Oral History of Bill Carter

Interviewed by:
Steve Trimberger

Recorded: July 13, 2015
Mountain View, California

CHM Reference number: X7545.2016

Steve Trimberger: We are at the Computer History Museum. Today is July 13, 2015. I am Steve Trimberger. I am interviewing Bill Carter, and today we'll find out a little bit about the history of the PLD business and a few other things as well. So Bill, welcome.

Bill Carter: Thank you very much, Steve. Thank you for having me here.

Trimberger: So tell me a little bit about yourself and growing up. Where did you grow up, and how did you get into this technology business?

Carter: I grew up in Phoenix, Arizona from an agricultural family. I kind of always imagined I'd be a farmer when I grew up. That's what I imagined through grammar school. When I got to high school, I discovered I wasn't good at math, but I enjoyed math, but I really enjoyed physics and chemistry, which were the two classes they taught at the high school I went to. And I kind of imagined I might become a scientist one day.

The summer between my junior and senior year I got a job working at a US health education welfare laboratory in Phoenix where they were trying to isolate the hepatitis virus, and so I was surrounded by a lot of scientists for the summer. And they had one person that was there that was an engineer. It was a guy from NASA, and his job was to try to determine how do you guarantee when you send a probe to Mars that it doesn't infect the Martian environment. And so his problem was, for example, I have a transistor. How do I ensure that our transistor, if it breaks on impact on Mars, doesn't have some microbes inside of it that would contaminate the atmosphere, and so interesting guy. I got a chance to talk with him at lunch and things like that quite a bit, and I asked him one day, so what's the difference between an engineer and a scientist? After some thought, he said, well, scientists seek out knowledge for knowledge sake, but engineers take that knowledge and use it to solve problems. I'm not sure if that's a good definition or not, but that's what he said, and I sure liked the idea of using knowledge to solve problems, and so that got me interested in at least exploring this notion of engineering. And that's where I got my start in that.

Trimberger: I want to follow up on something you said that I think was a bit subtle, and I just wanted to highlight. Well first of all, what year was this when you were doing that?

Carter: That was probably-- jeez, when did I graduate from high school? 1966.

Trimberger: So you mentioned he was worried about what if a transistor broke open on Mars.

Carter: That's right.

Trimberger: Today when we think about transistor, it's kind of an atomic thing and getting close to literally atomic. What was a transistor that he was concerned about breaking up?

Carter: His transistor was a metal can with three legs, and it was a bipolar transistor. That was pretty much all that existed at that time. I didn't even know that, by the way, when I asked this question. He was playing with Plexiglas as a substitute or a surrogate for the material inside of a transistor, but it was just an interesting problem. I have some kind of a solid. How do I know what's inside of that solid? And he was trying to figure out a way to diagnose what was there. So interesting problem.

Trimberger: So when you were a kid, what did you do for fun?

Carter: I did the same thing kids, I think, all over the US did at that point in time. I rode my bike everywhere. The neighbors-- we would play baseball in the neighbor's backyard. The grammar school I attended was close by, and so we'd go over and play at the fields there. Every summer I swam a lot. I

think the way my mother got me out of the house was to send me to the swimming pool, and so it was just normal kid stuff. Nothing exotic. I had no idea about science, or engineering, or anything like that back then.

Trimberger: So how did you wind up in electronics? How did you get exposure and start doing that?

Carter: There was a time probably when I was late grammar school, early high school, where I got into I would say ham radio, but it really wasn't ham radio. It was more shortwave listening. I don't remember what prompted that, but I explored the possibility of getting a ham radio license. And they had this onerous requirement that you had to learn Morse code at five words per minute, and I had no way of learning how to do that, but I did have a receiver, and my dad helped me put up a long wire antenna. And the receiver was as big as a microwave oven is today. It was huge, and it was just a short wave radio receiver. And I would put on my headphones and listen to broadcasts from all over the world. For me, that was just amazing that you could hear people talking to you from virtually any place on earth and you could hear it over the air. And so for me, that was really exciting.

Trimberger: Did your parents do anything particular to encourage or discourage your engineering bent?

Carter: Not really. A long time later I asked my dad about it. I think my parents took the approach that, if you get a good education, you'll figure out something to do. And they weren't too concerned about the specific field that I might pursue or the career I might have after that, so they gave me wide latitude in that regard.

I asked my dad a long time later about engineering, and he said he actually had been an engineer in school for about a semester. The graphic design class did him in. I found his book, and I was asking about the book. And it was a textbook for doing-- you could do great fonts. You would draw them by hand, and he said that class did him in, so he ended up dropping his pursuit of an engineering degree. He ended up actually getting a degree in animal husbandry, which, again, that made sense, because he came from a background where cattle was involved. And so that's what he studied eventually in college after World War II.

Trimberger: Who are your heroes growing up? Is there somebody that inspired you?

Carter: Probably my dad more than anybody else. Our family had some property in Phoenix. My dad was the second child in his family of four kids. His older brother ended up being the one who kind of ran the farm, although his older brother wasn't as technical as my dad was. And so my dad actually worked for the Arizona County Agents organization where he was the local expert in Maricopa County, which is where Phoenix is, in the growing of cotton.

And so during the week, he would teach people-- he would run experiments on people's farms and write papers once in a while, although I didn't discover that until much later, but he would advise people how to grow cotton. But then he would go out to our family ranch every weekend. He would talk to my uncle, and they would talk over what he would do the next week, and I was a kid who got to play in ditches and all the different things. My cousin had a horse, and we would ride that occasionally. I drove a tractor. And the ride out from our house, which was in town in Phoenix, out to where our ranch was probably a 20 or 30 minute drive primarily through agricultural fields, and my dad would teach me how to spot the different plants by sight as we would drive along. He'd say what's that? What's that? What's that? And I could pick them out. And so it was just fun. So I had a private time with my dad, and none of my other siblings participated in this activity. It was just the two of us, and I think that's probably the closest thing that I had to a hero. And a side that I think is kind of interesting is the property that my family had was in Glendale, Arizona, and it's now the spring training camp for the San Francisco Giants. And I hadn't been back to see the spring training facility, but it's on the property that I played in as a kid.

Trimberger: That's great. So at some point you decided you were going to go to college.

Carter: That's right.

Trimberger: And where did you go, and what was your major when you got to college?

Carter: So that's kind of an interesting story, too. I applied to a lot of colleges, primarily on the west coast. I wasn't too interested in going to the east. I was accepted at several places, and I decided to go to a school that's on the peninsula here in the Bay Area. And I didn't pursue aggressively-- although I was accepted other places, I really didn't pursue aggressively going to the other schools. I couldn't afford any of them, but I decided what I would do is I would get a ROTC scholarship. I would trade a few years of my life in the military for a college education, and that looked good. So I actually had a dorm room scheduled, and everything was good, but about April of my senior year in high school, one of the requirements to get the scholarship-- I had done all of the other paperwork, and got the right grades, and all the letters of recommendation, and everything, and so I was good to go, but the last thing I had to do was pass a physical. And I flew to Los Angeles with a friend of mine who was doing the same thing. And we got to this place, and had our physical, and I failed. My vision had to be 20/40 correctable to 20/20, and I was 20/70 in one eye.

And so here I was. I wasn't going to be able to get my ROTC scholarship. I had a room at a school, but I couldn't pay for it, and I was in kind of desperate trouble. And my brother was going to school where I ended up going to school at the time, and he said you ought to pursue financial aid here. And I said it's way past the deadline, and he said it doesn't matter. Just try. See what happens. And I tried, and they accepted me. And it was probably the best thing that ever happened to me.

I went to Santa Clara University here in Santa Clara, California, tip of the South Bay, and it was a small school. It was an intimate school for an engineering organization. It's not a big research university, but it's a place where they focus on actually teaching, and I got to know the professors personally. And I needed that one on one guidance at that point of my life, and it was really important. For graduate school, those other schools may have been perfect, but this was a perfect school for me as an undergraduate.

I graduated in 1971 with a degree in electrical engineering, and I had also continued in the ROTC. And I was commissioned the same day I graduated as a second lieutenant in the Army, but not on a scholarship. The difference was now I had a two-year commitment as opposed to a four year commitment for active duty, but they let me go on to graduate school. In fact, I looked around for a job, and in '71 that summer there was no way I could even buy a job. The job market in the Bay Area was terrible. The job market everywhere was terrible.

And so I stayed on, and I stayed on at Santa Clara doing graduate school there for two additional years, and they have a program to this day called the Early Bird Program, where classes are taught in the mornings from 7:00 to 9:00, and that most of the people who are doing the teaching are engineers who work in the valley. And when they finish class, they go to work. And a lot of people tried to get me to narrow down to what particular sub specialty within electrical materials I wanted to go into, and I wasn't going to be pigeonholed yet. I didn't know. I wanted to try a bunch of things, and it was during that that I actually bumped into three guys who were local engineers.

One guy was a guy named Ed Snow, who was a semiconductor physicist at Fairchild Semiconductor, and so in the daytime he was doing the work that he was teaching us the next day in our class, and there was no text. It was basically his notes that he was teaching because nobody had created that text yet. And the two semiconductor IC design classes that I took were from a guy named Hans Camenzind and Alan Grebene. These two guys were the co-inventors of the 555 timer, which is probably the most ubiquitous IC that I think has ever been developed. And I said finally this is what I want to do, because if you were down at the transistor level, you can do anything you want. You're not constrained by what somebody has done who developed the component that you're using to design your system. I said I'd like to have that flexibility to do just exactly what I want down at the transistor level, and so that really intrigued me, and that's what I ended up pursuing.

Trimberger: So did that lead to your first job?

Carter: Well, ultimately it did. So after I finished two years of graduate school, I had a military obligation to fulfill. It was only a short stint because the Vietnam War was winding down, and they had way too many officers. So after three months in Augusta, Georgia, I came back to the Bay Area and started looking for a job pretty aggressively. And the environment had changed. Now you could find jobs. Lots of jobs. Virtually everybody I pursued a job with offered me a job. So what a change two years had made.

One of the first jobs that I got offered-- I turned down-- was from a company that was in the memory business here in the valley called Intel. And there were a variety of reasons why that job wasn't going to be the best one for me, and I ended up not taking it. And I've gone back since and looked at the offer letter, and I've looked at the stock options-- which I really didn't understand what stock options were at that point in time-- and I've never computed the value because I thought I would make myself upset, but it was probably the best thing to happen to me, too. A lot of good things have happened to me.

I ended up going to work for a company called Scientific Microsystems. They were a sister company of Signetics. In fact, the SMS came before the name of the company because originally it had been Signetics Memory Systems. They were doing some bipolar RAMs, and that looked like a nonstarter for Signetics, and they ended up spinning it off. And it became a new company that was pursuing something I had never heard about, something called a microcomputer. And I kind of asked my first boss when I started to work there-- it was a start up. It was not as young a start up as I ultimately joined later on, but it had no product. We had a good idea-- I thought a good idea-- and we had a small IC design department with, I think, three engineers, four engineers, and a couple mask designers. So you were thrown into the middle of this, and you got to do a lot of interesting things, and so it was fun.

But I asked my boss, what is a microprocessor? And he scratched his head, and he said it's a way to replace gates with instructions, and that stuck with me. I've used that analogy ever since, because if you're not in a hurry, instructions can solve most of the logical problems that you've got. If you're in a hurry, maybe a gate is a good thing to use. Before that, most everything was built with a device from the TTL catalog wired to a bunch of other devices from the TTL catalog to make a system, and they were going to build a microprocessor that was going to do most of the computation. Interesting machine. Harvard architecture, bipolar, very fast-- for its day, anyway-- huge chip, worked-- went into production for a number of years. Signetics ended up taking over that business, and they ended up selling it, but good introduction for me to IC design. So I had a little bit of education in that, but this was the real practical first taste of that. They taught me well. And also it kind of an introduction to microprocessors and microcomputing.

Trimberger: And do you have a particular mentor at SMS?

Carter: At SMS there was a gentleman named Bill Price, who was my boss. Bill was a great guy. So the first thing I did when I got there was to help out with a microprocessor. I was coming in late in the project. Pieces were already being designed by different engineers there. My job was to put it together, to integrate it. And so maybe I digress a little bit-- when I got out of college, I wanted to become the next Bob Widlar, who was the folk hero in the valley for analog IC design. He was the guy who conceived of a lot of the basic building blocks that were used in all of the analog devices that I saw, and that's what I wanted to be. And Bill Price eventually told me I got to be the Bob Widlar of interconnect, just wires, because I did the piece that hooked all these subsystems together to form the chip. Good experience. Did a little bit of everything, and that was good, too.

The first chip I had, I guess, the technical lead, although I think they used the notion of lead kind of leniently. It was just one I was responsible for, but most of the design of the logic at least was done by somebody else before me. I was going to do the circuit design and follow it through to manufacturing, and I had a budget for power and things like that, and it was a tough design. I had to drive 300 picofarad loads with TPDs or prop delays from input to output of 10 nanoseconds, which was really fast for the day. And

told him, if I took, I guessed, a moderate beta for a transistor-- and I said, if I put all the current budget that they had given me into the base of the transistor driving a 300 picofarad load, I didn't have enough to even switch that capacitor in 20 nanoseconds with a reasonable beta. And I said I can't do this. It's impossible.

And Bill Price worked with me, and we did it. And I missed my 10 nanosecond goal by about a nanosecond, at least in the simulations. I think almost all the production stuff turned out to be that fast. And when he gave me a hard time about missing it by a nanosecond, I said, would you write that out with a decimal point and all the zeros and the one in seconds so you can see how far off I am? And he laughed, but he gave me crap for it for a long, long time.

Trimberger: Then what happened after at SMS? They went public. You got fabulously wealthy, and then they were done, right?

Carter: No, that's not what happened, Steve.

[LAUGHTER]

SMS, they gave me stock options. Again, I had no idea what these things were, and that's good because they turned out to be worthless. SMS was reacquired and sucked into Signetics. They went into the system business. They had a captive IC capability. They stayed in the system business, and the few people who were doing IC design work for them had the option of going back to work at Signetics. And so I had never really worked at Signetics, but they offered me that opportunity.

