Oral History of Steve Trimberger

Interviewed by:
Doug Fairbairn
Jesse Jenkins

Recorded October 16, 2017
Mountain View, CA

CHM Reference number: X8370.2018

Fairbairn: So, welcome. I'm Doug Fairbairn, and I'm here with Jesse Jenkins. And we're here at the Computer History Museum in Mountain View, California. And we're here to talk with Stephen Trimberger, recently retired from Xilinx, about his illustrious and lengthy career in electronics, software, and related activities. So, Steve, welcome. Glad to have you with us.

Trimberger: Thank you.

Fairbairn: So, you've been through this before. And one of the things we like to do is to get a little background on the people that we're talking to. What kind of environment, family life, and otherwise did you grow up in? And especially attention to things that might have steered you towards the career that you eventually chose. So, tell us-- let's go back to the beginning. When were you born? Where did you grow up? And tell us about your early family life.

Trimberger: I was born in 1955 in New Brunswick, New Jersey, or at least that's what they tell me. I was pretty young at the time. My family moved when I was still an infant, moved to Cincinnati, Ohio. And so, I lived there for a few years and then moved to Sacramento, California. That was '61 or '62.

Fairbairn: So, you were six or seven at the time?

Trimberger: Yeah.

Fairbairn: Was your father moving for job things?

Trimberger: Yeah, my dad was an engineer. And so, he had a series of engineering jobs. He was trained to be a chemical engineer and wound up moving to California to work in aerospace. And so, that was big, brought the family out. And that was real good for a while and not so good for a while after that [because of the aerospace bust]. But that was great. I lived in Sacramento in a classical family unit.

Fairbairn: Did you go to public high schools and public schools or--

Trimberger: I went to Catholic schools. I was in Catholic schools for grade school and then Jesuit High School in Carmichael, California. And then, from there, to Southern California to Caltech for college.

Fairbairn: So, going-- you went on to Caltech, but sort of how did you wind up there? What were you interested in? What things did you find interesting in high school? What kind of trouble did you get into?

Trimberger: I going to give you kind of the medium long version of this because I think it's interesting. Maybe other people reading the transcript will, too.

I was in high school in the early '70s. And there's this California in turmoil, all this social stuff. And being at a Catholic school, too, we had a lot of people coming in to tell us about all the ills of the world and that you should write a letter to your congressman about it, or you should join the march down at the Capitol building or come with us on the sit in. And I cared. I cared about those problems. And then somebody came in. We had these people coming through the school. I don't know who it was, but somebody came through and said, "Yes, there are all these problems in the world. And we are putting in clean water and sewer systems in these places in Africa." And this just stunned me. Here's somebody saying he's going to do something about it rather than trying to control somebody else into doing something about it or paying money. And that was--. That was when … I didn't realize it, but that's when I thought I had to be an engineer. Now, my dad was an engineer. So, that had something to do with it for sure. But I remember the difference between talking about it and doing something about it. I mean, kind of silly of me: I really didn't have the term engineer. I didn't really associate that. I didn't realize that was what I was going to do.

When I applied to colleges, I applied to mathematics. I think I was good in math and science. Liked that stuff pretty much. Yeah, math is pretty good. And so, I applied to a few places, and I actually did not get into Caltech, I was waiting listed. And I was pretty upset about it because actually. I didn't know much about the campus either. And then my high school counselor said, "Oh, here's this catalog. Look at this place. You might think it's interesting." I looked it up, and-- he actually gave me two of them. And one was Stanford's. Yeah, okay, Stanford. I know about that place, great place, heard all about it. Lousy football team, but otherwise they're good. <laughs> But that was seventies. But then there was this Caltech place that I'd never heard of. And I started looking through the catalog. And I said, "Jet Propulsion Laboratory. Mt. Palomar Observatory. I've heard of these places. I want to be there." And So, that became my number one, my number one school.

Fairbairn: So, did you apply to Stanford as well?

