June 18, 2009. Let's see, we were still talking about your PhD period at Illinois between 1965 and 1968. And I think yesterday we were, toward the end of the time we were talking about your advisor Metze and what happened with him...

00:01:03 ED: Right, Metze. M-E-T-Z-E. So my neighbor advised me to talk to Gary Metze. And I explained what I wanted to do and he said "That sounds interesting. I'd be happy to work with you. I can offer you a research assistantship" and I just switched. So I spent the first year with McCormick and that summer I switched, I think it was that summer rather than the following fall, I switched over to Metze. And Metze used to meet to all his graduate students for an hour, each one individually, and he'd talk about whatever you wanted to talk about. And he'd react, he'd listen and he'd go with you. And he was very creative and you'd say what's bothering you, what your problems are and he'd help you through it and make suggestions and he'd find ways to help you get done what you were trying to do. He loved working with students who came in with their own ideas. So the fact that he was young and heading up a reliability research area that he had just started on and wasn't fully up to speed in and inherited students from Seshu...

00:02:35 PE: This is reliability of what exactly?

00:02:38 ED: Failures in computers. What was the... There was a project that Seshu had called the Sequential Analyzer. It was a way to correct your faults and you know... I don't know the great details of it because I never got that close to that. But most of the students were there, the fact that a few outliers came to see him on random topics, you know, that just interested him. ...He didn't try to force us into the mold of what he had to deliver. The downside of that was he wasn't necessarily very good at deliverables and he didn't write the right research proposals or publish papers. I inherited all of that from him [laughter]. I loved working with students, particularly loved working with students with their own ideas. If something sounded interesting I would do it. I remember when I was at Illinois on the faculty, years later, and when Dave Hilter walked into my office and said he wanted to do a Master's thesis on Ethernet, cause we had two incompatible computers and he thought it would be fun to hook them up together and they said "Gee that's interesting. What's Ethernet?". I'd never heard of it before. Turned out there was a draft spec for how Ethernet ought to operate and no description of the support, hardware, software for doing it. That was the state.

00:04:12 PE: At Illinois?

00:04:13 ED: No, when I was on the faculty later. This was...

00:04:17 PE: Seemed awfully early for Ethernet.

00:04:19 ED: yeah. This was probably mid-70's,'76 something like that. Okay, so we're back in wherever we are, '65, '66 by this time. So I always worked with students in that mode, meet for an hour a week, you know, less if it doesn't take that long, more of it takes longer. Don't meet in groups, occasionally we'd have a, you know, group seminars and stuff like that, but one on one. Socrates said you could only teach one person at one time. And you know that way they were given full attention. And for a long time when I was the faculty member, I had graduated more PhD's than I had publications. Not smart. Not a good strategy for an Assistant Professor, but you know, that's how it was. Once I understood how something worked I lost interest in it, I was ready to move on. I didn't want to waste time you know, having to write it up.

I'm reminded of that famous cartoon on a roll of toilet paper that says "The job isn't done until the paperwork is over," but I hadn't internalized that yet. So working with Metze was a dream come true and the student group had a lot of fun sharing offices and talking with each other. Also the lab was air conditioned which was great, 'cause on Sundays and the summer time when young families would come into the lab and read the Sunday comics in the papers and sit in the lab because it had an air conditioner in it and where we lived didn't. So, I was working on optimizing NAND networks, NAND is all caps N-A-N-D. I think I started with an idea that I could do any mix of logical functions, because I still had my memory of the complete sets of operators from Art Burks [laughter] but we quickly dropped that. The adding different logical elements and mixing them together increases the combinatorial space of possible solutions very very greatly. So, we cut it down to NAND. NAND was complete then, we were in an era of late small scale integration, early medium scale integration and people were doing combinational logic with NAND gates. That's what it was. Some of them....

00:07:06 PE: When you say small scale medium scale integration, you're talking about the microchips?

00:07:10 ED: Yes.

00:07:12 PE: So, in that early period of the integrated circuit era — in the [microchip] sense, because there was another sense of integrated circuit before that which really meant co-op...

00:07:19 ED: Transistors...

00:07:20 PE: a module. No, they were... I can show you one sometime. They're big, big clusters of components. They were all hooked together on a kind of miniature plug board and the whole thing could be inserted and pulled out.

00:07:35 ED: Ah. Well, I sneered at that kind of thing because that was just a packaging problem. One of my students, Phil Ember at IBM, [chuckle] after he'd had enough of, at least for a while of all the architectural stuff, discovered packaging. And packaging is in fact a very, very critical way to optimize the machine, often ignored to one's peril. But that's an aside, yeah.

00:08:08 PE: So anyway, you were thinking about medium scale integration?

00:08:11 ED: Well I wasn't actually. I was on small scale, because I was really going to minimize combinational logic functions using NAND gates.

00:08:23 PE: One of the things I'm curious about — reading through your work made me interested in this — what level of interaction did you have with people who were actually designing chips? Physical chips, as opposed to the logical structure of it?

00:08:37 ED: At that stage, none. At Honeywell, they had a circuits department which did, they didn't design chips, but they designed the circuit boards. And they had a logic design group which was what I was in and they had the circuit group and they had a peripherals group and they had a systems group. And the systems group they wrote specifications. They were the elite. However, they appeared to be the people who didn't know enough about anything to be competent and they wrote the damnedest specs. But at Stanford I began to run into a lot of the integrated circuits people because the Solid State group was the dominant group in the department. But I didn't interact with them much.

Interview recording part 3

00:09:33 PE: ...

00:09:33 ED: Well, yes. But not for me. There was a very very strong, from Bardeen and .. and the others, so an extremely strong group. But I didn't interact with them. I had my thing and you know... Which is a good thing for a graduate student or a young professor to do, is to focus. Do the best you can in your area. Put all your energy into that. A little cross-fertilization is nice, but it can... Intellectually it's important, in a long career, it's important. It helps relevance. But in terms of maximizing your bean count for promotion and career in an up-or-out system, it's better to focus. That's how the world works. So my idea was to take all the, you know, prime-implicants theory and all of that stuff and just throw it all out the window and don't use it whatsoever. And look at the combinatorial space of possible networks, and start building it up one connection at a time, using a branch and bound method. And so the cost of a partial circuit was bounded if I knew it would cost at least so much to complete it, that was a bound, and I'd explore all possible alternatives. With that you can start increasing the number of levels in the circuit, you can put in fan-in, fan-out limitations, which the fundamental theory can't deal with. There was one method for multi-level which Eugene Lawler had done, but the computational blow-up when you went to the third level was huge. And you go to the fourth level and it's just out of sight. So I thought that maybe the space of possible solutions, using this way to constrain them, might be smaller than what you'd have to go through to find prime implicants and select them and put things together. So it was an alternative, didn't know the answer but it looked interesting. So we explored that.

00:12:04 ED: Well now I had to write a program. And although I had taken programming, I wasn't particularly good at it. We had a computer in the lab which was a Control Data CDC 1604B. It was a long time before I knew that Control Data actually never manufactured the 1604B. They did a 1604 and a 1604A, and we had a engineer on staff who managed the computer installation. If I recall his name was Ernie Neff, N-E-F-F. And he took a field service training course at Control Data, he was licensed to be a field service engineer but he was employed by our lab, Coordinated Science Lab. He was an engineer's engineer, a putterer, he was our own kind of guy. So when he was bored, which he often was, because he didn't have an awful lot to do, he would look at the machine and say "Gee, I could optimize that circuit a little bit." He'd start rewiring circuits inside the machine, which you could do with wire wrap in discrete packages you could slip in and out. And he minimized, minimized and rewired and packed it in and pretty soon he got a couple of empty slots. So he ordered some new spare cards and he'd add a feature that had never been part of the original machine.

00:13:37 PE: So you had a custom computer?

00:13:38 ED: Well not only a custom computer, but a custom undocumented computer. Which was very interesting. Illinois also developed a language called ILLAR for Illinois Assembly Routines. I don't know who created...

00:13:54 PE: ILAR?

00:13:54 ED: ILLAR, because it's Illinois. Unlike the Rose Bowl games that I went to, when the band was marching across the field spelling Illini, it was called the marching Illini, and this line goes up and down the rose bowl in Pasadena and this lovely young California couple sitting in the row in front of me, the date turned to her date and said "I thought there was an S in Illinois".

00:14:28 PE: [laughter] Right.

00:14:30 ED: But there are two L's. Where were we?

00:14:36 PE: The computer, CDC 1604B...

00:14:40 ED: Yes. ILLAR. So I didn't know the machine language for the CDC, but there was a guy on staff, Chuck Arnold who was employed and he helped people program. So what'd he do was he wrote a series of low level subroutines for me. Following the spec that I laid out, so they would do like AND operations and OR operations and various combining of strings of bits which were basically parts of truth tables. And then I could program in as if it were a language basically calls to those subroutines and build the branch and bound algorithm and I was home free. That machine was a transistor machine of a vintage where computer operators still had a... Old computer operators still had something of a fondness for the old relay machines which made noise. And you could hear how the machine was working, which helped you stop problems when things went awry, which they often did. So...

00:15:51 PE: Because it would start to make a new sound.

00:15:53 ED: A different sound. Yeah, and you can hear an endless loop very well. You know, it could be a rapidly converging loop that's just going through many iterations, but you know, short iterations. But you recognize these sounds. Uh-oh, the last time I heard that sound something bad was happening, and you go fix it and start over. So I had one of those speakers in my branch and bound algorithm I could hear the algorithm work. And you know, it would go 'b-deep, b-deep, b-deep', that's hard to type isn't it?

[laughter]

00:16:22 ED: The various sounds would start to speed up. It was like the computer was getting excited and that meant that it was doing fast finish-up steps. And it was about to emit an improved solution. And it would go very, very fast and then pause briefly and then you'd hear the printer go and the new solution would start coming out. Then it would go back and start back-tracking and...

00:16:51 PE: That sounds really exciting.

00:16:51 ED: It was fun.

00:16:52 PE: Crescendo, you know?

00:16:53 ED: It was like, yeah, it was like, alive. It was like having a conversation and things worked swimmingly. So then I started to put in larger functions and more variables and play with wider, you know, less constrained solutions and so on, and try to find stuff. I quickly found the optimum solution for a full adder, the eight NAND gates solution which was a cute circuit. It had been known before but it, I don't know if it was ever known to be optimum. So we're doing a couple of things that really improved. One interesting thing is that once I went to functions of a large enough number of variables, I believe 4 or 5, suddenly very strange things started to happen. It would print out “there is no possible solution,” and that's absurd because there's always a possible solution. And, you know, it would go through the whole search tree and there's no solution whatsoever. I wasn't limiting it in any way. And I started looking at what was wrong, I start tracing it back, tracing it back, tracing it back, and you know, if you draw a truth table, the columns on the left are for the variables. And they are a column of 0s and 1s for the variables and the left-hand variable has the half 0s on the top and half 1s on the bottom. Once there are enough consecutive ones to fill a word of the 1604, that string of 1s disappeared and got magically changed to 0s. And I wondered what the...

00:18:47 PE: Do you remember how long the word was on that computer?

00:18:49 ED: I don't. But it might have been 16 bits or 24 bits or something like that. So I... How could that possibly happen? So I start tracing it back, tracing it back, and it happened when the problem was input, first input. The variable truth table from the paper tape reader, which was the primary input mechanism. And, you know, I traced it back and the information coming from the paper tape reader... The paper tape was punched with all 1s and the information deposited in memory from the paper tape reader is all 0s. Turns out, the 1604 was a ones-complement computer and all 1s was the code for -0. The 1604 spec guaranteed that if you produce by some mathematical operation, a result of -0 the computer would automatically convert it for you to 0. Which of course, is what you want. Well, my word of all 1s was not -0, it meant that these elements were in the set. So it just wiped out that part of the set. And they did that, strangely enough, in the paper tape reader routine.

00:20:14 PE: Ah. Not in the processor?

00:20:15 ED: Not in the processor. Well, in the processor but in the driver, the I/O driver that drove the paper tape reader. Unbelievable! Who would have the mindset that the only thing processed in a computer was numbers? It's unbelievable!

00:20:34 PE: That's very interesting because, you know, in the earliest days of computers they were seen primarily as giant calculators.

00:20:41 ED: Yes.

00:20:43 PE: And, you know, by the time you were working on them though, it was pretty clear that they were simple machines that could do all kinds of other things.

00:20:48 ED: Well, we thought so. And it could. But you know, that was amazing to me. So, you know, it was easily fixed. And, you know, it was my first lesson in how the problem is sort of never where you're looking. The problem is not within your area of expertise, it's in the stuff that you relied on or did that was not your area of expertise that you wouldn't dirty yourself with trying to master or learn.

00:21:17 PE: That's what we were talking about yesterday. About the need to know something about a level below the one your working at and the level of...

Interview recording part 3

00:21:23 ED: Oh yeah. Oh yeah. And, you know, that came up years later when back in Illinois on the faculty when we introduced the microcomputer lab and people were really... Most microcomputer labs taught students how to program a microcomputer. We taught them how to build one. So we use a level of integration well below the state-of-the-art and they built the computer. We had one experiment where they designed the logic for a UART, Universal Asynchronous Receiver Transmitter, for controlling a teletype. So they had to know asynchronous logic design which nobody taught anymore. And making that work was all tiny problems, everything was tiny problems. I cried when they dropped that experiment. They said, oh well, we have UART chips; only two people in the world have to know how to design them, why don't we just drop that experiment. People don't know how to do asynchronous logic anymore. So there was that. When I completed my PhD my office cubicle was filled with printouts, floor to ceiling. I couldn't find the desk chair anymore. Then Metze came in and looked at it and said, "Gosh, what are you trying to do, write a history of how you wrote your thesis? Why don't you throw that stuff out?". [laughter]. I ended up publishing the thesis later under my own name without Metze's name on it. And the custom...

00:23:10 PE: This is the article I have here...

00:23:16 ED: Probably.

00:23:16 PE: “An algorithm for NAND decomposition and network constraints.”

00:23:20 ED: Yes, yes. The network constraints were added the summer after I finished the PhD, where I stayed at Illinois as a full-time researcher before moving on to Stanford. And that was the one article that I published on my thesis. I don't know when the publication date is but...

00:23:40 PE: It's 1969.

00:23:42 ED: Okay, it's a year after I finished. That's not bad.

00:23:44 PE: But received, manuscript received in 1968.

00:23:47 ED: Okay. So I published it under my own name without Metze's name on it. It was an act of total naïveté on my part. I wrote it, but he helped me write it. I did my thesis, but he helped me do my thesis; he guided it. You know, I had no idea that the tradition is that you publish with your thesis advisor. If anybody had ever hinted, that's how it would have been. Later, I don't know if I started the tradition, but other, you know, other people really picked it up... I think the good guys all picked it up. We switched to where the student's name was the first author and the faculty name is last author. Tradition had been that the faculty member was first author. Or worse yet, in decadent fields like chemistry...

[laughter]

00:24:46 ED: The lab director, who never had even seen the work, was first author. And we had one faculty member in Solid State who was first author on, I don't know, 150 papers. He didn't even read those papers. He was supplying material to 150 different research groups which published. Well, anyway. So I've often felt guilty about that; that was a mistake. I wish

Interview recording part 3

I hadn't done that. On the other hand, Gary did capitalize on the thesis. He was consulting for Texas Instruments and my algorithm, through that relationship, became part of the suite of design automation tools that Texas Instruments provided to their customers, designing with small-scale integrated circuits and they could use my thing. I must say, at their peril. Gary helped them clean it up and get quick sub-optimum answers so that it wouldn't get lost. So then I started interviewing around, if you want to go there, for jobs.

00:26:00 PE: Sure.

00:26:01 ED: Oh, there is, I could...

00:26:03 PE: Actually, before we move to that let me just ask you, and I realize it's a little bit out of sequence, but since you published this later, but you said you added the network constraints after leaving Illinois. But tell me a little bit about what that means. In the beginning of this article, it talks about the trend toward large-scale integrated circuits and that's really a motivator here because you say, “in the integrated circuit environment, network constraints will play an ever-increasing role with respect to cost criteria.” So, this is not network constraints in the sense of the internet or something like that.

00:26:36 ED: No, no, no.

00:26:37 PE: This is a network of processors.

00:26:38 ED: A network of... No, a network of logic gates. A network of NAND gates.

00:26:43 PE: Okay.

00:26:47 ED: I did that at Illinois, the summer after I finished my thesis. I got my degree in June and I stayed until... Through August. So, in branch and bound it's very easy to add constraints. When you're searching a part of the solution space that violates the constraint, you back-track. So, you just cut it off. So, you just periodically check for constraints and go away. So, adding fan-in constraints you can have a NAND gate but only with this many inputs. You can have an output... You can have a fan-out, you can connect the gate to only so many other gates. It's output can drive only so many other gates. Levels; you can use only a chain of up to three NAND gates. Three is minimum with NAND gates. Two is minimum with AND OR. But with NAND you've got to get the inverters on the third level, at least. But you could set a four level constraint. And if you squeeze fan-in/fan-out and levels too much, you can get an answer that there is no possible solution; it'll tell you that.

With LSI, the idea of designing with only NAND gates was... I won't say that's the where it's all going, but some people did that, some people had a mix. Levels was tight. Interconnect was the real, a real problem, because the interconnect area on a custom LSI circuit can be a real killer. The wires that connect one gate to another, as we talked about the brain and all. So, interconnect; the simplicity or regularity of the interconnect pattern was not something that I was ever able to address. That's a hard problem. Even now it's only done as a second stage. You do the best you can with what you've got. But it does teach you that you don't over-optimize one thing at the peril of the other thing. Something that I learned at Honeywell was over-optimizing the number of flip-flops and paying for it in combinational functions.

Interview recording part 3

00:29:23 PE: Part of what's interesting in this problem -- and I see this throughout the rest of your work, so I was going to bring it up at one point -- is this relationship between the logical structures you're designing and the cost, in the sense of cost per computation, or something like that.

00:29:40 ED: Yes.

00:29:40 PE: More than the money cost of the actual chip. But that's a really tight constraint on what you're doing. That seems to be the goal: to maximize efficiency with respect to that variable.

00:29:52 ED: Yeah, yeah. I have two thoughts on that, you know, I did like mathematics and I did like optimization and I loved Operations Research and various algorithms like that. I like geometry and spaces. So one thing you can do when you have this space of all possible solutions you can crawl around in that space. And you are inside it and you are living in it. And if you visualize it, you can make algorithms that crawl efficiently and you know what game is there to play. Optimizing is a game. And I intended to gravitate towards optimizing cost performance. The cost doesn't have to be dollars; it can be any cost function. What you pay for what you get. And how do you do that. I'm aware that it might be a bit obsessive. If somebody wants certain, I don't know who, but this is the purpose of an undergraduate engineering education: to teach you that it is not important to fully optimize anything. Very few things have to be fully optimized. There are some problems where the space is very sharply turned around an optimum, but a lot of problems are fairly flat around the optimum. And if you can do a near miss that satisfies other criteria, that aren't so formalized and you have that sense. You are actually a much more valuable engineer.

But if you work without understanding the space of possibilities -- which I admit is what most people do, they don't think about the space; they think about what they are doing at the moment, and they just do it -- you get into a real trouble. That's where disasters come from. You design an uncompetitive product or... But by not being efficient where it matters -- and if you understand the space, then you can begin to understand where it matters and where it doesn't, and work on what matters -- and also by focusing on what you are doing and ignoring aspects you might not understand as well, you know what's really important and when you screw those up you build a disaster. Referring back to the microcomputer lab that we did, that was the first lab where we gave the students keys to the lab and let them work weekends without a TA present.

00:32:46 PE: Undergrads?

00:32:46 ED: Undergrads, yeah. That was the administrator at Illinois, Ed Ernst, who was my mentor. He was Associate Chair of the department. He watched my back. And he let me do things that... God bless him. I mean, you know we broke all the rules. And it was wonderful. We created what is called a learning environment. We were given hard assignments, and the students did not have enough time in the lab. The lab had a TA and attendants for normal hours and students would drop in. I don't recall if they had regular assigned times in the lab. And we wanted to run a 24/7 lab. When they found that the students were dangling the building key off the window, in case there want a student in the lab that we didn't like and we tried to stop that. So, if somebody signed up for key, they were responsible, they could pass it to the next guy, but the next guy was responsible to them. They were responsible to me. As far as I was concerned they had the key. And it worked! I built a culture where they thought of themselves as the preferred and the elite and they loved it. Well, so some students didn't thrive in that environment. And they'd come in, and on Monday they will be just furious. They'd come storming in to my office and say, "I worked in the lab whole weekend and I could never make any progress. This lab sucks!". And I'd ask, "What did you do?" “We've been hanging a scope for some signal...” The thing they were designing was not functioning. They were trying to debug it.

And they look at the signal and they keep looking at that signal for whole weekend. That signal is not going to look different at 2'O clock and then looked at 1'O clock. And you know it never fixed itself, no matter how many times they looked at it. That's what I call the lamppost problem. And it's very tempting. You know somebody is walking along a street and you see a guy in the gutter and he is crawling around on his hands and knees, you walk past him and see what seems to be the problem. And he says, "I dropped a quarter". "Oh let me help you find it." So the two of them were crawling now for a while and not finding the quarter. It's under a lamppost where it’s lit up and not finding the quarter. And finally the helper says, "Where did you lose it?" And the guy says, "Well I lost it up there, a block up the street". He says, "Why are you looking here?". He says "The light’s better here". We look where there is light. And you can't. You have to get out of that mindset. Look where the problem is coming from. And some people have hard time doing it.

00:35:52 PE: Yeah. So, let's go back to Illinois. You have a few things, a few notes, don't know if you had a look at it. I don't know what they mean. Can't really interpret.

Experience with the Vietnam draft board

00:36:07 ED: Yeah. There are two topics that we could go to next. One is looking for a job after my PhD, and the other is my experience with the Vietnam draft during the PhD.

00:36:20 PE: Wow! Both of those must be very interesting.

00:36:20 ED: Well, let me do the draft; it was chronologically earlier. I basically was parlaying one type of draft deferment after other. When I was at Honeywell, when they wrote a letter to my draft board saying that I was a critical employee because I was designing computers for building submarines. And I think that's because Honeywell once sold an H200 to the Groton Navy base in Connecticut. But it worked. I got my deferment. That was after all my college. So now I was married and I was back at Illinois. A year after Ellie married me... Hopefully for love, I assume for love, but also for my promise of financial security -- a year after, of course, I quit my job [laughter]. And she was employed, supporting me through my PhD in addition to my research assistantship. So, I am married, I am over 25. They are not interested in married people, they are not drafting those people. So I let it lapse to 1A. One summer Lyndon Johnson decides he is going to take everybody who is over 25, married or not, with a 1A classification. Married or not, but without children. With a 1A classification, they have to take the Army's physical fitness exam to see how they are. So, I get this letter: report for an army physical.

Interview recording part 3

00:37:59 PE: So this was I guess, around 1965 I'm guessing?

00:38:03 ED: I would...

00:38:04 PE: Combat escalation in Vietnam was done...

[pause]

00:38:09 ED: It was when they were running out of people. And they were considering going deeper in the age pool.

00:38:15 PE: Might have been a little bit later.

00:38:17 ED: It might have been the summer of '66. That’s my guess. I didn't have to take any... What I didn't know at that time, but I soon found out, I didn't have to take any courses in the summertime. I had done my course requirements. I was doing full time research. And Gary was kind enough to put me on a 100% assistantship. And I didn't register in university, because I was not taking courses and I could save the tuition. University students draft office which was set up -- this was a kind of thing that through my whole career got me crazy at universities, and I fought it my whole career -- that office was set up to help advise students on their status with respect to military service. But Illinois is also a state university. And they are also an agent of the state. And this office is an agency of the state.

When I didn't take courses that summer, the draft office, without contacting me, wrote a letter to my draft board saying that I had dropped out of school and should be considered eligible for service. Bringing me to their attention. That hit right about the time when Lyndon Johnson was calling everybody up for a physical. So I get this letter: at 4 o'clock in the morning, I was supposed to be at the train station at Chicago to go take an army physical. Well, I freaked out. So I immediately wrote a letter to my draft board requesting a student deferment again, and I have to wake up at 2:30 in the morning or 3 o'clock in the morning to go to the train station and I get up and I am ready to leave and Ellie opens one eye and looks at me and says, "You can’t go looking like that. This is an appointment. A formal appointment. You have to put on a jacket and tie." I said, "For an army physical? You’ve got to be kidding me".

00:40:34 PE: First thing they'd make you do is take your clothes off.

00:40:36 ED: Exactly. Well, she was firm and she was also my source of social norms, I never had any idea what they were, and I was tired and I didn't want to fight although I was sure she was wrong in this case. So I wore a white shirt and a jacket and tie for my army physical. Well, it worked like a charm. While I was doing the urine test a confused other person taking physical started talking to me intensely about where do we go after this. You carry your valuables in a brown paper bag and you put it on a shelf when you are doing your urine test. So I put it on the shelf and this guy is talking to me and unbeknownst to me, his friend is lifting my bag off the shelf because I was conspicuous for having worn a jacket and tie to an army physical. So that's what that did for me. So there goes my wallet with my dollar and half in it and my 12 year old Timex watch, and my train ticket home [laughter], which is the only thing that mattered. So now I have to go beg the sergeant for another train ticket home. And he listened to me with great amusement. Then finally said," Okay, college boy, I'll write you a ticket home." So there was that.

The other interesting thing was the intelligence test. And they explained that there are ways of telling if you cheat. “Cheat” meaning trying to answer it badly. And when graded, if they could determine that you cheated, you would be inducted immediately. And there is no penalty for guessing wrong on multiple choice questions with 4 possible answers. And their passing score was 20. That means if you're random, you pass.

00:42:51 PE: Were there 80 questions?

00:42:52 ED: No 20%. If you are random you get 25%. it’s 4 choices, multiple choices. I could figure that out. You can't flunk. The way to tell you are cheating is if you get less than 25%, I suppose.

00:43:12 ED: Well, it turns out, a couple of years later I got my wallet minus the dollar and half back in the mail. Some woman, someplace in Indiana, she mailed it. Said that her son had been drafted and sent to Vietnam. She found a wallet under his bed and she was returning them. But she returned it from a post office box. That made me feel good. Whatever he did... He is in a sense in Vietnam in my place. I'm getting out of this. Well, within a week and a half after the physical, Ellie was pregnant with our first child and children were good for deferment. A couple of weeks later I get a letter from my Draft Board saying please stop writing letters to us requesting deferment. You are over 25 and you're... The problem with people over 25, they never drafted people over 25, except in World War II, as far as I know. And World War II they did not work out really well. They were old enough to be cynical and cant be indoctrinated very well. And they are in terrible physical shape. If you send them to the front they die quickly. So they are not very useful. Unless you need cannon fodder. So they never did draft that category. But...

00:44:34 PE: When was your first child born? That'll date this episode.

00:44:39 ED: That's interesting. Certainly, well. I had the week and half as an exaggeration, I do that sometimes. First child was born on August 5th, 1967. So it was the summer of 66. Exactly. So Ellie stopped work when Becky was born. And I had a year to finish my PhD. And I finished my PhD with 500 dollars left in the bank and headed out for a job.

Initial job search: Bell Labs, IBM, and Stanford University

00:45:07 PE: So how did you find a job?

00:45:09 ED: Okay. So, interviewing around, I had a variety of interviews. So at one point... I will talk about four of them, briefly. At one point, Metze came into my office, and said "You want to be a professor at Stanford?" and I said, "I don't know. I never thought about it". I had come from industry and I was going back to industry. And he said "Well, think about it. Ed McCluskey is going to call you on the phone". Well, I bloody well knew who Ed McCluskey was. And before I could exhale, [chuckles] the telephone rang. [laughter] You know, why say no to a chance to interview at Stanford? Of course you'd grab that.

I went to Bell Labs and I didn't like it. I saw some very applied groups, which didn't suit me, and I saw some very theoretical groups. And in one of the theoretical groups, the... I related to the applied groups more than the theoretical stuff. But the applied research wasn't very interesting. Theoretical stuff was very advanced. One of the guys said "Did you like writing your PhD thesis? I said, "Yeah, I loved it". He said, "Would you like write one a year for the rest of your life". I said, "I don't know about that". And he said, you are the wrong person for our group. Another person I saw there, took me into a room -- I was a pretty brash young kid, I was as cocky as hell -- took me into a room. There are 2 polished spheres, concave spheres 50 feet apart... Hemispheres, 50 feet apart in the room. It's got a light beam which is beading a Lissajous pattern. I don't know, some French guy I think. A Lissajous pattern, it's a close back and forth pattern that eventually closes on itself and repeats -- between the two circle mirrors. And he says, "This is the fastest memory in the world. It's a modulated light beam, on/off, and it's storing bits". As the light beam is traveling around, there are parts of space lit up and parts that are dark. And this thing can store... Two bits of information, something like that. Because light travels very fast. And he had built a Lissajous pattern that went back and forth a vast number of times in order to get enough distance... So the modulator could get enough bits on there. Because the modulator can only go so fast. Well, it’s perhaps a very clever theoretical invention. I am an engineer. I look at it... And he is very proud. Little nasty snot that I was, "I look at it, and I look at him and said "Wow. I see that it is very large and it must be extremely expensive. And it doesn't store very much information. Does it have any other advantage?".

[Laughter]

00:48:55 ED: And that terminated the interview.

00:48:57 PE: I like that.

00:48:59 ED: That was the end... That was me and Bell Labs. At IBM I met Dick Karp, IBM research at Yorktown Heights. And that was a thrill. I knew who Dick Karp was. He had worked on the traveling salesman problem for example. He had done combinatorial logic minimization. He had done arguably better work than I on any problem I ever attempted, which I only found out after I had attempted the problem. He hadn't read my thesis, he hadn't yet seen my seminar. He got a few words from me about my basic approach and he sat back in his chair and he said "Well, you seem like a clever fellow. I bet that you can optimize all three variable functions, because that’s really trivial. If you are extremely lucky, you might be able to do all 4 variable functions. And there is no way in hell that you can do all 5 variable functions". And I said." How did you know that?". He said "Well they blow up in the combinatorial space. It’s immense. It's an exponential of an exponential. It's just... " And if you're dealing with any kind of combinatorial function, no matter how intelligent, you can't do it. He was exactly right. Particularly the symmetric four variable functions, he couldn't do. Because we kept rediscovering the same thing. The other thing is once the function got sufficiently complex. He would say "Oh, I have to design this function at this place in the circuits". So it looks like it's going to be expensive, so I'll just throw an inverter in and postpone the problem for another level. And that made it look locally cheap, but globally expensive because what you are doing in the last stage of the back-track algorithm is exploring an infinite chain of inverters until they violate the cost bound. Well in a large combinatorial space, you can't do that on every leg of the circuit.

He knew this in ten seconds. I took 3 years to find it out. It's interesting; knowledge is power. He also said to me that IBM research is fabulous. It's just like a university with one major advantage. And he leaned over me and said "There are no students". And I was thinking no students. Is that an advantage or is that not an advantage, because Gary certainly seemed to enjoy the advantage of students. And you know I thought I would. Well sufficed to say that semester after that Dick Cark left IBM to visit Berkeley and he never went back. Students are not a disadvantage. Students are what it's all about and that is the greatest joy in the world. I interviewed at Litton Industries in Los Angeles where a friend of mine from Honeywell worked, Bill Mao. Bill's father had been one point, head of Chiang Kai Shek's Air Force. His father represented Nationalist China in the UN for a while until he got caught on the wrong side of some scandal and was summarily eliminated.

00:52:49 PE: Can you spell that?