So the first set of chips I did were all low power Schottky, and they wanted to build a floppy disk controller for this family of microprocessors. Floppy disks were just coming out. These were big eight-inch floppies, not the little floppies. And so I took a design that somebody had done, and I converted it to a new technology that Signetics was pioneering called I-squared L. So it was, in fact, a hybrid. It was the same process technology, the high-speed stuff. So the data separation out of a bitstream from a floppy disk was done in low power Schottky, but then after you got it down-- instead of bit rate to byte rate-- so we've dropped the speeds down by a factor of eight. All that was done in I-squared L it was much more lower power, much lower density. So it took a lot fewer transistors to solve the same problem, and that's what I ended up doing, but I had teach myself how do I-squared L. Nobody really knew how to do that. Signetics was about the only purveyor at the time-- probably the only purveyor ever, because it wasn't a great technology, but it was good for what it was.

It was an interesting puzzle to solve because it was completely different than anything else I'd ever done, and you had to worry not about fan out, but fan in. It was just backwards from-- did I get that backwards? I'm not sure, but it had one output, but you could have as many inputs as you wanted. And you could have a couple of outputs, but a few number of outputs. It could only go-- and each output was dedicated to one destination. You couldn't bus them like you would one output to several inputs. It had one output for every input that it goes to, and so you had to worry about how many things it drove, not the number of inputs coming into the particular gate, and so you had to kind of turn yourself on your head. So I figured out how to do that, and it was fun.

But I discovered I enjoyed doing the logic design more than I enjoyed doing the circuit design, and I looked around, and bipolar technology when I started was the primary technology being used, but we saw this other technology, MOS technology, was beginning to emerge. And I wanted to learn that. In fact, that's probably one of things that's been a hallmark throughout my career-- is wherever I changed jobs, it was primarily to learn a brand new technology. And so I wanted to do something in MOS. Bill Price said MOS, "moss," which we called it-- said moss is the stuff that grows on the north side of trees. He said you never want to go do that.

But funny story- I never wore a tie, and he gave me a hard time about that. And I said kind of in jest one time-- I says, Bill, the day I wear a tie is the day I turn in my resignation. Well, the day I turned in my resignation, the first words out of his mouth is, where's your damn tie? Because I didn't have the courage to put one on and go tell him I was quitting, but I ended up going to work for a company called Zilog. They were a competitor of Intel. They were working on their 16-bit version of their processor. They had just begun that, and I was going to go to a peripheral for that, and that's what I did. It was in MOS.

It was an interesting thing because this was probably the only chip where I did everything soup to nuts. I came there. The first thing they had me do was write the product spec, create it. I really didn't know what the process technology was capable of delivering, and so I just wrote down a wish list, took it around, and got it all approved by everybody in the organization. Then I took that hat off, and put on another hat, and said now let's see if we can put this and turn it into reality, and I did. I ended up doing the design, all of the logic and circuit design, for one guy. I had a couple of mask designers who did the drawing, the layout of the chip. And then when the chips came back, I was the one who wrote the debug program and then the test program, so it was kind of like a one-man chip soup to nuts from the product definition through test program at the end. And it was amazing fun. It was a ball for me to be able to do that, but I don't think something like that was even possible then. It was barely possible then, and certainly not possible anymore to have one person do the whole thing.

Trimberger: It sounds like it was good experience for you.

Carter: It was fabulous experience.

Trimberger: When was that?

Carter: I started that in 1977, and I stayed at Zilog. I did a number of other things there, but I stayed there for about seven years. And for me, it was great.

Trimberger: So it was at Zilog where you met the founders of Xilinx, right?

Carter: That's correct.

Trimberger: Tell me about that.

Carter: In fact, the guy who hired me into Zilog was-- in fact, it's funny. One of the guys who first interviewed me, he knew a guy at the company I had worked at before, at SMS, Scientific Microsystems, and his name was Gary Prosenko. And Gary invited me to come in, and he was doing something it was microcontroller, which became something that became the Z8, but I drug my heels, and eventually he filled the position by the time I got interested seriously, and that position was gone. Well, there was a guy, Ross Freeman, who needed somebody to design a peripheral for the Z8000, their new 16-bit microprocessor, and that's who I interviewed with.

And this is the first time I had met Ross, who ultimately was the primary founder of Xilinx, where I went next, but Ross-- the interview was the weirdest thing, because Ross knew nothing about bipolar design. He'd come from a PMOS background at Teletype, and I came from a bipolar background, and I was going to be doing NMOS. And so I could talk about what I did, but I couldn't talk about the technology that he wanted me to use. And he could talk about PMOS, and I thought that was cool, but, again, we couldn't talk technically one on one about things. I always remember subsequently thinking when they offered me the job it was, why? I wasn't the right guy in a lot of ways, but he did, and I'm sure glad he did.

Trimberger: So you met Ross there at Zilog. Did you also know Bernie Vonderschmitt there?

Carter: So Zilog at the time was kind of an example of a company that had a problem with lack of focus. There was a components division and a systems division. So we built boxes using the Z80 microprocessor in peripherals, but we also had an IC capability. And the company-- I got stock options there. It was eventually acquired by Exxon Enterprises, a subsidiary of Exxon when they were trying to diversify into some of the high tech stuff. And so my stock options went away since they wholly owned the company, but they gave us something they called performance units as a surrogate for stock, but we weren't doing very well. It was a struggle. Especially the box division was having some trouble. We had a change in management, and they brought in a new manager for the components division, which is where I was located, and Ross was my boss. He ran a big chunk of the components division, but they brought in a new manager for that, and this Bernie Vonderschmitt. And Bernie was a veteran of RCA.

He had been on the east coast in Somerville, New Jersey, and he had gone up through the ranks there starting with color TV. In fact, he held a number of patents on the original color TV technology that RCA created. He ran their solid state division, so all their semiconductor business was a part of a thing that was his realm. He learned the business on the job, but he went back to school and got an MBA so he could put formal training around the practical experience that he'd had. And then he decided to come out to the west coast because he saw a lot of interesting things happening there. And so he started, and he became the manager of the entire components division. And Ross worked for him, and I worked for Ross.

And so I didn't know Bernie very well, but I knew I had enormous respect for Ross. I had worked for him for a long time at this point, and he had a lot of respect for Bernie, but because I had Ross between Bernie and me, I didn't see Bernie all that much, just at the big meetings where you're giving your reports to the senior management. So that's about as much as I knew about Bernie.

Trimberger: So tell me about starting at Xilinx. How many people were there? What's your employee number? What was it like that first day at this company?

Carter: So like I mentioned, I worked for Ross for a long time, and one day we had a regular one on one, and I came into my one on one-- this must have been by late 1983, probably December of '83. And near the end of our one on one, he said, I'm going to tell you something, but you have to promise me you're just going to go home after I tell you that. And I really had no idea what he was talking about, and he told me that he was leaving the company to start a new company, and I was shocked. I had no idea.

And so I didn't like losing a boss because he and I had a very close relationship. He was as much a friend as he was a boss, and I did what he said because I'm sure I turned white, and I just went home. And so he, and Bernie, and another guy who worked for Ross, a guy named Jim Barnett, left soon after. They founded Xilinx, and I had no idea really what they did, but at that point-- so my boss had departed and his boss had departed, so suddenly I was reporting to the CEO of company, at least on an interim basis until they filled those intervening positions.

And so I did that for a while, and after a while, I discovered that Ross and Bernie had buffered me quite a bit from what was really going on at the higher levels of this company, and I could see why they left. And at that point, I made the decision that I had to seek employment somewhere else. Now, Ross and I had kept up a relationship since we were friends, and I had scheduled to have a lunch with him in Los Gatos, California. And I called him and told that I was going to be leaving the company, and he said, I'd like to change our lunch plans. Why don't you go to JC's Barbecue on Winchester Boulevard and pick up some hot link sandwiches? And I'd like you to come to my house. I want to talk to you about what we're doing. And I think the idea was that, since I initiated this notion that I was leaving, it was OK for him to talk to me rather than him plucking me out of the company. That might have been bad form. Ultimately Zilog decided it was bad form either way, but that's a different story.

But I went over to Ross's house after we went to the barbecue, and he showed me on his kitchen table the idea of what he was planning to do at Xilinx. And I remember very vividly walking back out to my car and saying to myself, this is the stupidest thing I have ever heard of in my entire life. I had enormous

respect for Ross, and so I didn't have the courage to tell him to his face what I thought, but I said this is crazy.

And the reason I say that is because I had been trained throughout my career-- every job I'd ever had in school and in my jobs-- is that the way you went in the semiconductor business is you solve a particular problem using fewer transistors than your competitor. Number of transistors implies chip size, implies cost and maybe speed. And so if you could solve it in fewer transistors, you win. You win in the marketplace. And what Ross was suggesting was probably the least efficient use of silicon that I'd ever heard of, and so it was going to be expensive. His product was going to be expensive. It was going to be slow. And I said, how is this going to win in the marketplace?

And so I kind of discounted it, but the more I thought about it, the more I realized so maybe there's something here I don't see. I really like working for Ross. Might be fun to try a start up, a real start up. Every company I'd been to, none of them were public, and all them had stock options, but they were at different stages of maturity, but this was going to be starting from ground zero. And I said, what the hell, let's give it a try.

And so I talked to my wife. I was-- let's see, '84. How old was I then? 35. And I had two children and one on the way. No, I had three children. One was very young, and I talked to my wife, and I kind of told her this is a real gamble. I had a mortgage. I had all of the stuff that goes along with responsibility, and she said, if it makes you happy, go for it. And I said thank you, and so I went for it.

And so I turned in my resignation. They walked me out of Zilog. I told them I'd be happy to stick around for a smooth transition, and they said they weren't interested in that. They wanted me to leave right now. And so I walked out the door. I had a week of vacation I hadn't anticipated. My wife and I went to San Francisco.

And I started-- I think it was a Monday in 1984, and I think it was March the 5th, so if you go back and look at a calendar, whatever the Monday is, it's first Monday in March. That's when I started at Xilinx. Our office was at 160b Albright Way in Los Gatos. If you go there today, they're building a building that's the expansion of Netflix. It was a very small little piece of a building. The company was founded in January, but they had no facility. The day I started was the first day they actually had a facility, and that's if you use the word facility in its broadest sense, but that day I arrived. There was three offices and a conference room. That was it, and a warehouse area in the back.

This was an era when phones-- there were different companies that would provide the wire to the building, and then another company would do the wiring inside the building. We had forgotten to arrange for the second part, and so there was a phone in the warehouse area, a pink slim line phone. I forgot what they call it, but that was the corporate phone. If you heard it, you ran out and picked up the phone. When I came there, didn't know what I was going to do for sure.

A bunch of people from our former employer that were friends of ours, which was actually physically not too far away-- there was a program on the line printer at Zilog that you could print banners, and they printed a banner that was Xilinx. And they brought it there, and that was our first sign literally, and we ended up having exactly that font made into a wooden sign etched on the front outside the building there, and that's where the font for the first sign came from-- was just the banner program of whatever program was that they had at Zilog.

And the people there all came to get a picture with us, but they turned their faces away from the camera. So we all faced the camera. They turned their backs to the camera so they wouldn't get in trouble, and so that was the first corporate picture.

Trimberger: So you mentioned the Xilinx name, and so I have to ask the most important question about Xilinx. Where does the name come from?

Carter: Where did the name come from? There are lots of stories as to the background of that name. I don't know if some of the informal stories or the hearsay stories are accurate or not. Two of the founders were from the University of Illinois, and so some people said it meant two ex-Illini-- but that's not the story I heard. I'm not sure if that was made up before or after, but the background of the story goes this way. They came up with a name for the company, and I'm trying to remember what it was. Logica. L-O-G-I-C-A.

And there's an entity in the state of California that you go take your name, and register it, and do all this kind of stuff. And so they took the name. They paid their fee. They fill out the forms, and they were rejected because the name had already been taken. And so the three founders got back together, and Bernie was the president of the company, CEO of the company. And Bernie said to the other two, come up with a name. It can't ever be used before because I don't want to pay another fee because that's just wasted money. Bernie was very, very frugal. I can tell you frugal stories about Bernie, but he said the only requirement for the name is that it begin with an A, B, or C, or an X, Y, or a Z. I don't know why that was the constraint. And basically he said, with that requirement come up with a name.

And I actually got a call from Ross when I was still working at Zilog. He was doing his market research on names, and he read me a number of names. And one of the names was Xlinks. And I said to him you can't call it Xlinks, because Xlinks reminds me of a product you buy at the drug store, and that's not suggesting something that's really good, so please don't name it Xlinks. Well, I guess they went back to the drawing board and changed it around, and it became Xilinx, X-I-L-I-N-X. The first X sounded like a Z.

And the story goes that the Xs at each end represent the logic blocks of an FPGA, and the magic sauce in an FPGA is the programmable links, or interconnect, that is between the logic blocks. And so it was X link X. They massaged it around, and it became Xilinx.

Trimberger: And you were the guru of interconnect, I understand, from a few years before, so you were there.

Carter: That's right. I didn't know the story of the name at that point in time either.

Trimberger: So you joined Xilinx despite the misgivings about the technology, had faith in the founder.

Carter: I didn't have the vision Ross had for about two or three years, but eventually his wisdom became crystal clear, but it took me a while. It was the fun of doing something brand new. We were going to do CMOS. None of the founders had ever done any CMOS design in their life before. I hadn't done CMOS, But I figured it was an easy step going from bipolar to NMOS because it was a lot less complicated. It only took two transistors to make an inverter instead of a whole handful in bipolar, but I said I could probably figure this out. And I guess they figured I could probably figure it out, too.

And I think that was part of my interest. Again, I mentioned before I like to learn new things. I have an appetite for something, and I think if I get bored with it, then I want to go on and learn something brand new. And this was going to be something much different than-- I had always been in the microprocessor or the microprocessor peripheral business up until this point. Now I was going to get a chance to something I knew nothing about.

Trimberger: So-- go ahead.

Carter: And so I think all of those were contributing factors, and I think-- Ross would always say to me, whenever you change jobs, there's a push and a pull. Well, that was the pull. There was the push. I really wasn't enjoying my life at the other company, and so this looked like a good opportunity. I didn't have to break in a new boss.