Trimberger: Yeah, and I got into Stanford; I didn't get into Caltech. So, I finally said, "Okay, yeah." I wound up paying all the deposits and stuff like that and ready to go to Stanford. And then it was the second week of August -- I got a letter from Caltech that said we have an opening, are you, basically are you interested. And okay so, sorry, I had to tell Stanford no. My life would have been completely different or at least a slight deviation to come back to Silicon Valley. So then I went to Caltech for that. And the interesting thing is, over the years, I wound up working both of those places [Jet Propulsion Laboratory and Palomar Observatory]. It was pretty exciting. So, anyway that's how I got down to Southern California.

Fairbairn: Yeah so, tell me about arrival at Caltech and what-- how do you decide what to study there, and sort of what were your major influencers in your college career?

Trimberger: Being waitlisted meant nothing was prepared for me when I got there. Something important like housing. And so, actually, I slept in the open-air Loggia ballroom of the Athenaeum the first couple nights. And it got-- it's pretty lonely. It's before Internet and stuff like that. And so, parents dropped me off, said goodbye. I didn't know anybody there. Anyway, I was there. But what happened was it kind of forced me to go out and find people and find friends and things. So, the first thing I did was I went down and see if the soccer team was practicing and found them and joined the soccer team and made my best friend that first week.

So, how did I wind up in engineering from mathematics? I decided to try everything. I said yes to everything. And I tried everything. And Caltech is pretty structured that first year, a lot of required classes and things. So, I took those. But then something would come up. Yeah, gee, that looked like an interesting thing. That's introduction to all branches of engineering. Hmm, it looks interesting. I'll take that. I'll do that. I'll take that class. I was visiting with this good friend I made, Ken Severin one day. And his roommate came in. Not his roommate, it was a guy from down the hall. I said, "John, where have you been?" And he said, "Oh, I went to Palomar." "What do you mean you just went to Palomar?" "Oh yeah, I'm on this project. And we're looking for near-Earth asteroids. And so, they just needed somebody that could help guide the telescope." And I said, "Well, that sure sounds fun but I never-- I've never done that." "No, they're looking for people, come on down." So, Ken and I both did that! And so, it was like that. I said, "Oh yeah." Here's somebody that came by. "Gee, you want to try this?" "Yeah, I'll try that." To give you a heads up on some of the theme, I told my kids too, say yes and then figure out how you're going to do it. If you say no, you're done. But if you say yes, you've got options-- you could keep going and maybe you do alright. So, that was great.

So, it was a couple months later, the next dark run, I was at Palomar and learning how to use the telescope. And we're looking for near-Earth asteroids. I changed my major out of-- well, I knew I wasn't going to be a math major probably within about twenty minutes. But I didn't change it right away and didn't know what I wanted to do. But at least mentally, I had changed my major from math and oh, maybe this astronomy thing was going to work out. Well, it turns out looking for things inside the solar system is geology at Caltech, so geology, geochemistry, geophysics. And then, after a while, I sort of ran into the computer stuff, not really accidentally. Honestly, my brother was in computer science-- my older brother, Mike, was a computer science major at Sac State. And I really liked that stuff. I thought it was kind of fun. So, I thought that might be it. But I really wanted to try all this other stuff. And when I wound up doing stuff on the computer it was ... All the other stuff was interesting. But the computer stuff was like breathing. It was just so natural. I had to do that.

Fairbairn: So, you-- what did you get a Bachelor's in?

Trimberger: So, at Caltech, the bachelors-- all engineering-- something interesting about the school, all engineering is Engineering and Applied Science. That is the name of the degree or at least was then. So, there was no separate computer science. Even if you were EE (Electrical Engineering), “Engineering and Applied Science” was the official major, what it says on your diploma. But then all these electives in your major would all be in EE. So, mine were some EE and some computer science. By the time I graduated, those last couple years, is when Carver Mead and Ivan Sutherland were starting the Computer Science department with a focus on VLSI. And so, I was just there. They were just starting, and I walked in and said I’m going to do that too and just started doing it.

Fairbairn: So, you went on and got a PhD, but did you think about leaving after a Bachelor’s? What was the transition to graduate school?