00:52:49 ED: M A O. Bill had also quit Honeywell, as all the good people did. He did a PhD at Purdue. And then went to Litton. So I got to interview at Litton and the most likely match was the guy with a lot of intensity and a gleam in his eye, and he is designing a massive parallel computer for pattern recognition. And he is explaining how it's going to work. I should remember the name of the computer, but I don't. It might be an ASP, I am not sure. There was an ASP computer, it might have been something else. But if that was a Litton computer that's probably it. So he is doing pattern recognition. He is explaining if you have all these parallel computing elements connected together, you can do a great job on the pattern recognition problem. And I am not getting it. And he is describing it in very general terms. Litton, of course, is a major defense contractor in Aerospace industry. And he says "Well, look... " and he takes a piece of paper and he draws two parallel lines and he draws a rectangle at one space spanning two parallel lines. He says, "You don't have to do perfect pattern recognition. You can do simple parallel recognition. And we don't want false negatives". We don't want to say that's not the pattern we are looking for, which means that they are looking for a particular pattern. We don't want to say that's not the pattern we are looking for, if it possibly is. He says, "If we see a pattern like this the two parallel lines with the rectangle across it, you don't know if it's a train or not. You can guess it's probably a train. So you just bomb it anyway".

[laughter]

00:55:02 ED: And I'm thinking "Oh boy, this man is a lunatic; I don't want to be here". And then I look around and see the other people and they are all dark, cloudy people and they scurry around and they don't talk to each other. And I said to Bill, "You know, not only is this project something I don't want to get anywhere near, but people don't seem very happy here. There is no sense of community or common purpose, nor they do any kind of group, and there is no socialization." I don't mean parties, I mean no relationship between people, "How does anybody work like that?" And Bill looks at me. He said "Time to grow up Ed. That's not what work is about". Well that was for me then and always what work is about. It's networks with people and what you do with other people. Isolation is not a very happy state and it's not a very productive state. I hate that.

Interview recording part 3

So my third choice was Stanford. Ah... IBM, Litton and Stanford were the best of category. IBM was the best industrial research position, and Litton was the highest paying job offer. IBM offered me, I think 16000, and Litton offered me 17 or 19 thousand. Bell labs offered me I think 8000, they really didn't like me. And I didn't like them.

Stanford offered me 12000 with the possibility of 16, if I had research money to pay myself over the summer. So money wasn't a big issue. But Litton was a top dollar offer, and I really didn't want to go there. So rapidly came down to IBM or Stanford. Ah... I didn't talk about my interview at Stanford, but I can come back to that. So what I decided, naively, is that I'd been in industry before and I knew what industry was like, I knew what working in industry was like, IBM was clearly going to be better than Honeywell. And it was research, it wasn't designing product and it was a company that actually did something useful and was going to stay in business. There was something happening, a very exciting place, brilliant people.

I didn't know what it was like to be a faculty member. I never taught before. I supported myself from college tutoring wayward junior and high school kids in algebra and geometry and trig. But I'd never really taught. So I decided that I knew less about being a professor than being a researcher, and so therefore I should go to Stanford. Because if it turned out to be wrong, I could go from the unknown to the known. Knowing where I was moving toward. Whereas if I went to IBM and it didn't work out, and I decided to try it... Well, I wouldn't get an academic position [chuckle] in good place. But if I had, then I would be moving to another unknown which might be worse. So my mother's accusation that I had most revival instinct wasn't quite true. There was a method to the madness. My interview process at Stanford, there was a buzz, before I was giving my seminar, that Bill Kautz was coming to my seminar, from SRI [Stanford Research Institute]. He was coming across the street to Stanford for a faculty candidate seminar. Bill Kautz would never stoop to that.

00:59:17 PE: Does Kautz starts with a C?

00:59:18 ED: K-A-U-T-Z. An eminent switching theorist and probably eminent in other things as well. This thing rattled me because I had never heard of Bill Kautz, I had no idea who he was.

00:59:32 PE: Stanford research is better.

00:59:35 ED: And Stanford Research Institute was you know... What were they? Some lab across the street. I had no idea of their theoretical dominance. And sure now, I give my seminar, people are amused, it was different from what they had seen before.

00:59:53 PE: Was the seminar on the NAND gates then from your dissertation work?

00:59:57 ED: Yeah. Kautz raises his hand as soon as we get into the question period. He wasn't coming for his edification, he had a mission. I didn't realize that. He is asking the first question on "Oh yeah, old guy in the middle, what do you want?". He says "Do you consider feedback?". "What do you mean?". "Well, do you allow a cycle of NAND gates connected to one another?" and I said "No, no. I am designing combinatorial NAND networks". And he said "Yes, I know. Do you consider the class of combinatorial networks with feedback?" and

Interview recording part 3

I said "No. If it has feedback it is sequential, it has memory." He said "I beg to differ. I have many combinatorial networks with feedback and they are rather good." And I said "Have you ever published that?". He said "No." "Where are these?". He said "They are in my desk drawer." I said "Well, I'm sorry, I am not familiar with that. I didn't know that that possibility existed." So, that was the end of that conversation. But I wasn't so stupid as to flatly... I didn't bluff. And I never... Well, sometimes I bluff, but that was for political reasons, so. But for technical reasons I never bluffed. It's wrong. And you get caught. Like Mother Courage said about her sons, Swiss Cheese and Eilif, "I raised my son to be honest, because he is too stupid to be dishonest and get away with it." [PNE - ED's reference is to the Bertolt Brecht play *Mother Courage*.] So it is good to know what you can get away with.

01:01:59 ED: That night I stayed up until three or four o'clock in the morning. My major interviews were the next day and by morning, by three or four o'clock when I went to bed, I had a proof that no combinational network with a feedback loop could be optimum. And I showed my proof to Ed McCluskey and Allen Peterson the next day. And they seemed impressed. They liked the proof. It was only three weeks later that I... You know, it kept nagging at me. It was only three weeks later that I discovered the proof was wrong. It was correct, if you only wanted to design one function. You could break... The trick to a combinational network with feedback is that the actual loop is always broken. For any set and input values, the loop is broken at some place. Some place, there is a know zero, like doing A or A bar. And that breaks the feedback. It doesn't have memory. But where it is broken is different to different points in time. So the feedback loop is actually doing something. So what I did was I had my counterexample, which was a five NAND gate network that I showed was not optimum and then I showed that the principle generally applied. It showed the existence of a network and then I showed that you could prove it, that it wasn't optimal. And by including induction you could imagine that was always true.

01:03:54 ED: So, what I did, was I fed it into my algorithm once I got back to Illinois, and it did exactly what I said, it designed a network which was cheaper. And then I said "Ah gee"... It was several weeks later, I said "Ah gee, wait a minute. You know, this five NAND gate thing... " I can say what the network is. You get four NAND gates connected in a chain around. There are two input NAND gates and the input that is not part of the loop reading from left to right is A B A Bar C. And to get A Bar you might have an inverter with A coming into it. So that would be the fifth gate. So with that, you got five gates. So I threw into my minimization algorithm this five function problem; design me a network that produces these five outputs, the outputs of each of the five gates. And it came back, I don't remember the solution it came back with, but the optimal solution had more than five gates. So if you want all five of those functions, that five gate network with the feedback is the optimum. I saw that Bill Kautz was right. It exists. Not useful, but it exists. Then I fed in, you know, I don't give up easily. So I...

01:05:22 PE: I can see that.

01:05:27 ED: I fed in all the problems with four of the five functions. I said "I suppose you only want four of these five functions" and for any possible combination of four those five functions my algorithm generated a feedback-free circuit with no more than five gates. Usually less. Well, sometimes less, like, it has to have four, because it is four different functions. Sometimes four gates is enough, sometimes five, depending on what you pick. So that was interesting. You know.

01:05:58 PE: So this was in the next few weeks? Not all while you were doing your interview with Stanford?

Stanford University, Assistant Professor (1968-1973)

01:06:02 ED: No. I discovered that I was wrong a couple of weeks later and I... You know, I was interested in the fact that I was wrong. I was always more interested in discovering the truth and then being right. A characteristic that I would say most of my students have as well. And I like that characteristic. So I wrote to McCluskey and I said "Ah, you know, I had a slight glitch in my proof. Here is what I know now." So they offered me the job and I took it. And we went off to Stanford.

Interest and involvement in political events (late 1960s - early 1970s)

01:06:38 PE: And that would have been in the summer of 1968?

01:06:41 ED: Yes. The summer of 1968 is a summer that many people remembered.

01:06:47 PE: Well, I was just about to ask you, we talked a little bit about the draft, but we haven't talked about your feelings about the war and the civil rights movement and the other huge political movements of the period.

01:07:01 ED: I empathized with the civil rights movement, having come from the William Lloyd Garrison School, and you know, a black kid came once, to the school, a student and soon there were more and soon there were more and the neighborhood basically went black. When I was young.

01:07:23 PE: This is in Roxbury? Yeah right, so.

01:07:24 ED: Yes. So, you know, when Jews moved to Roxbury, Christians moved to Brookline and Newton. And when the Blacks moved to Roxbury in great numbers, Jews moved to Brookline and the Christians moved to Weston and Lincoln and points beyond. Well, I am exaggerating of course. But there was a time in my elementary school where I thought that Christians were black and Jews were not. I didn't... A black family moved next door, they were perfectly fine. I liked playing with them. What I didn't understand is this first black kid who came to school, who was wearing clothes with holes in them and he had pants with a zipper in it that didn't fully zip up. And the other kids used to do everything they could to make him miserable. I didn't understand why. I'd say there were sensibilities in that regard.

Look, Martin Luther King I thought was wonderful, but I never, you know, I never lifted a finger on behalf of the civil rights movement. I was in favor of it. And I didn't understand why King went anti-war. I said "Damn, he has got a good movement going with this civil rights thing. He ought to stick to that. What is he messing it up by going into another domain?" At that point, I hadn't synthesized that these things are related and the people who are dying in Vietnam were differentially the underclass and differentially black. And it's the same issue. We think that the draft dodgers were all these people who marched around in support of the peace movement. They weren't the draft dodgers. Most of the people who didn't show up when they were drafted, were people from the black ghettos, who never thought about themselves as part of the system and the system always had a hard time finding them. We didn't count them in the census and so on. So they were non-people. And so, they would just not show up. And that is where most of the people who didn't show up came from, and then there were these other people. What happened... I was against the war in Vietnam. It seemed like a horrible thing. I had trouble blaming it on Kennedy who started it for the United States. Actually, Eisenhower started the United States involvement with an argument...

01:10:11 PE: Kennedy just built it up.

01:10:12 ED: With an argument that we had to go there because Vietnam was an important source of the constant intent. It was a straight capitalist argument. The dominos came later. Kennedy had a problem... You know I am oversimplifying of course, pardon my ignorance. But you know, he really did have to prove to the country that a Catholic president was not beholden to the Pope. And here was this country with a Catholic leader, who was dysfunctional of course. And we had the domino theory by then and they assassinated him and put in a non-Catholic leader. From Diem to Ky with probably other people in between them. Didn't seem to accomplish anything. I had trouble believing the domino theory. And Ho Chi Minh had trouble believing the domino theory, which you can learn about if you read Robert McNamara's book about the war [PNE - this would be McNamara's *In Retrospect: The Tragedy and Lessons of Vietnam*]. McNamara is not exactly a raving liberal, but he certainly destroyed himself as Secretary of Defense. And the book is an apology for, you know, trying to restore his public image, which is difficult to do. But if you ignore that, there are valuable lessons. Like during war it is very important to always maintain communication with your enemy. The North Vietnamese, when they finally met with them, said "Don't you idiots ever read history? Don't you know anything? Vietnam has spent its entire history fighting the Chinese. How you can possibly consider us to be in a bloc with the Chinese is unimaginable. They are our historical enemy. We are not a domino. We are doing something else."

01:12:19 ED: McNamara also said that toward the end of Kennedy's administration, they were on the verge of withdrawing troops and ending the thing. That was before the Johnson escalation, before all of that. They almost had it worked out, and on the eve of going public with that information, Kennedy was assassinated. Now, he refused to speculate about who assassinated Kennedy. I try to avoid empty conspiracy theories, but I don't believe that Lee Harvey Oswald thought it up by himself. I believe there is a reason why Jack Ruby killed Oswald. I don't know if it was the mob getting their vendetta against Bobby Kennedy, or if it was the CIA and military industry deciding that they needed to keep the war going. I don't know what it was. But I know that Bobby Kennedy was deeply bothered by these questions until his death. Not that Bobby Kennedy was so pure, he was the legal counsel for Joe McCarthy's Committee [PNE - the House Un-American Activities Committee], if I remember [PNE - ED is correct; however, Kennedy resigned after 6 months in this role]. He had many incarnations in his life, the last of which was super beautiful Kennedy for President, which transformation really happened when Martin Luther King was shot. Bobby Kennedy, I want to call him the aristocrat but he was really just Boston Irish with a lot of money, marched straight into the heart of most... What do you call it, the most volatile black neighborhood and got up on the podium and spoke. I can't, I'm sorry, I can't do the line without crying to this day. [crying] He stands up in front of an angry mob of black people and says "I too had a brother who was shot." God, I need a break.

01:14:49 PE: Okay.

01:14:54 ED: So he said "I too had a brother who was shot and I understand how you feel" and there was a real bond. It was like lightning struck at that moment. Did he do it cynically for this campaign? Maybe partly, because Bobby was many things, all simultaneously. He was the hard-knuckle strategist for the Kennedy administration. He took no prisoners. Did he have a personal transformation? I don't know. It's a great mystery. We'll never know.

01:15:29 PE: It sounds like you were well aware of all this at the time that we are reflecting on.

01:15:36 ED: Oh no. At the time... At the time... So in 1968 I'm driving across the country to my job at Stanford, circuitously, but we're getting there. Ellie and I are in the front seat and Becky is in the back. We bought a platform that lay on the backseat and hooked onto the back of the front seat. So it enclosed the space. Basically built a plate out of the backseat area. There were no carseats and there were no seatbelts. Becky was rolling around on her platform, playing with various things and having a great time, one year old, as we're driving across the country. And we thought nothing of it.

All the way across country we are listening on the radio to the Democratic National convention in Chicago in 1968. And at night you check into a motel and we watch it on television. And by the time we got to California I am ready to kill. We had been for McCarthy. But McCarthy was against the war, and Humphrey was not crossing the Johnson administration. Being the Vice President, loyal as he was — he probably would have been against the war [otherwise]. I didn't vote for him in the election. I couldn't. It was the third party candidate back there I think. Did McCarthy run as the third party eventually? I don't know. But I didn't vote for Humphrey, I didn't vote for Nixon. Nixon? Yeah.

01:17:28 PE: Yeah Nixon.

01:17:31 ED: So I was radicalized by the time I got there. Now I'm at Stanford, and Stanford is coming apart. None of this entered my consciousness at Illinois. I was doing my thesis and Illinois was in the Midwest. It was a relatively placid campus; it got active later. Stanford was already at boil by the time I got there. And when I first got there, I wondered... Oh, Stanford would give me a 5-year contract which was very unusual then. Junior faculty used to get three year contracts, and then get renewed for a new 3 years if they were doing well. And then get reviewed for tenure. So 5 years sounded great. Joe Petit was Dean of Engineering. He met the incoming faculty and said... This was his greeting speech, before the semester started. He said "There two kinds off universities in America now. Those are in deep financial trouble, and know it. And those that are that don't know it. And Stanford knows it". Stanford was, in my words, Stanford if anything, they were very consciously and professional managed as a business. They knew where they stood, nothing happened by surprise. So he said "We are about to enter a thing called BAP. BAP, the Budget Adjustment Program" and he...

Interview recording part 3

01:19:17 PE: One of those quiet phrases...

01:19:18 ED: Yes.

01:19:21 PE: ...about something terrible.

01:19:21 ED: He bypassed the usual speech, look to your left, look to your right you know, one of the three isn't going to be here in few years. He said "Unfortunately, none of you will be promoted to tenure. But we're glad to have you here and this will be a very valuable experience in your life". I was young, with a young family, but I am entering my career, and I'm now a faculty member at Stanford. And I said "Great! I've 5 years to do whatever I want to do. I am free!"

01:19:33 PE: Because the constraint of trying to achieve tenure would have been...

01:20:08 ED: The pressure is gone. I can do whatever I want. Well, that's the stupidest idea in the world. And as I have explained from my learning experience to junior faculty since then, whatever you think the odds are, here, wherever I was, what you want to do is this; because this is what you need to do to get tenure. And if you don't want to get tenure here and stay here, if you want to move to someplace else, or if you don't think you're getting to tenure here, what you have to do to try to get tenure here is exactly what you have to do, you need to get a job someplace else. So you just do it. And don't think about it. I think good advice. Nobody gave me that advice.

And Vietnam was at a boil. So I was in sympathy with the campus radical movement to some extent. I didn't throw rocks through windows. But I went to meetings and rallies. I knew the leaders of the movement. When they attacked Engineering... Now I have something of a conflict, because I am an Engineering Professor, and they're attacking Engineering. Bruce Franklin was a English professor who rallied the crowds knowing very cleverly where the line was, where he wasn't inciting to riot, legally, but he was. One of his favorite phrases which has to be spoken very, very fast without drawing breath is, "Stanford Engineering is the breeding ground of the expansion of the American imperialistic empire into the Pacific Basin." ... In many ways World War II was the physicists' war with the atomic bomb. And that split the physics community, I would say in half, I don't know if it's 50/50, but it really made a split in the physics community, which exists to this day. Vietnam was the engineers' war. We had engineering technology, we had the cluster bombs, we had napalm. which is more chemical then engineering I guess, you know. And we loaded our soldiers with so much equipment that they couldn't move. They were target practice after that, they were weighed down with the stuff. My God, you know, what is the purpose of that equipment?

01:23:02 ED: Seems like the purpose is to blow it up and make money. It's a technology war. The napalm and the cluster bombs are designed to maim civilian populations, they're not particularly civilian in war exactly. But they terrorize the civilian population. Strangely, Vietnam, I would say did split not the engineering community. It didn't rise to that level of humanity where there was a split in the culture of engineering. And I say that with a bit of shame in my profession. We had people in Los Angeles working in the weapons industry, wearing buttons that said don't knock the war that feeds you. So one day I'd be marching in a peace rally, and the next night I met with a group of volunteer faculty on the roof of the building that I worked in, watching for crowds and students who might be coming to throw rocks, so that we could call the police. It's a little bit schizophrenic.

The most militant act I did was really not very militant. I once drove a car to a blockade of Moffett Field [naval] airbase [PNE - in Sunnyvale, CA]. And I was in the front row. Front row since I was in a vanguard, in fact that’s where the chickens are. The job was to block traffic so that the marchers could block the gate. People are trying to get to work, trying to create a traffic jam at the gate. You want cars, your car's in front because if you have marchers with the workers’ cars behind them they might choose to run over the marchers. So we were protecting the marchers. So we get to block the gate with our cars, when the policeman comes over and says, catchphrase of the day, "Do you intend to be arrested?". You can either say "Yes" and stay there, or "No" and leave. And I was one of the ones who was, by pre-assignment, going to say "No" and leave. We had our choice, within the movement. So I did that. But I blocked the gate and was gone. I was in a crowd when Richard Nixon gave his reelection speech in San Jose and taunted the crowds afterwards by getting up on the foot of his car waving peace signs at the crowd.

01:25:57 PE: That’s famous, yes.

01:25:58 ED: And then claimed he was stoned on the way to the airport, not on drugs but by people throwing stones. I saw that crowd. That crowd was holding flowers and carrying candles. I know what an angry crowd looks like, I've been in them. That wasn't an angry crowd. That incident was made up. And it wasn't there where the crowd was. The dents in his car appeared halfway between there and the airport. He drove off the road and they banged up the car and then he went to the airport and said what happened. Ridiculous. He was taunting the crowd with his two fingers in the air, and he wanted to provoke violence and he failed. Somewhere along the line I had a student named Mike Aymar, who went on to a very successful career at Intel. He wasn't a grad student of mine, he was taking the course. And he came to me and he said "We need to start a symposium where the campus radicals and the power structure need to talk". He's an undergrad student. I didn't know if he was ever particularly involved in politics before or after that. But this came out of the blue and I said, “Okay, why are you talking to me?". “Because I want you to run it. I can't do it as an undergrad, they won’t come.” So I said okay. And we started a thing called the Science in Society Symposium.

01:27:49 PE: Did you realize I taught in the Science Technology and Society Program at Stanford?

01:27:53 ED: I did when I looked at your resume. I think yours might have been different. Yours might been legitimate.

[chuckle].

01:28:02 PE: The Program?

01:28:02 ED: Yours was the degree program, right?

Interview recording part 3

01:28:06 PE: Well yeah. It was... You'd get an undergrad degree in that, yeah. Anyway, there is a long heritage there, it goes back to this period. So I'm...

01:28:19 ED: Really?

01:28:16 PE: Yes.

01:28:17 ED: Well, we were not going anywhere with that. I'd be very interested to know anyway. It wasn't a course. It was a just a thing. I'd very interested to know if the roots are common, that'd be amazing.

01:28:31 PE: This whole... There's a bunch of programs called Science, Technology and Society, that all formed around that time. There was one at MIT, one at Stanford, others in other places that started with engineers principally. Exactly in this way! There were social issues, political issues, and they felt like they had something to say to students that wasn't part of the curriculum they had.

01:28:57 ED: Exactly. And to open that discussion in engineering is a difficult thing. Engineering is a trade; most of the students are in it to get a job. And they not trying to... All summary. Let me know if I overgeneralize. But there is a certain heads-down-do-your-work attitude. But not among all engineers. So we invited the radicals and the leaders of the radical movement and more towards the semi-respectable ones, not the foaming-at-the-mouth ones.

01:29:37 PE: And this is... Do you know which year we're talking about here? Probably wasn't immediately after...

01:29:41 ED: No it wasn't, it was...

01:29:46 PE: 1969, 1970, 1971.

01:29:46 ED: it's got to be '70-71-ish, I would think. I don't know if it was 1972, my guess is '71. Yeah, it's probably '70-71. Because I got involved in other things later. But among the people who came were Kenneth Pitzer, who's President of Stanford, Ed Ginzton who was very very young, Carl Djerassi, inventor of the first oral contraceptive .... And the radicals whose names I've now forgotten, but some of the student radicals and community radicals. And we had a topic for each, I think we ran four of them. We were drawing audiences of 200-250. I was forced into the position of being the moderator. And I have a rather loose hand as a moderator, I let people go at each other. And there was direct questioning and cross, back and cross in the panel and we had cultural differences in the panel. Carl Djerassi's daughter was Pammy Djerassi who did a poster in my house a long time, labeled Pax Americana. It was a beautiful... She was an artist. It had a bird and it was red, white and blue and looked fairly patriotic and it was a dove of peace. And you had to look fairly closely to realize that the blood splatter pattern was pretty evident. And it basically is a dead spread dove with a circle around it and Pax Americana written on it. Pammy Djerassi eventually became very depressed and died of an overdose of drugs. Carl was a sensitive person. He was an... [chuckle]

[pause]

Interview recording part 3

01:32:30 ED: He was an Austrian Jew who had escaped from the Holocaust, and he saw many sides of many issues, and he wasn't directly in the military industry. Ed Ginzton also spoke very intelligently, talked about the Varian brothers, one whom had been a bush pilot and the other had been a physics professor, both of them were very intelligent. And during World War II they made the klystron tube which was an essential tube for radar. And one day... Oh and Varian became very big, the military loved them and kept buying klystron tubes and Varian kept getting bigger and bigger. Even though they started by buying a chicken farm, figuring that they could eat the chickens if the first contract ended and they didn't ever get another one. So, one day they go to Washington, to make a pitch and the Defense Department tells them we don't need any more klystron tubes; not now and not ever. Technology has evolved , the war's ending and we've got enough klystron tubes. And Varian says "What are we gonna do?" And the guy in the Defense Department said "I don't care what you do. That's your concern." "Well", Varian said, "You've encouraged us to grow and be big and you said you needed our product and on your word we got bigger and bigger and you've a share in that responsibility" and the DoD guy looked at him and said "We are not a charitable organization. We run wars." And Ginzton said "Regardless of your morality, that is the reason why companies need to be diverse. Because if you live on one customer, you're vulnerable." And that's when they learned that lesson. And that was why they diversified. It was not from any strong moral conscience.

01:34:47 ED: So the dialog was interesting. And we had optional spaghetti dinners at people’s houses afterwards and it was very 60's. You know, it was nice. It was a beginning of a conversation, it was an edgy conversation, but it was a conversation. And it was probably a glimmer of appreciation of each side for the other. Or not, I don't know. But it seemed like it. Right about that time I met John Gardner who started Common Cause. And he was coming through and Elli and I invited him to meet him. And we became the mid-peninsula organizers for Common Cause. To put it together we did a... And the purpose of Common Cause at that time -- it exists today -- they were trying to get subscriptions of, you know, 250,000 people and they thought they could do... cause enough trouble with that. And their primary goal was called Open Up The System. Daylight laws. The business of government should be done in public. And the good things would come from that. And they were not endorsing candidates.

Well, our local representative was Garlic Charlie Gubser, G-U-B-S-E-R, from Gilroy, California. The garlic capital of the world. He was a garlic farmer. And he was a rock solid Republican, very hawkish and very corporate leaning. And I didn't like him. But he was considered untouchable, he won by huge margins. So we ran a Common Cause signature ad before the campaign, just before the campaign, when he's running for re-election. And we were on the edge of the rules, but it was okay because we were not endorsing a candidate and we were talking about process. The title of the ad was "Who does your representative represent?". And it was a series of paired quotes that we found, what he said, and there was a thing on what he said on an issue during the campaign, and what he does. And there was a series of votes, on related issues. And they were diametrically opposites, on one issue after another. All the issues that Common Cause cared about. And we got I think 1250 signatures for the ad, took out full page, and you know people paid small amounts of money and that was where goal... Large numbers of people paying small amounts of money. And we ran full page ads in the San Jose Mercury News and so on and printed the thing. And Garlic Charlie went... He voted that... He won the election. But he chose not to run the next time. So you know maybe we had something to do with that. We did a lot of work for Common Cause. When Johnson bombed Cambodia, we were near...

01:38:19 PE: Nixon bombed...

01:38:20 ED: Nixon. Yes. When he bombed Cambodia... They all blend together you know.

01:38:28 PE: [laughter] One bomb is like an another.

01:38:35 ED: The campus came apart. We had sit-ins and shut-downs and there are students who 30 years later saw me at a conference and said "You are the guy who led the class during that sit-in". They still blame me for it. They felt that's over the edge. John Linvill had a daughter, Candy. John Linvill was head of Electrical Engineering and he had a daughter, Candy, who was blind. And John's reading aid for the blind that he'd developed [PNE - the Optacon], she was one of the only people who was able to successfully use it. She was wonderful, brilliant and lovely and sensitive person. She was sitting in at the building. The daughter of the chair of Electrical Engineering. His blind daughter understood more about what was going on than he did. And I give her all the credit for it. Now, he didn't, I don't know... Whatever. So I crossed the picket line to go into class and chalk on the board "No class today" and I'm going honor the picket line and then I left. I explained that to the pickets and they said fine. They weren't pickets, they were sitting there blocking the entrance, they let me go in. And then I was talking with Mike. And we said "Why don't we try something. We are doing these bombs and we are causing all this havoc. Why don't we take up a collection in Engineering, among the Engineering faculty for medical supplies for Vietnam. Wonder what would happen." So we did. Inside of two weeks...

01:40:28 PE: You were sending to civilians in Vietnam? Or to...

01:40:31 ED: That was the question. There were two questions that people wanted to know. Faculty, who I had never would have imagined anybody could talk to anyone about anything like that, would sidle up to me and talk secretively and furtively in my ear "Who are you gonna send this to?". I had an answer for that "We are going to send it to the American Friends Service Committee." They were above reproach. Everybody knows that their medical supplies don't go to South Vietnam, they all go to North Vietnam. But they don't say that, they say Vietnam, it's for Vietnam. So there's that cover. You had deniability. You didn't know, but you knew. And the organization is, you know, it's not a radical organization. There are some people who call it radical, but it's a respectable mainstream organization. So, that's one question, how are you going to distribute the money.

And the other question is, can you absolutely guarantee that my name will never be exposed? They wouldn't get anywhere near it if there was any risk that their name would be exposed. Now with my reputation and my long hair..., and my attempt feeble as it was to grow a goatee, and long sideburns, and my identification with the radical movement, they still trusted my word. And I took great satisfaction in that. I was so schizophrenic in that period, you know. Being on both sides of everything. You know, during that period somebody would ask me what my religion was, my flip answer was “functional paranoid.” And you know, once we had assured them of security, they all gave money. So they had it, however much they would deny it publicly, they had it in their consciousness and they had it in their heart, and they were horrified and felt terrible about what was going on. Now the radicals can call names, and lord knows they deserve some of that, but these are not bad people. These are intimidated people. And someone said that you know, "Never underestimate how difficult it is to explain something to somebody when his livelihood depends on not understanding it".

01:43:26 PE: I've heard that quote before, that's a good one.

01:43:31 ED: And it's real true. I don't remember how much we collected but it was a lot of money in a couple of weeks.

01:43:36 PE: The quote is from Sinclair Lewis. I can actually give you the citation for that if you are interested in it. Because I heard somebody else say it recently, so I looked it up and told him.

[laughter]

01:44:01 ED: My recollection, it was somewhere between 10,000 and 40,000 dollars. It was a lot of money. It was... I felt very good about that. That was finally, I wasn't sitting around and talking, I was actually putting your money where your mouth is. That was something that mattered.

01:44:26 PE: I am not quite sure when the Cambodia bombing was, but my memory is after Nixon was re-elected in '72. So that would have been very last part of your time at Stanford that this happened.

01:44:37 ED: Yeah, he was elected... Yes, he was elected on a saying that he had a secret plan to end the war. The first thing he did was bomb Cambodia. So that was that. We can switch to the academics if you want.

01:44:56 PE: Yeah. Let's go to the academics. [chuckle] Maybe take another brief break. I'm gonna get another cup of coffee.

01:45:02 ED: Sure.

Teaching and research at Stanford

[Interview recording part 4 begins here]

00:00:02 Paul Edwards: Okay. Part 4 of Ed Davidson's oral history. So now we're gonna talk about Stanford teaching and research, and you were beginning to say, I think, that the curriculum at that time was completely unformed. You were in Electrical Engineering, correct?

00:00:22 Ed Davidson: Yes.

00:00:23 PE: Did the Computer Science Department exist? I think it did.

00:00:25 PE: Yes.

Interview recording part 4

00:00:27 PE: I think it’s around 1963 or something that it was formed at Stanford. It came out of Math and EE, like a lot of the early Computer Science departments. There were two lines in Computer Science, one from the electrical side and one from the logic side and math.

00:00:44 ED: Yes. Illinois was an exception. Their Computer Science, their Digital Computer Lab became the Computer Science department. So they had some mathematicians but they were primarily engineers, but ILLIACs. So they were very very different. Most Computer Science departments started from a math route and specialized in software and numerical analysis and theory of algorithms and so on. [The Stanford] CS Department existed, the head of which was George Forsythe, and he was asked at one point how Stanford went so suddenly from not having any CS department to having what arguably was the best Computer Science department in the world, so quickly. I think one reason is there weren't many Computer Science departments. But that's not the answer, because Stanford was... That department was just incredible. I was not part of that department. And Forsythe just said, “well it's not as difficult as you might think. All you need to do is hire the best faculty that money can buy and admit the top one half of one percent of the student applicants. And you are there.” Now that's not a luxury that the Big Ten schools can do. Illinois can.

00:02:25 PE: There are only a few places that can do that.

00:02:27 ED: Yes.