[LAUGHTER]

And so I figured, let's give it a whirl.

Trimberger: So maybe this is part of Ross's vision, but let's talk about there was programmable logic before Xilinx.

Carter: That's right.

Trimberger: I'm thinking about PLDs.

Carter: Right.

Trimberger: So what was the big deal? What was so different here instead of it's, oh yeah, he's just going off to do a PLD? What's happening here? What's different?

Carter: I think the difference is that-- so the popular name for the programmable logic that was popular at that point in time was PALs, programmable array logic. I think PLAs where the technology that they used, but they changed those letters around to form the name PAL. I had actually seen programmable logic at my first company, but they used bipolar PROMs to implement arbitrary combinations of gates. And so I had been exposed to it a little bit.

One of the different things here was it was scalable. PROMs are very inefficient for implementing programmable logic because you really don't need all those memory cells to implement some form of programmable logic. That's where PALs came in. PALs-- in fact, the first time somebody told me what a PAL was, they said it's a partially populated PROM. And I said, oh, OK. That made sense. It had an and plane but a limited or plane. And I said, oh, OK.

And so I could see that it was not as flexible as a PROM, but it was much, much more efficient than a PROM. Well, what Ross was coming up with was something that was much more efficient than a PAL, which was still something that, as you scaled it, it grew by the area. It didn't grow linearly, where Ross's idea-- in fact, the first picture I saw of an FPGA-- we didn't call them FPGAs at the time-- was the lower left hand corner, because you could scale this idea out to arbitrary tall and arbitrarily wide without really changing the fundamental structure of the device. You may have to change the number of resources that are there, but the architecture fundamentally doesn't change, and that was new, but for me, I think it was still just the interest of doing something new. And if we were successful, great, and if we weren't successful, I get a great education.

Trimberger: You mentioned you didn't call them FPGAs. What did you call them, and where did that term come from? Did you invent that?

Carter: No, I didn't invent that. A lot discussion before I came as to what to call this thing. The original name-- there was a number of names. One of them was a PGA, believe it or not. It didn't have the F. It was rejected by Bernie, because PGA implied something to him that was completely going to be a distraction, which was the Professional Golf Association. Bernie liked to play golf, but he didn't get to play golf very much because of his work. And so he said you can’t call it a PGA because it's going to remind everybody of golf. And so I guess they decided to can that one.

When I joined the company, it was going to be called a CLA, a configurable logic array, and we put together a bunch of documentation of our roll out of marketing material. And we actually had Regis McKenna and his firm do some market research for us. And they looked at it, and they said you can't call it a CLA, because CLA sounds like PLA, which kind of implies a PAL, and you're not distinguishing yourself over the existing technology. And so we were really bright guys, and so we changed the letters

around-- two letters-- and called it a configurable logic array, a CLA. I called it a Carter logic array to myself, but I wouldn't say it out lout.

[LAUGHTER]

But that's where the name came from. If you look at our original marketing document, it was called a CLA. Eventually my belief is Dataquest or some other market research firm was trying to create a category for what to call this product when it got big enough to actually get some notice, and we became FPGA. There was a gate array market. It was the GA market, and then it became the MPGA for mask program gate array, and we became the FPGA for field programmable gate array. We put the mask programmable into traditional gate arrays by creating the technology that we had.

Trimberger: So let's talk about the first FPGA, the development of that. What was your role? What did you do? Tell me about the design process.

Carter: So when I started, what we had basically as a design guide was Ross's patent application. That was fundamentally all we had, and so it had kind of a loose description of what he had in mind, but it really didn't get down to specifics. And so we had to kind of work on those as we went along. And so he had a number of different ideas for what the logic cell would look like in his patent application. He had a number of different ideas for what the interconnect might look like, those kinds of things, but we really hadn't narrowed anything down. And so we were starting with a blank slate.

And there was no model. Like with microprocessors you had traditional processors, or mainframes, or something like that where you can get some of the concepts from. We really didn't have a model to copy with what we were doing to change and create what we were doing. We had to come up with it from scratch using what Ross's invention as a guide.

And so we started off. So what's this architecture going to look like? Well, we didn't have a process technology, so we didn't have any models. We couldn't do any circuit design because we don't know what we're going to be doing, and besides, we have no idea what the circuit is going to look like anyway. So we said, well, let's start with the high level architecture. We came up with a high level architecture. We decided that it needed to be in a eight by eight array. I think it's because we were all processor people. That's where we came from, and processors spoke in bytes. We figured it would be nice to do a byte's worth of something in here, and if you could do eight bytes worth of something, sounded good. So let's make it an eight by eight array.

And we ended up kind of creating an architecture, putting down the amount of interconnect we imagined might be necessary, and we ended up drawing it on a blueprint, on a D size blueprint piece of paper. Just here's the architecture of our first chip. And then the job was to figure out, so is this architecture going to work? Is it adequate with the resources that it has? And what we did is we took the MMI PAL applications book. They had lots of simple applications that they showed how you could implement them in a PAL, we used those designs to say let's see if we can implement those in one of our gadgets.

And we did literally with red pencils and blueprint paper and would draw circles around this logic block would have this logic in it, and this kind of thing, and then we'd try to route it by just drawing lines with pencils through the possible tracks that were on this piece of blue line paper. And we discovered we had-- we never stressed the routing. We always had plenty. So what was the first thing to go? The routing, and that's why all the software guys hated us for the rest of my life.

[LAUGHTER]

But we ended up doing-- in fact, everybody in the company, including Bernie, had to take a problem from this PAL handbook and implement it using this paper just to get them exposed to what is the technology we're going to be in the business of doing anyway. So we had already at that point narrowed it down to

how many tracks of routing and how much things would be. We'd seen something. We had a good idea what the logic block was going to look like.

And so then we had to figure out, so is this feasible? So we had what we thought was an architecture that might work, but we didn't know if we could actually build one of these things. And as I mentioned before, we didn't have a factory, so we didn't have any design rules. And Bernie, in fact-- one of the business strategies that we had right from the very beginning was to focus on the piece of the business where we had value, and I guess in current vernacular, outsource the rest. This was pretty weird. Most people did not do that. In fact, I don't know of anybody who did do that.

We were going to focus on doing the architecting, the designing, and the selling. We weren't going to do the manufacturing. And I think part of the motivation there from Bernie's point of view-- he had a run a factory, several factories, when he was at RCA. And he knew that they were extraordinarily expensive, and the only way you can make it economical is if you run it 24 hours a day, seven days a week, 365 days a year. And there was no way our little product with a market of virtually nothing was gong to be able to fill a fab.

And so he said we need to find a partner to help us with this, and that was a problem because most people at that time frame who had fabs also had their own chips. And they filled their fabs with their own chips, and they treated their recipe for making chips as kind of the family jewels. It was really proprietary, and they were very unwilling to share it with anybody else. And we knew our technology was going to stress manufacturing, like I mentioned before, because it's so inefficient with its use of transistors, it's going to use a lot silicon. We had to find somebody who could really manufacture very efficiently.

So we looked around the world, and there were a number of possibilities that came up. I think at the time probably the best manufacturing was being done in Japan, and it turned out in a prior life-- when Bernie was at RCA, there was a company in Japan called Seiko-- Suwa Seikosha was actually the technical name of the company. They made the chips for Seiko watches, the things that made the guts for Seiko watches, which was primarily gears and springs, but they saw the writing on the wall that watches were evolving into something driven by a semiconductor device. And they knew it had to be low power since it's going to be battery powered, and so they heard about this technology called CMOS, which was very efficient with power consumption. And RCA was one of the leaders in the world in this area. They were one of the pioneers in CMOS.

And Seiko approached RCA and eventually licensed RCA's CMOS technology so that they could use it to build semiconductor fabs to build chips to go into Seiko watches. And Bernie was the guy who struck a deal. And eventually, in fact, at the time that we were founded, RCA was still in business, but they had a lot of their manufacturing was actually being done by Seiko because it was the same process. They had given it to Seiko, so when they had a shortage of capacity, they would go to Seiko. They already have a business relationship, and so it worked out pretty.

Well, Bernie approached a friend of his, who happened to be the head guy in the semiconductor business at Seiko at the time, and they struck a deal. And Seiko was not really in the business of being a foundry, although they were doing some work for RCA, but the whole notion of foundry was, I think, an academic term at that point in time. It really hadn't gotten into the real business world, and they agreed to manufacture wafers for us on the condition that they have rights to sell the product in Japan. So we limited them geographically to where they could sell the product. We would develop it, teach them how it works, teach them how to sell it, and they could sell in Japan. We could sell in Japan, too. It wasn't an exclusive deal for them. We could compete with them there, but that's how it started.

And so at the beginning, before that deal was struck, we did have a semiconductor process technologist on our staff. We begged him to create-- make up some models. Just make up some models that you think are reasonably close to state or the art models right now for the devices that we can use so we can get started doing some circuits-- just to get a sense of what things are like. And he was very reluctant

because he thought we were going to hold him to it. We had to swear to him that, if your models are not right, we won't hold it against you. We'll understand that, and so that's how the circuit guys got started-- playing with models that he made up. Circuit simulation-- this is an interesting story. I'm digressing probably too far, but it's a good story, so I'm going to tell it anyway.

Trimberger: Tell it.

Carter: So we were a company that had-- so the first three guys were the founders-- Ross Freeman, Bernie Vonderschmitt, and Jim Barnett. The next three were software guys, which may surprise you, because we're a semiconductor company, but I think Bernie-- more Ross understood that the software problem was going to be a bigger problem than the IC problem. And I think maybe that was because he understood the chip side better, but he didn't understand the software side as well, and so he was a kind of guy that, if he didn't understand, that that's where you put more of your resources so you could understand it. And then they hired a woman who was kind of did everything in the company except the engineering, and then I was the next hire. So I was employee number eight. Three, six, seven, eight. That's right. And then they hired another circuit designer, chip designer, a guy named Jim Hsieh, who worked with me, we were fundamentally the IC design department, the two of us. We started doing some simulations just to play with things. We had to learn CMOS because he had never done CMOS either, and so we started playing, and we had an account with a mainframe computer from a company. And I think the mainframe was probably in Minneapolis or something like that-- that we would submit out SPICE decks, but you didn't call them SPICE decks back then. I don't know what we called them-- SPICE files. But we would submit the simulation files using a modem with an acoustic coupler over a telephone, and wait an hour or two, and get the results back, and see what we did.

Well, it turned out we made a lot of mistakes in syntax, and so it would take an hour or two to find out you made a stupid syntax error that they would discover immediately upon running this job. So most of the time was not simulating. It was waiting in the queue to get your turn to simulate on their mainframe. So at about that time, we were reading one of the current electronics magazines. They had an announcement about a company that had come up with a version of SPICE that ran on a personal computer, a PC. We said that's impossible, but it was like $300. It was less than $1000, whatever it was.

We figured let's give it a try. We can use it, if for anything else, just to check syntax. And so that's what we did. We bought a copy of the software. We ran it. We would use it to check syntax. If the syntax was good-- because it would be able to fly through the syntax checking really fast, too, and then it would just take forever to simulate the little circuit that we would give it. At that point, we would submit it to the mainframe.

Well, it turned out the simulation from the mainframe after queue time came back in about two hours, and the simulation off the PC came back in about two hours. And they were the same result, and we said I don't think we need this mainframe account anymore, so we cancelled it. And so we did all the circuit simulation for the first family of FPGAs using a PC. And this was an 8088 based IBM PC. It actually had a hard disk. This was the first version of PC that had a hard disk, and that was we had four of them in the company. Three for the software guys, because that was going to be the target tool for our tool set, and we had one for the IC design department, me and Jim Hsieh, to run our circuit simulations on. And that's how we did our simulation.

Just thinking back to it, it's bizarre that we could actually develop a product with as primitive of tools we had. Well, then the problem was we had to get real models. Once the relationship with Seiko had been consummated, my job was able to go over to Seiko and to work with their IC design engineers, their process technologists, to come back with models and design rules so that we could lay the chip out. So the bosses had agreed to this deal, and by the way, the deal was a handshake deal between Bernie and his friend. There was not contract-- literally no contract.

Now I was introduced to the people who were going to have to transfer the technology, and we weren't old friends. We just met each other. Maybe we were married, but it was a shot gun wedding. It was not something that we chose, and they behaved like I would have behaved if I was in their shoes, and that is our process technology is our family jewels. Why are we sharing it with anybody else? Especially these guys. They have no product. And so it was a little bit of a challenge to get the information from them. It was compounded by the language difficulty because I don't speak Japanese, and their English was much better than my Japanese, but it wasn't good, and the person who was between doing the translating was a person who didn't understand technology. And so it was a difficult-- the bandwidth wasn't huge coming between the two of us. Fortunately at least when you start drawing pictures, that translates pretty well, so we can talk in pictures pretty well, but I had a real problem in the beginning.

I asked for the design rules. Well, they gave me their design guide. I took it back and looked at it. I said these are terrible. They don't even allow dog bones on contacts. Well, it turns out what they gave me-- they were using this particular process to do gate arrays for themselves that they used in all of their system products, and what they gave me were the rules for designing their gate array, and they didn't allow dog bone contacts. Every track had to allow for contact anywhere since it was a gate array.

So at the beginning, what they gave me-- what I was looking for, which I didn't know how to ask for was, what is the process capable of manufacturing? Not what are the design rules you use for your gate array. And it took a long time for them to understand my question. I had to make more trips back to Japan to work through the details of this stuff, and they really didn't know because they never built anything that was like this. They built their gate array. And so there were some interesting negotiations back and forth for what ultimately became the design rules that we used to lay out the first chip.

Another thing that we did-- once we had design rules. We had to start doing trial layouts to see how dense this was going to be because we really didn't know. And we made the decision at that point to do layout the old fashioned way, the way I had learned when I first started, and that was not to use a CAD system, a C-A-D, computer aided drafting system because they were really expensive. That's Bernie's motivation. For me, they were notoriously unreliable. You'd lose days and weeks of work due to a fluke. A bug would hit it, or a glitch would strike, and yikes, you lost a ton of work.