Trimberger: Yeah. So, yes, I thought about leaving. And I interviewed in Southern California. I had a couple job interviews. And the jobs that I interviewed for and the companies didn’t seem all that exciting. And it was real work and looked like fun. But it just didn’t seem all that exciting to me. So, what I’d done at that time was to give myself options. I also applied to graduate school a few places. I really liked the VLSI stuff going on at Caltech. I went in and talked to [Prof.]Carver [Mead]. And I talked to Ivan [Sutherland]. And I talked to Chuck Seitz. And I talked to everybody and basically they said, “You know, we like you, but we think it’s right to go someplace else for graduate school. Otherwise, you get too inbred.” Now, I noticed there were a few exceptions to that rule, but I was not high enough on some pecking order to be that exception. I did not like it. I actually hated it. I thought that was being very discriminatory, and I was being treated unfairly and so on. So, anyway, I wound up at UC Irvine in the computer architectures group doing data flow and working for Kim Gostelow. And all those things that they had said about “go someplace else and learn all the things that they know there that we don’t know here”: absolutely true! I mean true to a degree that I wasn’t even sure they understood. What was happening at Irvine? Well, there was data flow architecture, there was real computer architecture and the data flow group. And there was an artificial intelligence group. And there were data structures. Just -- well it’s a data structures group. It was all this stuff that I would not have picked up anywhere else.

Jenkins: So, they didn’t have that at Caltech?

Trimberger: No. You have to understand also Caltech then, as now, in their graduate program, they focus. So, it’s not like Carnegie Mellon where you have three hundred professors or something like that and whatever the size of it is. At Caltech, there were three or four professors. And the intent is to not become knowledgeable about the whole span of knowledge but be really good in some area. And so, computer science, their computer science at that point was-- it was the VLSI program. There was no operating systems, and compilers, and anything else, just VLSI. So, it actually played into my hands in a couple of ways.

I'm going to roll back a little bit because I really wanted to tell this story. When I was in grade school, the city of Sacramento library system had the Bookmobile. And the Bookmobile would stop at the bus stop one day a week so, kids getting off the bus could get books. And it was the far stop. So, I could get off at the near one and a shorter walk home, or I could get off at the far one. So, on Wednesday, I'd get off at the far stop and I would go to the Bookmobile and get a book to read. And I started out reading Sherlock Holmes, puzzle solving, maybe you could trace this whole engineering thing to that sort of puzzle solving. I really enjoyed that, really liked it a lot. And then one day, I was in there and I saw this book. And I thought this was pretty interesting. I picked it up. I don't remember what the book was. But I got it. Okay, I got a couple of them. And then when I got home, my mom noticed and said, "Did you finish all of Sherlock Holmes?" And I said, "Well actually." Frankly, I had. But I said, "This looked interesting." And she said, "That looks like nonfiction." And I said, "Never heard of it" and I had never heard the term before. I said, "Well, what's nonfiction?" Well, she said it means that they're true stories. And I said, "You mean all the other stuff I was reading was lies or at least false?" And so, then after that, I decided I'm not reading any of this fiction stuff. I'm going to read all real stuff, so I know things. And so, I wiped out the entire nonfiction section of the Bookmobile in short order. So, then by the time I was at high school, I was going to real libraries now and reading nonfiction of all sorts of subjects. I liked space. Of course, I was a space program junkie back then, too. So, I read a lot of that.

So, but anyway, to bring the story back up to the present, so then even when I was an undergrad I read Kleinrock's operating systems book. It wasn't anything that Caltech had. I found it in the bookstore, bought it, and read it. I read Nilsson's AI book. I read Aho, Hopcroft, and Ullman. I read the collected algorithms of the ACM. This is all cool stuff. And so, actually, by the time I got to Irvine, I kind of knew at least the language. And so, that was a good way to very…-- It really accelerated that part of the plan. So, I finished a master's degree in a year over there. And after that, then Caltech was ramping up the computer aided design part of the VLSI program. And so, I applied to get back to Caltech and was successful. It was well-timed. And John Gray was starting that. And he was… That was a good time to get in on a program, very early.

Fairbairn: So, what year did you come back to-- well, what year do you get your BS, or your bachelor's?

Trimberger: Finished in '77.