00:02:29 PE: And Stanford's location is a particular advantage in that respect.

00:02:33 ED: Well, yes. And Fred Terman... It's interesting, in these schools there are Deans, or you know, people started as Deans and people who were Deans at one point they moved up in the administration. There are particular people who put them on the map. Terman put Stanford Engineering on the map, and he knew how to exploit the land, which was Stanford's huge asset. You know people wanted to live there. It was an idyllic place to live. When I was recruited somebody from the AI lab drove me around the hills and he said, “you know it's very difficult to live here, because you have to plan a balance between your winter flowering plants and your summer flowering plants.”

[laughter]

00:03:20 ED: In your landscape.

00:03:21 PE: How awful.

00:03:22 ED: And I'd never heard of a winter flowering plant. You know, that and the fact that they had Napoleons at the faculty club for lunch made me decide that I absolutely had to be there. But you know, Terman wrote the ethos and he talked about excellence. There is a paper he wrote, a white paper that he wrote which was his Mein Kampf, as it were. And he said, how do you build excellence? And he talked about excellence, and you build steeples of excellence. What is an excellent university? It's a place that other universities say is an excellent university. And then he studied what did they react to? And what they react to is the number of stars in the faculty primarily. And it doesn't matter what percentage of your faculty are stars, it matters the absolute number. ...How do you hire the top faculty? You get into dilemmas like should you hire specialists in the Chinese art of some particular century a long time ago, the world's most renowned scholar in that, or should you hire, another example is a Nobel prize winning physicist, which was his example of the one you really want. And he's from Engineering, so he's not tilting. He didn't say Engineering, because we don't have Nobel prizes. The elite is the physicist. So he said you are asking the wrong question. It's not a dilemma; you hire them both. But you hire the physicist, you make him pay a third or quarter of his own salary from his research money, and you use that to hire the Chinese art professor.

[laughter]

00:05:30 PE: Have it both ways.

00:05:31 ED: So he built steeples of excellence. And very... You know you could say cynically but effectively.

00:05:40 PE: So apart from the question of horticulture, what were your first impressions of Stanford when you got there?

00:05:49 ED: Oh, it's such a vibrant, alive place. ... Oh Terman also built this Stanford Industrial Park, threw all that in. And loads of other places tried to emulate it, but they are not in such good a location.

00:06:05 PE: And they had the land.

00:06:06 ED: And the reason they had the land comes from Leland Stanford whose young son died on a trip to Europe and Stanford wanted a memorial for his son. He was a rail road baron. So the railroads had all this land. Stanford is one of the guys who drills the golden spike; he's in that photograph. So he went to Harvard and said you know "Can we name a building after my son"? Harvard said "Oh no, we don't sell our buildings". It's a different time. So he went on and started his university, and gave them all this land to have it, in the endowment. And Terman figured out how to capitalize on that. It's ironic that when I think of the railroad baron, his wife who was so sensitive. If you wanna get a good picture of her sensibilities you can go to Memorial Church, affectionately called MemChu by the students, and walk around the inside. Look at the quotes on the wall. They are all from her personal notebook of preachers of the day in it. And you know it's got stuff like you know, "Blessed are the poor for they shall win crowns and scepters in heaven that the rich cannot attain" and stuff like that. ... Some rich man guilt there, I think. But, be that as it may. Brilliant, Terman. At Illinois it was Dean Edward.

00:07:34 PE: Did you know him?

00:07:36 ED: Edward was Dean when I was a student. He was no longer Dean when I came back as faculty. So I knew him a little bit. A wonderful wonderful person. At Michigan there may have been early incarnations, but Jim Duderstadt was just amazing. When Shapiro was president of the university [of Michigan], Jim Duderstadt was the Dean of Engineering. And that was late. That was momentarily before I was hired. Shapiro was president when I was hired. He had gone to Princeton by the time I got here. Duderstadt was Dean of Engineering...

Interview recording part 4

00:08:11 PE: That keeps happening.

00:08:11 ED: Yes. When you're good, you're up for picking. And Duderstadt was Dean of Engineering, they made a hell of a one-two team. They built Engineering. I could talk more about that later, it's a matter of public record. I keep diverting.

00:08:30 PE: Stanford. First impressions, people...

00:08:32 ED: So there in the middle... That's where it's all happening. Everybody, if it's not actually happening in their backyard, everybody from where it is happening wants to come to Stanford. It's Mecca. So my five years at Stanford technically were phenomenal. And even though I believed every moment of those five years that I had no future there, I never regretted going there. It was a wonderful five years and a fantastic learning experience. I had a love-hate relationship with the administration, right down to the department head level. I even had some contention with Ed McCluskey. Although basically he was on my side, and I was on his side, and he supported me. But he was a...

He was a tough-love kind of Lab Director. He delayed my first research proposal for over a year, claiming that it should be rewritten to be coherent. Not the first one. The first one was just written, what they now call, a Presidential Young Investigator Grant. Then they just called it an Initiation Grant, you just get to continue your thesis which is all you can think of doing at the moment. But my first real grant, which was for pipeline work, he wouldn't sign off on it as Lab Director until I wrote it to be coherent. Something which I resented. Like my eighth grade English teacher who taught me how to write. He would mark me down one letter grade... He made us write themes of three paragraphs on one side of the paper and took off one letter grade for every punctuation error or erasure.

00:10:18 PE: Wow! [laughter]

00:10:20 ED: And so I was getting D's in English all of a sudden. And I didn't want that, 'cause I felt I knew how to write. I hated him. One day I wrote a theme on him and he called me up and we had a chat. The chat was that I misunderstood him and he was hurt. And he became a human being to me at that point. And I have grown to respect him very much. His name was Samuelson. I'm sorry, I'm digressing constantly.

00:10:59 PE: It is a long period in time and when we have a discussion like this, it’s gonna happen more and more if we keep talking all day. So, I'll have to think about when it makes sense to stop. But anyway. We were...

00:11:13 ED: Stanford was great. So I wouldn't say that they had no curriculum. They had a strong Computer Science curriculum, such as it was for the day. Electrical Engineering, McCluskey was there. He had been the director of the Computing Center at Princeton and a faculty member there and he came to Stanford and stayed and built the Digital Systems Lab and I was one of the first five faculty in there along with Harold Stone from SRI, Tom Brett, Forest Baskett, and a few later. ... When I left, I was replaced by two fellows who seem to have done rather well. One was Mike Flynn and the other was John Hennessy.

Interview recording part 4

00:12:11 PE: Yeah.

00:12:11 ED: Came from Stonybrook.

00:12:12 PE: Indeed.

00:12:13 ED: But it was McCluskey's lab and the rest of us were junior. Harold Stone being more senior because he had worked for a while after his PhD at SRI. So he was a non-tenured Associate Professor. Rest of us were non-tenured Assistant Professors. So we built what would later become a core of Computer Engineering discipline; it was just EE courses. And we had a partnership with Computer Science. My first semester there, I was stuck in a... McCluskey assigned me to a Computer Science course which was a graduate level Computer Science course in Computer Architecture. Well, I knew a lot about minimizing logic circuits, but I didn't know anything about architecture. I mean I built pieces of computers, as an academic discipline, I didn't. There's a good book, I don't remember what it was and I taught the course. There was a guy in there named Bruce Baumgart and whenever... And he's gone on to some note. And whenever people would... I guess there were advanced undergrad and graduate student in there. But hard thing to do if they day after you're a grad student. So when students asked questions and I didn't know the answer, it seemed like Bruce Baumgart always knew the answer. And at the end of my semester when I got my course evaluations, the evaluation comment that I remember most was, "Tell Bruce Baumgart to shut up".

[laughter]

00:14:07 ED: I always enjoyed his answers, because he, they were illuminating and he and I had nice conversations. And I remembered a philosophy course I took at Harvard where Jacques Barzun's son was a student in the class and TA spent all his time sucking up to Jacques Barzun's son and I didn't like Jacques Barzun's son. I guess I understood how the other student must have felt. The second semester I taught a course... I don't remember what the course was, but it was more in the heart of my area of interest and expertise, and it was in Electrical Engineering. So it was my first course in Electrical Engineering. There weren't semesters but quarters.

00:14:53 PE: Yes, I remember that. I hated that system. Don't know about you, but....

00:14:55 ED: There was another study that Terman did; whether the semester or quarter system is better. And they did a study and came out with, what I maintain today, is the very best answer to that question, which is painful and not observed, basically that it doesn't matter, that things can be divided into more chunks or fewer chunks and the quarter system gives students... You have multi-quarter courses where the students have an escape path when they made a mistake [laughter] or they've had enough. So they had more of them in a quarter system and fewer of them in a semester system, and you can repackage anyway. So it really ends up not mattering, but the important thing is to change from one to the other periodically. Because that's when you clean up the curriculum.

00:16:05 PE: Yeah, right. That's only happened a couple of times in my entire experience.

00:16:12 ED: It doesn't matter.

Interview recording part 4

00:16:13 PE: Berkeley did it once from quarters to semesters.

00:16:16 ED: Yeah. But Michigan went to trimesters

00:16:22 PE: It was gigantic. Yeah. Administrative...

00:16:22 ED: Oh it's a nightmare. And all the faculty's vested interests are getting gored.

00:16:27 PE: Yeah, yeah.

00:16:30 ED: You know we had a guy at Illinois on the faculty, Michael Louis, young guy. He would take a photograph of his class, where there was, there could a be 150 students there and then he'd demand that they all, on the first day of class take a photograph, demand that they all sit in the same seat for the rest of the term. But by the second lecture he knew everybody's name. Unbelievable. And I complained to one faculty member at Illinois that Louis writes his lectures two weeks ahead, and that's not fair for the rest of us, since I always wrote them a minute ahead. That was Arnie Dipert who I was complaining to, who was in transistor stuff, and he said "What's wrong with that? I wrote my lectures 20 years ago." [laughter]

So the second semester, I'm in this class and... I mean my own, I like this course, first lecture, a student comes, I catch out of the corner of my eye, maybe 10 minutes late and leaves 5 minutes later and I don't think anything of it, I just do my stuff. And I'm happy. I get back to my office, there's a sign on my office door. Says see me immediately, immediately — two underlines — John Linvill, Chair of the Department, Head of the Department, forgive me. So I go to Linvill's office and he says "where were you? when there's a sign, I sent you a message to see me immediately. That means immediately, not half an hour later." And I said I'm sorry, I was in class lecturing, I just saw the sign and I came over immediately. And he said, he went right by that answer and said "At Stanford when we assign a faculty member to teach a course, we assign you to teach a course, you teach that course. You don't send your TA in to lecture the course."

And I said, "I thought I just explained the reason I didn't get here was because I was lecturing the course." And he didn't process that and he kept going. I'm thinking what am I saying, why am I not communicating and I'm sure he was thinking the same thing. He wasn't listening to me, and eventually maybe on the fourth repetition, he explained that some student had gone to the class and saw some young kid up there lecturing. I didn't look my age. And came to the department to complain, he was paying the tuition and he deserved a veteran in his class and he was probably the guy that stayed there for five minutes. I would imagine so Linvill had the warning well before the end of the class. We parted friends sort of, but as I'm walking out the door, Linvill said
"okay, this is okay, but don't do it again." [laughter]

00:19:52 PE: Wow.

00:19:55 ED: Such were my conversations with John Linvill. It was amazing.

Interview recording part 4

00:20:04 PE: It was hard to get his attention, it sounds like.

00:20:05 ED: We started a logic design course... We did many courses, but we started a logic design course, a lab, and we found out at Michigan they had been using... I had never seen in Michigan. But Michigan shortly before they had a logic design course where they use these little plastic strips and put wires in them and you could stick a chip in there and you could bread board up a circuit very quickly because it had little spring clips that you need to wire and you got a circuit. You can have lights and switches and circuits and build logic. And actually see it. And you could... Ah, this was Stanford of course. So we mounted those little things in an attaché case which the students could carry around with them and wire up, and we put the course in. Payne Freret was my TA the first semester, I did that course. Payne Freret. Sometime after the end of the semester, Payne said let's do a capstone project at the end of the semester; the students will build a telephone dialer. That you set telephone numbers and switches and it would dial a... Basically rotary telephone by going click a click. And then he told the students, any of you who think it works write a program to dial my home phone number every hour, on the hour, all night long and he allowed them to that. And I will wake up when the phone rings and I will move a penny from one side of my night table to the other for you and I don't know how he knew which student was dialing and somehow he knew.

00:21:54 PE: Answer the phone. What was on the other end? Nothing.

00:21:57 ED: I don't know. Yeah, it was just a computer. I don't know how he knew who was who. But maybe they did it on different nights, I don't know. And not many of them got it working anyway, so it wasn't so much of a problem. And he kept track and if you got them right you get a good grade and there you go. So he had also built a clock. [chuckles]. The students loved it. I remember, you know, talk about a paper tape problem, I had one student, this is a Stanford student, he is a bright kid, you know Stanford has almost no undergrads. I think there were 50 undergraduates in class. It was not a Big Ten university in Electrical Engineering. A huge number of masters students who pay whopping tuition, that's the profit center. And then a bunch of the HT students. This is not a Big Ten university. So...

00:22:57 PE: The phone dialer.

00:22:58 ED: Yeah. So after the semester I get another invitation to John Linvill's office. And I think, "Ah, now he's noticed that there is a buzz about the lab course I created and how much the students love it. I'm going to get a pat on the head." Ah, you can guess that that was not the case. I got another lecture, in a fatherly way. He said -- and I think it says a lot about the value structure at Stanford -- he said “look, you are a junior faculty member here and you seem fairly bright and you probably should have a successful academic career and I just want to tell you that you are misdirecting your energies quite badly. You put your energy into starting a laboratory course, and worse yet, it'd be one thing if it was a graduate laboratory and it fed your research program. But it's an undergraduate laboratory. And those students who you train are going to go to some other university for graduate school and work in some other faculty member's research and you are just training people for that. And worse yet, labs are expensive, and you are costing the department money.”

So we used to have a phrase for the way in which Linvill ran the department, which was "the buck stops here". And you know money was to be spent on catalyzing research and hiring top faculty and I would say on the solid state lab which was his area and the beloved jewel of Stanford Department. This upstart lab was off teaching undergraduates and so on was not part of the picture. I mean lectures; lectures have been studied and they are deemed by education people who study such things to be the least efficient means of transferring information and knowledge. But they are bloody cheap. You stand up one professor in front of 300 students, you can't do anything cheaper than that in terms of filling time. And the information is there. Next to lectures, textbooks are about the worst. At least a textbook you can go to a page with the information that you want. Labs are extremely effective, but very expensive.

When we got to Illinois, we started the Computer Engineering curriculum, Ed Winston who was my mentor had a totally different attitude. And now it's a Big Ten university and serves undergrads, and undergrads are the children of the people and the taxpayers of the state. And that's what the state is paying for, for us to educate the kids of the tax payers whether those kids end up getting jobs in the state or in California. So that was our mission and he said laboratories are the loss leader for the department. That's where you do the work, where you hire, you attract the students to the curriculum, that's where the excitement is. That's where the rubber meets the road. But you don't make... The way the university budgets you lose money on every lab that you do, but that's your loss leader. That's the way he put it. Absolutely right. Absolutely right. He cared about education. And the state university, the state pays for education. So there's more of a balance of terror between education and research in a big state university.

00:27:12 PE: Interesting.

00:27:13 ED: So my accomplishment of which I was so proud, and Ed McCluskey was very pleased with, was written on my judgment book in the department office in red ink yet again. But in the meantime in research I got my continuation grant, and I had two graduate students first. Poor Sherman Lee had to work on yet another approach to NAND networks where he tried to merge the kind of combinatorial approach I was taking with some of the classic theory and develop optimal 3 level NAND networks. And he managed to do it and do some new theory on a heavily worn, you know previously worn road. Beautiful job. Ed McCluskey wouldn't let him graduate because his English wasn't good and the thesis wasn't cleanly readable. And people on our graduate student committee at Stanford had... The third reader on a thesis had a lot of weight; they could block a completion. Most other universities they don't care and the advisor just does it when he thinks the student's ready, and the process is corrupted.

Stanford was, everybody, all the readers read the thesis and they all critiqued it. So Sherman was held up on completion until a year after I left. He was already working at Amdahl. He was an Amdahl CPU designer for most of his career and most of Amdahl's existence. Very very brilliant guy. But you know maybe he was better off for it. Like I was better of for McCluskey delaying my pipeline research thing. The other early student I had was Ken Ouchi who was working with IBM research in San Jose and he had, we had some idea that the way MSI [medium-scale integration] was going, you could get more and more transistors in the chip, but the number of pins wasn't going up. So if you had a memory chip, maybe what you should have in the chip to get regularity of connection was instead of having a fully random access memory, maybe you should have shift registers. And maybe you could build a multi-dimensional memory, which were shift registers connected to shift registers. You could go two or three dimensions with that. And then the question is how many dimensions should you have and what length should the shift registers in each dimension be. Well that had to do... You could build all of these, they were nice and regular, easy to layout. The question was which structures would programs take best advantage of in storing code and data. So, he came to me with that topic and then Stanford was also a Joint Services university. So, you know, the great thing about those block prints is that you can pick up, like Metze did with me, you can pick up an outlier student with a bright idea, and support them right off. You don't have to wait a year till you write a proposal and get it funded. Which is wonderful, great way to do research. So Ken came to me with his idea, and you know, we did that. He did the thesis. So he was my second PhD student. He went back to IBM research labs and he holds the key patent on what eventually got called RAID [random array of interactive disks] disk technology, way before it was done in Berkeley. Unfortunately, I lost contact with him. I never saw him again, from the day he finished his thesis, and I have never been able to locate him. I have no idea where he is, and he doesn't pop up on an IBM directory. So I have no idea what happened to him. But he knew exactly what he wanted to do and how he wanted to do it, I was just trying to help how I could. It was interesting. Weird but interesting. It never happened, but a few articles were interesting.

00:31:36 PE: You made an interesting comment a minute ago where you said that McCluskey delayed your pipeline research?

00:31:41 ED: Yeah. Well, now the pipeline work which was a thing that I was, you know, still, I think most known for... There was a conference called the Lake Arrowhead Conference, which a group, which the West Coast area committee of the Computer Society. Ed McCluskey was the founder of the Computer Society, you know, creating it within IEEE. The Lake Arrowhead Conference was run by a committee where each year, they would... Well Lake Arrowhead is a conference center owned by UCLA in the hills. Rusted, double decker bunks and stuff like that. So the committee would pick two communities that don't necessarily talk to each other and announce that they were having a workshop that year for those two communities to get together and talk to each other. And then there would be some sort of papers and some huge amounts of discussion time. If you wanted to go to the conference you would have to write a paragraph on what kinds of things you would be prepared to talk about. And then they would decide whether you could come to the conference or not. Conference registration was something like 50 dollars, and they fed you and housed you and whatever. I mean, UCLA didn't charge much for the use of the conference center. And this year they were doing something on Numerical Analysis and Computer Architecture. [laughter] Two communities that don't talk to each other very much, okay. And...

00:33:46 PE: Remind me what year this is?

00:33:49 ED: This is right off, this is, might have been spring of '69. And it’s an off the record conference. Because a number of people come from industry and that way they can get them to say a bit more than they should, a little bit. The other thing is, I don't know what they charged us, but it was nominal. And UCLA charged such nominal amounts, that the only way that you could use up the registration fees that we paid was in liquor. So they ran an open bar in the evening with stuff like Wild Turkey and single malt scotch and oh my God, it just flowed and people talked and drank till three o'clock in the morning and the tongues were very loose, very loose. That was awesome. Driving down there and past the vista point, I pulled out and looked at the vista and it was all smoggy, you couldn’t see two feet, okay. But Mike Clint was at that conference and talked about his early days at IBM. And there was this guy from Control Data, who said there is this thing called pipelines, and people think... And of course pipelines existed in IBM machines and in Control Data machines. They are in the... The hard disk computers might have been first machine to introduce them. But they were ample enough in the IBM 360/91 and so on. And Control Data was fooling around with them.

And he said, people think that you make a pipeline by dicing the logic into... You know, slicing it. And stuff flows through, like an assembly line in a factory, which would be what I eventually called the straight-through pipeline, where the stuff starts at the beginning and then moves linearly through the stages and comes out the other end. He said, you don't have to do... You can do amazing things with pipelines. Now he is running into a boundary where he is not allowed to talk about it. So he is saying things in the abstract, thinking that maybe we will get it. Hey, you can slice them sideways, you can slice them vertically, you can have stuff loop back and loop through it.

And... I spent the rest of the conference not listening to anything that anybody said and drawing pictures on my pad of paper. And by the end of that conference... I had no idea what he meant, or what he was talking about, I never had a private conversation with him. But by the end of that conference, I had an abstract model all things that you could do with pipelines. And you could build feedback pads and stuff like that. So the idea is if you have a process that you want to do and you have got a factory, a stage, a workstation, has specialized equipment for doing one step of the manufacturing. Now, suppose you want to revisit that stage, you paint and draw a circle several times or you paint and then you put another part on and then you repaint the second coat, stuff like that, if you have this expensive specialized equipment you may want to look back and re-use that station. And the fact is, in a real pipeline, in the heyday of General Motors or Ford, Ford more the first then General Motors after I guess, it was all straight through. It was linear. It was a merger of tree paths that came together and ended up with the car coming up at the end of the line. There was no looping back. If you wanted to paint twice you had two paint stations. And that gets the best flow, if you can afford it, and if you have the demand for the product it justifies that expensive equipment.

00:37:54 PE: Even there, whatever is the slowest step in the process will determine the speed of the whole line. It is linear.

00:38:00 ED: Yeah. So then the question is what granularity do you want to build the pipeline on, if the paint process, paint drying process is the slowest thing. Maybe you build a long drying chamber and you have stuff, more than one car moving through it. Or you chop it into several stages, dry stage 1, dry stage 2. How do you balance that, how do you work the granularity? I basically considered only pipelines where each stage took the some amount of time, because I was going to decide where to put the stage breaks. And it’s logic after all. All you have to do is that when you decide that you want a break at a certain point, you cut the wires of the combination network, then stick a register in there. So as you make more stages, you put more registers in. But if you want to economize you may want to use feedback. And if you use feedback you have scheduling problems, which is you can't have a new task flowing through when an old task comes back to reuse the same stage. You have a, what I call a collision. And how do you control that? You have a collision vector which prohibits you from starting things when there would be a collision down the line, but you know the stage of everything that is in the pipe. Well nobody did that. They built straight-through pipelines. Or in the exception, like the divide pipeline in the 360/91, there is a dominant loop that it goes through several times, and uses the same stages in the same sequence, which has scheduling properties fairly similar to a straight pipeline. But I had this wonderful...

00:39:49 PE: The idea is that without this you just waste clock cycles until the process is done, that the process ahead of you in the queue is done, and restart it. Right?

00:40:06 ED: So how do you use everything most efficiently? And get the most output? But when you create more pipeline stages, there is more overhead associated with that because you are inserting a register and a clock, and that takes time. And so, so I divided this thing into... By the time McCluskey was satisfied with my proposal, this problem that we just discussed had about 8 major pieces, each of which could be one or two PhD theses. [chuckle].

00:40:45 PE: So after that Lake Arrowhead conference did you go back and then talk to him about this straightaway? Or...

00:40:49 ED: Oh yeah, oh, yeah, yeah.

00:40:52 PE: This really excited you.

00:40:53 ED: Oh, yeah, this is my... Yeah, this is me. That’s sort of my... Let me get rid of these damn trees that... [laughter] I can't stand anymore, and do something useful. Oh, yeah, this is what I want to do. And that is when Len Shar was... And you referred to him as my first PhD. He was actually my third, but he was my first pipeline student. And when I say I did this theory, you know, is hard to tell what I did and what Len did, and then others who followed him. But I was off and running. So the Stanford time was very schizophrenic between really productive curriculum building, at the undergraduate and graduate level, research that launched my professional career that was first rate and the secret of that is just what Forsythe said. You get the best students that you can get. Man, what a difference. And you learn from your students.

00:42:10 PE: And just taking Shar as an example, how did you hook up with him? Was he already there? Did he apply to work with you? Or...

00:42:18 ED: Oh, he certainly didn't apply to work with me. Who would do that? I was just a kid off the street. Ah, I don't know, he showed up one day. And he was from South Africa. I learned substantially that he was Jewish from south Africa. And he wanted to get a PhD and go back and work in South Africa, and he was eventually persuaded by his brother, don't come back. There is no, there is no life here. This is wrong. And I always wanted to work with foreign students and convince them to stay in America, because the thing is America... America is what is happening technically and this, I believed, this was the best thing for them. But from another point of view, if America can rape and plunder all the natural resources from all over the world, why not plunder the best brains too.

Interview recording part 4

00:43:15 PE: While we are it!

00:43:16 ED: While we are it. [laughter]. I was never in sympathy with the faculty that said "Don't admit foreign students because they are just here just to get an education and they would leave and we won't get the benefit", and the other faculty that said that the problem with foreign students is they get an education, they come here to get an education and then they stay and take American jobs. You know, one of those two has to be wrong. I think they are both wrong. I think what made America great, as Franklin Roosevelt said, was immigrants. When the DAR denied Marian Anderson use of the DAR hall and then Roosevelt was invited to address the next DAR convention and he did, starting his speech, "Fellow immigrants".

00:44:00 PE; Yeah. Whiplash.

00:44:03 ED: You know, it was nothing in the American soil that makes us brilliant or productive. And I realize I am talking about a bygone era now that the sun may be setting on the American Empire. But at that time, this is where it was happening. And you know, I mean, there is a reason why in my generation, a lot of the people who were the overachievers were Jewish kids and in this generation, they are Asian. That is the hungry generation. And they apply themselves. And once you are reasonable comfortable you don't have that particular motivation. You know, you may have curiosity, you don't kill yourself to get where you are going. It's a natural thing. So you need new hungry people.

00:44:56 PE: And Shar was one of those.

00:44:58 ED: Well, yeah. Very bright. So we worked on the scheduling problem. And we developed this thing with collision vectors and state diagrams. And we could... You could design any cockamamie collection of stages that you want. And we could march through... We could build a diagram very quickly. You build a collision vector, you make the reservation table and you build a collision vector and then you could march in, you could produce a state diagram and then as you go from one state to the next, you've started a new job at some time interval which is the weight of the arc between states. And we could find optimum cycles in the state diagram. Now if you have bad feedback paths, the optimum cycles in that are not very efficient. If you have good feedback path, they are quite efficient. The fact is that people didn't.. That was a lovely theory. People didn't build pipelines like that. They built simple straight-through pipelines that were easy to schedule, and they didn't use these controls. To my knowledge there was only one machine that really ever used this kind of a controller on their hardware, and that was the Texas Instruments Advanced Scientific Computer, designed by... Harvey Cragon headed up that project. And...

00:46:12 PE: Spell that name.

00:46:12 ED: Cragon? C-R-A-G-O-N. And he... He admitted afterwards that they didn't know much about my theory but just, they discovered it late and if they had used my theory to do the scheduling, then they could have -- I say my theory, it's the plural my, it's me and my students -- that they could have scheduled it more efficiently actually. We showed that in the end it was true. But the bottom line is that they shouldn't have designed it that way because, it's not, you have odd feedback paths and inefficiencies and it’s better to do it the other way. So now you think this is probably a useless theory. But Peter Kogge, who was Harold Stone's graduate student, watched this pipeline stuff and then after he went to IBM, and then worked in IBM for their systems division, he wrote a book on pipeline and a lot of my... The stuff that we had is basically Chapter 3 in his book. I didn't know that until I read the book. He told me later that in the Manassas, Virginia division of IBM, where they designed space-borne computers, they used all this theory to optimize the microcode of the computer. Because the code had to be very... You know in space it has to be small and compact and efficient. But at the level of code you had much more irregularity than you have at the level of hardware. And you do have these feedbacks and collisions. And so they scheduled... And basically the space-borne computers are running a particular set of routines that are.. They optimized the hell out of them. So they optimized what would have been software which was the hard code of the machine using all this theory. You never know when something will end up being useful. And if it gets used in another domain, in another environment, it’s just lovely the way that happens.

00:48:30 PE: Yeah.

00:48:34 ED: So, he did that. Thampy Thomas did the problem with what if you have multiple function generators. So you have several different kinds of operations which share some of the same logic and some of the output. So some of the stages are used for multiple functions, some are used for particular functions, any old path through that. And we found optimum mixes of operations that would extract the most design levels from that design. That was even more complicated yet, and it didn't do the transitions very well. Well anyway that was published there. So he did that piece. Arvid Larson A-R-V-I-D L-A-R-S-O-N, I think, did some work with Al Despain on CORDIC functions for Fast Fourier Transforms and then ended up working with me and Al Despain. Doing pipeline implementations of CORDIC and standard arithmetic implementations of the Fast Fourier Transform. Despain was...

00:49:55 PE: Spell CORDIC

00:49:55 ED: C-O-R-D-I-C.

00:49:58 PE: That’s an acronym for...

00:49:58 ED: It is an acronym [PNE - it stands for COordinate Rotation DIgital Computer]. [laughter]. It was a way of doing arithmetic by rotation is in trigonometry. But you're doing arithmetic. So you did several... And Despain was doing the CORDIC. And the standard was known before. And Despain was in radio science but also taught computers. Hence the interest in FFTs [fast Fourier transforms]. But Larson developed on my side, the CORDIC and the FFT implementations were examples. The theory was if you have a straight-through pipeline, which is what people were building, what should the granularity of the pipeline be? You could slice it fine, you can slice it coarse. Every time you slice it you spend some money on a register and you lose a little time for the register. But you increase the rate that you can introduce tasks. So there's a cost formula and a performance. So why not go for the maximum performance over the cost? The interesting thing is that the optimum point turned out to be where the space-time product of the time that you spend in a stage, space being cost, is equal to the space-time product of the registers. In other words, you should spend half your time and half your money on the logic of the computation and the other half on storage platforms, the registers between stages. And what you get out of that, well you don't get as much as you might think. You know like if you do 10 stages you might triple the performance or something like that. And we had that all worked out. You might not hit the point, you know that balance point for cost and performance at exactly the right time. It might not be possible to spend half your time and half your money on the logic and half your money in the registers. You might end up with time not equalizing at the same point as cost equalizes, and if that happens, then you settle for the geometric mean between the two and your pay off for pipeline into a certain level of granularity is less; it flattens out. And if those two points are very far apart, the whole performance cost thing flattens out to a step function. Nothing costs, nothing can give you any more or less benefit than anything else; it doesn't matter. That's if your registers are very expensive and very fast. Logic is very cheap and very slow. Not that it doesn't work very well.

00:53:39 PE: Would you like to talk a little bit about this paper. This is the...

00:53:43 ED: Yeah. Interesting that you mention it. Janak Patel came along at that time...

00:53:46 PE: Yeah. I was going to ask about him also.

00:53:48 ED: And he worked... Janak was amazing. He's a professor at Illinois now. And he always did things differently from everybody else. He sat back and let everybody decide which section of my proposal that they were going to do.

00:54:09 PE: Was he American or is it that an Indian...

00:54:10 ED: No. India. Larson was American.

00:54:19 PE: Did Patel start with you while you were still at Stanford? Or did he...

00:54:21 ED: Yes. Yes.

00:54:23 PE: Because I have his graduation date down as '76.

00:54:26 ED: He finished the Stanford degree at Illinois. Four students came with me from Stanford to Illinois; a fact that I'm very proud of. Left Stanford to go to Illinois, to stay with me. One who didn't was Bob Rau who switched from me to Mike Flynn. But Janak waited until everybody else picked what they wanted to do and he just took what was left. Which was what he always did. He figured that would probably be the hardest thing and he didn't care what he did as long as he could do something. So you said why would students work with me and how did it happen they worked with me. All the graduate students wanted to work with McCluskey, he was the senior guy, he's the guy with the reputation. Having his name on your thesis will get you the best job. He was working in reliability and some other things at that time.