We couldn't afford that, and so we ended up getting green gridded Mylar plastic and colored pencils, in pointers[???], and electric erasers, and that's the way the first chip was laid out. It's very cellular, which was nice. And we did some trial designs. We finally figured out the magic. There was a woman named Shelly Sze, who I had worked with at a prior company, and Shelly was a very good mask designer. And she came on. She wouldn't join the company. We begged her to join the company, but she wouldn't. She was a contractor, and I think she regretted the fact that she didn't. Her son, who at the time, would actually help, was going to grammar school-- would help her with the designs. He eventually became an employee of Xilinx before she did.

So we drew the chip on green gridded Mylar. We tried some trial layouts. So we started the process-- like I said, I started in March of '84. We got design rules probably in August of '84. We finally had something to work with. We had made up models, and I think we had some attempts at their modeling. Then we kind of modified-- it turned out the guess that our guy had done-- his name was [INAUDIBLE]. They were pretty close, and so we were on target with the kind of game playing that we were doing before we had the official models.

We probably started the design in earnest in September of '84, and my goal that I had been given was to have a chip ready to tape out in June of '85. And so we were scrambling as you might-- but we still had just this small group of people. We had a mask designer, two circuit designers. I did more logic than circuit. Jim Hsieh did most of the circuit work. He did come from a memory background, so he was a much better circuit person than I was, and we actually made our goal-- we wanted first silicon in June, end of June. That was the goal.

We taped out the first pieces of-- the lower level is taped out in May, I think, of '85. The upper levels, the last layers-- in fact, I took-- so our layouts were digitized. I went over to the shop that did the digitizing. I picked up these nine track reels of tape that were the PG tapes, pattern generator tapes, to take to Seiko to build it, and I had a duffel bag that must have weighed 50 pounds filled with tapes for the last few layers that I carried. The value of the company was right there in my lap the whole way. I wouldn't part with this stuff, as well as single layer plots of all the layers to show them so they could look at what they made-- Seiko could look at the mask they made and compare them with our single layer plots to see if they resembled one another, but it was amazing that we were able to hit that goal.

Probably about the time that we started in earnest, we actually, by the way, added two more circuit guys to help get the detail work done. One of them, in fact, was a guy who had done a lab design for me, had gone on to Berkeley to get a degree in chemical engineering. He wanted get into EE, but he couldn't. Program was impacted. He couldn't get in. So he got a degree in chemical engineering, but he took as many EE classes as he could, and I hired him because he was the perfect utility infielder. He could write software. He could layout, and he could do circuit design and logic design, and so he was a great addition to the team.

And another guy who had CMOS experience, first one, who came on more primarily to design our output drivers and our input protection. We had never had any concerns in our prior technologies with latchup, and I think latchup was a big fear that the people who are uninitiated had. And so we wanted a guy who had fought with that stuff before on our team, and so we had that, and we added, I think, a second or third mask designer-- second, maybe, and a third mask designer to the team. But we worked like crazy and were able to get the product taped out about on time, and we got first silicon in June, June the 38th.

[LAUGHTER]

That's what I would tell people. I made my goal. I was not quite making my goal, but I got close, and so we had our first wafers delivered back to us from Seiko in early July of 1985.

Trimberger: And what process-- how big was the die?

Carter: So the process technology-- they didn't really call it the same way that we did. Probably equivalent to the way that a US manufacturer would've called it. It was something between the two micron and a two and a half micron technology. We called it two and a half micron. The gates were drawn two and a half micron. And so it was not the leading edge, by the way. That's another maybe story I'll tell you a little bit about Bernie.

Bernie said the only risk we could take, since the product concept was risky, was that, just the product concept. Everything else had to be vanilla. I don't want you doing anything too exotic, and so our process technology was not leading edge. It was a mature process at Seiko. Everything we did was pretty conservative. He said I also want you to use stock process. No special process changes. And it turned out to do programmable interconnect. It would be nice to have a p channel transistor with a different threshold. And this was kind of skirting Bernie's, so I had three or four different designs that we had come up with for how we could solve this problem. One of which required this unusual threshold.

It turned out at Seiko, when they implanted-- if you did not implant a transistor a p channel transistor, it was the right threshold for what we wanted. So the process technology was precisely the same. We just had one change to the p channel mask. We blocked the transistors that we called native transistors, and it had the right threshold for what we needed to do in our circuit work and saved us a bunch of transistors. So we were very vanilla.

The other thing that Bernie told me-- this is, I think, a great story, too. Early in the design process, he came to me and he said, Bill, I know engineers really like to sharpen their pencil and do it just right. And he says, I don't want you to do that. He says I want you to use the 80/20 rule. Do the 80% design, and the

last 20% that takes 80% of the time, do your best, but don't spend 80% of the time. He said two things. I want it to be good enough so that it works, and is useful to a customer, and useful enough that they'll use it, and tell you what they like, and tell you what they didn't like, but beyond that, don't gild this lily too much. Just get it out.

And I was a little offended. Not a lot offended because he was right, but that's what we did. But then years later when that first product was obsoleted after a 19 year product run-- it now had been shrunk multiple times-- I went back to Bernie, and I said, Bernie, for a rough draft, that chip had a pretty healthy life. So we would joke with each other about things like that. So you asked a question, and I went off the track. I'm sorry, Steve.

Trimberger: No, that's the right track. So I was actually asking about die size, and then you mentioned it got shrunk. So what was its die size when you started? What was the die size when you were done?

Carter: So I, at that time, thought in English units more than anything else. When I said two and a half microns, that's probably not native language to me back then. It was about 307 mils on a side, so 0.307 inches on a side equivalent, which was by far the biggest chip I had ever done. It was huge, and I had done some huge chips in my career, but this one took the cake. And the process technology we were using was probably a generation or two too early. The part wasn't really practical. With a die size of 300 mils on a side, that's really not something you want to put into production, but we had the ability to do linear shrinks, and it would be precisely the same product. it just would cost less because the die is smaller. It ran on the same power supply so we could drop in the same socket, and, oh, by the way, it went and got faster. And so everything good happened when you shrank the product. And eventually that chip-- I don't know how small it got, but I think was less than 100 mils on a side. In my career, it was literally the biggest and the smallest chip that I ever had a hand in.

Trimberger: What was Ross's role in this initial design?

Carter: So Ross's role was basically my sounding board. He didn't do the detail design, but he sure contributed to it every day. If I had issues, I would talk to Ross, and he would give me guidance, but the other thing that Ross was doing at that time, as start ups seem to do a lot, is the senior executives of a start up-- they spend a lot of their time talking to people about money, getting funded. And so Ross was gone a lot, and he was also the guy who was negotiating deals with partners.

And so interesting story there, too, is either one of either Bernie or Ross was always at work on the premises every day. They never travelled together because they wanted one of the two of them to be around to take care of the day-to-day issues that may crop up, that did crop up. So what that turned out for me, is when Bernie travelled and he wanted a technology savvy person with him, he'd take me. And so I had the good fortune to be sitting next to him on a plane for more hours than you could imagine, but I used that time wisely. I would pick his brains, and it was probably the best gift I've ever been given-- was the opportunity to spend so much time with Bernie one on one. Nobody could interrupt us, and I had him, and I could just talk with him, and pick his brains, and learn, and learn, and learn. And that was just a beautiful, beautiful thing for me. Wonderful gift.

Trimberger: So I'm going to go a little bit out of my sequence here, and because of this opportunity, I'm to take advantage of it here. We know both Ross Freeman and Bernie Vonderschmitt have passed away.

Carter: That's correct.

Trimberger: You just mentioned Bernie. What questions and answers should I propose to Bernie? Can you speak for Bernie and give us some of his wisdom, the pearls of wisdom you encountered in his vision on that?

Carter: One example of a pearl of wisdom, and I think this one is quintessential Bernie-- there was one trip that we made to Seiko, and this was not early on, but probably in the second year. And we went to Seiko with the sole purpose of telling them they need to raise their prices, and most people would say, why would you go to your supplier and tell them to raise prices? It just doesn't make any sense at all. And I asked Bernie. I said, so Bernie, why are we going to Seiko to ask them to raise prices? And he says, well, the deal that we struck with them-- and I don't remember if the monetary units were in yen or in dollars, but the currency fluctuation had changed from the time the deal was originally written to become a situation where Seiko wasn't making money with us. They were losing money on the wafers that they were shipping to us by selling them at the price that we had agreed on due to the currency fluctuation. And so our goal was to go back, and talk to them, and tell them they needed to raise their prices to compensate for that.

And Bernie's wisdom to me was, in a partner relationship, you want your partner to have a vested interest in your success. In fact, we literally tried to get Seiko to invest in Xilinx as a start up. We wanted them to be an owner. They had a corporate policy that they couldn't become an owner unless they were 100% owner. They could acquire you, but they couldn't own a little piece of you, and so they never did become an owner of the company. But he said a deal should always be one that benefits your partner 51% and you 49%. Now, it's not 52%, 48%. I don't want you leaving money on the table, but I want to make sure they want you to be successful. They will benefit, and that was, I think, powerful wisdom to me because most of the businessman I had been around before that had always said you kind of take advantage of your partner. And if you get a good deal, great. And I didn't understand the notion of long term.

That's what Bernie taught me. Think long term. It's not what the immediate impact of this decision is going to be, especially in business, but what's the long term impact? What's going to happen 10 years from now, 5 years from now? And he said the other thing is our business model is predicated on partnership. We want good partnerships. We want to be a good partner. That's going to be one of our skills, and so that was, I think, something that Bernie taught me I thought that was extraordinary valuable as a businessman.

Trimberger: Well, I want to get back to product and the development of the technology again. We talked a little bit about the first three hires were software people, and I had questions about a semiconductor company now getting into software and applications. Was that understood at the start that that was something that had to be done or would be a big investment? And just how did Xilinx approach that?

Carter: I think it was well understood, because the tools that our customers were going to use to use our product were not going to be built by anybody else because nobody cared. This was not something where they could build it and it could be used by a broad spectrum of chip companies like us. It was just going to be for us, and so we knew we had to do it ourselves.

I think one of the reasons we were successful is we set easy milestones. We didn't make it too aggressive with what we were trying to do in software. We started off very simple. The first piece of software was not automatic place and route and all the other things you would see going on in the gate array world. The first product was supposed to be a physical editor. It was everything was done by you, Joe customer. You did it at a screen, but if you wanted to find the programming inside of one of these logic blocks, you'd find it. If you wanted to say how logic blocks were hooked together, you took a thing, and you stitched it together to create the track until it went to its destination. And there was really nothing automated in the process.

The first product we came out with actually was better than that. We exceeded our goal. It was you could specify a source of a net, the destination of a net. It automatically would find a way to do it. It may not be the best way, but it would find a way, and that was beyond the spec. And I think because we set modest goals for ourselves, because they were not going to be hard to achieve, that we were able to come up with something that was useful for our customers at the time the silicon was available. But we knew that that was going to be a problem because it was a weird new kind of product. Nobody had been trained on

the use of our product in school or in another company. This was going to be brand new for everybody who was there.

And so we also had to do the same thing in applications. So we had to have people who had the capability of going out and holding the hands of customers to get them started on this. It kind of becomes the college education. This is the masters class in how do you do design with this kind of a product. So we knew we had to do that right from the start.

Interesting stories-- at the beginning, so here we are. Our tool is PC based. We didn't know if our customers had a PC-- or our potential customers-- and so how do you demonstrate it? How do you go in and show how it works? Well, you've got to go on the road with a PC-- oh, and a monitor. And this is not a flat screen monitor. This is a big CRT based thing. So we actually had to have custom boxes built for our field application engineer. We had one guy to go out into the field to show people how to do this stuff. And so it was kind of bootstrapping the whole world because the infrastructure just didn't exist for our customers to use our product we had to come up with all that.

The first product was delivered on a floppy disk, a 5.25” floppy disk. So that was the software suite. It wasn't very sophisticated. Later on when we had to start using CDs as a way to ship them, customers didn't have CD players. So we ended up giving away CD players to customers so that they could install our software. So it was one of these things where we're kind of pushing the envelope all the way along the way.

The PCs that we had-- we actually required that they have a color monitor, and that was not common at the time. Most of the stuff was monochrome. And so we were pushing the envelope all across the board with the technologies that we were employing to get this product into the marketplace.

Trimberger: That incremental approach-- did that also apply to your silicon development?

Carter: What do you mean by incremental approach?

Trimberger: You talked about with software light, easy goals, and take small steps. We talked about the development of 2064, the first--

Carter: Right, first product, 2064. And it has a sibling, the 2010, and you'll wonder why the naming changed.

Trimberger: I would assume because it's a smaller number it's a smaller device.

Carter: No, it's a bigger device.

[LAUGHTER]

What happened between the first one and the second one is we hired a marketing guy.

[LAUGHTER]

And that's not to disparage the marketing people all around the world. It's just the first one was called the 2064. It was XC2064. The XC and 20-- I don't know where it came from. The 64 implied that were 64-- an eight by eight array-- 64 of these logic blocks. So that kind of implied the capacity of the device. The next part-- in fact, the documentation in house all called it the 2100 because it was a 10 by 10 array, and it had 100 logic blocks, but when we went into the marketplace, it became the 2010 because it was supposed to be 10,000 gates.

The problem that we had is customers didn't understand the capacity of our devices by looking at logic blocks because what's a logic block. And so they didn't know how to compare it with anything because there was no scale. They hadn't developed any experience with it, and so what we tried to do was to compare it with what maybe you could get from gate array, because we figured some of our customers may be familiar with gate arrays, and so if we kind of imply the capacity based on a similarly capable gate array, this is what it would be. The gate wars began at that point in time where nobody believed how big the capacity of the devices were. They actually had to use devices to come up with their realistic idea of what the capacity were themselves for the most part, and different competitors would inflate gates, and it was a problem.

Incremental approach, you mentioned that. In fact, that was exactly it. It was what Bernie had said. It was get a product, make it useful enough, get a few customers to use it, learn what they like, learn what they don't like, and learn what the next process technology gives you. So the kind of things that came into our mind-- we would take existing products, and we would shrink them, but if we were going to develop a new architecture, we would take the knowledge we learned from the prior architecture. We would take a look at how our customers were using our products for their other capabilities we should add that were common to the applications they were using them, and, oh, by the way, does this next process technology allow us to do something that the last one didn't. And so those kinds of things were in the back of our mind when we started to develop the architecture for the next generation of product, and that was the plan from the beginning.