Fairbairn: And you came back to Caltech in '78?

Trimberger: '78.

Fairbairn: Yeah, okay.

Trimberger: So, and by the vagaries of university bureaucracy, my master's degree from Irvine was dated '79. So, for a while-- for a short period of time, I was actually at both places. But I don't think either one of them minded too much, I guess.

Fairbairn: Okay so, we got into the VLSI program. And at some point, you found your way back up to Xerox Palo Alto Research Center. How did that happen?

Trimberger: Oh god, yes, so, this was when this Mead and Conway book was being developed. So, of course, Carver Mead at Caltech, Lynn Conway at PARC, were collaborating on the book and pulling these two organizations together. And then, of course, Ivan Sutherland was at Caltech, and his brother Bert was up at PARC in SSL (System Science Lab], I guess, yeah. And so, there were these strong links. And then we started getting some really interesting speakers down to Caltech like Bob Sproull and Alan Kay. So, this seminar series was just spectacular. So then anyway, that was going on. This —actually got started when I was still an undergraduate. And it's another one of those things. I was talking to my good friend, Jim Rowson, and one of these meetings this fall, start of my senior year. And he said, "Oh yeah, I've been up to this I spent the summer at this place, Xerox PARC." I said, "Nah, I hadn't heard of that place." But he started to describe what was going on there. And I couldn't put it together in my imagination. He said, "Oh yeah, they've got a computer there with a graphic display. And you can type, and you can see what you're doing, what you're typing," which okay, you've got to remember, this is '77. And my imagination, I said, "Well, they have one computer up there that did that." That was what my imagination told me. So, that's another one of these episodes, sort of like getting to Palomar, that I said, "Gee, that sounds great. How do I get a piece of this action?" He said, "Well, you know, there's this guy Doug Fairbairn. And-- you know Lynn Conway?" I said, "Oh, yeah." "Oh, yeah, so what? , So you should just apply." I said, "Well, gee, will you take my resume up there," or whatever. I expect that's how it happened. And then--

Fairbairn: I don't remember how it happened either, but you just showed up at my house one day.

Trimberger: This is also a possibility. This is life as the second oldest son. My brother is two years older than I am. And so, we were sort of natural playmates. And so, what you learn, as the younger one, is you just do it. And yeah, your brother is out there with all his friends, and they're all two years older than you are. But you just do it. You just join. And you just work as hard as you can. And things work out. And so, yeah-- oh, yeah, these guys are going to Palomar? Yeah, I'm in. You guys are going up to PARC? I'm in.

Fairbairn: So, you came to PARC (Palo Alto Research Center) in '77 no?

Trimberger: '78. [Mis-memory It really was '77]

Fairbairn: '78.

Trimberger: And that was a HUGE eye opener. <pause to collect thoughts> You've got to remember-- again, for all you guys reading the transcript or out in video land, you have no idea. You have no idea! Nobody else in the world had the graphic displays, and WYSIWYG editors, and mice, and laser printers, and local area network like this. We had email stuff. Why? Because at Caltech, we had one computer. And everybody had an account on that computer. And that would work. But they had nothing like PARC.

Fairbairn: Right.

Trimberger: But-- and oh, yeah, we had a visual editor. VI wasn't quite there yet. But no, so, I get up to PARC and see what's going on there and just-- it was just jaw dropping. It was … <pause> That summer, I walked twenty years into the future. That's what it was. And it was imagine. Here we are in 2017. What if it was 1997, and you walked twenty years into the future and found everybody with a cellphone? Everybody, right? Not just a cellphone, smartphone. Oh yeah, wherever I am, I just you know-- maps, who cares? And that was like PARC. You get there, and networks, and mail, and fileservers, and printers with multiple fonts. It was just-- there was even a color printer. And so, to be there to see the just-- just stunning-- and pers-- actually, I should roll back. I just missed the most important one, personal computer. I was the only one on this computer, amazing, just way over the top. And this-- it's amazing how fast you get used to that stuff. So, actually, I did.

Fairbairn: So, what did you work on that summer?