Some people didn't like the topic and McCluskey wouldn't take all the students, he only took about 50 of them. And I was lucky to get 3 or 4. So my speech to the students is, "Look, I know everybody wants to work with McCluskey and for all of those reasons, all of which are valid. And I know I'm the new kid on the block, and you don't know if I can be of any assistance to you or not" [chuckle] "Or whether I'm any good." I said "But I'll tell you this, I'm an Assistant Professor and I'm only going to have a few students. And if you want to get 20 minutes a week with McCluskey, half the weeks being canceled and half the time when they're not canceled, he's going to be on the phone and be one of his army of students, that's fine. But if you want to be my student, here's what I'm doing if you find it interesting, and I will spend as much time and attention on you as you wish. Because I've got the time to give and I am launching my career and you will be very important to me." That was my speech.

00:56:48 PE: It's a good sell.

00:56:50 ED: Caught a few. [laughter].

00:56:53 PE: Okay. So...

00:56:54 ED: Oh then Thampy Thomas, who was also from India.

00:56:58 PE: Well maybe since you just brought up Patel, maybe you can really talk about this one first. This is “Effective Control for Pipelined Computers” (1975), by Shar, Patel and Thomas.

00:57:08 ED: Yeah, that's a collection really of what I just talked about. That's the multifunction units which Thomas did and the pipelines with internal buffering and what... You know, as I said, my publication... I didn't publish like I should have and that was very very stupid. What you're holding there is a three-page paper, probably from CompCon I think.

00:57:50 PE: Yeah, that's right.

00:57:52 ED: It's a digest. So there I've got one publication and three student's PhD Thesis's in three pages. [chuckle]. I mean, this is insane. And when I gave the talk at CompCon, because I gave the presentation, I had on the order of 60 35mm slides to go through in a twelve minute talk. I could deliver it, nobody knew what in the hell I was talking about. Worse yet, on the way driving up to CompCon I was driving behind a truck and a mattress fell off the truck. Right in front of me on Interstate 280 going to San Francisco where the conference was, and a spring sticking up out of the mattress hooked on to the axle of my car and I was dragging the mattress. So I lost some time getting there. So I got to CompCon after my session had started and had to load my 65 slides into a carousel in about 10 minutes before I was going to stand up and give my presentation. That didn't work very well.

00:58:55 PE: I imagine. [chuckle]

00:58:56 ED: People were impressed how...

00:58:58 PE: That’s about 15 seconds a slide...

00:59:00 ED: Nobody knew what the hell I was talking about. People were impressed that I could talk that fast. Young and stupid. What Patel did... Now Patel did give a talk at ISCA III.

Interview recording part 4

The Third International Symposium on Computer Architecture. Third annual. I don't think it had the "I" at that time, it was before it was called international. And it was about adding delays to pipelines in order to improve their throughput, their performance. Gene Amdahl was giving the keynote talk at that conference and he chose to devote some of his remarks to our paper. Which I was delighted at. After he got to the conference, he looked at the table of contents for the and he picked out paper. He was at Amdahl Computers at that time. And he said that he was delighted that the academics who were concerned about the fact that in a high performance computer, a pipelined computer you have to add delay. Normally academics don't get to that level and it's absolutely critical to understand this and deal with it and know how to do it. Otherwise your computer will either be inefficient or won't work at all. Now the fact is that he... I loved the compliment, and I've quoted it before and people were impressed, and they patted me on the head for it, and so on. The fact is that he guessed wrong about what our paper was about.

[laughter].

01:00:56 ED: He was talking about circuit builders, Amdahl was talking about circuit delays. So between two stages of the pipe you have multiple paths through the combinational logic, and you don't want skew between those signatures. You'd like them to move as a wave front, so that they all arrive at the output register simultaneously. Because that's where you have the least skew problems. And so if you do, like Seymour Cray did with the wire on the four input NAND gate to make it slower, so that it was the same speed as the five, so that the signals move the uniformly, that's padding out the short paths, so the paths have equal. We weren't doing that. What we were doing is we were building these stupid feedback pipelines that couldn't get scheduled regularly because they had collision patterns at different times whereas if you built a straight-through pipeline that equalized it all so that it was one uniform loop, you could schedule every so many clocks and do a constant schedule and make it so all the stages were busy all the time. And you could achieve that by adding dummy stages at certain points in the pipeline flow. These are stages that do no work, consume the clock and simplify the collision vector, so that in fact you could have a constant start time that kept everything busy. So the question is how can you optimally improve performance by inserting delay stages in the pipeline. That's what Patel did. And did it optimally.

01:02:46 ED: Bob Rau started with me doing this thesis about clocking. How can you clock the pipeline? Well the obvious answer that would occur to anybody which is what we thought was right, was that you take the least common denominator of all the pipeline stage times and clock at that rate. But you have to clock it at least that rate in order to him the exit right. But question is what would happens if you clock at twice that rate, or four times that rate. Rau proved and answered that question, which is that you could have the transient benefit in a start up that you would never gain a steady state throughput advantage. It's only when you have gaps with no work to do. If you had enough work to do always, there's always tasks waiting to be done, you didn't gain by over clocking. And he said if you clock at twice the speed, you take this collision vector that you've developed for half speed and here's how you would transform the collision method and get a collision vector for twice speed. This is a beautiful thing. Damn, I wanted him as my PhD student. But I was leaving and Mike Flynn was coming in and he switched to Mike and did another thesis. Then we got him into Illinois as a faculty member. He's brilliant. All my guys were brilliant.

Interview recording part 4

01:04:19 PE: I was going to ask you about this one. The multiminiprocessor system you did with Len Shar. ["A Multiminiprocessor System Implemented Through Pipelining", 1974]

01:04:25 ED: Yeah. Well that's interesting because that was actually an appendix to Shar's thesis. And he just had this idea that once you build a pipeline, if you have trouble within... Ah, let's see, if you're trying to run a program and get different operations going in pipeline fashion, you have little fits and starts and problems. Because one, sometimes you hit an instruction that can't compute because the result of the prior instruction... It may depend on the result of the prior instruction and that one hasn't been finished yet. The pipeline is basically, you know you started a job and you haven't finished it yet and you start the next job before you've finished the previous job. Well if they're dependent, you have to have the output in order to start the next one, you can't do that. But what if you run multiple programs with the naive but reasonably good assumption that instructions in two different programs are not dependent on one another. They accept through input output, so forget that. That doesn't happen very often. So now you got enough independent stuff to keep the pipeline full. And now you can always find work to do. So how would it work in practice? So what we did is we took a Hewlett Packard minicomputer, I forget what it was, a 1600 maybe. It was one we had in the lab at the time. And we did a model of it. And we made a guess what would happen if you took that simple computer and pipelined it and what it would look like and how you could get a multiple stream processor. Which is the beginning of the direction like the Control Data 6600 Peripheral Processing Units and what later the Alliant did, and what later was done in... Well, what earlier had been done in the Honeywell 800 with the multiple programs and, but it wasn't pipelined, and later evolved ...at University of California at San Diego into multithreaded architecture, which is actually very hot now. ...Then the question is what's the difference between this and multithreaded. And the answer basically is that multithreaded ... doesn't do a strict round robin and it can pick and choose instructions from multiple different threads or multiple instructions from the same thread, all this kind of running through at the same time as pool for multiple threads and it's a very flexible scheduling thing. I think I've got that right. ....

01:07:43 ED: But these ideas keep coming...

01:07:44 PE: Systems that you've used...

01:07:46 ED: Yeah. It may not have been. It may have been an abstract thing. But it was patterned after a simple processor. And the idea again was cost effectiveness. And we said hey, you can use pipelines to achieve a multi processor. Yeah, HP built mini... It's a multiminiprocessor. HP was building mini processors. Before microprocessors.

01:08:10 PE: So the way you put it in the paper, "Choosing your appropriate architecture for a multiminiprocessor system and illustrating some inherent cost advantages of configuring a system that appears to the users as a multi processor but is in fact a single pipeline processor".

01:08:26 ED: Yes. So it's a single pipeline and you can schedule jobs from multiple threads or multiple programs. And if the programmer thinks of each program as running on a processor that he or she owns completely then it virtually it appears like a multiprocessor. So that was the beginning of that idea in this academic formalism. You know sometimes people ask me if I invented the pipeline. No, heck no, it existed long before. But I did some early weird things with it.

01:09:07 PE: [laughter] Okay.

01:09:09 ED: And the same with this, you know. People get credited for inventions. Rarely is it one person that invents something. But the media and publicity like to pick out somebody. So they do.

01:09:30 PE: Just pausing for a second here.

01:09:35 ED: We could do reading if you want. One time there was... There's the lunch.

01:09:44 PE: Okay.

01:09:46 ED: Of the junior faculty who were there, Harold Stone was the senior amongst us. And he started to think that once a week, no senior faculty allowed, all the junior faculty would go to lunch at the faculty club. And the only topic allowed for discussion was research problems that we were having difficulty with. And we were not allowed to say that we couldn't find a parking space when we drove in and we weren't allowed to bitch about senior faculty or the administration. We weren't, it wasn't a social thing, it was that topic and that topic only. And that was rigidly enforced. And that was wonderful. We got to pick each other's brains and make suggestions to each other. And learn about what each other was doing. So there were amazing tactical things that happened there. I haven't seen that done in that form elsewhere, and it's a serious mistake. I've mentioned it to people, I don't understand. You know maybe they're so busy now that they can't even contemplate that, but they're making a serious mistake. It was wonderful. And when you get the senior faculty around you can't very well do that, well first of all some of the junior faculty don't like to expose themselves. And second of all, if the senior faculty wants to bitch about something, you can't stop them very well. I mean somebody once said what's the difference between a tenured faculty member and a terrorist. [chuckle] And the answer is that you can negotiate with a terrorist.

01:11:37 PE: [laughter] That's great.

01:11:40 ED: So that lunch was a terrific thing. Another... Yeah, go ahead.

01:11:52 PE: You were going to talk about Edsger Dijkstra.

01:11:52 ED: Yeah. A couple of odd things happened. That are perhaps worth mentioning. One time I was told that Edsger Dijkstra was coming by interviewing for a faculty position in computer science and would I like to meet with him. I said, I knew who he was and I said "Oh, my goodness, yes". And they said "Well, why don't you take an hour? Do you want an hour and a half". I knew then that something was funny. He's interviewing Computer Science, I'm a junior faculty member in Electrical Engineering and I'm getting all this time for a slot. Well unbeknownst to me Don Knuth had been in Europe that summer and he had run in to Dijkstra, and spontaneously in a burst of enthusiasm, offered him a faculty position at Stanford. You know "Why don't you come and see us. And we can get... Maybe you would be interested to come to Stanford." He didn't offer it to him. He said maybe you'd be interested.

When he got back he mentioned it to the Computer Science faculty, they said "You did what?" Dijkstra is known to be, whatever his professional reputation with the flags and synchronization... my characterization of the flag stuff that he did for synchronization was that he invented the master slave flip-flop in a software. But ... and I'm sure theoretically, there's more to it than that. But on a personal level he was known to be somewhat prickly and difficult to get along with. So nobody wanted to talk with him. They were stuck; they couldn't fill his schedule. So I got time with the great master. So I'm thinking what do I ask him. So I popped the question to him, wondering why, since he was from Europe, why is it that so many of the really fundamental computer inventions came from Europe? Like the index register and micro programming and Boolean logic, you know a lot of really fundamental stuff. And yet all of the commercially successful machines at the time are American. How can this be?

01:14:39 ED: So he said, "The European educational system encourages the nuts and the weirdos. It gives a tremendous amount of freedom and time for private study and contemplation and going your own path.” And when you get to a certain point, like he didn't say in Zen Buddhism but what my anthropology course taught me is in Buddhism, when the pupil turns to the master and says, "I know more about this stuff than you do and who the hell are you to be telling me what's right and what's wrong?" The master is supposed to say, "Ah, now you are the master". That's a point today I'd love to get to with my graduate students, that was victory. So Dijkstra said "It encourages the fruits and the nuts. They're undisciplined but some of them turned out to be geniuses. America has a form of education in engineering that produces graduates that are interchangeable parts in industry. That when someone hires an American engineer from a reputable university, you know what you're getting, you know what they've studied, there is uniformity to it. And they can work in a team and understand each other and they have a certain discipline. So you tell them what to do and they do it. And you can get the job done." That's interesting. In British computers, it's so imaginative, a conglomeration of stuff that doesn't make practical business sense. But it can be interesting and some of them do.

01:16:34 ED: Bernie Lacroute, who I met years later, was the financial person behind Sun Microsystems and a number of other companies, mid Peninsula. He's the kind of people that you meet if you hang out at Stanford. I got to spend a little time with him and I told him the Dijkstra story and he said, "Wow, that's not really true. What it is, it's part of it. That's a small part of it." And the other thing is the venture capital community that we have, which is high flying and risk taking. And so the gap between... And not only high flying and risk taking but they're incredible business people, and there's a whole business infrastructure, a profitmaking, hardcore capitalist structure that makes for better or worse with the idea people and the implementers, and it's this combination, which is at its peak... Was at its peak in the Bay Area, and it just all functioned. So the gap between concept and delivery was very short. And in rapidly obsolescent technology, that just moves on, you've got the business people who on deadline will ship it out the door before it becomes obsolete. When the day comes you ship. And then you’ve got the engineers who optimize until doomsday and never ship anything.

01:18:24 PE: There's a famous story about Steve Jobs. I think when they were still working on the Macintosh before it had come up, he storms into a roomful of engineers one day and he writes on the whiteboard, “Real artists ship.”

01:18:42 ED: Aha! Yeah he's a tough taskmaster, known for throwing tantrums and all the rest. I could do two Steve Jobs stories.

[pause]

01:19:06 ED: So Lacroute championed the venture capital side of it. And he's quite right. He also said a company that becomes a classic of its era, a product idea that becomes a classic of its era, does not come from looking at what's on the market and figuring a clever way to make it 20% more efficient. Or really even... It has to be like an order of magnitude, preferably two orders of magnitude, better to move the market. Because the market doesn't like to go to a startup that may be gone tomorrow. That's risk. I mean no computer installation manager ever got fired for buying an IBM computer. [chuckle] You take a risk man, you could be dead. So that’s that. He said "The real successes are the products where nobody ever knew that they wanted something like that or could use anything like that until the product existed."

And his two examples were the Mister Coffee coffeepot. There was nothing like it before. Suddenly everybody had a Mister Coffee coffeepot. And the Xerox machine. You already have it on a piece of paper, why do you want to copy the piece of paper and want to have two copies of it? They didn't preconceive, how common that would be? What is wrong with ditto masters, anyway? Or like my thesis which I had to get typed on multiple sheets, and if you make one typographical error on a page, you have to re-type the page because you couldn't erase. My secretary quit halfway through. My thesis is unique in the world in that the first half is in Elite type and the second half is in Pica. Because my secretary had personal problems and she was done, and I had to type the second half of thesis but I couldn't rent an Elite typewriter. So I just changed type fonts. So the ... I put in for a format check in two boxes and they never noticed that one box to the other, the type font changed. Nor did anybody else ever. Ah, gee, I'm digressing. So that was the Lacroute story, and he said that Sun Microsystems was an example because of the open architecture. And Apple was an example because of the personal computer. The two Steve Jobs stories that I love...

01:22:01 PE: Did you have any contact with him?

01:22:02 ED: I never met him personally. No, I never met Jobs. Just Alan Kay who did summer at Stanford...

01:22:08 PE: What about the Homebrew Computer Club? The famous...

01:22:11 ED: Oh, sure. Dr. Dobbs Journal?

01:22:13 PE: Yeah, sure.

01:22:14 ED: Absolute. Jim Warren! Dr. Dobbs Journal eventually became legitimate but the original was a kind of counterculture thing and Warren was a hippie and he wanted to serve the industry and so he was called "Doctor Dobbs Journal of Computer Calisthenics and Orthodontia: Travel Light without Overbyte". B-Y-T-E. Yeah, Jim was around there. That's my Jim Warren story. Allan Kay was doing the stuff at Xerox with the personal computer and the story was that Steve Jobs was having trouble getting the Apple thing done. And he took a tour of Xerox one day, and by the end of the tour, he had hired the key people that did the thing right at Apple. A good and right thing to do. Xerox Corporation didn't know what to do with computers and they weren't gonna capitalize on the technology; that would be a damn shame. The other Steve Jobs...

01:23:38 PE: There’s a book about that called *Fumbling The Future*, about PARC and the personal computer.

01:23:44 ED: But there's a culture. And if the management... Like I found at Honeywell. The management of the corporation cannot deal with culture; it isn't going to happen. It's got to go somewhere else. And that's a good thing, to move it. The other Jobs story is that when Digital Equipment... Ken Olsen didn't like personal computers. Although they started with small computers like the PDP-8. For some reason, he just wouldn't go there. He wanted to do real computers.

01:24:26 PE: Yeah. He's famously quoted as saying, "There's no reason anyone would want a computer in their home."

01:24:32 ED: Von Neumann once said he could foresee a day when he might be as many as 10 computers in United States. Who knew? Who knew? God! It was an incredible industry to be a part of. It's just wonderful. Fireworks going off everyday. You didn't have to use drugs. [chuckle] It's just all there. So when Olsen was forced to do a personal computer at Digital, the computer called the Rainbow, and it was basically stillborn on the day of first delivery. It was a disaster from day one. Steve Jobs sent Ken Olsen a dozen black roses, just thumbing his nose. Years later when Apple was going down the drain and Intel was just killing...

[several sentences deleted here at ED’s request]

01:28:45 ED: The story is that when Apple decided to quit making their own processor and buy another vendor's processor for their machines, the hot processor... The best design and my favorite as an engineer at that time, was a Digital Alpha. It was a wonderful machine. I think one of the reasons was one of my students...

01:29:20 PE: First they went to Motorola and went to the...

01:29:23 ED: Yeah.

01:29:24 PE: The Power PC chip.

01:29:26 ED: Before they went to Power PC and then Intel.

01:29:33 PE: Intel, yeah.

01:29:39 ED: They wanted to use an Alpha and Apple proposed... I guess this would have been after Jobs had left and then come back to Apple, because this wasn't that long ago.

Interview recording part 4

01:29:53 PE: Yeah. Yeah.

01:29:56 ED: And they wanted to use the Alpha and they made their presentation. And Ken Olsen remembered the black roses. He said "I will not give Apple the benefit of anything that Digital produces."

01:30:19 PE: Wow.

01:30:19 ED: “I would not do business with that man.” And the story was that that decision drove Apple into the arms of IBM [the PowerPC] and then Intel. That worked okay for Apple, but that was a shame. A very beautiful computer, one of my favorites. And probably my two favorite computers will be IBM 360 model 91, which was a disaster, a commercial disaster which had some lovely features in it and the Alpha. They were wonderful engines beautifully tuned. Well, not the 91; 91 was very progressive and Alpha was just a beautifully tuned system. I think one of the reasons was a former student of mine named Joel Emer who you know helped that along, and he is about to get the Eckert-Mauchly award this year.

01:31:21 PE: Oh, really? Great.

01:31:16 ED: Next week.

Departure from Stanford

01:31:21 PE: Okay. So leaving Stanford.

01:31:28 ED: Yeah. So in my 5th year at Stanford, this is now late 1972, early 1973, Illinois Electrical... The Computer Science Department had started at Illinois and they were going along well. We had been... McCluskey put me on a COSINE Committee C-O-S-I-N-E all caps by ACM, writing a book on the standard Computer Engineering curriculum. So I was part of that, I was part of the laboratory part of that. Taylor Booth chaired that part. Eventually got a Taylor Booth award in education which was nice. Illinois Electrical Engineering decided to start a Computer Engineering undergrad degree. Computer Science wasn't happy about it, but one department can't very well control another department's degree programs.

And my advisor Gary Metze called me up and said “what do you think about coming back to Illinois and helping us with this program?” So I thought "Okay, my five years at Stanford are almost up and I don't have a future plan. That might be a good move". I liked Illinois technically, but we didn't enjoy living there as a graduate student. Talked to Ellie and she said "I know you don't have a future here and I know particularly you like the California culture that much." Almost every faculty member on the block we lived on during the 4 years we live there had gotten divorced and married some other people. So a lot of instability. It was the high drug era and so on. Which I wasn't part of, actually. It was one thing I didn't go into. Or as Mel Brooks said "Well, I smoked but I didn't exhale" [laughter] So...

01:34:01 PE: You were thinking about Illinois.

Interview recording part 4

01:34:05 ED: Yeah. So she said "I will go back with you. But you got to, and I will go back and live with you", which we didn't enjoy didn't know if we would. We did as it turned out. "But you have got to promise, no more politics. It's done. Do your job, do what it takes to have a career and a life. Somewhere in Stanford everybody was starting companies, so I talked to her and said "You know what, maybe I can start a company. That's something interesting." She said well "You can start a company if you want, but you let me know beforehand, so I have time to file for divorce." She denies she ever said that. I oversimplify stories. But I got the message. So I said okay let's play this out. I had done some consulting for Hewlett Packard. So you know maybe I could go to Hewlett Packard. So what I did was expand the memory address space from 16 bit addresses to 20 bit addresses for one of their machines. It had to expand because memories were getting bigger. Worked with them for the summer and came up with the same proposal that their engineers had come up with, because it was the only way to do it. But they didn't want it because it would cost something; well, you can't retrofit something like that into a machine without it costing something. That was the only way to do it. Eventually they did it. By the time they did it the machine was old and obsolete. It had another life.

01:35:43 ED: So I went talked with David Crockett who was a former Metze student, who was my friend and mentor at Hewlett Packard. But was Steve Allender whom I directly consulted on the project and said look "Here is a deal. I got an idea for a tiny computer." You know I don't even really remember what was in it. Probably had something to do with pipelining, "And I want some time to develop it. So here is what I will offer you." Because I knew that I could get the job, so that's the best. What can I do that is better. So I said "I want to pick three, four, five people to work with me, and I want anonymity from the higher administration for three years. I don't want them to know what we're doing. So you just cover me in your organization, and I don't want have to make presentations or pitches for 3 years. At the end of 3 years I will make a pitch; if people go for it we do it the HP way. And ... if they don't go for it I will do something else if you want me to." And Steve looked at me and said "Hey, can't do that." He said "You think if I tell you yes, that you've got a deal. If upper management doesn't know about it, you got nothing. I could leave HP tomorrow and what have got then? It's not how it's done." I said "Okay, I am not... " You know the idea wasn't mature enough for me to sell it successfully, I don't think right now. So you know I maybe don't want to do that.

01:37:30 ED: So I went to John Linvill. And I said "Okay, I remember Dean Peterson's talk. I am in my fifth year now and I know he said none of us is gonna get promoted. I never expected to get promoted. I know that I have burned a lot of bridges here and have agonized a lot of people." I said "But I have contributed to the curriculum and I have done some what I think is excellent research. I haven't published much, but I can redo that and I pay my own way in terms of research funding. Can you tell me... " Ah, and I said "And I realize that I am probably useful to you and you will probably reappoint me for two more years at the end of five years, which is the maximum time you can keep me without tenure. If you were to reappoint me for two years at the end of this year, is there anything I could do in those two years that might give me a shot at tenure?" Now that sentence is very carefully constructed. I am not asking for any guarantees or anything like that... And I was met with utter stony silence. I mean he talked, there were words. But there was no content to the words. And I left that meeting and I said "Okay, I am done at Stanford. I am not... I've got a real opportunity here. Professionally it makes lot of sense. I am not gonna wait out the two years, I am gone.

And I am going to Illinois where they want me for who I am and what I know I can deliver. And they know me. And they want me anyway. And Stanford never understood me." It was always a clash with the place.

01:39:46 ED: So I made up my mind. And I clued my students in and this is probably the way it's going to go at that time. Tom Kailash was the professor in systems division. He was from India. Couple of my students were from India and they went and talked with him. And Kailash said, "Don't worry." He said "I know Ed, and Ed just built a house on Stanford land. You know he is here. He is not gonna leave this and give all this up. You got nothing to worry about." I appreciate Tom Kailash and I respect him. He was mentor to tremendous number of students and faculties at Stanford. Elder statesman in the systems area, outside of computers. But I was always saddened that he mis-estimated me so and to think that I would guide my life by the fact that I owned a house. That's an interchangeable commodity, who could care. I don't sell myself that cheap. And it just thrilled me... Well I say 5 students came with me from Stanford to Illinois. Actually I am not sure how many came with me. I could count might even 3. Look Dan Weller, Thampy Thomas, Janak Patel, Len Shar finished, maybe the others I talked about finished. Okay. And then Fayé Briggs stayed at Stanford and finished his master's degree and came to Illinois for PhD. Dan Hammerstrom, beautiful, sweet, wonderful person started with me in Stanford, worked... You know we knew each other during the masters degree. He was on the verge of starting a PhD and he was drafted. And I left. I thought he had gone to Vietnam. Turned out he did technical work for the military in the United States the whole time and when he got out he came to Illinois.

01:42:06 PE: That’s five.

01:42:06 ED: That’s five. But he didn't do...

01:42:10 PE: Nice path for him.

01:42:11 ED: Three did a Stanford degree. And one fine day we got on a airplane and flew back to Stanford and did three final oral defenses in one day, one after the other. And John Linvill was on Dan Weller's exam committee and Weller presented this stuff about optimal pipeline search which if you wanna search on a two stage pipeline you don't want to divide the list in half each time, do binary search. Turns out you optimum thing to do is divide it like a Fibonacci series. The only practical usage of the Fibonacci series I know, God probably knows another because a lot of things in nature use it. And Dan used the word associative at one point and because it came in context as encountered, as an example of another kind of thing you can do, for associative searching. And it came in as part of what he was doing and Linvill asked the one question, the only time he ever spoke during the exam. He said "Associative search. What is that?". And Weller started his answer with a phrase "Well, it's kind of like a telephone book." and at that point Linvill said "Oh! Okay". [laughter]. That was the end of that. You know you don't know what page to look at in the phone book, you know some name, and you just poke around until you find it. That's what you are doing. You know it was fun. That was a great day. So they all passed. That was nice. I was very flattered that they came. Dan Hammerstrom one time had to say something nice about me or someone and he said "I thought when I got out of the service and I thought about going to Stanford for my PhD, ending my leave of absence, I could have just walked back into Stanford. And having all the benefits of the Stanford PhD. Or going to Illinois which I didn't know and didn't particularly want to live at and getting a chance to work with Ed Davidson. And there was no doubt in my mind which was I wanted to do. That's what it is all about."

01:44:48 PE: That was a great story.

University of Illinois (1973-87)

[Interview recording part 5 begins here]

00:13 Paul Edwards: So this is Part 5 of Ed Davidson's Oral History. And we are now about to talk about your time at Illinois, from 1973 to 1987.

00:25 ED: Yeah.

00:27 PE: So when you went there... I think we talked about this previously, it's called the Coordinated Science Laboratory.

00:33 ED: Yes.

00:33 PE: We talked about why it is called that the other day. Tell me what it was like when you first got there.

00:41 ED: Well, so my academic appointment, my tenure, I had nothing. Didn't have tenure actually, that's another story. My academic appointment was in Electrical Engineering and I was associated with the Coordinated Science Laboratory which was a research lab that sat on a, that had a block grant of Joint Services Electronic Program which is what really bound us all together. And that's long gone now but the lab lives on. And so my research home was there and my academic home was Electrical Engineering. Actually I think, it is a very nice kind of split because you have two different administrative structures that you report to and you can nicely play one off against the other. It works very well. And it was a very heady time. There were several senior faculty, Gary Metze, Franco Preparata, who was more algorithms and theory, Bob Chen who did a variety of things. His statement was you should never do research in any one area for more than five years because then you've done all the things that are easy for you to do and you should move on. I don't like that style of research. I think that one of the things that makes Illinois great is that you can work on a problem and polish it and tune it and become as good as you can be over a long period of time in a kind of a... Saying it's sweeping it out of the way doesn't do it justice. But it's not in the eye of the maelstrom as Stanford is. And it doesn't have all the pretense and aura that Stanford has. So it's a different style. Ed Jordan was a long term Chair of the Electrical Engineering Department. He was Dean Everitt's protege, and he chaired that department, I don't know how long. It must have been at least 30 years. And he was one of the key inventors of the log-periodic antenna.

03:16 PE: What's that?

03:16 ED: It was the major TV type antenna, the one that sort of looks like a series of rods that keep getting narrow and narrow. And they point off in a certain direction. That's log periodic scaling, it makes it a very efficient antenna. So you know, there I was... When you leave Champaign-Urbana, you can drive an hour and a half in any direction and see nothing but corn fields, corn soil. And it's flat as a billiard table. And the reason is that all of the local farmers in Champaign county like to say that they are growing farms on Minnesota topsoil. Illinois was a swamp naturally, it was malaria and swamp. And it was the residue of everything that washed down from the upper states. When in the westward movement of the United States, Kansas and Nebraska were settled long before downstate Illinois and it was only after they were running out of relatively poor farmland that anybody said, gee, can't we do anything with Illinois. And someone discovered that if you dug a ditch a foot deep and long enough, you could drain the entire state, because it was just flat. And once it was drained it was topsoil as far down as you could dig.

04:59 PE: Where is this ditch?

04:59 ED: On the side of the road. They're all over the place. So like most states have school districts, Illinois has drainage districts.

05:07 PE: So it's like Holland in the middle of the United States.

05:10 ED: Well except there's no ocean. It's just the swamp seepage. Yeah it's like boggy lowland.

05:16 PE: Exactly.

05:16 ED: Yeah.

05:19 PE: Tough for planting.

05:19 ED: Woe betide the farmer who plants too close to the drainage ditch. They protect those drainage zones, because you get flooding if you screw it up. So you know it's known as the corn and soybean state. It's a high competitor in wheat. When we got there a Provost of the university met the faculty and said, look here a few guys the best and brightest and very honored to have you here and great to welcome you to the University of Illinois, but let me tell you what our University budget really depends on at because it was highly appropriated by the state. The state collected the tuition that the students paid and then appropriated the University budget. So we don't even keep the tuition here, have very little control over anything except the research budget, what's given in the research and what the states gives the University. Unlike Michigan, which was incorporated before the State of Michigan, and sets its own tuition, as long as they don't charge in-state tuition so much, the legislature isn't antagonized. They get the legislative piece and they get many other pieces, so they have much more control. So he said what determines how well we do in a given year financially, really has to do with only two things which the legislature reacts to. One is how well the football does and they usually don't do very well. And the other is whether the agriculture faculty has found a cure for this year's corn blight. Because every year it's different; it's like flu. And he said, so the rest of you don't really matter on that score, but we are trying to build a great university and keep it great and you have a role to play there. That put us in our place.

00:07:19 ED: Jordan said what made it great — a sleepy town without a heck of lot of going on other than the university — is that the only reason to be there is because you love your work. And you can do your work there at this university. And because that's the only reason to be there, you are doing it in the company of other people who love their work. And that's what builds that culture. There is a lot to be said for that and there is a feeling that we are doing a social service by providing an education to these particular undergraduate kids. At the time I was there by 50 odd, 97% of the undergraduate student body was Illinois born and bred. And the state would get very upset if it departed from that. Well Michigan doesn't have that. I think Michigan has about 30% in-state and they can charge any tuition they want for the out-of-staters, so they make their money on that. Maybe Michigan is 70% in-state, it's one of the two [PNE - the latter is correct]. Unfortunately that leads to a narrow demographic in the undergrad student body. But that's okay. A lot of the students are first generation college. But they are smart, and that's good. So the time that Intel was doing the microprocessors it was a point of pride for us, that the dominant population in the engineering group that did the design for Intel and others was Illinois, Purdue and Berkeley. Not Stanford, they didn't have the undergraduates. You know it wasn't the kids from the elite schools in that sense, it was the kids with the rock solid undergraduate professional training. So we had a role to play there. And then you know the graduate students were good and the research money was good and there was the Coordinated Science Lab to run away and hide in.