In fact, if you look at the original business plan information, it was keep coming up with new families of products that exploit the improvements in process technology and knowledge that you've learned, and that's what we did. And it said on there also the notion of adding dedicated functions when appropriate, and we didn't do that for a long time, but that eventually is what we did. And if you look at a modern FPGA, that's pretty much what it is if you squint your eyes.

Trimberger: You don't have to squint too far these days. So about that time, about the transition from the 2000 series to 3000, you started moving into management.

Carter: More into management. That's right.

Trimberger: So tell me about that transition. How were you prepared for that? Did you leap into it? Do you grab that opportunity, or were you forced into it? Tell me about that transition.

Carter: I had to face some of that before when I was at Zilog. When I started at Zilog, like I mentioned, the first thing I did was the one-man chip soup to nuts. Really a lot of fun. Loved that, but as time progressed, the company got bigger. I don't know if it was ego or what, but I was given the opportunity to have more responsibility in terms of management, and I took it. One thing that they did have was a good management training program there, and I think part of that was because we got Exxon at one point kind of behind us. And they realized that you have to grow managers inside a lot of the time, and the way you do that is give them the appropriate set of skills.

And so I had learned some skills. I wouldn't say I was good at it, but I had learned some. And I liked the visibility of a manager. I liked to see what was going on, but quite frankly I didn't enjoy the personnel piece at all. Doing reviews was like, ugh, the worst thing I could possibly think of doing. I would do anything else besides reviews. The people who worked for me knew that because they didn't get their reviews on time ever.

But when I came to Xilinx, all my managerial responsibilities were lifted. I was a member of the technical staff. Jim Hsieh and I were colleagues. We didn't have a boss. Ross was our boss, but like my prior employer, the organization grew. When the 3000 came along, Jim took over primary responsibility for that family. I had now to manage the group a little bit more, and so it just kind of evolved that I took on more and more managerial responsibilities.

When Ross Freeman died-- so this was in October of '89-- and Ross was the VP of engineering, and he was responsible for both software and hardware. Bernie came to me and asked if I thought I should be the VP of engineering, and I told him no, because at that point, it had become clear to me that the biggest problems that we were having were probably more on the software side than they were on the chip side. The chip side was actually-- it's terrible to say, but pretty simple. I had the easy problems. Some other guys had the hard problem.

And I thought it was more important to have somebody who understood software development much, much better than I did. I knew very little about it, and so we ended up pursuing someone else that I worked for who came from more of a software background, but eventually I did become VP of engineering. And then I became chief technology officer, which actually was a nice way to get me out of management in a lot of respects. I didn't enjoy management, and my last job as chief technology officer was to hire the new chief technology officer, and I would go to work for him. And I could go back to being a contributor, although I discovered that my skills had gotten pretty rusty in the intervening years.

So I did do a lot of management. I can manage. I don't like managing. Somebody told me it's kind of like if you can manage, but you don't like it, it's like a person who can write left handed when they're right handed. I can do it, and it's not real legible, but you'll get the idea. I'd much rather be writing with my other hand. I kind of pined for the day of getting back to doing engineering work, and that's kind of where I ended my career.

Trimberger: You mentioned Ross Freeman died, and I think this is probably a good time to talk about Ross. And so that was in October of '89. The company was only five years old.

Carter: That's right.

Trimberger: This must have been traumatic. Tell me about that experience and what it did to the company and the products.

Carter: It was a real hard time. So the Loma Prieta earthquake, which happened like five miles from where the physical facility was, happened just before that. They were within days of each other. Ross had been sick for a while. Ross was a friend, and so I saw Ross. I would go to the hospital and visit him, and he seemed in good spirits, and I thought he was getting better. But probably the Sunday before he passed away I went to the hospital, and I'd never seen him like this. He could barely talk. He could barely breathe, and I'm sure I went home. My family saw me before and after, and I think they saw a pretty dramatic change, because it had gone from something where he was sick, but this will pass. He had been sick in the past. This will pass, and he'll be fine-- to finding out that he died. For me, that was devastating. Ross had been my boss combined at Zilog and Xilinx for 12 years.

He was good personal friend. Our one on ones got to be mostly during walks. We would go at lunch together and go walking. In fact, before that, we ran. When I was at Zilog in the early days, we would do the par course at noon time, and we would have a good conversation until about the last mile when things got competitive. Breath started suddenly disappearing, but that was the best time. Again, it was kind of like Bernie on the plane. I would have Ross to myself, and so we would talk about work, some work related things, but a lot it was just life. So Ross had become a really good friend. And we didn't do a lot of personal things outside of work together. I had a family. He didn't, so my sphere was kind of a completely different universe than the one he was in, but we were good friends at work. We talked all the time, and we walked and talked a lot. So for me personally, it was just devastating to have Ross leave, because he was my bedrock. He was the sounding board for all the problems and issues I ever came across. Ross would be the first guy I talked to.

And it was a tough time for the company, because he was the technologist. He was the guy who conceived the notion of FPGA. He was the face of the company to investors. He was the face of the company to the engineering organization. In fact, one thing about both Bernie and Ross, Ross in

particular, Bernie probably just as much-- I shouldn't say in particular-- is neither of them were big egos. These were senior managers at what became a pretty successful company. In Ross's lifetime it was definitely growing. We hadn't gone public yet, but it was a going concern, and we're doing well. He had invented a new technology that had never existed before, and we were making a successful business out of it. And so he never took the credit for himself. If it was possible, to shine the spotlight on somebody else, he did.

And he had what I called-- in fact, in his eulogy, I talked about this notion of a quiet confidence. He was confident in himself. He knew how smart he was, I think, but he didn't have to-- he knew it, and that was all that was necessary. He didn't have to convince anybody else of it. For me, it was a pleasure because he wasn't a big ego. A lot of times he would give me credit when I really didn't deserve, and I'd tell him so. Bernie did the same thing.

And so the two of them, for me, created an environment to work in that was just wonderful. It wasn't the big egos that I read about in Microelectronic News or other things that were around at the time. These were just two wonderful people who cared about people. In fact, they genuinely cared about everybody who was an employee. They were family. Bernie treated the guys who came in early in the morning-- they were the only guys there doing maintenance. They were cleaning up the place. Bernie would be the first guy at work, and Bernie had a relationship with them. And they were probably contractors, not employees, but he knew them. He knew them by name.

He recognized everybody was important. Ross did the same. He went out of his way-- I could tell you stories of examples where they did personal things for employees or employees' families that was way above and beyond what a manager would do for an employee, and that made a huge difference. We all felt part of a family, and that was special.

Trimberger: Any Ross Freeman anecdotes you could share? How did he approach problems? What made him so good?

Carter: Ross, he had one ability that was uncanny in my opinion. Well maybe it's uncanny because I just couldn't do it, but he had the ability to forget decisions. And that seems funny, but he would take the data that exists at a time, analyze a situation, come to a decision. And if that decision ever confronted him again, rather than saying, oh yeah, I thought about that five years ago and the answer was this, he would reanalyze the situation in light of the current situation.

And so for me, I would do the former. I would say, oh yeah, I thought about that. That's a dumb idea for this reason, or that's a good idea because of this reason. And he would look at it with the current environment, the semiconductor process technology we were using, or the customer base, or just whatever was relevant input to the decision, he would rethink it. He knew the process, but he would insert the current parameters and come to maybe a completely different decision, one that was a good decision, but wasn't a good decision three years ago. And I always thought that was something that I wished I could do, because I wouldn't forget. He would just remember the process, but forget the answer, and I thought that was really wonderful.

The other thing Ross would do-- where was I going with that? On our walks, it turns out Ross was a dreamer, and I mean this literally. He had dreams, and he was one of those kind of guys who remembered his dreams. And so he would tell me his dreams when we would go walking. Some of the time they were relevant to what we were doing, but most of the time not. Most of the time they were just bizarre dreams.

One dream, one of my favorite dreams-- Ross was in Europe-- I think in Rome. He happened to be at the Rome airport, and so he had probably been in Europe doing some customer visits or something like that. And he had this dream about being in the Rome airport, and there he saw the Pope. And the Pope was coming up with his entourage, and this Pope had this pile of luggage, just tons and tons of white leather

luggage. And all over the luggage were decals like you sometimes see people put decals on their luggage. These were decals for Primo beer, which was a kind of that was sold in Hawaii at the time. Now where this came from, God only knows, but this was an example of the kind of dream that he might have.

Some people said a lot of his early concepts of the FPGA actually came to him in dreams. Now, I didn't dream so much. In fact, I would be stumped with problems all the time. And one thing I discovered is I came to believe that my brain had a foreground mode and a background mode. And if I couldn't solve a problem in the foreground mode, I would ship it off to the background mode and go on to something else. And the time I communicated with my background part of my brain was in the shower.

I would get up, and I've gotten a reasonable night's sleep because I put the problem away. I didn't worry about it overnight, and I'd get in the shower, and beside my shower was a tablet of paper and a pen, pencil, something to write down. And I'd be in the shower lathering up and go, yes. And I'd jump out of the shower and write it down before I'd forget it, because I would forget it immediately if I didn't do that, but that's the way I kind of attacked some problems that were sometimes onerous for me. Just kind of put them in background and let the brain chew on it, and in its good time it'll tell me the answer.

Trimberger: You wouldn't mind if people said your ideas were all wet.

Carter: I wouldn't mind if they said my ideas-- as long as they were good ideas.

Trimberger: So I'm going to switch gears out of technology and talk a little bit about business. So Xilinx went through multiple second sources, and you were responsible for some of that.

Carter: Yes, I was.

Trimberger: Could you talk about second sources back in the '80s and early '90s. What was that all about? How did that work?

Carter: So the notion of a second source was the fact that customers were reluctant to buy a product from one company if they couldn't get the equivalent product from somebody else. They didn't want to be solely dependent on one supplier. From their point of view, that made perfect sense. And so it was not uncommon for one semiconductor manufacturer to have somebody else make equivalent product. They licensed them and do a deal.

And we were in the same situation. So here we were a company that some customers were getting interested in. We had customers. People were building things. Most of the early customers were not the who's who, but the who-are-they kind of companies. These were little guys who were technology focused, probably poor businessman because they didn't ask about second sources. And they were the ones who used our products in the beginning, but as we got to a more mature level and we were talking to IBMs, and HPs, and Digital Equipments, and those kind of companies that one of the things they would say to us is it's hard for us to have our product line depend on a company that we can't spell the name of with a technology that's unusual, and there is no second source. So one of the things you need to do is to find somebody to second source your products.

And so Bernie scoured the earth, and we ended up striking a deal with Monolithic Memories, MMI. MMI was the inventor of the PALS, and so they were highly regarded in the world of programmable logic. And it was kind of a feather in our cap to get them to endorse our technology by second sourcing our product. MMI was run by a guy named Irwin Federman, and I didn't know Irwin personally, but his reputation was that he was a very good man. He and Bernie were kind of kindred spirits in that way. They were not big egos in Silicon Valley, which was kind of the exception, not the rule. And so this was a perfect second source for us. They worked out well.

And so my job in that relationship was to transfer the technology to them, and I think they were shocked by the primitive nature of our technology. They said, where's your netlist? And I said, well, we don't have a netlist. I have a schematic, but I don't have it electronically readable. I have it human readable. They asked for different things that I just didn't have, and they couldn't believe that I couldn't provide them with-- they thought I was stonewalling them. I don't think they believed me, but eventually they did begin to manufacture our products, and they were a good second source. But then a bad thing happened. MMI was acquired.

It was acquired by a company called AMD, Advanced Micro Devices. Advanced Micro Devices was a common second source, a second source for a lot of different people. One thing that I've heard-- I don't know if it's true or not, but usually they never second source for a company twice. They second source with them once, and the relationship got so bad that nobody would come back. And that's kind of how it worked out for us. It went from being the ideal second source to the world's worst second source, at least in my mind.

And one of the biggest problems we had at the time was, again, the market was minuscule. We needed somebody who was going to evangelize like we were evangelizing to help make the market. And MMI was willing to do that. They were investing some energy in trying to grow and can teach people about what this technology was all about.

AMD had a history of kind of calling where you identified a customer and then swooping in and taking the business with a better deal, and they could afford to do that because they were bigger. They weren't dependent on one product line to keep their company afloat. So we went from the situation with somebody helping us make the market to somebody destroying the market.

And this was, I think, one of Bernie's proudest moments. Bernie went to AMD. He talked to Jerry Sanders, who was the CEO at AMD and Jerry was kind of the antithesis of Bernie. He was more of the flamboyant kind of guys, and he convinced Jerry Sanders that the best way for AMD to make money in the FPGA market was not to participate in it, but to buy stock in our company and let us go out and grow the market. And that's what they did.

And I think it was a five year-- they had rights to our patents. That was a mistake that we made in our MMI deal. In the MMI deal, we should have put a clause in the contract that said, if you are ever acquired, the rights to our patent technology will return to us, and you can't pass it onto the buying entity. We made a mistake and didn't do that, and so in the deal that Bernie struck, he basically prevented them from being-- they still had rights to the technology, to the patents that Ross had invented and some others, but they couldn't do it for five years. And Bernie figured by then we could move on far out that it would be very difficult for them to enter the market, and it turned out that was exactly the case.

And so I think if you were to ask Bernie what he was most proud of in his business career, I think he would say that. That was probably the shrewdest thing he had ever done. I still have a-- on my bathroom counter, there's a little Lucite Plexiglas thing from the people who did the underwriting of this deal. It has AMD, Xilinx, and Lufkin Jenrette, I think, was the company that did the underwriting. That was kind of a miniature version of the prospectus or something that went along with this that's encased in Lucite that I got from Bernie to just remind me of what he thought. And it reminds me of Bernie every time I look at it.

Trimberger: Let's talk about competitors. Who are the competitors early, and who are the competitors later? How did you deal with them? There was something you did in engineering to deal with competition or was this straight business marketing sales?