Trimberger: We started working on a VLSI editor. So, ICARUS that Jim [Rowson] had worked on, the IC design editor, Jim and Doug [Fairbairn] -- so then, we sort of well, let's do the next thing. And let's do a design system that-- not just the drawing and graphics part, let's do an editor that can use-- that can do some of the VLSI concepts that we were working on at Caltech. So, that was let's do a-- someone might say it's redo ICARUS, but do it big. And ICARUS was sort of repurposed SIL [a schematic editor on the Alto], right? So, repurposed drawing language. And this was let's do something real in the design area. So, we worked on that for a summer. And we suffered from the classic Brooks' Mythical Man Month Second System effect, where we took a much bigger piece than we could manage. We actually did some really good work in there -- it wasn't as clear until much later when we actually started to appreciate how we were managing data-- we were actually doing multi-process stuff and inter-process communication and things like that, which we sort of had to do because we ran out of memory on the machine. But that necessity is the mother of invention.

And then back to Caltech, going forward twenty years in time isn't so bad. Coming back twenty years in time is excruciating. <laughter> I mean I've got to share this computer with everybody else. And I print something, it looks ugly. I got to … let's see. Oh, yeah. I've got to remember "I" for insert and type a few things and then print to see what it looked like. The text editor was-- It was just hard to live that way. And that burnt that memory in like nothing else. I would have to come back and try to build some of this technology into the stuff we were doing at Caltech. That was kind of the mission, everybody's secret mission besides doing some real thing you might want to be doing for a PhD. Look, we've got to somehow get WYSIWYG editors. We've got to somehow get these other print capabilities. Somehow, we've got to get this stuff.

Fairbairn: So, what did you finally get your PhD in?

Trimberger: My dissertation was "Automated Performance Optimization of Custom Integrated Circuits". At that time, when these things were rolling out, EDA was sort of being invented. And there were people doing things like timing analysis. Randy Bryant was at Caltech. And he was doing his timing simulator-- transistor level simulator with timing. And I decided I didn't want to know how fast my circuit was; I wanted the fastest circuit I could make. Why turn it back to me, and then say get your hands on it and change things around. So, I wrote code that did transistor sizing in a symbolic editor-- stick-like editor. And so, that would size the transistors, flesh them out, re-compact them, and give you back a cell that was faster than what you'd given it.

Jenkins: Did it do it iteratively to where it would do a trial, make some estimate of what that one would do, and then reshuffle and go back or--

Trimberger: No, it didn't. And actually, I did this a couple ways. Part of the reason for not doing that was that would be really time consuming. And I didn't want to do that. And the other part was it would be really hard. And I didn't want to do that part either. So, what I did was an analysis of the chain. The first thing I did was just analyze the chain and say okay, I can see from the primary inputs to the primary outputs, I can see what had to be-- and then optimize along that path. I did that. And then-- and yeah, that's all doable. But that was really hard. It took a long time. It didn't work all the time because you actually had to crawl around the network and find all these paths and then optimize. And optimizing some branches would de-optimize some others. And so, that required a lot more thought, a lot more constraints that I didn't want to do. And so, what I wound up doing was the simpler model of start at the back of the chain, what is the load I have to drive, size for that, and then work your way forward and build the-- basically, exponential horn on that. And then all the intermediate nodes all had their loading too. So, that would get pulled in and taken care of at the same time. So, it was much simpler, ran really fast, and really gave great results. And then, like all the symbolic layout stuff, did not get adopted by anybody. That was another one of those big lessons. Gee, you could do great stuff but if nobody is going to adopt the methodology, you're not going to make a difference. It gets back at "Gee, do I want to make a difference, or do I just want to do fun stuff". I kind of wanted to make a difference. I don't want to do stuff.

Fairbairn: So, you completed your PhD, and what was the next step?