00:09:40 ED: So if you liked being a professor and doing research, it was a great place to do it. But in the five years that we were gone, they had built the Krannert Center for the Performing Arts. So that the New York Philharmonic no longer had to play in the basketball arena. We heard a concert there one time. I think every note of that concert is still echoing in that place someplace. The Krannert is a wonderful facility, international people that play Chicago on the weekend, they come down midweek to Champaign. Chicago Symphony used to record their records in the Krannert Center because the acoustics were so much better than what they had in Chicago. Nice cultured town, you get island fever, but a wonderful place to raise kids. We didn't even lock our house. Well eventually our neighbor's house got broken into, and after that we locked the house, but it never occurred to us to lock the car sitting in the driveway with the garage door open. You know, a nice place.

10:51 PE: So what about work life? You got there and you had these Stanford students who came with you or came soon after and Patel was one and there's another paper here, I have marked the most important ones, there's a little bit laid out here... Oh, this is not Patel.

11:15 ED: Kumar?

11:16 PE: This is Kumar & Davidson, "Computer system design using a hierarchical approach to performance evaluation" (1980).

11:22 ED: Yeah.

11:24 PE: I don't know if you want to talk about it straightaway because this is five years after you got to Illinois. Seven years after you got to Illinois.

11:32 ED: Yeah. Well I kind of talked about the group that came with me from Stanford and they happily finished. Kumar was in India. He applied for graduate school at Illinois and something that fouled up with his visa, with the timing of it. So he missed the deadline in Illinois because his student visa didn't clear through in time. So Illinois did not admit him. He got accepted to a Ohio state and I... And he started the semester at a Ohio state and he phoned me up because we had interacted during his application process. And he said, "I'm dying here". [chuckle] I really would like to be at Illinois and what you do is more interesting. It was you plural, but what you're doing is more interesting than what I can find here. Can I come and talk to you and see if there is a way that I might be able to transfer. So I said, sure, but don't just come, bring all your possessions with you. [laughter]. One of the advantages of an Indian applicant and having Indian faculty that you are working with is that the Indian higher ed system, which is based on the British system, and it's a huge country with a huge population and everybody is rank ordered by the academic system. So when you see a student with the President's gold medal you know what you are getting. You don't have to ask anything more. So I knew...

13:33 PE: He had that.

13:34 ED: I don't recall if he had the gold medal or the silver medal, but he had a stellar record. So he came and he said he had no place to live or stay here, he had no resources and so on. So I said, you know I had recently moved to town at that time and I didn't have encumbrances. So I said well we've got a spare bedroom and you can stay at our house. And so he did. Got everything on a bus, turns out he had to change buses in Indianapolis and all of his worldly possessions including his passport and documentation and visa did not make the bus transfer at Indianapolis. And when he got off in Champaign, he had nothing except the clothes on his back. He was sure he was never going to see that again and I kept telling him don't worry it will show up, and 2 days later it arrived back from St. Louis and he was amazed. He had never heard of anything disappearing and then reappearing. In India, if it disappears it's gone.

We had a great time with him staying at the house, and he got to listen to Arlo Guthrie on the phonograph singing "Alice's Restaurant" and decided that he had not landed on the moon and this was something he could relate to. His English was flawless, better than mine you know, good British English. My kids loved him. They were little ... the older one was first grade, the little one was nursery school. And they used get up in the morning and run in and jump on him while he was sleeping and he kept assuring me it was okay. He had brothers and sisters and it made him feel like he was at home.

00:15:31 ED: Kumar, I would say, was genius material. He was just an overwhelming intellect. He did a Masters and a PhD thesis. The Masters thesis was deterministic simulation where he took as his case study the IBM 360/91 computer that I mentioned before, absolutely gorgeous pioneering and defective machine, it was not a commercial success. An early aggressively pipelined machine and responsible for the repetitive multiplication way of doing division. The Tomasulo algorithm reservation stations, a just incredible innovative machine. He broke it down and he did simulations of design variants of the machine, to try to find how it performed, what all the hang ups were and so on, trading features in and out. I'm mixing the Masters and the PhD, but I guess that's okay. When he got the hierarchical thing, he really had a space where he could trade, trade stuff in and out. He had the good switch from the Tomasulo algorithm for scheduling and reservation stations to a Control Data scoreboard kind of thing. He could change the speed of the memory, he could introduce more or less pipelining in the floating point unit and the fixed point unit etc., etc. He could vary a whole variety of those. He built the simulator. And that was his hierarchal approach because he had levels of coarse granularity gradually working into fine granularity and the purpose of that is try to avoid local minima. And he had a cost model.

00:17:57 ED: So these were a couple of things in there. One is that the overall style was a precursor to conversions to optimum that was later called simulated annealing. And he did it. He didn't aggressively market stuff. He published it, but I don't think he even got recognized for that. The other is how did we ever know what the effect of changing these things in what was then the most complicated computer on the market and how did we... You can never make a cost model. How can you make a cost model of a commercial machine? The one thing that even companies that publicized their design because they were proud of them or publicized their design because the machine was a failure and now all the design people had to do after that was write papers, they don't publicize their cost model. IBM won't tell you what things cost, they won't tell you how they benchmark their machines when they do designing. That's the family jewels. If you know that, it's not that you have the secret to beat them, but you have a secret to design a machine the way they design a machine on their metrics and compete with them on their metrics, whether they are real or fake.

19:20 PE: What they think are the most important.

19:21 ED: Exactly. So you could beat them, in the way that they sell their machine to their customers. But the 360/91 they published little bits and pieces in different places and he found them including costing discussions and integrated them together into one thing. Some time after he finished, a long time after, several years after I got a call from IBM asking me to come give a seminar at Yorktown on Kumar's study of the model 91. Because they'd read it and they wanted me to present it. And I said you don't want me to present it, you want Kumar to present it. He knows more than I do about the details and they said no, no we want you to present it. I didn't know why, but it's an honor and I said okay. Turns out then there was a flurry... that there had been some legal thing with IBM where I had been involved in some consulting on the other side through which I... Yeah, involved on the other side as a consulting job. And IBM typically, when you gave the seminar there, paid an honorarium of $100 and they were concerned about the consulting that was technically unrelated and so on and that was okay. But some bureaucrat was worried that IBM would pay me this $100 and it would be like they were buying influence.

21:09 PE: For 100 bucks. [chuckle]

21:10 ED: Exactly. One of my friends at IBM, former student who had been charged with inviting me said to that the administrator, "I know Ed Davidson and he cannot be bothered for $100." [chuckle] And he said okay.

00:21:32 PE: So let me just say a thing about this paper. Because it's...

00:21:36 ED: Let me come back to the IBM seminar. Because that was kind of weird.

00:21:40 PE: Okay. Finish that.

21:44 ED: They had something specific in mind. A game was going on. I didn't know what the game was. The game was that sitting in a room where all the designers, the key designers of the 91, including Tomasulo, and they didn't know where the hell we got this stuff and if I knew what I was doing, and they didn't want to take someone who had just been a graduate student and shred him. They would be willing to shred his adviser. So they were gunning for me. "Do I know what I am talking about? Where did we get the information?" I wanted to talk about the elegant model and simulated annealing and the area that how it had worked. They wouldn't let me get to slide 2. They wanted to know details of the model 91 and why I was concluding this and that and so on. So they said "Okay, this switch, CDC or IBM switch, what's that about?" "Well, that's the priority whether you use the CDC system or the Tomasulo algorithm in it, that you're scoreboarding a Tomasulo algorithm." Somebody said "Tomasulo algorithm, what's that?" And I looked at him, you everybody knows what that is, academics have been talking about it. It was classic this beautiful out of order scheduling with the reservation stations and the common data bus and the way that it worked. And so I said "That" and one of them looked at me and said, "Oh, Tomasulo algorithm", is that what you call it? What I didn't know it was that Tomasulo was sitting in front of them and he was tweaking Tomasulo and having fun with him. [chuckle]. So I am answering all this stuff seriously...

23:25 PE: You are the straight man in this kind of...

23:25 ED: Yeah, they are gaming. And, you know, where did the cost model come from? That was where I could point to a paper, we got it from here and said "Oh, that's very interesting. Where did the memory model come from?", because memory was interesting. Kumar discovered that there were two major flaws in the 360/91. One was that the floating point unit was very elegant and highly pipelined and very fast. The fixed point unit had very little investment in it. It was degenerative, it was not pipelined, it was relatively slow. Well they were designing a scientific machine and they thought scientific machines do floating point and so we will put all our money in floating point. What they forgot was that the fixed point, or what they seemed to have forgotten or they didn't take adequate accounting of, was the fixed point unit does all the address calculations for the accessing of the floating point numbers. And so this gorgeous floating unit is spending all its time waiting for the addresses for next operation. And that, and it can't... It's choked, it's not balanced. So if you cut out some of the floating point unit stuff, scale it down, make it cheaper, put more investment in the fixed point unit, the machine got faster. Then there was the matter of the memory, because it's waiting for memory all the time. It's got a 12 cycle memory in the machine, that can do several operations per clock. That's very slow. If you start making the memory faster, it turns out that it illustrated the hierarchal approach, because there was a local minimum, optimum at 7 cycle memory. But the thing really flew at a one cycle memory. And it could fully take advantage of the one cycle memory. We found out. We found the local and we found the optimum. And it was worth the price, you know for what we thought was the price for it. And the IBM guys were very intrigued with that, because they knew about the local. And because...

25:54 PE: They didn't know about the other optimum?

25:56 ED: They knew about that too. Well... But they couldn't get the one, that technology at that point. The machine was designed for a faster memory. Which was a thin film memory technology that IBM was developing, and it didn't get ready in time. And they were on the ship date, as we talked about, and they had to shove this slower existing dependable memory in and they knew it was unbalancing the machine, but they did it. Well they could have done it with cache, but there was no cache on that machine. Well that's how you get a one cycle memory on the cache.

00:26:41 PE: The C-A-C-H-E cache.

00:26:45 ED; Yeah. Later the 360/91 was not a commercial success. It was very expensive for what it could deliver in application. It was so unbalanced that for example when the Stanford Linear Accelerator got their 360/91, they ran it for almost a year before somebody noticed that the switch on the operator's console, it was a switch that was hidden, it was under the desk, for the maintenance engineer was switched to non-pipeline mode for almost a year. Because that's how you do certain maintenance, because it's easier to check what's going on. And nobody noticed. And then they finally, oh flip the switch. And it got a little bit faster, no as much as you would like. So I am concerned about this, because I did pipelining. I like pipelining to work. So the recovery from that was the model 195. Okay, they didn't do a cache, because cache was introduced in the model 85. That was the cache machine, that was a business machine which was expected to have large amounts of data. So they had a cache, not a good cache, early cache. It had very large blocks because you can read those directories at that time and that made it inefficient because you couldn't put many different blocks in the cache. So they learned to make smaller blocks. So they made smaller blocks and more of them and they built the 195 with the improved cache on the 85 and the CPU with some redesigns from the model 91. Now you got your fast memory. But it turns out if you have a fast memory there is a lot of overdesign in the 91, that's designed to try, however unsuccessfully, to overcome the slow memory. And you don't need a lot of that stuff anymore. And I don't know if they pipelined fixed point. You know we were able to show that the way they ported the technology, come out with the 195 was again this mixture of stuff moving in the right direction, but to a combination that they didn't fully understand how to play together. And they missed the mark again. It could have been better. Although the 195 I think was moderately successful machine.

00:29:23 ED: In years when I taught graduate computer architecture, I taught that advanced graduate computer architecture, quite a large number of times before one student finally came up to me said, "You know when you teach this course, you teach an awful lot of different computers, right". Yeah. "And you describe the details of the machines and what is good about them and what is wrong about them and so on, but all of the machines that you talk about are failures. What are you trying to teach us about computer architecture?" Commercial failures. And I hadn't thought about that before and hadn't been asked that before. Being an academic what I was teaching was stuff where I could find source material that described what I wanted to talk about for machines that had interesting features which are usually the first ones to introduce those features and those are usually failures. If they are really successes, they don't publish a lot of the internal details. And the first ones are the ones where the ego involvement of the designers is such that they want to tell people about it. After that it's all past, it's not new and they don't describe it that much. And the first ones are likely to be failures because you got to get everything right. So that's where the material was. And so what I wanted to do is, if you live in this space of possible designs, it doesn't matter what's a failure or what's not a failure and the numbers change, but the models carry over to the new numbers and if you understand how to balance and what the functional space is, you can take this information from the failures and pioneering ideas and put together a success. That's what I was trying to teach.

00:31:28 ED: So I got very unhappy with the overdoing of the quantitative approach, when I would give a seminar about caches and show curves and functions, and show the functional dependence of the performance and the cost on different parameter choices and the way the curve, the shape of the curve, that doesn't change as you change parameters. But the actual performance and the actual cost changes with technology. And then people would say, "Okay, well you've done this study, how big a cache should I design?" They want the number. Well I can give the number for the sort of toy academic parameters that I plugged into my model. But if you are an industry and you know what the real stuff is this week, why don't plug those into my formulas and my methods and get the right answer. You don't want my answer from my toy academic study. That's garbage. And that's all they wanted to hear, they didn't want to understand the thing, the creation and how to use it; the tool. It's mindless. So I applied the Hennessey and Patterson quantitative approach and I think I was one of the many many people using that quantitative approach. We used McPherson and tried to optimize things and numbers better than pounding fists on the table and arguing. But I don't believe the actual numbers in the bar chart. I want to do it with the real numbers. And so to me it's the functional. So when I gave the, when Pradip Bose who was one of my students who went to IBM research and started...

00:33:28 PE: Just spell the name please.

00:33:30 ED: P-R-A-D-I-P B-O-S-E. He was at IBM, Yorktown, still is. And he started so many things in the industry, but he started the ISPASS conference. I-S-P-A-S-S. International Symposium on Performance Analysis of Systems and Software. I gave the keynote there and that was in the heyday of the Hennessey and Patterson book and Hennessey...

34:12 PE: Can you put a date to that, when it was?

34:16 ED: I can. But it's in my resume, I'm sorry. [chuckle] Probably I was invited... April 2000. That was couple of months before I retired. That's pretty late. So the Hennessey and Patterson quantitative approach, which did a wonderful transformation of the curriculum and the field, and those guys were prophets for doing that, and it was about time. That's better than not having any quantitative approach. But I thought it had gone too far. And I thought the way... You know, usually the original prophet has some idea and the zealots who follow don't quite get the idea and they take it to extremes and the extremes are damaging and bad. And it had been very easy by then to do simulations without having any real simulation model. Do a toy simulation model and generate bar graphs and publish them. And so in that, I titled my keynote toward a more functional approach. Where I was trying to play off the quantitative approach wording and say okay numbers are fine, we need to know the functional relationships, we need to know what Kumar talked about as the frame of validity of a model and not get outside the frame of validity and use the same numbers which don't apply outside that frame. You are going to crash and burn if you do that. And just understand the function. And I liked the double meaning of the word function, it’s functional and has functions and has a functional space and it works. The bar chart method which had become popular after that...

36:23 PE: After...

Interview recording part 5

36:23 ED: After the quantitative approach. If your goal is to the maximize publications, because he with the most publications gets tenure, the quickest way to generate a research paper in that era with the peer-review community, was get six random ideas for how you might solve a particular problem, simulate them all with a model that you have not designed and have not even vetted for your purpose, that you got from somewhere else, probably the toy model is provided with SimpleScalar, a wonderful simulation facility done by Todd Austin. But his toy model that he did to illustrate the tool was used as a real computer model by too many people who never even knew what it did. And was never intended for that purpose. Okay, so you generate six random ideas, simulate them all, draw a bar chart and I guarantee you that one of the bars... And on whatever toy example you want to do, I guarantee that one of the bars will be higher than the other five. And now you publish that the result which is, that's the best way to do it. This is content free. There is no information there.

37:43 PE: It's very interesting. What I was going to say about this paper...

37:46 ED: It resonated a huge amount with the audience and they came and confessed to me one by one they used bar charts. And I said hell I've used bar charts. Bar charts are good if they have real information. That wasn't the hit at the conference. The hit of the conference was this guy from the defense department in Australia who gave the last talk and who was forced in. He didn't really want to take the paper, that somebody was financially dependent on him and said we'll let this guy talk because he wants to come from Australia. And he started showing slides that had no content, he was droning on and on and going way over the end of the conference, until somebody, somebody merciful went out into the hall and pulled the fire alarm and we left and that was the end of the conference. [laughter]

38:34 PE: I'm going to remember that.

38:38 ED: But what did you want to ask about the paper?

38:40 PE: Well, it's more of a comment than a question. But I just, this is point reading through your work, reading this Kumar-Davidson paper that I start to see this pattern here that you are, you know as time goes on... These machines were already complicated when you first began working on them. But at this point, they have become so incredibly complicated that there's no way to deterministically analyze what they do. So here, and especially later on, we begin to get into statistical analysis of the outputs as a way of getting at what is going on inside them. And I think that this a fascinating thing, that these machines are so complicated that we can't really tell exactly what they are doing and optimization becomes more of a trade-off among things, then you just see what happens.

39:37 ED: Yeah. Well beyond that you also can't tell particularly if they are successful machines and transformational machines. You can't tell what anybody is going to try to use them to do. You don't know the application space. That's the hard thing. And no engineers want to get into the application space. That's... Oh, that's dirty.

39:55 PE: you always see this with benchmarks, that they give you a list of performance benchmarks for some processor and then somebody will come on and test them in applications and you get really different results from what the benchmark seem to say.

Interview recording part 5

Shouldn't be happening.

40:08 ED: Yeah. Well the applications might not be properly tuned. The benchmarks tell you the performance of the benchmarks. And to the extent that the community has agreed on what the benchmark should be, they are at least the standard metric to get to use from machine to machine. So you have a consistent point of comparison. Whether it is rational is another question. And everybody argues... So the, so Konrad Lai and oh, his partner in crime at Intel, whose name I have now forgotten caught me at a conference one time and said when you start a startup company, the most critical employee to determine success or failure is the benchmark. And he is the guy that develops the benchmark with suite for your new product. And he said that guy is critical. Because what's really critical is, as you might guess, whether anybody buys the machine. That determines success or failure of the company. So how do you convince somebody to buy the machine. Well attached to marketing, you have somebody with some technical savvy and also awful lot of showmanship skills, who does benchmarking, picks out the benchmarks that make the machine look best competitively and then somehow through slight of hand and slick talk convinces the customer that those particular benchmarks are the ones that represent his workload best. And that is characteristic of the successful startups. [laughter].

41:59 PE: Let me pry you back to Illinois and let's talk about people now. Who were your colleagues? You talked some about students you had while you were there. Now, your most important colleagues and...

42:16 ED: Yeah. When I started Gary Metze, Bob Chen, Franco Preparata were there. Bob Chen however was Director of the Coordinated Science Lab. So he was taken out of most of the teaching and certainly out of curriculum development. And Metze and Preparata didn't really get along. In fact Metze and Chen didn't really get along. And now we want to start Computer Engineering and it has Ed Ernst. Ed Ernst was chairing the Computer Engineering degree program which was an Area Committee. Most Area Committees were in their area. In Computer Engineering that area was also a degree program, because there was an Electrical Engineering degree and a Computer Engineering degree. So it was more than an Area Committee, but its chair had no administrative title. Except Ernst happened to be associate chair of the department. But he didn't know computers. He is in radio science, he was just a good guy. Was it radio science or electro magnetic? Well they are closely related.

[laughter]

43:33 ED: So I went to Illinois without tenure, because Illinois does not like to hire anyone with tenure; someone who doesn't already have tenure some place else or a very senior person from the industry. So I came without tenure with a promise that I would be promoted to tenure the next year. That was Jordan's promise to me. Well I am vulnerable. You know at that stage you're vulnerable. Ernst says me... Oh and coming at the same time was David Waltz who is in Artificial Intelligence from MIT. And he is junior. So I am little more senior and you can't have the other guys Chair the committee because they don't belong. So Ernst says to me, would I be willing to chair the Computer Engineering committee. I said "You got to be kidding. I am an Assistant Professor, I'm vulnerable. You're telling me that the senior faculty in the area don't get along and you want me to run the committee, to push them around? I'm going to die." He said, "I'll have your back. Do it." I said "Okay". So I tried to do it without antagonizing everybody and anybody, and the way I did it was I would have a meeting of our committee and I would say, "These are the issues that we have to resolve today and we will resolve them today and we will stay in this room until we resolve it. And you can come to the meeting or not as you wish and if you don't like what's going on you're welcome to leave anytime you want. But whoever is left in the room, when we get consensus among the people who are left in the room, that will make the decision." And we moved forward. I thought that was very creative. I invented that myself [laughter].

00:45:36 ED: And we had a battle between entrenched camps because we were stomping on faculty from other territories who taught courses that were related to us. And we were trying to change degree requirements. We had a potential enemy in Computer Science, we had people who taught the Logic Lab, Morrie Babcock was one of those, actually he taught it. By the end of the semester the Logic Lab students learned how to design a NAND gate, from transistors and resistors and capacitors and stuff. We could get over by the end of the semester, the students were building a little four function calculator. And Babcock took me aside and he said "Look, I know what you're trying to do". He said "But you... You know, you are a smart young guy, but you're dead wrong". And he said "There are only two things about computers. There's devices which we talked about before, the components, the building blocks, and there's wire. And any idiot can wire things together, there's no knowledge in that. The knowledge that students have to understand is the device and how the device works." So he taught devices in what was a Logic Design lab. I have no problem in students learning about devices but that's a device lab. I'm teaching a Logic Design Lab. You know, truth and labeling. So I thought he was just an old dinosaur and we had fun. We took the lab and you know kicked him out and we took it and we taught it. We took no prisoners. We introduced the full curriculum. And we kept hiring.

47:32 PE: Were you patterning this or anything else on the other Computer Engineering programs around the country that you were...

47:38 ED: Well there was this segment of the Stanford curriculum I was familiar with. I mean Dave Waltz and I introduced some AI courses. You know, we only had so many faculties that we were developing enough that we can teach. And yeah, there was the COSINE group from ACM that produced a standard curriculum, we took bits and pieces out of that. Jacob Abraham joined us next year, he was one of the McCluskey students from Stanford, he came in the reliability area, but he understood computer architecture. He eventually introduced the VLSI design course patterned after the Mead and Conway stuff. We started a computer organization course. I introduced a, I developed that. I introduced the graduate architecture course which is like reading papers and talking about machines. Dick Brown from physics, who was interested in computers, he was sort of the resident computer guru in the physics department. He used to do infrastructure for their Fermi Lab experiments and so on, he got involved. He said when he did teaching, he'd rather teach Computer Engineering than Physics. So he came over, he started. He kept his appointment in the physics department but he'd teach our students. And he and somebody else developed an assembly language programming course that we taught in Electrical Engineering. Well now it's got the word programming in it and Computer Science teaches programming; we're in their turf. So they started getting edgy. The problem is their assembly language programming course taught how to do arithmetic functions and recursive algorithms and things that computer scientists do in higher level languages. And so they're trying to teach assembly language so you understand the structure that the compiler would be implementing for the type of algorithms that they do. When we taught assembly language we focused on the I/O drivers. And the parts of the machine that normal programmers don't touch as a way of teaching how the machine was working.

00:50:15 ED: There was one experiment that Brown designed which I thought was sheer genius. We didn't have much money behind these labs. So we had a small computer and we had a console with a CRT screen and a keyboard. [chuckle]. So what we said is we want somebody to be able to type in a description of something that they wanted displayed on the screen. And your job is to write a program that reads in the keystrokes of the information being typed in and display that thing on the screen; that image, that picture. Well it sounds pretty degenerate, simple, you understand one piece and other and you just do it. The trick was, and why this was a fabulous learning experience, is that the project had to be done by two people, two students. One student would focus on the input from the keyboard and store the information that they read in some intermediate format in the computer memory. And the other person would read that stored intermediate format and generate the character graphics image that was going to appear on the screen.

Now, the trick is we told them they could use any intermediate format they wished and it was up to the two of them to decide what they wanted to use. And we didn't give them any more guidance than that. And what they learned as Engineers is that when they think they have agreed on an intermediate format, they have no idea how mismatched they have been in that alleged agreement. And how they must communicate and check over and over and over again, does this mean this, does this mean this, how you're you handling this case. In the design process they don't meet one day, design an intermediate format, go away each to their pieces, throw it together on the day of the demo and expect it to work. It's the scene between two human beings. Like the cable harness and the paper tape drive, that's a scene between the guy who designed and specified it and the guy in the factory that harnessed those signals together.

53:03 PE: I like to talk to my students about standards a lot and what I always say is, you might think that when you see a standard you've seen the end of it, but that's only the beginning of the story. Then you've got the implementation and it's almost universal that implementations begin to diverge. And then you have to figure out whether your implementation really represents the standard.

53:25 ED: Yes. And the standard doesn't fully specify. And the standard for most things doesn't say “You have to build it this way.” The standard for most things says it has to behave as if you've built it this way and the standard's incomplete. It's a human process. So to me I have always been depressed that more women don't go into Computer Science and Computer Engineering. Now I'll be a male chauvinist. My conception still is that more women spend a bit more time at home with kids and family than most men do. And there may be stages in their lives when they're not working full time because of this. Now I know about the egalitarian marriages, I was not part of one, but for someone where it applies. In computers there's such a market. And people can work full time, part time, they can work from home, they can work at a different city, you can solve two body problems by not just physically being there, particularly today. It's a great field. And it's relatively clean, you don't have to get dirty. And there's immediate gratification. So do it in pieces and there's a billion pieces, you never run out of pieces. Surveys have said that women tend not to go into it because the high school kids they knew that played games were geeks. And there's a lot of stuff now the geeks are the ones who ultimately win, some of them do and some of them are just degenerate, there's no guarantee. But there's also the feeling that Engineers and Computer people relate to the machine and they don't relate to other human beings, and more women are more concerned than more men about relating to other people. I hope I've given the impression that everything I've ever done in my life is relating to other people.

55:43 PE: That's very clear.

55:44 ED: It's human, it's I mean a team. It's nothing like us against the world, you know. That's the drawn...

55:56 PE: Yeah. I am gonna need to close up this segment in a couple of minutes. We have not talked yet about some other things you did while you were at Illinois. This, what I know nothing about. We can see the research. And especially the Center for Supercomputing Research and Development. Was that the same thing as NCSA?

56:16 ED: No. Interesting. We were the other one, we built the CEDAR. NCSA bought machines other people built and sold the service. I would like to spend a little time talking about some of my other colleagues in the VHSIC program, and the CEDAR.

56:34 PE: Yeah. Okay.

56:37 ED: And we're done for today. Okay.

[Interview recording part 6 begins here]

0:00:02 Paul Edwards: This is Paul Edwards interviewing Ed Davidson, this is Part 6. We have had a hiatus of more than three months it’s September 29th, 2009. Both of us had things to do this summer and now we're back together talking about the last part of Ed's career. And we are going to start at Illinois, we have been discussing here work at the Coordinated Science Laboratory and you had a plan for where you wanted to begin. So please go ahead.

0:00:32 Ed Davidson: Well, I talked about coming back to Illinois from Stanford and I talked about the curriculum work in Computer Engineering.

0:00:46 ED: And we didn't get much into the research at that time. The first phase of the research was finishing with the students who came with me from Stanford. And we gradually developed a group in Computer Architecture and Reliability of about five faculty. Jacob Abraham came the year after I did, or a couple of years after I did. Gary Metze was leader of the group. And there were people in other areas that joined, David Waltz in artificial intelligence and some theory people as well. And then we, our group was running on basically $100,000 or $150,000 a year from the Joint Services Electronics Project, which was post World War II funding, that I perhaps talked about when I was at Stanford. There was another lab at Illinois which was the Coordinated Science Lab which was the JSEP Lab. And we were developing the curriculum.

Interview recording part 6

0:02:04 PE: Science and Engineering, what's the P.

0:02:07 ED: JSEP was Joint Services Electronics Program, joint services being Army, Navy...

0:02:13 PE: Electronics Program, right.

0:02:15 ED: Air Force. CSL was called the Coordinated Science Lab; it had been the Control Systems Lab during World War II and did some early radar and computer stuff. And then it broadened with solid state controls and computers, they didn't know what to call it, they wanted to keep the initials because they had all the stationary printed. So it became the Coordinated Science Lab. A very uncoordinated lab, of about 35 faculty and one computer, a PDP-10. It was a delightful place to work because it was a place where you could do research. It was a research lab, separate from the academic department. It was a problematic place to work because the College of Engineering Administration was run by the departments. Well, it was run by the Dean, but to the extent that he never consulted anybody. In the department the Coordinated Science Lab was neither fish nor fallow; it kind of didn't fit. But we had our own way there and we had a hideout where we could do research. And it was nice because you had one boss, the department chair, to report to for the teaching and the curriculum and the education and another boss for the research. So we kept in two nice piles, and whenever you're sick of one you could hide out in the other place for a while.

00:03:47 ED: So, in that way, you know, we finished some of the Illinois research with... I finished some of the Illinois research with Janak Patel, Thampy Thomas and Dan Weller, who had their finals in one glorious day when we all flew back to Stanford and just drafted one final exam after another. And that was that; they finished their Stanford degrees. And then so starting in 1973 we were going in that mode until about 1979, Jacob was doing reliability work as was Gary Metze, I was doing architecture stuff. In the process we hired Janak Patel, one of my Stanford students who finished at Illinois and then went to Purdue and then we hired him back at Illinois as one of the faculty. And Bob Rau, who preferred to stay at Stanford and work with Mike Flynn for a PhD rather than come to Illinois with me, finished his degree with Mike Flynn and we hired him on the faculty at Illinois. So we had a nice little group. And Mike Schlansker came just after that. So we had...

0:05:01 PE: S-C-H-L-A-N-S-K-E-R.

0:05:06 ED: Yes. So, we had six of us by that time, and Jacob Abraham took a specialty course taught by Lynn Conway on the Mead-Conway approach to the VLSI design. And we haven't been focused on the technology of the design. I had introduced the microprocessor laboratory which I talked about a bit, where students did detailed design of a microprocessor system but it wasn't one chip; it was the next level down in integration. And Jacob brought back the Mead-Conway VLSI course. I had the pleasure teaching it one semester, although I don't think the students had a great deal of pleasure taking it from me. And I maybe did it a second time, just to see if I could get it right. And we were moving in there. So, Jacob came back one day and said there is this new program in Washington called VHSIC, Very High Speed Integrated Circuits. Everything at that time had to have a V for very something in it, and so they came up with that acronym, funded by the military, huge amounts of money being invested and DOD had put in a restriction that industries had to partner with the universities to get the big grants.

0:06:45 PE: Now was this part of the DARPA's strategic computing program or is it a little before that? Think it might have been a little...

0:06:52 ED: Might have been a little bit before... They were certainly related.

0:06:57 PE: Some of the same people moved into strategic computing later but...

0:07:00 ED: I think so some of them did, yeah. I never had a DARPA contract. This... Our contract monitor ended up being in the Navy, JP Lettelier, who was just fantastic at Office of Naval Research, which predated the National Science Foundation.

0:07:19 PE: Yes.

0:07:19 ED: And NSF was largely patterned after how well Illinois ran that show.

0:07:24 Yes.