Carter: Who are the competitors? Well, in the beginning, it was probably MMI because most customers lumped the two of us together. They couldn't distinguish, even with our changing the name of what the product was. They still couldn't distinguish what we were doing over what they were providing, and we had a lot of sales work to do to teach customers what the distinction was, but as people got to know us--

and I think also some people began to look at the profitability of our company. They began to realize maybe there's something that we have. They were still pretty small as they looked to get into the business.

Altera had taken the MMI technology, and they had moved from being fused base in the beginning to being EPROM based, so it was reprogrammable. Now, it wasn't reprogrammable in the sense that you could do it situ. You had to take the chip out of the board, put it under UV light. It had to have been manufactured in a windowed package, which was pretty expensive, but at least you could do that, and then they could sell a cost reduced version that was in plastic. But they were a competitor, and they figured out a way to make an array of PALs on some of their products, and so that was competing with us. They were trying to address this scalability issue with that architecture, the PLA based architecture, and so they were probably our primary competitor.

And then in the mid '90s-- maybe early to mid '90s-- they actually jumped into the FPGA business, too. So they began to make something that was very similar, if not identical, to not the specific architecture, but the general architectural of an FPGA that we were using. They called it something different, but looked like an FPGA to me. And so they were probably our biggest competitor at that point.

Earlier than that-- in fact, this goes back to your question about second sourcing. We got to the point with Seiko where Seiko was concerned that we were becoming too dependent upon them. This was, again, a good partner. They recognized that we were consuming a significant amount of their fab at that point in time-- their fabs because we were on a variety of process technologies-- and they encouraged us to go find someone else to manufacture wafers, too, in the instance that they couldn't provide them to us. They didn't want to leave us high and dry. And so we started with that. We actually started with a company in Japan, a company called Yamaha. You may know them for their motorcycles and pianos, but they also were in the IC business, and they became a second source for our early products.

Later on, we needed a second source for-- I think it was the 4000 family. This is our third generation of FPGA. We came to the conclusion we might need somebody who had manufacturing capability in Europe, because at the time, there was concern about onerous tariffs on bringing products that weren't manufactured in Europe into European customers, and Europe was a significant customer for us. So we were looking around for who could we find that had fabs in Europe?

And we ended up at one point in making a deal with the microelectronics division of AT&T, and so they became a second source for our 3000 family of products. And they were going to be the pioneer manufacturer for our 4000 family. They had rights to the 3000 family. They were going to second source us. They were going to manufacture for us for the 4000 family, and they would get rights to the 4000 if they grew the marketplace in the 3000.

We didn't think they lived up to their agreements, and so we refused to give them rights to the 4000 family. Our wafers were there. They finished processing, but they wouldn't give them to us, and so here we had 4000 first die first silicon available, but due to a business relationship problem, we could get access to them. And so I think before that happened, when the writing was pretty clear that we were going to get into a difficult situation, we went back to Seiko and taped out the 4000 again to Seiko.

And we ended up incurring a delay. I don't recall how long it was. It was a number of months before we got first silicon on the 4000 because we had to wait for Seiko to be able to catch up and manufacture them for us. So that was a relationship that went bad.

It taught me a lesson. I was involved in the negotiation, and my personal opinion is the quality of a business relationship is inversely proportional to the thickness of the contract. So if the contract looks like a phone book, the chances of that relationship being successful, I think, is pretty low. By this time, we had a contract with Seiko. It was a handful of pages. I think it still was fundamentally based on mutual trust

and respect more than legal words. And that taught me personally a valuable lesson. If you're going to have a partner, let's make sure it's a partner who understands partnership.

Trimberger: So in that case you're using the fabless model to advantage. You can go wherever you want.

Carter: That's true, but the problem was there weren't many choices. And it turned out, being inefficient with silicon, FPGAs are a great product to fill a fab. They use up wafers pretty fast, and it in addition to that, they are a great product to bring up a new process, a logic process. Most people used a memory product to bring up a process, but the logic process was-- it began to diverge from the logic process. That became quite different, and the FPGA had a lot of the benefits of memory in the sense that it stretched the design rules. The die were large, but there was good physical visibility. By looking at the outside behavior of FPGA, we could guess pretty closely where the defect was occurring, and you could show a process engineer a picture of the die and say go look right here and see what's going there, because that's most likely where the issue is coming from.

And Seiko-- so in the beginning, like I mentioned, they were not to fond of having us in their fab. There was a time when they lost their recipe, and so they were manufacturing a lot of bad wafers, not just for Xilinx, but for all their customers, mostly internal customers using gate arrays, and the gate arrays are not a very good product for this because they don't run them in any individual mask set in high enough volume to really recognize the problems. And we came across a significant problem. We were able to not cry wolf right away. We analyzed it pretty carefully. We went back to them only when we had a good idea of what the issue was. And we helped them save-- we found the process problem with them much sooner than they would have found it on their own.

And the guy who was the fab manager at the time did a 180 degree change. He went from being a person-- so why are they running these stupid wafers in my fab to being one of the biggest advocates of Xilinx. Eventually he was given the task of bringing up a brand new fab in another location in Japan, and he paid to bring our mask set into his fab first. That was the first product that ran in that fab. It wasn't Seiko's own, because he knew this was a great product to use to bring up the fab.

Bernie and I were the first two Caucasians to actually tour that fab before the construction was finished. Interesting story we changed into our bunny suits in a conference room because they didn't have a place to do that kind of stuff. And also Japanese bunny suits didn't come in sizes that actually fit a fat guy like me. So it was pretty comical seeing us, I'm sure. I'm glad nobody took pictures. And I forgot the question that got me into this, Steve.

Trimberger: Well, we were talking about competitors, but where you got is an interesting point. So you said earlier that you were using an old process technology, and now we're talking about FPGAs as kind of process drivers.

Carter: That's right.

Trimberger: Today FPGAs are almost the first on new processes. Is that how that transition happened? How did that transition happen?

Carter: I think the transition happened consciously probably around the 4000, maybe thereabouts. We were generating revenue. We were making profit, and we could afford to take risks. So as I mentioned earlier, Bernie was very risk averse at the beginning, and now his appetite for risk went up a little bit, and our partners began to see the value also in using us to bring up a lot to process. And it was perfect. That gave us early access to the best processes, and the best process benefited us hugely. And us running in that particular fab benefited the manufacturers as well because they got good visibility on the process. So it was something where it became desirable to run FPGAs in a fab.

I was mentioning earlier the FPGAs use silicon pretty quickly, and the semiconductor industry, as I'm sure you're well aware, is extraordinarily cyclical. There are times when there are gluts of fab capacity or where fab capacity is extraordinarily short. Our customers at the beginning, why they required a second source is they felt that, if process technology ever became short, our partners would kick us out of the fab, and they couldn't get product. They didn't understand the relationship we had with Seiko because we had those problems, but we worked together to make sure in good times and in bad times that they didn't take too much and we didn't take too much. But every fab that had a glut in capacity on earth approached Bernie asking if they could manufacture wafers for us. And Bernie was reluctant to give them a business reason for why that was not a good idea because usually that would be something that would be critical of their process.

And so he decided the way he was going to do that was to have us tape out to every partner who approached us who had a serious potential of being a partner-- to tape out one of our FPGAs for them to run to see how well they could manufacture it. And so the guinea pig of choice became the 3042, one of the midsize members of the 3000 family. And I must have taped that sucker out to 10 or 15 different fabs. It was kind of like the fab du jour. A new set of design rules would come in. I'd have to have a staff person go in there and figure out how to manipulate how ours were drawn to be what was appropriate for their masks. And we'd go through this exercise, and then we'd tape out, and then they would disappear because we probably had given them a test program at that point in time, too. And if they had difficulties making it, they didn't come back. But you name it, we taped out to it. Not the huge companies, but anybody who was moderate size, who had an excess capacity at some point in their life seemed to ask us to it them a try.

Trimberger: I like that story. So let's talk a little bit about customers. Who were the first customers, and what were their requirements? Did they ask for custom FPGAs? Did you ever build custom FPGAs that you admitted to?

Carter: Well, so it was on the road from the beginning. It was this idea that, if a customer used our product and they discovered that their product went into huge volume, is there a way that could cost reduce it? Because FPGAs were pretty expensive. And so to answer that we created a product called the HardWire, which was, in fact, in the business plan right from the very beginning because we wanted to be able to answer that question.

And Bernie, being the honest man that he was, wouldn't put it out there as a product unless there really was a product. And so we built something called a HardWire, and the concept was that we would take a bit stream from an FPGA-- the bit stream that was their version of our FPGA. So it gave us the specifics of what they wanted to implement, and we could build a device that did precisely that and only that that was mask programmed. And we had a product for the 2000 and 3000 family that did that. And we prayed as a manufacturer that we never had to make one.

Well, it turned out some customers were interested, and they would tell us that, at some point in their product development cycle, they anticipated going to a hardware, and we kind of go through the exercise of getting ready for this. And we would say, now, when you have your bitstream finalized-- you're not going to change it anymore, because we can't accept a change-- then give us the bitstream, and we'll go ahead and make you some wafers. And so we brought wafers up to the metal layers, and we're all ready to go.

And we had a real problem with customers committing. This was kind of like they would date us, but they wouldn't get married to us in that sense. They would not commit. You'd say give us-- you said you were ready with the bitstream. We're still checking a few things. Give us a few more weeks and we'll commit it. Virtually nobody ever committed to doing a HardWire device.

There may have been a few, but it was on the brochure to say, if you want to do this, we have a solution for you, but they all discovered they really wanted the flexibility that the traditional FPGA afforded them.

They liked the ability to be able to change it without revving a board if they discovered they had made a mistake. And so we probably made a few, but I don't think we made very many.

Trimberger: Besides just wanting lower cost, how did you handle other requests when customers would come in and say, I want-- whatever they would want.

Carter: A special?

Trimberger: Yeah, a special, or how do you convince them to wait two years for the next chip or what do you do?

Carter: We were reluctant to do specials because Bernie came from a background-- he also in his life had done ASICs, and he understood that, if you do a custom product, it's a real problem to manage. You never could make the right number. You either make too few or you make too many, and if you don't want to upset a customer, you always make too many. And then it's not as cost effective as it was, and so it turns out to be frequently not good business, unless you hit a real home run. And so we were reluctant to do that. There may have been some specials we did. I don't remember them. Maybe that's me just repressing them.

[LAUGHTER]

I don't know. You had asked me another question about-- in the prior question you asked me about something else, and I'm trying to think. There was a piece that I flashed on right at the end of the question, and I've forgotten it. Do you remember?

Trimberger: I was talking about FPGAs as-- we were talking about custom devices, about HardWire.

Carter: Custom devices. It struck a chord with me about something else, but gosh, I'm losing it now. I'll think of it at some point.

Trimberger: In the shower.

[LAUGHTER]

Carter: In the shower. I'll be all wet again.

Trimberger: So let's see. We talked about customers and about competitors.

Carter: You asked the question, who were the first customers?

Trimberger: Who were the first customers?

Carter: And I never answered that question, so I don't have to think about it in the shower. I thought about it now.

Trimberger: Yeah, there you go.

Carter: The first customer that I could find was a company on the East Coast called GTECH. I found an invoice for software. We actually sold the software tools at the beginning. I think it made software people feel a little bit better-- in fact, maybe a lot better that all their hard work was actually being compensated for and actually paying for the software, but GTECH was in the business of building lottery machines. I don't know what they look like now, but back then they were these bright blue machines you would see

them at 7/11's here in town. And the problem that they had is they had the problem with customs. Each state would want a little bit different product game that they wanted to play, and the states also wanted to refresh the games periodically.

And so they wanted a way to be able to do that, and they took advantage of the fact that the FPGAs-- the board wouldn't change, but they could change the behavior of the FPGA by just changing the contents of a PROM or however they program their device. And so they were populated by a bunch of techy kind of guys, not business kind of guys. And so they were the very first customers that I can recall, at least they bought software. But like I mentioned before, most of the very early customers were not the ones I would have gone to. They were the little companies-- because their qualification processes were usually pretty minor. I discovered the hard way that going to DEC [Digital Equipment Corporation] or going to HP [Hewlett Packard Company]-- they had this really long and onerous qualification process because they didn't want their products to die because our products were unreliable, and so it took years-- literally years-- to get a product qualified to be manufactured in their products.

And so-- again, not being a businessman-- I would've pursued these people at the beginning because that's where the big bucks were, the high volume stuff, but it was high volume in a long time. Bernie knew we needed to generate revenue in the short run to keep this sucker alive, so he focused on the small companies who were into technology. That's where we had our first success.

Trimberger: So let's leave Xilinx. You left Xilinx.

Carter: I did.

Trimberger: And at a relatively young age. And when you left, what did you do? What was the push? What was the pull? What are you doing now?

Carter: Push and pull. You're right. You'll always have push and pull. So the pull was probably most important, because the push was there, but the push was pretty minor. The pull was that, throughout my career, my priorities-- they did evolve, but they didn't change a lot. So I had my first personal horrible experience-- for a long time my wife and I thought we lived in Camelot. Everything seemed to be just going good. Good kids, all these things happening-- until my father died, which was, again, at a pretty early age. This was when I was still at Zilog.

And my priorities at that time had been work, family, me. Other things mixed in there, but of those three things, work always came first. Family got second. I got third, and as a result, I didn't do much of anything that was kind of on my wish list. I didn't know the notion of a bucket list at that time of my existence, but I didn't do much, and there's a little bit of a frustration. Well, when my dad died, the first two changed places. Family became first, and work became second. I probably say, in my mind, work was not second, but my behavior was work was second.

And so like when I started at Xilinx, when I was offered the job-- so I had worked for Ross for five years, and Ross knew that my greater family had a family vacation in Southern California the last week of July, first week of August, every summer. And so when I started at Xilinx, I said, Ross, my family comes first. That vacation is inviolate. Will you take me under those terms? Because I knew a start up meant long hours to unreasonable hours, and I said, no, my family's going to be coming first. And by the way, I'll work weekends when necessary, but if my kids have got soccer or the kind of stuff that just happens, they're first, and then I'll figure out a way to squeeze Xilinx in some time. But it's not going to be an every weekend thing. It's going to be on weekends when it's really necessary, which at the beginning, was most weekends, but he was willing to take on those terms.