Trimberger: So, you know, I just joined the Caltech pipeline to VLSI Technology. <laughter> It wasn't automatic. I had interviewed with a few other places. I interviewed at Daisy. They were building-- actually building hardware to do EDA because back then, again, not everybody had these personal computer thingies. And workstations were eh, kind of workstations, but they're pretty expensive. And so, Daisy was building the hardware, and the software, and selling the package. That seemed kind of interesting. These guys were a startup, but man, they were working hard! They were working really hard. And when I saw what they were doing, I said-- my thought was I know why they're working too hard. They're solving two problems when they only really need to solve one. All you needed to do was do the software and buy the computer from somebody. And they were building computers. And they were really proud of that. And yeah, there's a lesson we'll get to later. But anyway, so--

Jenkins: Didn't Cadence do the same thing?

Trimberger: Cadence was using existing hardware, using--

Fairbairn: Cadence never used-- never built their own hardware.

Trimberger: Mentor was--

Fairbairn: Mentor did. Well, Mentor bought Apollo.

Trimberger: Mentor bought Apollo. But before that, it was Mentor Graphics Corporation. They were doing graphics displays. And they bought Apollo for their--

Fairbairn: No, they started with Apollo from the beginning. They labeled them Mentor stations. But they didn't build any hardware.

Trimberger: So, anyway they-- we got off that one-- I don't know what we were talking about. Oh--

Fairbairn: So, you were looking at Daisy and what other companies did you consider?

Trimberger: Oh, at that time-- trying to remember all the other-- there were a lot of startups.

Fairbairn: Daisy, Mentor, Valid, did you--

Trimberger: Yeah, well, and Cadence didn't exist yet.

Fairbairn: No.

Trimberger: So I talked to those guys. I had observed this about these guys working really hard, and they're working on stuff that I wasn't sure about. And then-- and I talked to this guy, Doug Fairbairn, at VLSI when I was interviewing there. And I picked up a lot of really good things in job interviews, both sides of this by the way. But anyway, talking to Doug, and Doug said, "Pay attention to what you're going to do every day." And I thought about that real hard. What is my normal everyday going to be? And so, that's how I wound up at VLSI. I had great respect for the people there. They seemed to be doing a lot of the right things. They were smart people. They were enjoying what they were doing. And so, that was that decision maker as much as-- well, I told Doug he had to offer me some more money, but other than that… <laughter>

Fairbairn: So, what year did you join VLSI?

Trimberger: '82

Fairbairn: '82

Trimberger: It's funny, I had not quite finished my PhD yet. This was late '82. And I actually hadn't defended it yet. And I-- but it was really close. So, I defended my thesis on January 6th. I got the last signature and started at VLSI on February 6th of '83. But I had actually already signed the employment agreement in December of '82 with VLSI because there were some rumors that it was a good time to go public. And by signing earlier, I'd get a sweeter stock deal. I knew what I was going to do, so--

Fairbairn: What good timing.

Trimberger: It was very good timing and spoiled me, man. <laughter>

Fairbairn: The next month you go public.

Trimberger: Yeah, yeah, got there-- so, yeah-- defended the thesis January 6th, finished, got the last signature on January 25th. I was at VLSI on the 6th of February. And I think the company went public before the end of the month, right?

Fairbairn: Yeah.

Trimberger: Yeah, it was good.

Fairbairn: Isn’t that what everybody does?

Trimberger: Yeah, I kind of like this Silicon Valley thing. <laughs>

Fairbairn: So, take us through the VLSI stuff fairly quickly because I really want to get to Xilinx and all of your hundreds of patents there.

Trimberger: VLSI, my role was: “let’s do some EDA.” I mean it’s hard to kind of-- it’s easy to be flippant about it. And so, I will be. Back then, nobody really knew what the right thing to do was. So, no matter what you did, it was great. It was so much better than having nothing at all. And so, that spoiled us for doing tools. We started out doing some tools, actually building around this symbolic layout model that I had done with-- that leveraged part of my dissertation. So, things-- the composition editor and so on. So, we did a bunch of that. So, but what was happening was this whole industry is going toward gate array, cell based, gate level. And there are a lot of good reasons for that. You build up from transistors to gates, and there’s Boolean algebra you can apply to gates. You can’t do that at the transistor level. That made it really hard. So, the library thing was a really good thing to do. So, but anyway--, it was kind of nice because I started out doing a lot of transistor level stuff, tools, and simulators, and things like that. And then transition to gate level stuff, and another one of those situations where I was personally not involved in those. But I was, again, oh yeah, that looks interesting. Let me dig through that. Gee, can I help you with that? Can I-- how does that really work?