0:07:26 ED: Very research sensitive, technically confident organization. So, we had never done a research proposal for anything larger than $50,000 at that time, usually $35,000 or so. And we were living off of those. I had some funding from NSF as well in those days. From... John Lehman. So Jacob said that the amount of funding we could get was just astronomical. Illinois was a big place; had a lot of faculty and a lot of students. So we could do a large scale effort, we had the resources available in terms of space and students and faculty. But we never tried anything like that. And Jacob had met Dennis Buss from Texas Instruments who was interested in partnering with us and he went down...

0:08:43 PE: B-U-S?

0:08:43 ED: B-U-S-S. We went down to Austin and talked with Dennis. And I remember in that meeting he was saying, well what can you bring to the table. Here is the stuff we wanted to do, what can you bring to the table? So, well we can bring reliability and performance. [chuckle] there was no lack of confidence in this group. And we were pretty good in reliability and pipeline and parallelism, architecture. And oh, Illinois was a world leading center in reliability with what Gary Metze had done and Sundaram Seshu before him and now Jacob. So, they thought it was interesting and that if fit. And that the detail content of the proposal wasn't that important. What was important was the partnership and pitching it and showing that we could mount the scale of effort that was required and how much money did we want? So we said, well how much can we get? And he said, how much do you need? And we started talking and we said, well let's figure it out. You know, we could each supervise seven or eight graduate students. So, for six of us at that time that would be pushing, you know, 35 or so graduate students. Gary Metze later dropped out, so there were five of us and we settled on a 35 graduate student number. They'd need a computer, we couldn't run on a DEC 10 and Digital was just coming out with the VAX, which was clearly becoming the academic standard group computer. So we said well...

0:10:39 PE: Once more on the date for this. This is around...?

Interview recording part 6

0:10:43 ED: The contract started in the fall of 1980.

0:10:46 PE: Okay.

0:10:46 ED: So, this was fall of 79 when these discussions were going on.

0:10:51 PE: Okay.

0:10:53 ED: So, Buss finally took us off of our academic ramblings and started to put pen to paper and write down numbers. And we figured overhead and we figured publications, you know. And he knew the budget categories. We had no idea so he said, well, you know, how much would you need for this only for travel and publication, and so on? And we started adding it up piece by piece and we got up to million and a half dollars for a five year project with no trouble at all. And that was so far off scale to anything we had done, we started laughing, there were tears rolling down on our faces. Then we said, well is that enough? And he said well, do you need more? And we invented a slightly more glorious program and we settled at $2.5 million. And we wrote the proposal. And we went and gave a pitch on it and I went on sabbatical for a semester in spring of 1980, Winter term 1980. Spring term I guess the rest of the world calls it. Michigan calls it Winter term.

0:12:12 PE: Just because we only have winter and summer here.

0:12:13 ED: Yes. And while I was on sabbatical, the grant came in. It was, you know, like a short letter that said "funded." There was no discussion, there was no arbitration, there was nothing they funded it to the full extent. And I came back from sabbatical mid summer and we looked at each other and said, "What are we going to do now? This thing starts in a month". And we talked to... It was a transformation that's hard to believe today, you know, I think. The 35 faculty ran on a DEC 10 computer you could put in the hall and use... Go to a terminal room and type stuff for the DEC 10. We didn't have any room. We had six faculty offices and a couple of offices for graduate students. No computer room certainly. We went to the lab director and said, we need... I don't, know 2000 feet of floor space, some large percentage of the building. About half of the second floor of the building to be honest; it was a six floor building. The top four floors were smaller than the first two and we wanted half of the second floor. He says, "I can't do that" and we said, "Well, we can't deliver on this contract unless we have the space. Because we've made a lot of commitments and you signed a proposal and we've got a lot of people starting in a month."

[pause]

0:13:58 PE: You were talking about the building, second floor of the building.

0:14:02 ED: Okay. So, we said we needed to do the work and we don't want to default on this contract. And they dawdled and we said, "Well, we really want to do this and what we did is, we went on a road show and talked to four or five other universities about would they like to hire five faculty and 35 graduate students to do the project there." And the upshot of it is that few places went "We can't do this", although they all wanted to talk about it. Carnegie Mellon wanted to pick the top and the more senior faculty out of the group and do the project with

Carnegie Mellon students and just forget the Illinois students, leave them behind. And University of Texas said they'd take the whole thing. Because Texas at that time was a little short on research money. But they had space because they owned oil wells and the oil well money at that time had to be spent on buildings. So they had lots of space.

0:15:23 PE: So this was kind of a game of administrative chicken, where you tried to get Illinois to say yes and...

0:15:31 ED: Well, it, it was an ego trip. It was an anxious ego trip, and I like to believe it really was motivated by how could we do this work. And to be sure some anger and petulance that Illinois having been so excited for us was not giving us the resources we needed to do the work. But it ended up being chicken and we kind of didn't want to go to Texas; it would have been a tangent. And we knew to get going and Illinois cleared out half of the second floor of the building and gave it to us. And we said "deal." [laughter] That's all we had to do. I had to move the shop, and you know, move some people around and squeeze some other people. And then the other faculty in the building said you cannot have a group that has a computer, because we are one and all. And we said "Look, we need 100% of this computer and besides, we're buying another one since the computer room's so big [laughter] that we can get." We borrowed a 1700 and we bought a small Hewlett Packard machine. So we said "We made our commitment, so we can saturate 100%. This is not a lab facility, this is a group facility." And that was in the beginning of group computing in Illinois. And we had terminals in our own offices. Terminals. Not machines. But the luxury being able to sit in your office and type on the computer was just unbelievable. If other people had things they wanted to run on our computer, they could ask us, and we would let them. We had to negotiate separately for our own backups. And we found to our horror one time when a disk crashed that the people we had paid to do backups never actually backed up anything. So we lost like a month of research. [chuckle]

0:17:44 PE: That's incredible.

0:17:46 ED: You know, this was early days. Early days. But that made our group. That put us on the map. And it put CSL on the map in this area. Very seriously. We had a high percentage of papers and the major conferences on reliability and architecture. Very, very heady time. So, that was also settled before the fall semester began. In fact, it was settled about two weeks before the fall semester began. And in those two weeks my co-faculty and I went through all the applicants for graduate school, as well as the students that we had working with us already. And we picked out the 35 that we wanted on the project and called them up and offered them assistantships and two weeks later we had 35 research assistants on the floor working on the project. A very, very heady time.

0:18:58 PE: What was it called? Did you have a name?

0:19:02 ED: The project? You know... [chuckle] It's amazing I was never very good with names. And being in the Midwest and sort of laid-back and becoming a Midwesterner, we didn't think in those terms. We were not marketing people. And that's really stupid. [chuckle] Because people need a name to hang onto. We did make up names, but they were never the right names. And a lot of the research that I did which... I have it in my short one-page bio, did early work in the fields of... and I mentioned the right names, all of which we've given to those research ideas by other people concurrently or subsequently. We never got the right names. What did we call it? I have no idea. [laughter]

0:20:00 PE: It sounds like the name wasn't so important.

0:20:02 ED: Well, it wasn't to us. The work was important. And so, there was stuff done on reliability techniques, how to recover people... In the VHSIC world people were concerned about single event upsets and how small particle radiation could upset the state of a computer without destroying the circuitry. So that you'd have a recovery possibility. And Jacob Abraham and Janak Patel did some very fundamental work on how you can keep running and recover. Lots of people published papers on reliability involving redundancy and efficient redundancy, or recovery techniques; checkpointing and recovery. None of that is the hardest problem. The hardest problem is figuring out that something has gone wrong, detecting that the computation is no longer correct, it's been corrupted. And very few people have done work on that. And our group did. It was some of the really solid pioneering work. I can brag about it. I wasn't part of it.

0:21:33 PE: Well, talk a bit about what you found out from doing that. How you were able to detect corruption?

0:21:39 ED: Well, I don't know... It was early work and, and fairly pioneering. I don't that much of those techniques became standard. People sort of muddled through, and you know, reliability, I don't know how much is done in computing today, but reliability it seems to me that it's mostly, if you notice that something's wrong try it again. But Janak developed a technique called RESO, Re-computing with Shifted Operands. For a lot of the arithmetic operations, you can shift the operands over and try again and you get the same results shifted over basically, but it's using different hardware on the circuit elements. And if they don't match then you get something wrong. That was a clever idea. It's time redundancy and people don't like to spend much on redundancy. And we developed a number of coding techniques. In a related contract on a single event upset in memory, we developed a coding technique in memory for distributing the information in the circuits in an unconventional pattern and putting in just the right amount of checking to be able to detect the areas. This is a fairly efficient technique. Whether anybody used it or not I don't know. We did it, we published it. So we were not particularly good... Well, I wasn't certainly at marketing and getting people to use this stuff. Janak Patel did one of the early cache coherence pieces of work. Jim Goodman is generally credit with the first publication on the coherence problem involving a solution which he calls Snooping Cache. Janak, in about the same timeframe, developed a protocol that was adopted as the first standard for reliable cache.

0:23:58 PE: What does coherence mean in that context?

0:24:01 ED: [chuckle] I should have a tight definition off the top of my head, but the fact is I don't. It's... Coherence is in situations where you have multiple writers of the same information. Different processors or processes may be reading and writing the same word of data or information in the memory. Data. So it's important that when you read you get the most recent updated copy. And when you write based on the information that you read that nobody else has read in the interim getting an improper copy. So that if you intend to write you need to lock the information up so that other people can't read until you write. So, there's a consistent ordering throughout the system of reads and writes that makes sense. So someone said that the name was really mis-named, the coherence problem should properly be called the consistency problem, and memory consistency should probably be called memory coherence, that they got the word backwards. And don't ask me what the other one is. [laughter] But that's basically it. You can reconstruct after the fact a coherent timeline of reads and writes that makes sense. A snooping cache is where the multiple readers and writers share a single bus. So they can snoop on each other's reads and writes and know when they got to wait and when they could go. Okay. I worked more on a architecture side. One of the things we did during that time was a thing I called, or our group called structured memory access computing. Jim Smith... about the same time was doing decoupled access and execute. And he... As we sorted out afterwards he had gotten there first, and we had developed some modifications, that the structured memory access was something that we did. And Guri Sohi, who's now a professor at Wisconsin, was one of the people who worked on that. And the idea was that...

0:26:58 PE: S-O-H-I.

0:26:59 ED: S-O-H-I. The idea of that was that when you do heavy pipelining... I should have talked about that first, but I guess I can talk about it afterwards. Much of our work at that time was what I tried to call, with humor, "making the world safe for very long pipelines." We were trying to figure out how to get the pipelines longer and longer, and one of the difficulties there is that you have large wait times when you need something, because your clock is very fast and you're getting data from outside. So anytime you have a dependence problem or a branch problem or a memory access problem you have to wait for a large number of clock cycles before you get the information you want to operate on. So pipelines don't work well in that situation because you have trouble keeping them full of useful work. Because you're starting new things before you finish old things. So how do you get the information? How do you keep it flowing? So, in the structured memory access we... Jim Smith had done the idea of decoupling the access process from the execute process. The access process is what in those days would typically run in an I box, the instruction handling box, which fetched instructions and decoded them and figured out what they're doing and the execute would typically be going on in the E box which later became a collection of functional units; additions and multiplications and operations on data.

00:28:39 ED: Well the I box needs to get fairly far ahead of the E box. You can do branches in the I box if they're not dependent on data that you haven't calculated yet. So how do you organize such a system? Jim Smith came up with the idea, Jim Smith's group came up with the idea of decoupling the access process and the execute process, taking a conventional program and splitting it, so that the access parts and the execute parts were essentially today you would say they were in different streams and run in different boxes, and then they would coordinate every now and then. And you would like the access process to run ahead. And what we said "What if you had only one kind of loop?" Since a lot of the numerical stuff that we were looking at was loop oriented. So we said... "what if it was an infinite do-loop?" And you just go ahead and access. Don't worry about when it's going to stop because typically the major test in the do-loop is whether you want to stop after this iteration. Sometimes you can compute that ahead of time. Do from equals 1 to 10, and you know you do it ten times. But "do while," "do until convergence" and stuff like that, which is a more important case, you don't know. So we said "Okay, we'll do an infinite Do loop and then we'll fetch a few iterations ahead, we'll get memory errors, because we violate bounds, or whatever happens. But we'll know eventually that we meant to stop at a certain place and we'll throw everything that came after that. So instead of waiting all the time, at the end of the loop we skid ahead a little too far before we came to a full stop and we'd throw the skid stuff away." Make a lot of sense. And I get very excited about that. That's like the making the world safe for long pipelines. And we did that in a large number of ways. There was some earlier work at Stanford by Arvid Larsen, with me, where we looked at cost performance of pipelines. I don't recall whether I talked about that before.

0:30:57 PE: I have it in my notes here, but I don't recall from the transcript...

0:31:02 ED: Well, let me say briefly...

0:31:04 PE: Yeah, I think you should.

0:31:04 ED: The performance. As you cut the pipeline finer and finer... Oh imagine what a pipeline is. You're doing some functional operation; like in a manufacturing analogy, building a car in a factory. And instead of building one car at a time you do an assembly line, like Ford did. You begin the next car before you've finished the earlier car, and you develop specialized work stations that do one step of the manufacturing process and the car flows through, and gets built in one specialized station after another with specialized people at those stations. So on a computer you build specialized stations and the work flows through. So you have reading... You have fetching an instruction, decoding an instruction, reading from registers, performing an operation, storing the result; typically is a simple pipeline that people do today. But basically if you look at it as one large combinational circuit with a register at the end, and then you decide "How many pieces would I like to slice this circuit into?" If you slice it halfway through in time and put a register there, you can now operate it as a two stage pipeline. Slice it in three stages; insert two registers and you operate it as a three stage pipeline. So basically what's happening, as you slice it finer and finer, you have the combinatorial time, the time for the gates of the circuits, for the execution which is getting divided finer and finer, that's getting divided in half, in a third, in fourths, and so on. But each time you divide it you're inserting a register. And there's some time taken for operating the register. So you're adding, you're slicing the pipeline thinner, but if you're cutting it into N pieces you don't get N times the performance, because you're adding these register operation times as an overhead there. Result is that performance goes up like an asymptote. It rises fast and then it levels off, and approaches....

00:33:21 ED: Cost? Nobody knows how to model costs, so we took the simplest model with Larsen. We said "suppose cost goes up linearly; all registers cost the same." So each time you cut, you're adding a register cost to the cost of the system. So cost is going up linearly, performance is asymptoting. So the question is where do you get the maximum cost performance? The costing is simplistic because there are other problems, other control problems get more complicated so it costs more than you think. Not all slices through the logic have the same number of bits or the same size register. Some have big ones, some have small ones. So there are other considerations. Let's take the first model. Cost goes up linearly, performance goes up to an asymptote. The asymptote by the way is the register overhead. You've sliced the logical into nothing, and you've got nothing but register overhead. Now you can't get better performance than that. So there's an optimum number of stages, which turns out to be... Now I recall, I think we did talk about it before, but I didn't have the optimality condition stated correctly. What you want to do is to spend the same amount of time in the registers as in the logic, and you want to spend about the same amount of money on the logic as the registers. The problem is... And then you would, if that comes out to be five stages, you don't get five times the cost performance; you might get two times the cost performance. And you don't get five times the performance, you might get three times the performance. Not as great a bargain. But in some cases you can get out to five or six stages before it dies.

00:35:22 ED: If you don't, you might not be able to cut at a point where you're spending both an equal amount of time in the registers as in the logic and an equal amount of money on the registers as in the logic, and if you equalize one before the other, the one at a small number of stages, the other not, not for a larger number of stages, the optimum point is the geometric mean between them, but the gain that you get in performance cost is flatter. And the ultimate thing if registers are very expensive and very slow, but logic is very cheap but extremely fast, the cost performance groove, or vice versa, cost performance groove turns out to be a step-function. Which I used to say is either a designer's dream or the designer's nightmare. Nothing that you do is wrong and nothing helps. [laughter] They're all the same in terms of cost performance. And cavalierly what we said is "What happens if you don't get enough performance this way?" So we said, "Well, in that case you have to do something degenerate", which we called parallelism. Build two copies of the thing and now you can double performance for double the cost, again ignoring memory bandwidth, ignoring overhead in control and so on. That basic model.

00:36:54 ED: So in VHSIC, along comes Phil Emma and Pete Hsu. They started working with Bob Rau but after a year or maybe two on the VHSIC project Bob Rau decided that he didn't like academia. He resented being an Assistant Professor, he didn't like the way Assistant Professors were treated by the institution, without respect. In Illinois at that time if you're an Assistant Professor, at least until you graduated your first few PhD students, you had to have a senior faculty member co-sign your thesis. You didn't get credit as the advisor. These are indignities he didn't appreciate. And in my subsequent administrative life I used a principle that I first heard of from Nino Masnari when he was, M-A-S-N-A-R-I, I believe, when he was at North Carolina. He had been somewhere else before that, maybe Dartmouth. But I met him when he was at North Carolina. He's a Michigan grad by the way .... And he was chairing the department. And he said that what he did was he believed in the principle of juniority. [laughter] That the arrogant senior people with all the power can bloody well take care of themselves. And one of the major jobs of the administration is to protect the junior people. Make sure they get to teach the good courses, make sure they get to teach the courses where they can get the best graduate students, and so on. You're launching their career, they need a defender. They need a friend in court. Make sure you don't overload them with the conditions that will guarantee you keep them from getting tenure. All of which was rampant at that time.

00:38:52 ED: So I did that in my three years as department chair. And when I was running the Computer Engineering committee at Illinois, we ran it in a very egalitarian way, because I'd started running it as a junior faculty member. So I was sympathetic to the problems at that time. Bob had these ideas that he wanted to do, which eventually... And he went to industry. He went to TRW. We started an array computer project, and eventually the larger group that he was part of got into political troubles. They were in Palo Alto, they insulted some people in Los Angeles and their group was closed. [laughter]. And Bob left and started Cydrome Corporation to build this computer. One of many innovative computers at that time...

0:39:51 PE: Is that C-Y-D-R-O-M-E.

0:39:54 ED: Yes. Thank you. And that was pushing VLIW, Very Long Instruction Word...

0:40:04 PE: another V.

0:40:05 ED: Yes. And incorporated some degree of what was eventually called software pipelining, where you arrange the software so that you have a large number of things to do at once.

0:40:18 PE: Yeah, that's the overlap with [Pentium chip designer] Bob Colwell, who did a good deal of that in the later part of his career.

0:40:23 ED: Ah, well, I claim that Peter Hsu, when he was my graduate student actually started with Bob Rau and then worked with me, invented the concept, not the name. [laughter] We were just scheduling. Loop scheduling I think we called it. The name I believe came from Monica Lamb whose PhD thesis was later at Stanford. Better than it? No question. And Ju-Hotang worked on that. J-U hyphen H-O-T-A-N-G, with me after Pete Hsu at Illinois. So... I'm losing my track, I'm sorry.

0:41:15 PE: You said VHSIC and then you went into VLIW.

0:41:29 ED: I opened too many left parentheses I'm afraid. [laughter] Pete Hsu. Bob Rau. Yeah. So Cydrome and other machines of that vintage, Elxsi being one of them, started by Thampy Thomas, one of my students and Len Shar and Kumar worked there, and became vice presidents in its last days, they did a parallel machine. Really pioneering machine with an operating system that was "Better than Unix" and that was the time when nobody would buy anything but Unix. Eventually they implemented Unix. They didn't have a marketing organization, they relied on Prime Computer to do their marketing and Prime went bankrupt and they had no marketing and the machine by that time was technologically obsolete. Nobody was building these big board computers any more. So they went under. And Cydrome went under. But Bob went on, worked at Hewlett Packard eventually got the Eckert Mauchly award, very brilliant guy.

00:42:52 ED: He had done some early work with me before he went to Mike Flynn, something that we never published. We'd always say, how fast should you clock a pipeline? Suppose you're in a situation where for economic reasons different stages take different amounts of time, how fast do you clock it? So we said, "Well, take the least common denominator of the stage times". One stage might take two clocks, one stage might take three, others might take one, and so on. And depending on the pattern of usage, maybe you don't use them all in order once each. You might get some good schedules this way. And we did a whole theory around that that Len Shar developed. And Bob was working with me about that time. What he proved is, suppose you doubled the clock speed, make the clock period half of the least common denominator of all the stages, could you gain any performance that way? And the answer was you could gain transient performance under stall conditions. They could have permanent payoff, but you would never gain any maximum steady state performance. He proved that. It was a very nice theory. I wished he'd come with me. [laughter]

00:44:08 ED: So, that was that period. And Phil Emma, who I inherited from Rau, which is how I got off into Rau and Pete Hsu, who I inherited from Rau, Phil said "Why don't we pick up the old Larsen theory and see what happens when there are data dependencies and branches that we can't eliminate?" The data dependence such that it's within a pipeline flow and you don't have independent stuff that you can schedule in between, and the branches... branch prediction was just beginning to go then. So you can consider the branches to be branches with branch prediction or you could consider them to be mis-predicted branches, and evaluate that way; the ones that caused the pipeline to choke and recover. And what he shows is that the effective links to the pipeline was up for maximum cost performance. Performance shape is now different. Performance shape as you increase stages goes up to a peak and turns down. So in the presence of data dependence and branches, if you make your pipeline too long the performance is worse, not just hanging on the asymptote. Now you throw in the cost considerations, it goes down a whole lot. It is now, with that understanding, very difficult to build a pipeline longer than about five stages in any practical situation, and have it be the effective link.

00:45:53 ED: This came up years ago when Pentium... When Intel did the Pentium 4, which had a... By then the boxes were in different places and the pipeline for doing an instruction was of various different lengths depending on what type of instruction it was, how many boxes it went through. So, it's hard to just calculate one pipe length. But give or take, the pipe length was up to, I don't know, fifteen to twenty stages. It looked obviously too long for their memory speed, the cache miss penalty and so on. And lo and behold, after they did the Pentium 4, some of the key designers published a paper saying that they could double the performance by doubling the length of the pipeline again. That seemed obviously wrong. I couldn't imagine anything that they could do. Now the overhead for speeding the pipeline and recovering from errors is through the roof on cost. And given Moore's Law, Intel wasn't particularly worried about cost at that time and they were just trying to find a way to use all these transistors that were popping up all over the chip. So overhead was not a problem unless it took time. They were way over the curve. And doubling again was just terrible. Well, we knew that when we first saw what the Pentium 4, it looked like it was overdone already, relative to Pentium 3, and it was not long before academics all over the place started jumping all over the paper saying, "Well, you haven't considered this effect, and you haven't considered that effect" and it turns out that it was wrong. Intel has backed off of that approach.

00:47:39 ED: So, making the world safe for long pipelines, and lessons learned there from... Let me do one other piece. Jo-Hotang took a look at the Cray II, this was after we did the structured memory access stuff. Structured was, we knew what access pattern was involved in the loop, and we would create an access mechanism that we'd just run ahead, and keep grabbing data for the loop. After that, the Cray II was announced and the problem with the Cray II, relative to Cray I, was it didn't have chaining, which says that you could not take the result... You're doing a vector operation. So you're operating on 64 pieces of data in a pipeline function unit, so it can chug for 64 times. If your vector registers are linked 64, could you say "Add vector 1 to vector register 2, produce the result, put the result in vector register 3". Well, 1 and 2 are each 64 pieces of data, you're doing an add component by component, you're generating a 64 component result, you can keep the pipeline busy for 64 times just doing one instruction. Looks great, Cray I did that, with shorter vector registers I believe. But... And they had chaining so that you could do that. A standard operation just all over the place and these scientific codes just multiply that. Multiply two numbers and add them to a sum. Multiply two more, add them to the accumulated sum. It's a dot product type of operation. Very, very common.

00:49:42 ED: So, if you can crack multiply add, you're in great shape. So one of the ways that Cray I did that was with chaining so that as the first multiply result was produced, it could go... In addition to going to vector register, it could go to the adder. And so these multiply results would come out and just go from there. Multiply to the adder and end up with the dot product that you can store in a scalar register then. Cray II didn't have chaining, which meant that you couldn't start the add until you had filled the vector register with multiply results. Then you could add them all up in a reduction operation that takes all the vector elements and register and adds them all together. What can you do?

00:50:47 ED: So, we showed, Jo-Hotang showed, that taking some of the early theory that Pete Hsu developed, that you could develop a loop where you start enough operations, you get enough loop iterations ahead of yourself, so that you would do multiply, multiply, multiply and then you would get to the add instruction just at the time that the first multiply results were coming out and they would go to the adder, you no longer need to chain. But now, if it takes for example, 5 iterations collapsed together to do that, you need some multiple of 5 iterations rather than some multiple of 5 times the vector register length before you reach steady state in the computation. But we were able to show that for the Livermore FORTRAN kernels, which were standard, simple benchmarks at the time for scientific codes, we could run all of those at peak possible speed on the Cray II without chaining. So they didn't need chaining. We had to use more registers, but they had enough registers. And maybe there were two that way but all the rest of them we could do.

00:52:31 ED: Years later when I was working on the Ford project, I went to Cray on a trip with Ford and halfway through the presentation they're trying to sell Ford a machine and I'm there as Ford's resident academic [chuckle] along for the ride. And one of the Cray people looked at me and said, "I know who you are. You're the guy that wrote that paper about the Cray II with the software pipeline approach." We called it cyclic scheduling, which is what Rau called it. So, he said, with the cyclic scheduling approach. And I said, with a big smile, "Yes I am." and he said, "Did you know that we implemented it in a later version of our compiler?" and I said, "No I didn't, that's great" and he said, "Well, it was a total disaster." [laughter]. I said, "I'm sorry, but that's impossible, it works". He said, "Well, it doesn't." and I said, "Why doesn't it work?" He said, "We never get the applications of our customers. We never reach steady state", and I bit my tongue and didn't say, "Well then, they're buying the wrong computer, aren't they?" If they're not reaching steady state, they don't have enough iterations to justify cyclic scheduling. They don't have enough iteration probably to even justify vector computing, because their loops are only going through 3 or 4 iterations before they quit. So, the amount of data that they're crunching...

0:54:36 PE: Which customers are they talking about?

0:54:37 ED: Well, their customers.

Interview recording part 6

0:54:39 PE: The National Laboratories and the climate centers and weather prediction, these are the main customers.

0:54:47 ED: You would think so.

0:54:48 PE: You'd think that those models would do a lot of that.

0:54:52 ED: Maybe they had to recompile their programs. So that they went through larger slots of data at once. Maybe the type of tuning that we did on the Ford project, we discovered how to make it work. But it took 5 years and 30 people. One of the lessons that we learned from the Cedar project, which we haven't gotten to, is that there are two major problems in parallel computing. One is that there isn't any parallel software, application software, and the other is that there isn't a good enough... Well crossbars which is to connect everything to everything else is very, very expensive. Jim Smith once reported the number of miles of wire in the Cray II crossbar. It's a large number of miles. Unbelievable. It's a hard problem. So one of the ways of doing that is if you have a machine that doesn't require large amounts of parallelism, you do it in chunks. And you get by one chunk with a small amount of local data and you move to the next chunk. So you're not stressing the memory too much and so on. I don't know how they wrote their code but their customer applications weren't making it.

0:56:14 PE: Interesting.

0:56:18 ED: So with Emma's research and with the limits to where pipeline was going, I began to feel that making the world safe for long pipelines was maybe not the right thing to do, and that there had to be something else. And that you had to find a way to do it with shorter amounts of work-in-progress. Parallelism was coming in and that looked interesting. Another thing was that there was a conference, one of the ISCA conferences; ISCA - International Symposium on Computer Architecture - was in Scandinavia. I believe that it was the one in Sweden and Josh Fischer was presenting his VLIW work and Maurice Wilkes was in the audience. And I had known Maurice, because some time before I got the, had the honor of presenting him with the Eckert Mauchly Award; he was one of the early recipients of that, and blew me away when he stood up and said how meaningful that award was. He had been recognized with so many awards by that time, and to say how meaningful that award was because he had worked with Eckert and Mauchly. So, I talked to Maurice about my ideas about structured memory access and my excitement about the VLIW work that Fischer was doing and he smiled...

0:58:05 PE: That was Bob Colwell's first big job, with Josh Fischer.

0:58:09 ED: Oh! Really.

0:58:11 PE: Yeah, they had company for about four or five years.

0:58:14 ED: Yeah, they built the VLIW computer, well VLIW was... Multiflow. And Fischer and Rau, after both companies collapsed, ended up at Hewlett Packard. I would say together, except that Fischer never actually moved to California. [laughter]. So, where was I? So Wilkes. I met Wilkes in the courtyard of this beautiful place where the conference was, and I talked to him about my excitement about this stuff. And he smiled at me in that cherubic way that he had, and said, "Well I think that's all wrong. And it's not going to make it" and that took the air out my balloon.

0:59:11 PE: I think what's all wrong, exactly?

0:59:13 ED: This long pipeline stuff. The complicated management in the computer pushing very long instruction words we had to link these things together and optimally schedule, doing the long pipeline stuff, we've got to manage it right. Doing the fancy branch prediction where you've got to get it all right and so on. And I said "Why?" He said, "Well in the history of computers up until today, most of the successes have been from being able to do very simple things, very, very fast. Not in finding... Not in developing things that are potentially are extremely fast but require complex compiling and tuning, and complex structures in the application codes in order to exploit them." And, you know, at least through the next decade or so he was quite right. [chuckle] And this stuff really didn't make it. But we had fun and we learned things.

The Cedar Supercomputer

01:00:38 ED: When the VHSIC Project came to an end, Rau left, Schlansker left with him. I was no longer chairing the Computer Engineering Area Committee, which happened around the time that Janak joined us. I was co-chair at Brown for awhile and in the interim between I had two sessions, Dick Brown died, he was a great guy to work with. A physicist who worked in high energy physics. We seduced him over to Computer Engineering, because he'd always worked with computers. He did bubble chamber analysis and built special purpose computers for it. He and I made a great team together, I loved working with him. So, the project was winding down, Dave Kuck was in Computer Science. And he and Duncan Lawrie and, who was like Kuck, a compiler person, and Ahmed Sameh, who was an applications guy and Dan Gajski, had been working on proposing the Cedar Computer and they traveled around the country and spent two years of their lives, a couple of years of their lives, giving seminars touting their approach to this cluster parallelism in the Cedar Computer. The name Cedar is another one of those names, like the tree, Cedar, which was named because one day they were, decided they had to have a name, they were sitting at a conference and looking at the horizon and there were some Cedar trees outside and Gajski said "Doesn't matter what the name is, call it anything. Call it Cedar" and that's what it became. [laughter]

1:03:04 PE: Does it stand for something, or is it....

1:03:05 ED: No, never did.

1:03:06 PE: Okay, was it spelled with capital letters?

1:03:11 ED: No, capital C-e-d-a-r, it was the tree.

1:03:15 PE: Good. So this is, would be... Can you put a date on that?

1:03:20 ED: You know, I'm trying to remember, my guess is it was 1984, probably late, probably Fall of '84.

1:03:27 PE: You said the project was winding down, and it started in 79, actually 80.

1:03:31 ED: 80. Yeah, 80 through 84, I guess, maybe this was 85. Probably, yeah... Late 84 early 85.

1:03:39 PE: Okay.

1:03:43 ED: They suddenly got their funding, Dan Gajski went to Dave Kuck and said, "You know"... And Gajski's the hardware guy, and he said to Kuck, "You know, I've been thinking about this and I don't want to build that machine." And that was a shock, because there it was, ready to go and one day Dave Kuck walked into my office and said "Do you want to run the hardware group for Cedar?"

1:04:17 PE: So he really didn't want to build that machine.