And I think he had his fingers crossed-- my family went on vacation to that same beach every summer. I drove them down. I flew back. I drove back down. I drove them back. So I got two long weekends-- in fact, if it was a two week vacation, which happens sometimes, I got a long weekend in the middle

weekend, too, so this before cellphones. So they couldn't get me that way, but so my family would go on vacation, and I went to some of the vacation, but eventually it worked that way.

And I never did work weekends. I've heard of managers having their staff meetings on weekends so that they wouldn't be distracted from the work during the week. If that was the case, I wouldn't be there. That just was not in the cards. My family did come first, but I was still-- there was just this nagging desire to do things that I just didn't have time to do. I didn't play golf. I didn't do any of these because they were just too expensive, primarily in time. It wasn't the money, and Xilinx had matured.

Probably the push-- if there was push-- was politics had emerged. I'm not a politician. I don't play that game very well. I liked it in the beginning when there were no egos, and we were pretty successful in maintaining that, primarily by hiring great people. The people that we hired were first class, and a lot of them had shared values with the founders of the company. And I think that kind of kept the spirit of the company alive. It kept the culture that I grew to know and love there, and that culture was not gone, not by any stretch. It was still there, but it was diminishing.

And it was inevitable. We were getting to be a big company, and I think that happens when you have big companies. And it's not that I hated it. It's just it was a good time, and I wasn't going to cause any projects to slip because I wasn't on any projects.

[LAUGHTER]

And so I took the opportunity. And I kind of said to myself I wanted to do 20 years and I did 20 and half years. Well, not quite 20 and a half years. Bernie had died. Ross had died. My best friend at Xilinx-- oh, jeez-- Jim Hsieh had died early on. And so a lot of people who were really good friends from the very beginning, they weren't there anymore. I had good friends, and that's not to belittle the relationships I had with the people that were there. It was time for me to go do something a little bit different, and so I chose to move on.

And I didn't move myself up to number one. I got to be number two, and at the beginning, for a long time, in fact, I did what we're doing now for new employees. I taught the history of Xilinx class. It was kind of a mandatory class that they had to attend for the orientation of new employees. It was a way for me to get back, see from my friends that were still there, try to impart the culture as I knew it to these guys that were coming in. And I think the other thing is it just taught me how old I was getting because these people were getting to be my children's age. And so I had a good relationship with Xilinx. I think my badge still works. Shh. Don't tell anybody.

[LAUGHTER]

But it really was time for me just to do a little bit different. I thought about a start up. I realized I'm not the start up guy. Ross was the start up guy. This is a joke from the beginning. Ross was the father of the FPGA. I was the mother of the FPGA, because I'm the gestation guy. Give me an idea, and I'll go turn it into something real. It'll be long and painful, but that's my skill set. It's not coming up with a brand new idea. I'm not a great manager. I don't enjoy managing. So jeez, the last thing in the world I want to be is a CEO.

And I said the experience I had in my 20 years at Xilinx was as good a story as any screenwriter could ever write-- any screenwriter who was a geek-- and it was perfect. It was a wonderful time. And I said I don't need to try to catch lightning in a bottle. It won't be caught. I caught it once. It'll never be caught again. It's time to just go do something different, and so I started doing things like exercise. My exercise had been limited throughout most of my career to my daily walk. I had a run at the beginning until my knees gave up, and then I walked. It kind of gave me a break from the stress during the middle of the day to get out. So I did more exercise. I lost a few pounds. That was good. I went back to my roots. I grew a garden.

[LAUGHTER]

And so I hadn't had a garden since I was a kid when I thought about being a farmer for my career. And it was fun getting back into that. I dabbled in a bunch of things. One of the things I mentioned before is I really enjoy learning something new, and it's good in the fact that I give myself new challenges all the time. The bad news from my wife's perspective is that, when I learn it, I don't necessarily pursue it aggressively. I want to go learn something else new, and so my hobbies kind of come, and go like crazy, and then I'll say, ah, I know some of that. Let's go do this.

But one that's stuck-- it's fading little bit, but I'm starting to resurrect it. As I mentioned to you in the beginning that one of the things kind of interested me originally was ham radio or this technology. And it was my first taste, and i think when it came time to choose a discipline in engineering, I chose EE because of my experience with radio and the fact that I could open them up and see components that came from that discipline. And so I said, maybe that's what I should do.

And so I was reading an article in one of the trade journals about the current state of ham radio. This is after I had retired. And I said, ah, that's cool. Let me read the article. So I read the article, because most of the time now I just look at the headlines, and if they don't interest me, I just go on. And this one interested me, and I went and looked, and it said, well, number one, they gotten rid of the code requirement.

[LAUGHTER]

So I could become a ham without having to learn Morse code, although that's one of the things I want to do-- is to learn Morse code. And the technology had really changed. There was this new company down in Texas that was doing something called software defined radio. And I had heard something about software defined radio, because in my career at Xilinx, a lot of the sophisticated military radios were using digital signal processing techniques, but implemented in gates to get the speeds. This is going back to I can trade instructions for gates or gates for instructions. This is where they were taking instructions in a DSP processor, putting them back into gates because they needed the speed. FPGA was a good source or a good way to do that, and I said this is cool. I know a little bit about SDR [Software Defined Radio], just enough to be dangerous. I always wanted to know more about signal processing. I thought I'd teach myself that, and it was actually too hard. Maybe I have to take a class for that one.

But there was company doing software defined radio for hams, and I said, this is cool. And then I read further down the article, and before that, what they were doing is the radio was fundamentally-- the receive side was a big A to D, big fast A to D. And they got it down to a manageable rate, and then they shipped it off to a PC, a personal computer, literally an IBM PC, where they did all the DSP computation. And then they would send it back to transmit. They would do the computation for the filters and all that kind of stuff in the PC, and then send out the information to be converted from digital to analog, put it through a power amplifier and send it back out the antenna, but they had realized that they had a real problem with that because they wanted to do more DSP than was practical than you could do in a PC, a reasonable PC, and they had a lot of trouble. People had to have very sophisticated PCs to be able to do this.

And they came up with a new architecture, and the new architecture was based around one or two, depending on the size of the radio, Xilinx FPGAs. I said, wow, is this serendipity or what? This is karma. It's coming around, and so here it was, the technology that kind of got me into this in the first place coming around to using the technology that I had been involved with for the prior many years. And there-- I believe they're high end radio. There are two Virtex, two FPGAs. Very sophisticated radio. In fact, the front end of this is, again a big A to D, probably the best A to D they could buy at the time. Very expensive.

It basically digitizes the radio spectrum from not quite DC, but the kilohertz range, up to maybe less than 100 megahertz, but you could look at the whole radio spectrum like it's a spectrum analyzer. And you can choose I want to listen to that, or you could break it down into up to eight different receivers picking out different pieces of the spectrum to take a close look at.

And so I bought a radio. The radio I bought is based on their old technology. It's not the new one because I haven't gotten myself to the point-- the goals that I set with this old relatively inexpensive radio to invest-- I think the high end radio is now around $7,000, which is probably the best radio on earth for ham-- and supporting my former employer, but I haven't broken down to do that just yet, but that's where I'd like to go.

More recently, one of the things that I mentioned in the interview earlier is that I've written software. In my education, I did software-- early career, Fortran. Master's degree-- I did IBM 1130 assembly code. At Xilinx, I did bat files for PC to be able to run batch SPICE files overnight that the fundamental-- that cheap SPICE that I was talking about-- it couldn't do. You did them one at a time. So I wrote a little bat file so I could queue up a bunch of jobs to go out, and go home, and see the results when I got to work the next day. So I'd done a little software, but this was a chance to learn something more.

And so I don't know if you've heard of Arduinos. This is the maker movement. This is C like language that they program them in, and it's a little microcontroller. And so I'm bound and determined to teach myself C. And so that's kind of where I'm at right now. I'm getting there. I'm not very good. I have an Arduino next to my computer right now. It's the dumbest application of a microcontroller in the world. It's an LED dimmer. Instead of having no microcontroller, it's just a pot in a series with an led. I'm using the microcontroller to read the position of the pot and converting that to an analog signal to drive the LED. I can dim that with this big-- not expensive, but a little microcontroller in the way-- but I did it myself. And so I'm learning a little about a little bit about software. So I'm trying to keep my fingers in technology because I enjoy learning new things.

Trimberger: Engineers and their toys.

Carter: That's right.

Trimberger: I've seen you up at Santa Clara, also.

Carter: So I'm a graduate of Santa Clara, as I mentioned earlier. Got my bachelor's and master's degree there. I owe a huge debt to them because they took me in when I was-- I couldn't go to college. They actually paid me some money, and gave me a job interview, other things so I could actually get a degree there. And it turned out that that decision, like I said, was probably the most fortuitous in my life. I met my wife there, got married there, got my master's degree there in that order. My kids are all baptized there. My youngest daughter got married there two months ago in May. So I have a deep affinity for Santa Clara.

I got started in their school of engineering as a member of their advisory board. They started something they called the Center for Science, Technology, and Society. And this was kind of a center that was going to see how you could employ technology to benefit humankind. And as I matured-- in the beginning, I was mad at Santa Clara because they didn't teach me enough about the specific thing I was doing. They didn't tell me a word about microprocessors. Later on, they never taught me to think about management, and I said, gosh, I wish they had done that, too. Later on, I said those philosophy classes that they made me take that I put up with kicking and screaming-- why didn't I take more of those? These are where the real important questions in life are.

And so the Center for Science, Technology, and Society was a way to see technology used to do good. And so I threw my hat in the ring, and I became one of the founding members of the advisory board there.

I got a chance to meet with some great people-- Regis McKenna, who had critiqued the name of our company, was on that board. So I had known Regis by reputation, but I got to know him personally.

Doug Engelbart-- so Doug and I became friends. And so this was an opportunity for me to rub shoulders with some of the real big names, and least in my mind, in Silicon Valley. And I was exposed to some interesting things. The center has changed its name to the Miller Center for Social Entrepreneurship, which is probably a much more accurate name for what they do. I'm now a member of the trustees of the university. So that's a big change.

It's a place I love. It's a place where I give my time. That's where I volunteer most of my time right now-- is at the University. And I love it. It's done so much for me, and now I get to be part of it and try to make it grow into being-- maybe the reputation I think it should have has kind of been hidden behind Stanford and Berkeley here in the Bay Area.

The school with the most employees at Cisco is Santa Clara. Now, they're not all engineers by any stretch. They're a lot of other things, but we contribute to the valley in a lot of ways, but it's certainly not as well known as I think it should be, and by gosh, my goal is to help change that. So I'm very involved in Santa Clara.

Trimberger: You experienced-- you lived a piece of cultural mythology. I'll call it that-- the Silicon Valley mythology-- start up, entrepreneur, strike it rich, so on. I believe historically this will be a piece of American mythology that ranks with the Wild West, the California Gold Rush, Texas wildcatters, Silicon Valley technology. Talk to me about the contrast you see between the mythology that you see and the practice. What was it really like to be in that?

Carter: So I didn't get at the very leading end. Most of my peers or managers at the very beginning were graduates of Fairchild U, but then I'd see those guys are the real beginning of Silicon Valley. But Bill Price, who I mentioned, too-- he was a graduate. Published some interesting things. One day we went out for lunch to the Alpine Inn, now known as [INAUDIBLE] on Skyline Boulevard. We went for lunch, got back about 3:00 after several pitchers of beer. This was kind of a rite of passage back in the era.

In the beginning, it really was wild and woolly. It was hard driving guys. There was a lot of cross pollination. Secrets weren't kept very well if they were secrets at all. Like I mentioned, Ed Snow taught me semiconductor physics, and it was the stuff that was proprietary-- not proprietary, but it was just the leading edge stuff that was being developed at Fairchild at the time, and he was spreading it. He was spreading it new guys like me, and that was part of the ethos. It was you shared stuff. At Zilog, I had a total of one patent, and I did it at the end because I think it was because personnel was saying we need patents. And I didn't see the point in it. I had a lot of patents at Xilinx because we wanted to protect ourselves because we didn't have a fab, but other things.

And I could even debate that even to this day, but it was a wonderful time. People knew people. Don't get somebody too pissed off because he'll be your boss. It was the most fun growing up with that. It was, like I said in the beginning when I started, the way they made masks for chips was to-- Rubylith was kind of a plastic with a red material on top of that you would use X-ACTO knives to cut out the shapes and peel off the ruby. Some of the mistakes you would make is you'd peel off something that was supposed to stay because it wasn't clear. You'd see things cut, and you'd look at the ruby. And so I came in right at the end of the Rubylith era.

The first chips I was involved with-- they did have a CAD system. It was so unreliable. What it did is you would take your green gridded Mylar, put it on a table, digitize it, and every time you got a polygon, it would create a punch card, an IBM punch card. And so for every polygon was a punch card that talked about how much to the left, how much to the right, how much on diagonal, which diagonal directions. And when you closed the polygon, it would spin out a punch card.

And so your chip was described by a box of punch cards, and if you wanted to edit it-- so you would change the layout-- you actually had to literally go find the punch cards, take them out, put them into a key punch, and type in the changes for all the-- and so again, all of our stuff, all of our digitizing was being done by Signetics when I was at Signetics Memory Systems-- now Scientific Microsystems. They still did our fab for us in our masks. Well, they were slow, and I was the interconnect guy. I think I was, like I told you, the Bob Weiler of interconnect, and we were digitizing the interconnect.

And it was taking forever. It was really slow, and lots of errors were made. So I was going back to these decks of punch cards to edit them all the time, and I thought to myself, there's got to be a better way. And so what we ended up doing-- me and a technician-- we took our layouts of the interconnect piece, and we took a tablet of IBM punch card coating paper. And we sat there and measured dimensions, counted squares on pieces of grid, and wrote in the numbers on this coding pattern, because the key type operators were a lot better and a lot faster than the people who were digitizing the stuff. And so I would literally turn in pages and pages of these pages from a keypad into a keypad operator. They would type them up, give me back a box of cards, and that was much faster than using the alternative. And so that's what we did.

And so it was wild and woolly. It was so much fun. The wagon wheel of a bar that was notorious for people exchanging information I never went to, but that's because we had other bars that we went to that were the ones that were in the crowd that I was in. What was the name? There was a bar here in Mountain View. It had popcorn on the floor. People would be there for lunch from all different companies, so you'd see everybody there. I thought of this the other day, and I've forgotten the name. The most compelling thing-- popcorn was free. You threw the shells on the floor-- it was peanuts. You threw the shells on the floor. You'd get a pitcher of beer, and when you went to the bathroom, above the urinal was a sign that said return rental beer here.

[LAUGHTER]

And that was the way the valley was. It was kind of irreverent, lots of fun. You hear the stories.

Microelectronic News-- I alluded to that earlier. It kind of the Star, wherever the newspaper is that you would buy at the checkout stand at the grocery store now that has all the stories about the stars. Microelectronic News was the equivalent for Silicon Valley. It was written by a guy named Don Hoefler. At the beginning, it was printed on bright red paper so the black ink on bright red paper could not be photocopied, because the density was about the same. You put it on a photocopier, you just got this black page.

So it became the most popular reading in any company, and it would have a routing list on it. And you would just wait for your turn to get to read the newsletter. It had all the juice in the valley, and Xilinx made-- that was probably one of my defining moments-- to read my name in Don Hoefler's newsletter because we got sued. And that was the kind of stories he would print. It wouldn't be about the products. It would be about the scuttlebutt, and it was fun reading it, and so that's how you knew the juice of what was going on in the valley, and it was fun. It was just really. It wasn't a business yet. It was more fun than that. Too many engineers, I guess, in control and not enough business doing their thing.

But it evolved to becoming a business, and that was maybe part of the push to for me-- is I liked it when it was fun and just a little bit wild and woolly. We were not too much worried. I wanted to make money. I wanted to beat my competitor, but what I see-- I'm not in the know with the environment in Silicon Valley today. I read stuff, but I don't know it. But back then, people started companies not to have it acquired. Xilinx-- the goal was never ever to have it acquired. It was to do something nobody else had ever done before and drive it.

We were very competitive. You can read about that, I'm sure, in the newspaper, but we were extraordinarily competitive, but it wasn’t about making money. I don't think Ross-- I don't think it was in his

vocabulary this notion of going public. That wasn't important. It wasn't to sell it because this was his baby. This was his conception, and he wanted to see it successful, and I think Bernie was the same way.

Now, did the people who were there at the beginning make some money? Absolutely they did, but that certainly wasn't my goal, because I didn't believe it, as you may recall. I said this is going to be an education, and they're paying me for tuition instead of me paying them. I'm going to learn CMOS, and programmable logic, and maybe I can use that someplace else, but that's, I think, the difference. And I think that's really changed. I think we were naive and young and just wanted to change the world, and it wasn't about having a big company buy us and go off and do another-- I don't recall the notion of serial entrepreneur being in the lexicon back then. It was just people who had idea. They wanted to see if they could make it work.

Trimberger: Was there ever a time when it wasn't going to work? Was there a time when Xilinx was about to go out of business, or was that just sort of always up and to the right?

Carter: So I'll tell you a related story. This is a Bernie story, too. Soon after we had silicon-- so the original round of funding-- I wish I had my notes-- I think it was $5 million plus a $2 million lease line, and that's what we had to start Xilinx. When I started with the company, we had no money. We were close to a deal, but I was able to-- in fact, in the beginning, you didn't get stock options. You bought stock. So if you wanted to become a member of the company and you want to participate, get out your checkbook. You're going to be a buyer. And so that's what I did. I bought stock. I bought stock at a cheaper price than the first investors did because I got in before they did-- not a lot cheaper than them, but cheaper. And where was I going?

In fact, when we had our product and they introduced it to the marketplace, which was Halloween of 1985. That was the EE Times-- or not EE Times. EE magazine. It was the cover story. Maybe it was EDN. One of the magazines. We still had half the money in the bank, so we had been able to take a product from Ross's conception into being able to make it and sell it using $2.5 million. Bernie was frugal.

Well, soon after that, Bernie started to search for the next round of funding. And I'm being the, again, non-businessman that I was. I went to him and said, Bernie, we don't need the money. Why are we going out and asking for money? Why are we going to dilute the shares? And he says, listen to what you said. When is the best time to get money? It's when you don't need money, because that's when you're in the strongest bargaining position. If you don't need it-- now we actually have a product. I can show value that was more value than the original investors put into this company because then it was just a pipe dream. We don't need the money. This is the time to go raise funds.

And after he explained that to me, I said, hm. That makes sense. I understand that. If I'm buying a car, I don't want it to be in short supply. I'd like you to have a glut of them, a bunch of them on the lot. And so he was a brilliant man, and it was those kinds of decisions that I think most engineers at that time wouldn't have made, but Bernie was a businessman. He had done this, and he had learned through the school of hard knocks and then his MBA school to put it through.

His plan was, as I recall, he was going to-- I think Ross and Jim went out looking for money before Bernie joined the team, and they said you've got a nice idea, but you need somebody who understands business as part of the founding team. And so it was Ross's assignment to go recruit his boss, who hadn't been his boss for all that long, by the way, to join the start up. And I think Bernie's aspiration at the time was not to found a semiconductor company. His aspiration was to be a financial planner, work with people to invest their money-- invested because he liked playing that game. That was really, really fun for him, and they talked him into it. He did get his certified financial planners license or whatever it was. He took the test and did all that stuff, but he never used it.

And he was going to do the start up, get it off the ground. He was going to do five years. He was 60 years old when we started the company, which at the time, I thought was ancient. And Now that I have exceeded that, now that I'm a card carrying member of Medicare, he wasn't that old after all.

[LAUGHTER]

But he was going to do it until he got to be about my age, and then he was going to quit because he would have launched it. And he did it for the next 20 years. He quit when he was 80, and so that's the way the environment was back then. It was a lot different than what I read about. Now, what I read about in the paper is probably not completely accurate because everything I've ever read the paper that I knew specifics about what was wrong. At least maybe there was a nugget of truth, but most of it was wrong. So most of what I read about probably is wrong, but the environment was a lot different I think than it was now-- than it is now-- the way I believe it is now, anyway.

Trimberger: We've got to wrap up a little bit, but I'd like you to just sort of retrospective looking at your career and all that. Is there something you would have done differently? If you could change something, what would you change?

Carter: Oh boy. Probably I would have put family first sooner. That would've been a change. I think I put my family through a lot, being married to engineering for as long as I was, because it was so much fun. My first, I couldn't believe it. They were actually paying me to do this stuff, and it was fun, and was much more state of the art than what I was doing at school because it was a pretty primitive school that I was going to. And so my first circuit simulation was done on the job at SMS, but I'd change that maybe.

Do I regret not going to Intel? No. I don't think I would have been a good culture fit there from what I've read. The first job was good. I learned a lot. Bipolar was what I was trained in, and so to do bipolar-- in fact, I'd never done low-power Schottky, so that was something new that I got a chance to learn a lot in. I-squared L was new to everybody, and that was fun.

Zilog was a good experience. Doing a chip soup to nuts on your own was-- I can't imagine anybody ever doing that again, unless it's really a tiny chip, but that was a ball. Management-- I have a love hate relationship with management. I liked helping to make decisions, participating in the decision, but I didn't like a lot of the baggage that came along with that. If I could have been a consultant and never have to do a review, that would be fine. If I could be the one who gave staff meetings to my organization to talk about what we were doing and to kind of pump them up and get them enthused, I'd do that. That's fun, but there's some aspects of management that I really didn't enjoy, and if I didn't have to do that, I wouldn't do it again. But on the other hand, I learned a lot.

My wife probably taught me as much about management as the training classes I had at Zilog. Her language was different because she was a full time mom, but a lot of the skills that she learned were skills that, in doing her job-- and I don't mean demean anybody by saying that. They're similar. The vocabulary is different, but the concepts are very, similar and she taught me a lot.

I probably could have been a better dad, been around the home more. That's probably maybe going back to the first comment, but my kids all know me. And so I went to all their soccer matches. I saw all their graduations, and so I guess I did OK in those kinds of things. I was probably missing for more weekends than I would have liked, and travelled way too much. That's probably the thing I regretted doing, too, although I learned a lot.

The flip side of that is I was exposed to cultures that I never ever would have been. I never traveled until I got to Xilinx. First business trip ever was going to Japan to pick up design rules. And so I was going to a brand new culture that I knew nothing about, and so for me it was a huge experience because I was kind of key to the company, but it was also, wow, people live differently here. And I need to do this. In fact, that's one of the things that I've kept up-- is not business travel-- it's travel. That's one of the things that

was added to the list after my retirement-- was to travel a lot. So my wife and I travel a lot more than I travelled before, and virtually all of it for pleasure, except for the time I went to Australia for my boss, the CTO or the chief technology officer. No, I think I was working for him when I did, too, so I don't know-- so travel. Is there anything else I would have changed?

The companies I worked at were all good. I had great experiences. Zilog in the early days, the stories I could tell.

Trimberger: Hold it. I thought we'd asked those. We might have to come back.

[LAUGHTER]

Carter: And just going up with the business-- if I could've been in five years earlier, that would've been fun-- and gotten about the same. That would've been fun. But I don't have any regrets. No big regrets. I hired good people. People have asked me what am I most proud of, and they probably expect me to say the FPGA. I was the guy who helped design the first FPGA, and it's probably not. It's up there, but probably the thing I'm most proud of in my career is the people I've hired.

I've hired some brilliant people, or had a hand in hiring some brilliant people, and they taught me a lot. That's kind of important to me. I've told you I really like to learn things, and that's where I learned most of the stuff-- is from the people that I've hired. They kept me young because most of them-- many of them-- were younger than me, and they had been taught much newer stuff than I had been taught in school.

But most of them are really just good people because that was-- I rejected more people that I interviewed for offering them jobs because of culture fit than I did for technical capability. Most of them were technically capable. They were good, because they wouldn't have gotten-- the resume probably would not have gotten to me by then, but when I talk with them and try to understand what made them tick, if the things that excited me excited them, then they had a much better chance of being hired by me. And so maybe that was a truism that somebody told me is always hire people that are better than you are. It makes managing easy. So that's the part of management I liked. It was the hiring. Is there anything else I would change. I don't know.

So has is it all been Camelot? I talked about Camelot. No, I've had some difficult times-- Ross's death, Jim's death. There was another engineer, a product engineer, who died early on in his career at Xilinx. Got shot in the subway in Boston-- a guy named Tom Wauch. And so we've had tough times. When Bernie died, that was a crushing blow, and when Ross died, it was a really crushing blow. And so I've had hard times, but it's life. I can't change that.

So there's not much I regret. Professionally I've lived a pretty charmed life. Like I said, I couldn't do a start up that would be as good as what I did. I don't think it's possible, and so I have very little that I regret.

Trimberger: Do you have a short word of advice for budding young engineers?

Carter: It's corny, but follow your passion. Do what's fun. In fact, the hard part for me was discovering what was fun. I had to do a lot of things. People tried to pigeonhole me at the beginning, and I wouldn't be pigeonholed. And I found something I enjoyed. I wanted to be the best analog IC designer on earth, and I spent a total of no days doing that.

[LAUGHTER]

I did something I had never heard of called microprocessors. What the hell was that? But I think it's find out what you like and do it. And I think if you're going to be a successful engineer, you've got to have an appetite to want to learn new things because the technology changes really fast.

This is a story I'll go back to. So the people I was with at my first company-- Berkeley grad, brilliant guy, much better trained than I was at CPU design. I knew a little bit in some of the grad school classes I had taken. He had taken tons of classes. He'd built chips. He'd done a lot of things. He got obsolete about the same time I did, too, in terms of the technology that was there, because technology moves. And the technology moved on and so for any of us-- and I'm not talking about him or me. It's just everybody who was doing that work at the time-- if you wanted to progress to the next thing, you had to teach yourself something or be taught something. You had to go to school or teach yourself something. You had to have that appetite to learn or you're going to get left behind, and for me, I was fortunate that I enjoyed that. That was fun. That was part of the mystique. I enjoyed learning.

And so I'd say if you naturally have an appetite to learn, that's probably a good sign. Find out what you really like doing, but don't close your eyes to that. Go try other new things. That was another good thing about my very first job. Since it was such a small company, I did everything. like I was Bob Weiler of interconnect. That was chip part, but I did product engineering. I did test engineering. I did a lot of other things because there were three of us, four of us, and so you had to do other things. And I met people, and those people helped me get other jobs. That's how I moved on.

And so one of the things they asked me before this interview is to do a resume or a CV, and through my career I had one CV, my first one. And after that I never ever wrote a resume again, until I made a very bad mistake. I became an expert witness in a trial, and they needed a CV. And I said, oh, jeez. Do I have to? And they said, yes, you have to. And I learned two valuable things. No, maybe just one valuable thing. Never be an expert witness. It's not a fun thing. But the good news was I had a CV to give to the Computer History Museum, so I didn't have to go to scratch and actually create something. And it's not quite current, but I didn't spend any time at all making it current.

Trimberger: Well, Bill, I wanted to ask you if there-- I've asked all the questions. Is there anything that you wanted to just add here in our last few minutes?

Carter: You're a pretty comprehensive questionnaire. I think the stories-- I could tell more stories about probably every employer I was at. Good anecdotes, bad anecdotes. I never answered the question about did we ever come close to-- I told you about buying-- we never did, by the way. Not that I'm aware of. Bernie might have kept it quiet.

[LAUGHTER]

But probably the closest we came to dying was when AMD became the second source and maybe the situation with AT&T, but they weren't going to kill us. They were going to hurt us, but I don't either one of those would've killed us. No, I think we were under prudent management. One of the other things that's different today-- Bernie wouldn't have thought of going public without being profitable for at least three quarters in a row. If you're not making a profit, in his mind you had no right to ask anybody to invest in your company. And so that's different than it was before.

Do I have anything else to add? Probably not much. I guess I'm really thankful for this opportunity to reminisce and talk about some things I haven't thought about for quite a while. It was wonderful time. Like I kind of alluded to, it was a wonderful time in my life, and now it may interest somebody in the future, and I hope you can tolerate the duration of this event. It was fun. Too much fun for me to shut up.

[LAUGHTER]

Trimberger: Well, Bill, thank you very much for your time.

Carter: Thank you, Steve.

END OF INTERVIEW