1:04:20 ED: Kuck, he really didn't. No, no he was not up for it. He liked doing his academic research and publishing at that scale. You know, he'd done some serious soul searching. And VHSIC was winding down. It was going into VHSIC Phase 2 where you had to partner with a defense corporation and build applied stuff, and that wasn't my thing. And poor JP Lettelier, he really tried to get us partnered. And, you know, I just didn't want to go there. The money for academia was less, it was going into implementation phases and I didn't appreciate the application. Other people, for whatever reason, didn't go their either. Jacob went onto a fine career in reliability, and you know, they found other sources. And I decided, well there's another phase. I'd gotten hooked on large scale research by that time, and talked to them a little bit and I said "Yeah, I'll do it".

[pause]

01:05:40 ED: So I joined the Cedar project, that was the end of teaching for some time. The four of us didn't teach during that period, except occasionally a little bit. We were not going to have any graduate students, we were going to focus on engineers, hired to build the machine. I had a group of six or eight full-time engineers working on hardware. We took Alliant Computer Systems, that was a decision that was made before that, A-l-l-i-a-n-t, built by one of the teams that was featured in Tracy Kidder's book *The Soul of a New Machine*, and they left and started. That was the conservative group, they left and started Alliant. The radical group was... started Convex [Computer Corp.]. So the Alliant was an eight processor system with a shared cache, shared bus, shared cache snoopy protocol. And backing memory, and we were going to take four of them, link them together to run one application with a shared global memory that we'd decided early on, my group decided early on, would not be a backing memory, but would be a separate memory that the processor could address directly outside of the cache space. And it would do...

1:07:14 PE: That's unusual.

1:07:14 ED: Yes, highly unusual. The reason being that the... We didn't know what the performance of the shared cache would be, with all those processors pounding away at it. We didn't know what locality it would have and how often it would miss. And the backing memory speed was not exactly blinding. We could build a global memory, customized for block transfers that could run at a much higher rate. And it took several years after the Cedar was functional before the software people made a discovery that we'd been telling them all the time, which was that if you want a block transfer, the rate from global memory to everybody was faster than they could get out of the cache. And they should exploit it for that and develop a software for, you know, do these kind of accesses over at global side and do these kind on cache side. But, software moves slowly and carefully. You can't just re-write it all. You know, even with those excellent compiler people, they had their own alligators in their own swamps to deal with.

01:08:31 ED: So we were off on that trip. So we built this global memory, screamingly fast, global interface board, we stretched the back plane of the Alliant to provide room for the global interface, put four more guides in their chassis, built a new chassis. Our boards... So we had three boards: the interface board in the Alliant chassis, the switchboard which was a crossbar piece, two stages of those to get the global memory, and then the multi-bank global memory, global memory board. Three types of boards, a bunch of each. We were pushing speed. The boards had a mixture of TTL, ECL and MOS on them. I don't think anybody was crazy enough to do that at the time. It had a mixture of packages with pins and these new small surface mount packages, some of each. Of course we had power planes, we had every power supply you needed for all those types of circuits. We used the surface mount because they were smaller, we could get more of them on the board, and with that our boards ended up... And we did some custom gate arrays to boot. Jeff Konicek designed some beautiful ECL custom gate arrays for the crossbar switch. CQ Zhu, a visitor from Fudan University in China, designed the controller for the global interface that had to sequence through ungodly amounts of states. He built it with seven semi-independent sequential state machines that interacted together. How anybody could do that, and figure out that it would actually function, is beyond me. But it did. There were almost no logical errors in the design. They were just incredible. These people were good.

01:10:42 ED: So, when we finished with our boards, our three kinds of boards, they were, the interface board and the network board was like two feet square and fourteen layers thick, with on the order of a thousand components, seven hundred components is my recollection. We broke the computer-aided design systems and the design houses, so they couldn't place and route the boards properly. Our databases were too big. They did the final stuff by hand, they did it in pieces and stitched it together and made it work. We called them, we asked if they could, we asked the board manufacturer, which was Tyco, before Tyco got into all its financial troubles, if they had ever made a two foot long board, and they said "Yes". "Have you ever made a two foot live board?" "Yes." "Have you ever made a fourteen layer board?" "Yes." "Have you ever made boards with surface mount?" "Yes, we've done it all." Wonderful. We forgot to ask them if they had ever done it all at the same time, in one product.

1:12:11 PE: I was wondering while you were telling the story.

1:12:14 ED: You know, I was an academic, what do I know? The boards came in, they looked nice, we plugged them up, we eventually got stuff debugged, which was largely mechanical flaws of a one of a kind situation. We had trouble with the back plane, it was noisy. We hired a noise consultant, every noise consultant we talked to recommended a different kind of fix. Everybody's got their own favorite kind of termination, and for the life of me, I still don't know if any one is better than any other. But eventually we tried something that worked and that was the end of that. The machine started working. It generated a lot of heat. So we stacked two chassis on top of each other in the frame and put a cold frame in between them with cold water running through it, which now meant the room had to be very carefully humidity controlled, otherwise we would get condensation on the cold frame which would drip through the circuitry. Not very good. So we had that, we had a halon fire extinguisher in the room, in case we had a fire, which won't spray water all over the place. This is all very expensive. And to boot we were in the theoretical and applied mechanics building which periodically would break massive concrete beams in their beam testing facility on the first floor and the whole building would shake. And we discovered that this, in fact, would not crash from a building tremor of that kind, which was great. So we didn't chew up our discs. It got functional. The software people doing their software, the applications people doing their applications. The... And then it became dysfunctional.

1:14:07 PE: And did this machine have a name, speaking of names?

1:14:10 ED: Cedar.

1:14:12 PE: This is Cedar.

1:14:13 ED: Yeah. So and then the next day it wouldn't work, same thing, and we'd start looking. What we were having, which we eventually discovered, was called a barrel cracking problem. The boards being as thick as they were, the holes... What they do is they route the layers of the board, they squish them together with insulators between them, and then they drill holes when they have to make it connect from one way of the board to another, and if there are cross points that have to connect on the different layers, the hole drills through the wires, they plate... Hopefully it drills through in a way that there is a piece of wire sticking out into the hole or at least to the edge. They fill hole with some plate of barrel in the hole that makes the connections from layer to layer. So under thermal stress, the barrels would crack, because the boards were too thick and the holes are too small of diameter which we later learned. You fixed it by filling a hole solder. And now, that hole doesn't crack and then, the next one does.

1:15:27 PE: Yeah.

1:15:28 ED: So, the machine would work on Tuesday and on Thursday, it would fail and we'd find which holes were cracking and then, we use these thousands of holes on a board. I have one of the boards in the other room, if you like.

1:15:38 PE: Oh, yeah.

1:15:41 ED: And then, we'd find them and fix them and the next day, it would work and the next day, it wouldn't work. So the... We did kind of show eventually... Well, Tyco, a year after I left re-manufactured all the boards. They wouldn't admit that there was a problem until then. And then, it was stable. But the application people who swarmed to Cedar, who were the ones who were going to make it work, they're like a flock of birds. They will go wherever the fastest machine is. So, a lot of them were gone by the time it worked. Some of them came back and did further work with it. But... And the system software, of course, was slow because there are certain things you can't do with system software in terms of creating it until you get a working machine to work on. And it got better and better and eventually, there was a good machine which was now too obsolete to be of interest for being the fastest computer. But it did demonstrate the cost of parallelism which was the form after four parallel machines, each of which consisted of a processor, could do meaningful parallel application work. And that was the purpose of it. Was it worth all those millions of dollars? I don't know. But we agreed when we started that we would not have graduate students. We would just work with professional engineers. But we're academics and it wasn't long before we found all these interesting research problems that we could do. It's was like Bob Rau said after he went to industry. All the time that I was in academia, I wanted to do interesting research and I didn't know what problems to work on. And then, I went to industry and I saw all these problems, they didn't have any time to work on them because I got to get a product out the door. It would be nice if there was some in between. That's what some of these large projects give you, this in between. And they give you enough money, so that even if you're on a development project, you can... It's in the noise to hire the half dozen graduate students to work on stuff. So we agreed we weren't going to have any graduate students. Then we found this all interesting problems and we started well, a few graduate students each just to keep us happy. And by the time I left the project, each year we were graduating more PhDs, the four of us were graduating more PhDs than the entire rest of the Computer Science department.

1:18:28 PE: Wow.

[laughter]

1:18:31 ED: It was wonderful.

1:18:33 PE: There's a real lesson in that. That's great.

1:18:35 ED: Anything you wanted to do, you could do on a moment's notice because you never wrote a proposal for it.

1:18:38 PE: Right.

1:18:39 ED: It was in the noise of this big project.

1:18:41 PE: Yeah.

1:18:43 ED: The government could have funded the research directly, but they don't fund it at that scale. And I think that's why the development projects are so popular. I once said that on a panel and I never got a big development project contract again for that. I said it in public. I don't know if that was the cause.

1:19:05 PE: Yeah.

1:19:06 ED: But I did admit it. One day, I walked into Sameh's office and he was scribbling on transparencies with colored pens. He was drawing rectangles and arrows. And I said, "What are you doing? There are no words on this. What are you doing?" He said "I'm designing an algorithm." "How did you design an algorithm?" "Well, this is the data flow." So, he was comparing one type of algorithm to another, not in terms of the computation which is what I had always thought you do. The computation is free. It's the data dependencies and the motion of data around. It's the whole problem, particularly in this parallel machine. You've got to get enough independent strips and have them local and be able to work them while you got them before you give them up in the memory hierarchy. And that was just fascinating to me.

01:20:16 ED: Then, I sat in a staff meeting -- this is when the Cedar is semi-functional -- I sat in a staff meeting one day and Sameh's group had targeted 10 applications as the things they thought they could really crack for the Cedar. And I realized that if they could get three of them working really well in a way that nobody had seen before, maybe with some new algorithms, they were geniuses. They were golden. This was a brilliant result. Laurie was running the software group, compiler and an operating system. And he had [Constantine] Polychronopoulos and David Padua working with him on that. That the next level down, he had a whole group. And he would say "Well, Dave you can tell me what to deliver and when to deliver it but not both." And I also realized that software has things like releases. You can get it half right or half implemented and you can fix the other stuff in the next release and the early one that's got semi-featured is worth something. You can use it for something. And I'm thinking, there I am in the hardware group with these thousands of things that have to come together and if anyone of them is wrong, you got a pile of junk. You are nowhere. And I was thinking, I'm in the wrong business.

01:22:06 ED: And then, I was also on the Beckman committee at that time. We were not in a good state. We had ample space but not nice space in theoretical and applied mechanics building, with the beans getting smashed all the time. And Beckman instruments and the Beckman from Beckman Instruments gave a huge grant or biggest grant they ever gotten in one piece I think at that time and they were getting to build the Beckman Institute for Advanced Research. And I got put on the planning committee. And the planning committee was chaired by people from chemistry and psychology and they were looking for a nice place for chemistry and psychology people. And they were in control of the committee really. And what our committee got to do periodically was look at the plans for the landscaping around the building.

1:23:05 PE: Seems not very important.

1:23:09 ED: No, it doesn't. See which bushes would look nice. And then, Solid State came into the act and they were around that time building a new Solid State facility moving out of Coordinated Science Lab, in addition to Coordinated Science Lab. But they wanted a piece of Beckman, and we had project that was taking off with no real home. And I wanted a piece, okay. My view was that we would move to Beckman... We were doing advanced research on something. We would move to Beckman and that would be our home. Illinois, we were short on space at that time.

1:23:58 PE: When we say we, you're talking about project...

1:24:00 ED: Cedar, yeah. For a piece of Beckman. And they said "Well, that's we're considering how much space do you need and I say how much space do you need? So, that's too much the old story." Their official plans were to provide second offices for important faculty on a rotating basis. A massive building with incredible facilities that could be planned to be used for a vacation home for the faculty one day a week. It just struck me as ludicrous. Then, the departments said, "We won't give up our faculty." So, we said "Okay, any faculty member who writes a proposal to be done in the Beckman Institute, the overhead would go to the home department." Well, how are they going to run the Beckman Institute? This is a nonviable plan. Well, I could get nowhere with my arguments. I didn't understand at that time that the academic dialog is not won by coming up with a better argument. That each person has the result and reason is not part of the negotiating process. And I just become increasingly frustrated. I'm just a dumb engineer, what do I know.

University of Michigan (1988-2000, emeritus since 2000)

01:25:30 ED: So, that was very frustrating. And then, Michigan called and asked if I'd be interested in being Department Chair at Michigan. So, the Beckman experience and a lot of my administrator experience in this reasoning process reminded me of two things. One is a friend of mine once mentioned to me that we never choose between alternatives. We only choose between our perceptions of the alternatives. And I thought that was very profound, because how do we know what the alternatives are. We just you know... And I've mentioned that to a friend of mine in history and he said, isn't that rather obvious? Another dumb engineer story that I'm very fond of is Ron Rohrer — not that he is a dumb engineer, but he's an engineer, R-O-H-R-E-R, I don't think I talked about him before. But he was working in computer-aided design. He was an enfant terrible when he was a graduate student, I think at Berkeley doing computer-aided design of circuitry. He then worked for Carnegie Mellon, and then, a small company hired him as a Vice President. And shortly after he quit the company to go back to Carnegie Mellon where he passed through before, I met him, I think at Illinois at that time. And I asked him why he was giving that up because he had no specific plans for Carnegie Mellon at that time. He said, "Well, I realized something. When they hired me at this small company, they were in some degree of technical trouble and they realized they needed somebody, and I had a sharp technical reputation. So they hired me as their savior to be sitting on the board and as a VP. And then, I'm at a meeting one day and I'm not convincing anybody of anything and I realized that they had never actually adopted any of my suggestions. And I look around the table and I see there's me, I am just a dumb engineer and the guy running the company is an engineer, but he's an insecure engineer without the courage of his convictions, and in his insecurity, he has surrounded himself on the board with lawyers. Everybody else at the table is a lawyer. And he said... "

1:28:30 PE: This is beginning to sound like a joke.

1:28:31 ED: Not a joke.

1:28:33 PE: Okay.

1:28:34 ED: It is not a joke. And he said "And I realized that engineers are basically very simple people. We believe there's something out there called reality and that when you violate reality, bad things happen. And that what you do to get out this bad situation is you fix the problems that you know, the mistakes that have been made. You fix something." He said lawyers have no such concept. They are trained in argument and they believe that when you make mistakes in the past and these mistakes are getting discovered, what you have to do is construct more clever arguments for why the mistake was actually not a mistake. You're defending the mistake. And in that environment, in that situation, he found it impossible to move forward. And he quit. And I think a lot of this goes on in industry — not necessarily lawyers, it could be people from business school who run the corporations. People who come out of sales and marketing, marketing more than sales, with management training who run large companies. And they're... They have one great advantage. They will get the project out the door when it's supposed to get shipped. Come the day they're going to ship something. Engineers will not ship. They will improve forever. And they need someone to crack the whip. They're not necessarily good business people. You need that training. But management people, I believe, in my own uninformed heart are trained primarily to manage their own careers. And they are not interested in an engineering culture in the way that Hewlett Packard were in the old days, they don't know how to stroke engineers to make them work harder, they don't value personnel. So they're happy to change a 100 percent of the personnel on an engineering team if they can it... Get cheaper people. And then they lose their history, they have no idea where they've come from or what they're doing or why. Like universities trying to export clever ideas. The person with the idea is what needs to be exported, the one who understands it. The idea is not something you can write down on paper and it just works with people who don't understand it. It's complicated.

01:31:21 ED: Now, I'm not saying that academics should run the company. I never did like Plato's Republic, because we'd do a bad job of it, unless we happened to be talented business people. But it takes a village... Quoting Hillary Clinton's phrase. And it does and a village with some mutual respect for the various strengths. It's hard to find and when it's found, it's easy to lose and it's hard to reconstruct. And I just think that the precious people are precious. Motorola discovered that with the 68000 which was a day late and a dollar short because the guys doing the microcode told Motorola one day that they thought they were worth a large amount more money, because they had designed all this complex microcode that worked and would make Motorola a lead position over Intel. And Motorola management said, "Well, we don't pay engineers that much." And they said, "Well, if you don't, we'll quit." And they said, "Well, we don't pay engineers that much." And they quit and that cost Motorola a couple of years in lag time for getting the 68000 out, the first legitimate computer that they had built as a microprocessor. It was a real computer and they lost the lead to Intel from which they never recovered. These people were precious. Nick Tredennick was one of them, T-R-E-D-E-N-N-I-C-K. And I forget who the other one was. They both went on to reasonable careers. I believe in people.

01:33:03 ED: So, Michigan called. Do you want... We've done a first round looking for a Department Chair, they said. We have an edict that although we were very prominent in the 1960s in computing, we have lost the lead; we don't have a reputation in computing now. The Computer and Communication Sciences department from Liberal Arts has been merged into the Electrical and Computer Engineering program in EECS, becoming the Electrical Engineering and Computer Science department, and absorbed another control computing curriculum at the same time. And the mandate for the new Department Chair is to put Michigan on the map in computing. Do the integration which had just happened a couple of years before, but was ready to go now. A new building had been built, people had moved into it, ready to move. They had money to hire faculty. Okay. So question; are you interested in being a Department Chair? No, not really. [Laughter] Answer. Oh, good. We wouldn't want anybody who actually wanted the job.

1:34:35 PE: Right.

1:34:36 ED: Think about it. This is Michigan calling.

1:34:40 PE: I don't want to join any club that would have me as a member.

1:34:42 ED: Right. So, Bernie Galler made the call, G-A-L-L-E-R. And he's the one who said he wouldn't want anybody... This is Michigan calling. And I said, "Whoa, what's Michigan?" I've got to find out about this. Cedar wasn't going so well. The barrels are cracking. This stuff wasn't functional. I've been there, done that.

1:35:10 PE: Soldering thousands of holes.

Chairing the Dept. of Electrical Engineering and Computer Science (1988-2000)

1:35:11 ED: I was so totally ticked off at higher administration in academia. Rarely found someone I would respect although there were several, Dean Everett being one, and Jordan being another. And I said, "I've spent all of my life bitching about administrators. I think academia's running wrong. Here's all these resources at Michigan. I'm interested in society and socialization and building teams. Why don't I try it? And I don't have anything else to do now because we're just fighting these barrels; the hardware design's done. You just have to make it work somehow. Beat Tyco over the head and get it going. The action's over." So, I decided to do it. What Michigan had was they had been chaired... The President of Michigan had been Shapiro who left just before I got here. And Duderstadt had been Dean of Engineering had then become provost, D-U-D-E-R-S-T-A-D-T. And Jim and Shapiro had made a heck of a team. Shapiro's specialty was economics, particularly regional economics. And he and Duderstadt would go to Lansing and preach about how a state recovers from the debts economically as the automobile industry is collapsing. Nothing like it is today. But it was... That was when it was happening and the state responded. And every lecture to the legislatures that Shapiro gave ended with how the university would be instrumental in creating new technology, new base, new industry base and resurrect the state. And Lansing was persuaded and every time he came to lecture they gave Michigan money. So getting money. Duderstadt would make the hard-core technical pitch for what we wanted. We had a development fund. He got a development fund which, was spent in Engineering. He engineered the movement of computers in to make that central piece, but it hadn't yet been integrated. He saw that as the next step, became provost. So the other thing that Duderstadt noticed is that Michigan, having been relatively early to be large and excellent in engineering, had a rather elderly faculty, one-third of whom were to retire very soon. So, there were both new resources and a hell of a lot of retirement slots where you could do massive hiring without enlargement.

1:38:26 PE: Yeah.

Interview recording part 6

1:38:28 ED: So, Chuck Vest was Dean of Engineering. He hired me and he said, I would be...

1:38:36 PE: He went on to be President of MIT.

1:38:39 ED: He went on... As soon as I got here... Before I got here, Shapiro went to Princeton. Duderstadt became President of the University. Vest left engineering to be provost of the University. Duderstadt not being one who would be so shy as not to want two engineers in the top two offices. Although I will say that as President, Duderstadt bent over backwards to develop the basic sciences and liberal arts and not continue to funnel. He said, "Engineering's been done. You've got to take care of what you've already been given." So, he was fair in that regard. I deeply regretted not having a chance to work more closely and longer with Chuck Vest. He left too soon after I came in and I lost my base of support in the administration and never recovered from that. But Chuck told me that I could have 16 faculty slots in my first year and if they were spent I could get another 16 the next year.

1:39:46 PE: Wow.

1:39:48 ED: So...

1:39:49 PE: A lot of hiring.

1:39:50 ED: Yes.

1:39:51 PE: I'm just thinking about all the job talks.

1:39:52 ED: Yes. So, I'm walking into a faculty of about 70 with 32 hiring slots in my pocket and a few retirements coming up. So, my friend David Crockett who has been my mentor for my entire professional life after Illinois, after being graduate student... We overlapped for one year when I was a grad student at Illinois. A brilliant manager at Hewlett Packard, said that the most important thing I had to do was to tell the faculty what my priorities are because nobody can do everything. And to tell them how I intended to run the department. And that if I did that and people understood that, they would evaluate me based on what I had said. And that's how administrators and managers work in industry, if they last. And unfortunately, I didn't do that publicly enough. I worked in faculty recruiting. I said, "Okay, I'm going to make this a first class Computer department, Computer Engineering, Computer science. And I will... The other parts are already strong and I will take care to keep them strong and give them enough balance that they're not going to rise up and revolt, but this was the edict from above. And so, we'll spend a lot of the effort in there." And I will leave it... I didn't think the curriculum was in good shape in computing. I didn't think that the faculty had a sense of what a curriculum should be. Bricks laid in an orderly fashion in the wall with enough options for students to build specialty tracks and so on. And rather we had a collection of courses where whatever faculty member was assigned to teach that course in a given term taught whatever he felt like teaching. And so the pre-requisite structure didn't function well. Nor did the faculty seem that anxious to build it. I should've communicated better that that was their job, because I wasn't going to do it; not while recruiting, you know, interviewing over 50 faculty a year. I hired... Ended up hiring 33 faculty in three years. And we did get on the map. When I was first in there, I looked very young when I first went to Illinois and I asked for a student file one time and the office said they don't give those to the students, they only give them to faculty. Five years later, after we had built all the curriculum and so on, students were coming up to me and asking me if I needed help getting up the stairs.

1:42:52 PE: Well, everyone should have such problems.

1:42:56 ED: When I came to Michigan, there was a guy named Bill Dow, D-O-W, who was the granddaddy of Electrical Engineering. He built it. He had the faculty teach the first transistor course ever taught in any university in the nation. He was pushing 100 years old. He was in his late 90s. And he and George Haddad who had run the department for... George Haddad had run the department for a couple of decades before I got here. Tom Senior ran it in-between for a year when they couldn't find anybody in the first search round. And I was told by the secretary that one day Bill Dow would come bursting into my office... Oh, Bill Dow also started Willow Run. He started many, many things. Real pioneer. Tough, bare knuckles guy. I had a letter from him in the file to a faculty member saying he was sorry that faculty member couldn't take a sabbatical that year because they already had one faculty member going on sabbatical. Feisty guy. He would come in... Burst into George Haddad's office and talk whatever he wanted to talk about and George would indulge him. And Bill still had ideas, positions on things. When Tom Senior was acting chair Bill would go bursting into his office and look around and say, "Where's the Chairman? I need to talk to the Chairman." And he never acknowledged Senior's presence for the entire year he was running the department.

01:44:38 ED: So the secretaries told me one day Bill Dow was going to come bursting into my office and whatever I was doing I just had to drop it and indulge him because there's no way of shutting him off. And one day he came charging into the office, looked around saw me sitting behind this big desk, looked, his eyes widened and he looked at me and he said, "Are you the new boy?" And I said yes, and he asked me a little about myself, he told me a little bit his agenda, he was very kind and generous. And when he found out that I had done my undergraduate work at Harvard, he said, "Oh, that's where they have the big bass drum." I said, well as a matter of fact I played the bass drum in the football band, which is what I did while waiting to take computer courses in my senior year. And he said, Illegitimi non carborundum and turned on his heels and walked out. Well, it turns out INC is inscribed in all of Harvard BM literature and Illegitimi non carborundum is inscribed on the big bass drum which translates as "don't let the bastards grind you down." Good advice and that's sort of always been life model is that: try to hold my ground and stay the course.

01:46:11 ED: I became aware that there are two cultures in engineering; the impact culture and the intellectual difficulty culture. And they don't really understand each other very well. Dean Drucker, who was dean at Illinois when I was there followed Everett it is Dean, he was very much enamored with the intellectual difficulty culture. He liked to fight... Well, he was from theoretical mechanics and he liked departments like Theoretical and Applied Mechanics and he liked Computer Science because it had the word science in it and he liked science. And physics, which was true a quirk of fate was part of the Engineering College at Illinois, and Mathematics. I heard a definition of Engineering that I have always loved, which is: engineering is a cross between science and economics. The science tells you what's possible and you need that. The economics tells you what might there be a market for it and what will cost to produce. Okay, so...

Interview recording part 6

1:47:37 PE: Actually, hang on one second. Let me...

[Interview recording part 7 begins here]

0:00:00 Paul Edwards: Okay, this is Ed Davidson, Interview Part 7.

00:00:11 Ed Davidson: So, when one of them is missing, it's in my opinion, it is either second-rate science or second-rate economics. Because if it was first-rate science, it would be done in a science department and if it was first-rate economics, it would be in an economics department. But there is something unique that happens with the blend. We should make stuff. Now, I am not saying that we don't build theoretical models and I have spent most of my life building theoretical models. I came from a math origin and I loved my first course at MIT when I was in Honeywell in Operations Research where I learned branch and bound and combinatorial optimization which I have used for most of my life. But it has to go some place. So, my friend Jacob Abraham at Illinois used to say, when you interview somebody for a faculty position in engineering, they are going to say that they are engineers, but theoreticians have to find jobs when they finish PhDs. And industry doesn't want for many of them and they tend to gravitate toward academia where they produce more theoreticians. And in the last analysis, the world desperately needs theoreticians, but it doesn't need very many of them. And there is no place for a second-rate theoretician whereas a second-rate engineer can do good and productive work; you only need a few of the most brilliant theoreticians, it's hard work. So, you want to discover when somebody is interviewing are they really theoretician posing as an engineer. So you say "Did you ever prove a theorem?" And some engineers will never have proved a theorem, but in those days most PhD theses contain some theorems they have got proven, which there is something nice about that. So, if they say yes, that does not make them a theoretician. So, you say that second question if they say yes... But if they say no then they are no kind of theoretician at all. You try to figure out what they have done that might be useful. If they say yes, then you say "Was it important?" And well, it's their thesis and they just did it, so of course it's important and then you say, "Why?" And if they tell what it is good for, that's good. If they tell you it's good to prove this other theorem, they are theoreticians. If the end result is theorems rather than something that you do with them in the real world, it is not engineering, it should not be engineering. And that's important to be mindful of because otherwise if the practical engineers all go to industry and the theoreticians all go to academia, we implode as a department. But there is something in academic research -- going to back the impact versus intellectual difficulty culture -- there is something in academic research that loves complexity. And we should normally seek simplicity. I used to have a sign outside my office door when I was at Stanford that said, "Eschew obfuscation."

0:03:44 PE: Yes, my dad had the same thing.

0:03:46 ED: Did he? That's a wonderful sign. And there was a comment made on that, that if you believe the truly profound things are inherently complicated, it only takes a small leap of logic to believe that something that is inherently complicated must be profound. And you go to complexity, because that's how you convince your colleagues that you've got a mind, you can deal with complexity. But if it is not useful for anything, it is horrible. The peer review system supports that. Academia is filled with academic cults that have created on artificially made-up problems where it is a closed community that writes letters about how brilliant each other are and they do very well in academia, and they create their own journals and conferences and so on and they publish.

0:04:37 PE: That's true.

0:04:39 ED: So, I won't suggest if it's a college with engineering that they not promote anybody to tenure unless there was at least one letter from industry. So, we say what do we do to relate the industry better? Well, man I was dumped out of the room immediately that was not to happen. Now, I respect smart people and I have known theoreticians that do beautiful work. That's very important. I don't know whether the computation complexity crowd ever did anything worth doing or not. Average complexity is more important than worst complexity. And they eventually discovered that and went into that. But I think in academia you have to be aware of the intellectual difficulty crowd for doing artificial stuff and just working on each other. And you have to be aware of third rate applied research as well, tinkering. You don't want to do that. And there is something in between.

00:05:51 ED: When I went to the ISPASS Conference, I gave a keynote and talked about... That conference is International Symposium on Performance Analysis of System Sciences, I think. So, it is a performance analysis thing. And I have spent most of my life on performance analysis and then tried to invent widgets that could capitalize on what I was seeing. And I basically gave my talk on some trends in the research on computers that I thought were damaging. And one of them was the bar graph and simulation type of research, where you do research by saying, "Here is a problem, a legitimate problem. Here are the first six solutions I could think of." So now with easy computer resources and simulation tools, I will simulate each of them on a given set of... Some set of benchmarks, which are also plentiful, and on those particular benchmarks that were chosen of those ideas, some of them will have higher bar graphs than others. And I publish that as a result.

I claim that's no research at all. I don't know what it is about those applications, or about the system model being used, that causes those ideas to be better than other ideas. I don't have an idea of the frame of validity of that result when it moves into another kind of system or another kind of application; I have no idea what the range of applicability is for that result. I don't have any functional model underlying that that's says because of these properties this will work better than that and what the advantage will be and where do you go, so there is no theory. There is no underlying theory, there is no functional relationship. And I spent my life working on functional relationships. So as with the theorem prover, I think that's important.

[laughter]

0:08:35 ED: A mechanism of causality. The world doesn't... Our professional world doesn't get that interested in that kind of thing. So I pleaded with these people, don't do bar graph research. I mean you can do a bar graph because it is a good visual, but don't have that be the research; it's too cheap.

00:05:51 PE: You know, I'm teaching a research methods course right now. And one of the things I have been saying to these students is how important it is to not just learn some bunch of methods that are going to be like a machine where you crank the handle and the result comes out, the bar chart thing, but have some idea why you are employing the methods.

Interview recording part 7

0:09:17 ED: Yes.

0:09:18 PE: ...what theory are you trying to prove or disprove or extend or whatever. Because that's the only way to show a research result has any meaning.

0:09:26 ED: Yes. And unfortunately, in a lot of fields that aren't in engineering or sciences, but in the liberal arts, as they have moved toward quantitative publishing, they have suffered from this disease as well. And it's a damn shame. I like numbers, but you have to... Yeah, well, all right.

00:09:52 ED: So many people came up to me afterwards thanking me for doing that and striking a bit of a note of terror that the other research is hard and it takes longer and it is harder to get published, and confessing that they use bar charts. I said "Don't confess. I use bar charts too but that's not the answer." And when I give a seminar on a cache hierarchy and somebody wants to know how fast the second level cache should be then, they want a number from me and they want to take that number away and build it that way. I have no idea what context they are putting it in and I am not going to give them a number. This is just data to illustrate the idea; this is garbage. This is academic garbage. I don't want you to take those numbers. Look at the functional relationship. Figure out whether it fits your environment and do your analysis, or use my tools and do an analysis but change the numbers and change the model.

[pause]

00:11:10 ED: A faculty member at Michigan who we hired, Todd Austin, did this simulation package that everybody uses to generate these bar charts, pointed out that the computer model in there was constructed as a simplistic model to illustrate that his simulation framework in which people could construct simulators was functional and how to use it. And I would venture to say that most of the research that has been published with the simulation framework called SimpleScalar, most of the SimpleScalar bar charts that have been generated have been generated on that toy model that he created. And people who use it, most of the people who use it have no idea what's in that model. They've never looked at it. If they don't construct their model they have no idea what... They are generating garbage.

0:12:23 PE: Yeah, yeah.

0:12:24 ED: The fault is not in the people who do it. The fault is in the peer review community, which needs to look at that and not accept papers of that kind.

0:12:33 PE: Okay.

0:12:35 ED: But once you have that going on in engineering, in computers, and in well in hierarchy will be in computers, in engineering, in the quantitative sciences and in the non-quantitative disciplines that have gone quantitative, publication is too easy. You could scoop up stuff and generate a publication. The pressure to publish goes way up. So that now, for example, it is very difficult to hire a faculty member who does not already have six publications on his thesis topic. I rather have one publication that make sense. Before I was in the review community the editors, the program chairs and the deans, we need to ask what is the significance of what you've done and evaluate it as best we can. And if we can't evaluate we should find outside people who can, that we trust. That's how you build a strong faculty. Recently, I had the honor of going to Texas to the ISCA Conference, which I haven't been to in a number of years, to present the Eckert Mauchly award to one of my students, Joe Emer. And that, I think, was also after the last time...

0:13:57 PE: Spell the name.

0:13:58 ED: E-M-E-R.

0:13:59 PE: Okay.

0:14:01 ED: And a Berkeley student came up to him and talked about a research problem that she has, because she is working on something that she thinks is very interesting but it is complicated and relates to several different kind of fields. And her advisor is putting very heavy pressure on her to figure out how she can slice her research results up into the maximum number of pieces.

0:14:33 PE: Yeah.

0:14:33 ED: So that they can be published separately.

0:14:35 PE: Yeah.

0:14:36 ED: And I would call this "salami-style research publication." Maybe I didn't invent the term but I have used it long enough I think I have in my dotage. That is a horrifying objective to me. And very often I have noticed when you ask somebody what they have done and they describe something interesting and you ask them where it has appeared, they will give you a list of 12 papers. And you have to read all 12 before you can begin to synthesize what the heck it is about. That's a horrible thing.

00:15:25 ED: So, that's some of my griping. [laughter] A cult that is unfettered by the reality of the outside world is a very dangerous thing. In academia, it is not easy to police quality and you can't do it but by counting. I like to think as Department Chair, I promoted some people who have become excellent faculty members who maybe were marginal by normal criteria, but they had something, but there is a spark. They would do great in the classroom and they were doing competent research. You can't just be great in the classroom. But they had something real to offer, and we didn't promote people who were weak; even though that was occasionally an extremely painful thing to do. And then the Dean always gets blamed for that, the Dean in the college. But I can say after being Department Chair, the Department Chair who writes a cover letter, can write a support cover letter that telegraphs that this person should not be promoted [chuckle] as well as a support cover letter that says this person might ordinarily not get promoted, but if you don't promote I will quit. And that gets taken into consideration. It's important when people have that kind of responsibility over people's lives and they have a committee to hide behind, so it is not one person making the decision. And it should be a group decision because different people see different things. I never wanted to promote someone who had only taught graduated course. I wanted them to have taught at least core undergraduate course and done it well. That didn't go. I couldn't convince my Executive Committee that we should do that.

00:17:20 ED: But they soon taught an undergraduate course. And I rewarded people who taught well. We had nice differential salary things. I have my favored way of determining what an institution's value on a faculty member is. This wasn't in my notes, I wasn't going to say anything about it. But if you want to know what an institution values, how it values its faculty and what it values, take the faculty and the department and plot them on a graph where the vertical axis is their salary and the horizontal axis is years since PhD. And you will find several upward going lines that most of the faculty are clustered on, perhaps three of them. And you look at those faculty names and say, what's special about these who sit on the top end of the curve? What's special about these in the bottom and the middle and courses in between? And the difference will tell you the value structure in the university. And it often has to do with research dollar volume. I never understood why solid state was so highly priced by Michigan and differentially rewarded. They are very good people, they do excellent research and I am not faulting that. But they are often differentially rewarded for their research dollar volume, neglecting the fact the university spends more on each dollar taken in; it's running at a loss. The more money they bring in... Well, they spend more per dollar brought in than any other area. Why? Because it is a lab, the large expense of laboratory science. That's very expensive. You've got to be careful with that. And if you reward research dollar volume such if you will go broke.

0:19:35 PE: Yeah, yeah. Interesting.

0:19:38 ED: They don't reward, for example, in that way or pay as much respect to software systems. An area that I didn't work in, so I can say that. Theoreticians produce the most PhDs per research dollar, cost the university nothing. Software systems produces people who build software systems, who are hired in massive quantities by industry, desperately needed, a very valuable hardcore engineering discipline, they produce very economical PhDs. Some of whom make a lot of money and give money back to the university. But they don't need much cost, they don't need much space, they just have few stools in the room with a computer.

0:20:37 PE: Yeah, yeah.

0:20:39 ED: They don't have big expensive labs to run. They are a great bargain. They are a model of efficiency. Now, I am not saying that we should fire all the solid state engineers and give massive salaries to the software systems people. But we need to factor in a broader set of evaluation metrics when we evaluate things. It is hard to sell that, because the people running the institution and the powerful people in the institution are people who have been heavily rewarded under their status quo. One chemistry faculty member, not from Michigan, who was greatly lauded recently for his amazing accomplishments was asked, what is the secret to success in academia? And he said "Never give up a square foot of laboratory space, whether you need it or not". Because he who has the most square feet of laboratory space is cock of the rock, and that's what establishes your reputation. My mother was deeply disappointed when I chose to go into academia. I thought she would be pleased. She was pleased when I went for PhD and I said, why? She said, "Well, in industry there is a bottom line and you can look at the bottom line and say what you have contributed and what you accomplished. And academia is just a place where the currency is who can puff themselves up to look more important than the other guy. And it is artificial and so you never know, it's all bluff and... " Well, she hadn't been there...

0:22:36 PE: Yeah.

0:22:37 ED: Either in industry or academia. She has a point, but I can tell you that industry people who come to tell me it must be great to be in the university because there is no politics there... They are living in la la land, and university people think that industry is great because there is no politics there and so on. It just doesn't make sense. There's politics everywhere, people are people, and the best technical idea doesn't necessarily win in industry. It's a question of was on the inside and who has the ear of management and who can pound on the table, and it's great that startup companies can go more on ideas although they have to sell it to venture capital people who leave them dry. [chuckle] The success comes from a balance of terror.

[laughter].

0:23:37 ED: You have to factor it all in. And a great book on that subject, it was written by Nate McCabe called *The Gamesman*. It's still available in paperback, used copies and new. And it really shows how engineering people function. How people function in engineering corporations and what the game is. And how the gamesman who can play... work the environment for the greater good and for great success is a person, he says and has wounded me personally, he says, not like the typical academic for whom life is a succession of setting aside academic ideals, but someone who feels very comfortable with these conflicts and never expected anything else of the world and factors them all into consideration and knows how to deal with it and get things done. And man, that is a great book. His model for the gamesman was Dick Hackborn, if you look back and I have met Dick and I have worked with him, he would not remember me, but Hackborn was one of David Crockett's mentors. A great man.

0:25:05 PE: So that was your experience of being chair of CS.

0:25:17 ED: It was the hiring and the scrapping and working there. We hired a new Dean, Peter Banks, after a succession of acting Deans. I was on the search committee, unfortunately. Peter was very charming and we hired him away from Stanford, he had been... He was going to leave Stanford because his discipline he had gotten crossed wires with a few people, there was a little bit of contention going on there. And we were able to get him. Shortly after he became Dean, Mike Flynn came to give a seminar, he is one of two people who replaced me when I left Stanford. He was the old guy from IBM who would pass through Northwestern and Johns Hopkins and started a company called Alen when he came to Stanford the year I left. And the other was some kid who had just finished a PhD thesis at Stonybrook. I think his name was John Hennessy.

0:26:26 PE: Oh, that kid. [laughter]

0:26:28 ED: That kid. He has done rather well. Brilliant guy. Flynn and I have crossed paths a number of times over the years. We are... I like to think that we are friends. And he was giving a seminar at Michigan and he was looking at the bulletin boards and we talked about

Michigan and Stanford. He said... I was thinking about Stanford and the faculty and after I had been there for a number of years and try to figure out what it was about the Stanford faculty that made them so great compared to other places and he says "They are not that much smarter than everybody else. They are smart. And it’s not a lot of magic going on, but the one trait they all have in common is boy, they know how to smell where the next piece of cheese is." And then he looked at it and he said, here you hired our friend Peter Banks. And I said, well, is that good or bad, he said "Well, I think it's good, I think it's good." I said well is it good for us, or good for you?

[chuckle].

0:27:42 ED: And he says, well, I think it's good for both and I said, "Oh my God." Well, Peter didn't work out as Dean. He then went to ARM and he didn't work out at ARM either. He is a very selfish guy and he didn't do us any favors.

0:27:56 PE: How long did he last here?

0:27:58 ED: Couple of years.

0:27:58 PE: That's not a lot for a Dean.

0:28:00 ED: No. So shortly after Peter became Dean and some faculty in the department went to see him, nicely representing at least one from each division and that's all I know about who they were, saying that I was coming apart as Department Chair. And that although they bore me no malice and thought I was doing everything I could, I was trying to do it all by myself and things weren't working so well. My email was backed up, I wasn't responding to people, which was true. I wasn't doing some of the things that needed to be done. And that's why I said, I was never as good as Dave about explaining to the faculty what I was concentrating on and why. In the meantime, the way the department was running and our competitive computer science departments had one TA per faculty member. And we were doing that and we were terribly overloaded in teaching computers, relative to the number of faculty, even though we had done a good bit of hiring. Students were just swarming into the field. And...

0:29:19 PE: So, we will just get some dates here, so Peter Banks, roughly...

0:29:26 ED: Fall of '90. Yeah.

0:29:36 PE: Okay.

0:29:38 ED: Because I started in January of '88.

0:29:40 PE: Okay. So, that's a year and half after you came.

0:29:45 ED: Two and a half.

0:29:46 PE: Two and a half, right. Because you came in January.

0:29:49 ED: Yeah. And I had given up smoking when I moved here which is tough thing to do when starting being a Department Chair, I know I was starting to sneak cigarettes and my blood pressure was up probably on diabetes medication, blood pressure medication and cholesterol medication. I was killing myself. The committee structure, I would say, was not fully functional in the department things.

0:30:20 PE: Is this meeting that you were describing...

0:30:22 ED: Well, let me lead into the meeting a little bit.

0:30:24 PE: Just the date, roughly.

0:30:27 ED: Well, '90 in the middle of the term, just after I arrived.

0:30:30 PE: Okay.

0:30:31 ED: Well, within weeks after I arrived.

0:30:33 PE: All right.

00:30:36 ED: So in that calendar year, we... All right, the department head eventually spent most of its discretionary funds on teaching assistants, because the teaching assistant budget is far too low. And we moved a million dollars a year into teaching assistants. And that's how we managed to teach our courses. The college had no recognition, no official recognition that lab courses took more teaching assistance than non-lab courses. No official recognition that teaching a course with a huge enrollment or teaching a lab course was any more or less load than teaching a seminar for 10 students, etcetera. One course a semester, that's....

Ford Motor Co. — research on crash test simulation

00:31:31 ED: So after we sent out our contracts for teaching in the fall of '90, we were told that the state appropriation wasn't coming through and that we needed to cut our discretionary funding. Or cut... I had to cut a million bucks. The contracts were out. There was no money to cut. So I ran a million dollars in the hole, and I told them I was doing it. I said unless you have... I'm not going to take back the contractual obligation to a graduate student. And I'm not going to screw the faculty with no TAs. Your budget formula is wrong. And they said they can't fix my problem because I am between 25% and 40% of engineering, depending on what metric you use. And if they fixed my problem, I have to screw everybody else. ...And they can't do that. That is politically untenable for me. So I said, "Okay." So now a million bucks in the hole, Peter Banks says, "There's some faculty saying that they really need to get a different Chair for the department. In the meantime, Ford Motor company had come by and they wanted to do some research on... I think ways that... They asked if there was any way that University of Michigan could help them on their thorniest problems. One of which was crash simulation on big parallel computers. And I looked at that and I said "I'm going to do it. I'm going crazy as Department Chair. What I'm going to do in '90 is I'm going to restructure the department to take more care of itself and I'm going to lead this research project because I am dying. That's my lifeblood and I'm practically out of blood." And even though I can't handle all aspects of the Chair job, I'll do this in addition or whatever.

Interview recording part 7

0:33:35 PE: Wow.

0:33:36 ED: And I'll serve out my remaining two years. I came on a 5-year term, and then quit after five. So, this is the end of three... Near the end of three. Peter says, the faculty are complaining about me and he is gingerly leading up to what they're saying and what he is saying. And I cut him off and I said, "Peter, if you are beating around the bush to ask for my resignation, you could have it on your desk in the morning. No regrets." [laughter] And he said, "Well, think about it. Think about it. Maybe you want to activate some of your friends and so on. And I thought about it overnight and I said, if I activate my friends, maybe I win, maybe I don't. But I put them on the line and I really... Thinking it over, I really don't want this job any more. I have done what I came to do. I don't want to organize the curriculum. I can organize the curriculum better as a faculty member on a curriculum committee."

0:34:33 PE: Yeah.

0:34:34 ED: "And I want to get this research going. That's what I love and that's what I want. I'm going back to another big project." So, he said, "Okay." and I was done. And in January of '91, George Haddad stepped back into the Department Chair, for a little while, until they could do a real search. And then, they hired Pramod Khargonekar. I did the Ford project, which I loved. Ford said... This is another bit of academic gamesmanship. Ford said they wanted to... I brought in 12 faculty representing different groups to make the pitch to Ford because they said the sky is the limit. We have a lot of problems here. And everybody made their own pitches and they said, "We want to do this crash simulation stuff and maybe you could bring in some theoretical stuff with it. And ... maybe we'll talk to some of the other people about doing some other things." So, they picked what I cared about and I said, "Okay, I'll head it up" and they were happy with that. And then, they said "This is our family jewels" because they had their own crash simulation program, software and they wanted... And the industry to provide crash simulation software was not doing the job. And they wanted to get their internal staff to just beat the hell out of everybody else and that was going to be the key to their design process for new cars. To run it just a hell of a lot faster. To do a... I think it was a 50 millisecond car crash. Cars sitting, moving forward at 35 miles an hour with the front bumper touching an immovable object in the simulation. Moving at 35 miles an hour and in something like 40, 50 milliseconds, the car is in fact moving away from the wall rather than towards it. That's how long it takes. Then, they're looking at stuff like is the engine sitting in the driver's compartment, is the gas tank ruptured, things like that? Has the dummy cracked its neck, whiplashed? It took 40 hours to run that on one Cray processor... one 30-40 millisecond simulation.

0:37:15 PE: Yeah.

0:37:17 ED: So an engineer has an idea for modifying the material in the metal of the car and they gin it up, feed it into the simulation. They kick off the simulation and three days later, he gets an answer. Because he's not the only guy on the machine.

0:37:34 PE: Yeah.

0:37:37 ED: And he doesn't remember what the question was. He can't design that.

0:37:41 PE: How many problems can you test that way? In a reasonable timeframe?

0:37:44 ED: And the simulation was also critical because the way they do it physically to build the model... Because they have to build the model with the structural property of the finished product.

0:37:53 PE: Yeah.

0:37:56 ED: That model was built before this production line, because it's testing out an idea. It costs them a million dollars to hand make a car.

0:38:02 PE: Right.

0:38:03 ED: That doesn't even have an engine in it. It has a block of equivalent weight stuff and so on.

0:38:07 PE: Yeah.

0:38:07 ED: But still, it's a million dollar product, hand made. And then, they smash it.

0:38:11 PE: Yeah.

0:38:12 ED: Now, it's garbage.

0:38:13 PE: Yeah.

0:38:15 ED: So, computer time, they can't afford to pay for computer time but they can't stand the turnaround time. The models are becoming... Just becoming accurate predictors of the prototypes. So now, they could switch. So their goal is to do the simulation in an hour, instead of 40 hours. Can we do it? Well, I did parallel computing. I discovered that the application guys have the easiest time. I knew about architecture. Hell, yes. I could an architecture specific tuning or code, I didn't care how complicated it was. A little bit brazen. But we said we'd use a lot of grad students, we would back up on grad students again with money. And we have four other faculty. And Ford said "Okay. Now there's one hitch. This is our crown jewels. You can't publish." How I can use graduate students? These are guys after a PhD. So I said okay, double the budget and I'll spend half of it on your problems and half of it on academic problems that we get a vision on...

0:39:35 PE: Somewhere, yeah.

0:39:36 ED: As a result of the work that we're doing for you.

0:39:39 PE: Yeah.

0:39:40 ED: That we can abstract and follow up on other studies that you might not be interested in and we'll publish those, and they bought it. So I had another 5-year run and I stretched this out because they gave me so much money that... They didn't give me any more money, but we have five years in money and we stretched it to seven toward the end. And to be sure, they had an internal group working. They had a Cray engineer whose name I should know, but I have forgotten, a wonderful guy who is on site at Ford, day one. We met a lot. We came up with a tremendous number of ideas. We re-coded the code. We had the code on campus. We re-coded it, re-wrote it constantly. We tried out innumerable things on their computers and on the computers that we bought with an NSF grant and center for parallel computation in Michigan which we started with Bill Martin from Nuclear [Engineering]. And so, we had some fast computers here. We were in a partnership with them with Steve Chen's company Supercomputer Systems Inc. Which unfortunately went broke before delivering. They made a couple of technical mistakes. So IBM and Ford and we and some others were... Well, IBM and Ford were bankrolling and others. And we got to go to the meetings and sit in and watch all this stuff happen. It was wonderful. And then, we went with Cray when things folded up and we did some work with Silicon Graphics. We made a slower version that ran on some SGI machines. With the help of the guy from Ford and their internal staff and some of our ideas... We never wrote production code. We would modify the code but any change that went into the official version, they put that in at Ford, and vetted it independently at Ford.

0:41:46 PE: Yeah.

0:41:47 ED: But the end result was that we ran... We converted completely from a serial code that ran on one machine to a code that ran on 16 processors of the Cray YMP, and it ran in 12 minutes.

0:42:03 PE: Wow, very good.

0:42:06 ED: And George Carignan who was the Associate Dean for Research asked me one day when this was done and finally being leaked publicly, "What was the brilliant insight? What was the flash of inspiration?" Looking for something that could become a major Michigan press release. And I looked at him with this dumb look on my face and "He says, was it... One great thing or a couple or three, or was it a million different tweaks?" And I said "It was a million different tweaks." because that's what it was. We made it work. And you could see the air go out of his lungs, and it deflated like a balloon. Not the kind of thing you can publish and crow about that much. We published a little bit finally. The publications are not good on what was done there, but it set a way of doing it. ...Oh, and the revised product was used on the Ford Windstar. That was the first car that was made with this. And it worked finally very well. They did major changes. They could try out material changes. They could make a more economical car that met all the standards. And you know, an engineer can sit there and try out five, six, 10 ideas a day.

0:43:43 PE: Yeah.

0:43:44 ED: They can move.

0:43:45 PE: Yeah.

0:43:46 ED: Decisions get made much more quickly.

Interview recording part 7

0:43:47 PE: Yeah. That's the best scenario.

0:43:48 ED: Other than the management bureaucracy of course. Well, once that was known to exist, the industry caught up. So the people providing commercial crashed, produced stuff that would run that way. Ford's internal models, material models, they still kept secret, but this was primarily over the commercial stuff after that. In the meantime, we published just a whole hell of a lot of academic research. We built bounce models for performance. We did a whole theory of architectural specific performance evaluation and improvement. We did a series of tools to analyze, inherent parallelism and applications, looking at being able to inject or remove communication costs of various kinds depending on memory hierarchy. So there were abstracted models for types of architectures it could be built. And that stuff worked very well. It was used it... Geith Abandah, that was one of the people who did that. And he worked at Hewlett Packard, he came up with some of that. ABANDAH, G-E-I-T-H. He is now at the faculty of University of Jordan, from where he came, where he had done his undergraduate work under another student of mine, Basam Kahala, who said "Go to Michigan and work with Davidson", I'm very glad he did. I cried when he went back to Jordon. It's my mission as with Len Shar from South Africa is always to get people to work in the United States.

Final years in the UM EECS Dept.

00:45:53 ED: So, cache access predictability with Murali Anmavaram and D. G. Srinivasan. The Ford Project was great. And as it came down, as it finished up, Pramod Khargonekar became chair of EECS and asked me if I would consider going to, being Associate Chair for Computer Science and Engineering, recognizing that usually people who chair the whole thing won't take the subordinate position. I never felt that way. And you know, once again like with the Cedar Project, and like with leaving Cedar, the Ford Project was winding down. I had made various alliances with people in application domains, like Atmospheric Modeling and so on, and tried writing research proposals with them. They were not getting funded, the application community was saying, there is no fundamental new algorithmic research here, let the computer people fund it. The computer people were saying there is no fundamentally new computer architecture here, let the application people fund it. And we couldn't...

0:47:10 PE: That's a classic bind. Was it...?

0:47:12 ED: No, we couldn't get, we were not good enough, political enough, strategic enough to land another big one. Interestingly enough, the grant that we had got from NSF for the Supercomputer Center... I met the person who actually made the final decision, that we would get that funding in that program. And I didn't remember him at all, but he had taken a course from me as a student at Stanford, and remembered finally the way I approached things and that helped influence his decision to chuck the money Michigan's way even though we had not been a game player in that domain since... Now everything is at San Diego and Michigan is part of the San Diego Consortium. But we don't have big machines on campus anymore, because we don't build many big machines anymore, they are most the clusters and PCs that's not terribly interesting to me. So I think, we got a handle on this. You know, I know we are getting closer at this time, but you know I could talk about the bounce modeling that we did, it is well documented and well written up, so and the cache access predictability. The cache miss taxonomy, which I love, that D. G. Srinivasan did, that basically nobody was really interested. And although to me it solved, totally, once and for all, how to evaluate one cache prediction model after another or pre-fetch model, and she ended up doing her thesis in prediction. As did Murali Annavaram who is now at USC, University of Southern California. DJ's at IBM research. It's been a good trip.

0:49:18 PE: So then you retired in 2000.

0:49:21 ED: Yeah. So I did take the Co-Chair position. My last year as Co-Chair, I loved working with Pramod and supporting him. He eventually left and went to Florida as Dean. They had been recruiting him hard for some time. During that time the teaching overload in Computer Engineering was terrible, and I felt that I was getting no audience for that in the College of Engineering, and I was getting increasingly frustrated. I wrote some models of teaching load, which showed that the average Computer Science and Engineering faculty member was more than twice as loaded as the average in Engineering. Despite our hiring, that we needed support. And we got support for hiring adjuncts but not... And we had support for hiring faculty, but we weren't going to lower our standards on that. Money could have gone further and we were not getting money. And I felt that I had no audience from the college, I got a strong relationship between IBM and CSE which has then since fallen apart; no backing from the college on that. In my last year on the faculty, I wrote 13 research proposals with people in various academic application domains, none of which were funded.

00:50:50 PE: Wow! That's depressing.

00:50:50 ED: And so I couldn't go back to research, not in the way I wanted to do it. And I said to Trevor Mudge, "I could go back, you know, I could give up that and just be a faculty member and kick around", I said, "but I don't want to end up like one of those other guys who just draws his salary, goes home, teaches his course and goes home." I said "It's not right. I don't want to be a has-been." And Trevor looked at me and said, "Oh, well, that's okay, you've paid your dues, you could be a has-been" you know, and he is not a has been, he never was. But I just didn't want to do that, you know. And if I could have done the research on a small scale and, you know, do whatever teaching I had to do to keep going, I would have done that. I was totally frustrated with the administration. I walked into Pramod's office and I said, "You know this isn't fun anymore," And I said, "I just wake up in the morning and I just hate coming to work now." And he said, "Well if that's how you feel you should retire." I was 60 years old, which is the youngest you could be and retire with full retirement. And the stock market was great, I had enough money to retire on. And I did. And so at the end of academic year 2000 I retired.

00:52:30 ED: A year later Pramod came to me and said, "You know, I think you retired at the right time." I said "What on earth are you talking about?" He said, "Well you retired before the bubble broke" and I said, "Yeah, I retired before the bubble broke but I stayed fully invested, I rode it over the cliff." [laughter]. But fortunately, I have done some expert witness work and that's been lucrative, in a few weeks, a year expert witness work, I can earn more than I ever did as a faculty, well for year. So, you know, that's fine, that's kind of keeping the wolf from the door.

Interview recording part 7

Lessons learned

And I talked about, you know, thinking back on my career in academia and in Computer Science, I think one of the greatest things about working in computers is that there is almost no gap between concept and realization. That you can try out ideas so fast, and it's a great field for people who demand immediate gratification. Now you work hard, because there is always a lot to do and the tough part, really is, in thinking up and describing the right problem. Smelling the cheese as it were. I don't like jumping from one problem to another before you milk it. I like getting the understanding. And there is a lot of people very, very smart people who do; they know the next fashion in Washington and they are there with the first proposal. In fact they are more than that. They are there being on the committee that's going to define what demand and proposals are going to be in. But if they jump around too quickly they don't... They're just always picking low-hanging fruit, which they can do very quickly, because they are so smart and so well connected. And I think there is a shame in that, I think that's a waste in resources.

00:54:54 ED: One of the nice things about working in the midwest is we're a little slower and you can take a problem and work on it for a long time. But you can test things out and you get feedback, very quickly. Like the Kleinschmidt printer with the row of F's, I hope I talked about that in the first interview that we did. The rolling, there is nothing like rolling that roll of F's up above the carbon, and seeing it on the page. Computer is a great field for that. And a friend of mine from Stanford, Frank Weinhold, was in chemistry and he said to me, "You don't know how lucky you are to be working in computers instead of chemistry. In chemistry you come up with a new compound that you can synthesize, and then you have to figure out... First you invent how to synthesize it, then you have to figure out what it might conceivably be good for. So you spray it on a fly and you see if the fly dies, and you painted on a building and see if it holds the color better in the sunlight, and you give it to somebody with a disease and see if they get better. And you know, the chemical similarity of one compound to another doesn't well dictate what domain it might be useful in". And, a friend of mine in Mechanical Engineering said, "You don't know how lucky you are to be in computers because the model for something behaves like the reality of that thing. A NAND gate does an NAND function. You build the material, concrete, composite, and you can make a theory of how it's going to fracture. But each one you make is going to fracture little differently, it's not... It doesn't follow the model, and you just never know, and you can't get reproducibility" And in computers we can.

00:57:04 ED: It's been sad for me to see it become more of a commodity industry and less of an inventive industry. When everybody was inventing their own instruction sets, people were upset about that and wanted standards. And I spoke at a conference and said, "Don't be upset. This is a golden era, everybody is inventing stuff. It's great. You know, you are trying things out. And don't try to standardize anything that's... That once it's all standard... There is a time for standardization in industry for commercial reasons, for economic reasons and for interchangeability and that's a good thing. But, don't stifle the periods of creativity because it's not always that way." And I think for a long time we have been in a down period with the dominance of microprocessors over other technology and permutations and the dominance of Intel, in the microprocessor field, and the dominance of Microsoft in software applications field, things become hard to innovate, in practice. And you can work in the application domain and I think we don't do enough of that in academia, and I think we are doing more of that now.

Interview recording part 7

00:58:31 ED: But two things have happened recently. One is that you can emulate Intel machines on other kinds of implementations. And you can have chips manufactured in Asia with the advantages of mass manufacturing by these companies that put several different projects together to get volume so that they can negotiate price. There aren't very many foundries now because Intel has made them so expensive with Moore's law, the need to keep on Moore's law and priced everybody else out. IBM is dropping out or dropped out. I don't know what they are doing for Intel now. TI, I don't know what they are doing. Motorola I don't know what they are doing. But you can get stuff made, and you don't have to be beholden to Microsoft because you have these virtual computers that you can build, and you can emulate the standard, under another kind of engine. And there is all this hand-held world, the appliance world, that has gone wild, which is a place for innovation. And there are academic problems, like power conservation, to come along with it. It's a very, very exciting time to be in the computer field. Not like when I left it, in 2000, when it was boring. [laughter]

So, one of my remaining frustrations is that there are not more women in computer science. And, you know, I have read studies that say that women in high school see the geeks and they don't want to be a geek.

1:00:33 PE: Yeah.

1:00:33 ED: And, they don't want to spend their lives working with machines. They want to work with people. One of the things that I hope my story illustrates is that a life in computers has to do with working with people. One of the things that made Dave Crockett a genius in management is when somebody wanted to leave Hewlett Packard, he was the guy that had to talk to them, and he said, "Why do you want to leave?" And he said, "Oh, I got a better offer, and it looks more interesting." And then he'd talk to them some more. And it wasn't the money, which was the one thing that he wasn't allowed to do, cause if you start rewarding people with money when they want to leave, everybody's going to play that game. He would ask them what they saw in the new offer that was technically more interesting. And they'd describe it, and he said, "Well, is that really what you want to do, or is there something else you want to do?" And nine times out of ten, the guy would say, "No, what I really want to do is this." And he would say, "Oh. How about if we set up a group at Hewlett Packard for you to do that?" And he would put them in a place in the management structure where there was somebody over him who would be sympathetic to that, and off they'd go. That was in the good old days. But it is people. How do you keep your best people, and how do you get them doing what they enjoy?

01:02:03 ED: So the architecture that I spent most of my life doing is not of terrible interest right now. But some of the things that I wish I was doing are of terrible interest right now; parallel computing. I don't understand why our techniques for working with the applications and tuning them to parallel applications and multi-threading and that sort of thing didn't catch on, but they didn't. Maybe we didn't push them far enough, or hard enough, or market them right. And since retiring, I have worked... Talking with some folks, three guys, who found a technology for building a free space crossbar switch. It can connect hundreds of things on one side to hundreds of things on the other side by shooting an electron beam through a vacuum, through a small vacuum. And this can be built in a four inch by four inch by two inch rectangle.

1:03:21 PE: Wow.

1:03:23 ED: Using some of the technology that was developed for flat screen TVs. The advantage of all the optical stuff that they're doing is that the beam is immensely steerable. So you can steer it to a different point to a target array in under a nanosecond. And you can change your mind every cycle, and steer all these beams around.

1:03:53 PE: Wow, that's amazing.

1:03:54 ED: And you can produce it pretty low cost. But, we haven't been able... I've been advising them. And we've talked to a bunch of people, but there are no venture capitalists these days who want to put money into that. But, for ten million dollars, we could get up and running. I don't know why, 'cause like I said, the two problems that I thought were most important in the parallel domain, which is where people are going now, were the latency of the interconnect and the data movement, and the parallel software, which is beginning to come out now with multi-threaded systems.

1:04:42 PE: Yeah.

1:04:44 ED: So now that I'm like old Bill Dow. I can rattle around and bitch about why the world isn't doing the right thing. There's a prayer that I like very much, which is part of the Jewish Yom Kippur liturgy, which we recited yesterday in temple. I'll do part of it. To me, it's what it's all about.

> Birth is a beginning, and death is a destination, and life is a journey. From childhood to maturity and youth to age, from innocence to awareness, and ignorance to knowing. From foolishness to discretion, and then perhaps to wisdom. From weakness to strength, or strength to weakness, and often back again. From health to sickness, and back, we pray, to health again; from offense to forgiveness, from loneliness to love, from joy to gratitude, from pain to compassion, and grief to understanding. From fear to faith, from defeat to defeat to defeat, until looking backward or ahead, we see that victory lies not at some high place along the way, but in having made the journey, stage by stage, a sacred pilgrimage. Birth is the beginning.

1:06:26 PE: That's beautiful. Okay, we'll stop there.