So, I learned a lot about a lot of these things and wound up writing a textbook about capturing a lot of these concepts. It was actually too early to have written a textbook. It was fine for when it was. And it really was more focused on the transistor level stuff. Probably, if I’d waited two more years, you would have had to throw that all away and start the textbook looking at gate level stuff. But that was good because I learned about all the different simulators and all of the verification design rules checks. I’d done some of that at Caltech and did some more at VLSI. I’d learned about all that stuff. So, it was very-- it’s actually very interesting also being a EDA guy at a semiconductor company. Semiconductor companies, I don’t know, maybe it’s an over generalization to say they don’t value software. But VLSI was a bit more open-minded about that. But still-- and it was recognized that software was the wedge that got you into a lot of these accounts back before there was an EDA industry. And so, that was pretty important. So, that was one of those lessons I took from VLSI. It’s also nice being at a company-- okay, it was a startup. I got there pretty early. I got to see how these companies were organized and grew. And I knew a lot of the principals. And then--one of the lessons-- actually it started before that, but something-- CEOs are real people. And they’re not physically larger than everybody else. And they’re not ten times smarter than everybody else. They’re people. And you get them talking about their kids, and they love you. So, that’s another thing I learned about-- okay yeah, these people-- these executives are very approachable. And I-- it made you more comfortable throughout my career in dealing with a lot of people with lofty titles at least and people that I had heard about, just reputations, and then finally get to meet this person. And it was pretty exciting. So, that was one of those opportunities at VLSI to meet all those people.

Fairbairn: So, you're there for a few years. And then in 1988, is that right, you--

Trimberger: '88 was when I left.

Fairbairn: Decided to move on.

Trimberger: Doug, you might want to plug your ears. Here's what happened. So, when I was looking for a job originally in '82, I'd asked a friend of mine, Teresa Bronk, who was a CPA, I said "how do you evaluate a company." And she told me a little bit about cashflow and things like that. And that was okay. So, I could look at some of these companies and think about cash flow, and revenue, and all that. So, I knew a little bit about that. I knew I didn't know a lot. And there was this guy at VLSI who'd been at Intel, and he'd been around the Valley a lot. He was very much more senior, Scott Nance. I said this guy probably knows if we're off the rails or whatever. And so, he's my canary. So, we were going, and then one day, Scott announced he was leaving. And that rang the bell for me. I've got to-- found time to be kind of serious. I was a little uncertain. VLSI was up. VLSI was down. Up quarter in a down year, down quarter in a up year, I didn't know what was going right or what was going wrong. But if my canary's gone, then it's time to start looking more seriously. And so, I did. And the fact that Scott went to Altera, I didn't think that I would follow him to that place. But that's when I started looking and said okay, well, what else is there. And by that time, there was an EDA industry.

So, I looked at Synopsys, and Cadence. Mustafa Veziroglu was over at Xilinx. He said, "You ought to come talk to us." And I've got to tell you, I had heard about Xilinx. Oh yeah, there's this company. And they're making this gate array. And you program it by writing into a memory on it. And I'm thinking this has got to be huge. And it's got to be slow. And I knew-- one of the things that you knew from the ASIC business was the way you succeed in the ASIC business was you have to have the fastest device. You have to have the biggest capacity device. And you have to have the cheapest device. And what was that FPGA? It was slower than anything. It was-- you couldn't get very many gates on a chip. And it was ridiculously expensive. But you know what? I wanted to play with one of these. This might be my only chance to do it. I better get over there quick because these guys aren't going to be around long. <laughter> So, that's what it was. Partly, it was, "I'm going to get over here and play with this thing." The company will probably go out of business, and then I can do something else. But in the meantime, I get to play with the toys. <laughter>

Jenkins: And had you had children at that time?

Trimberger: No. So, that was the other thing. I'd just gotten married. So, that was the year, '88, I got married to Laura, new job, and found out I was going to be a dad. So, yeah big changes in that year! So, it stopped being the carefree life of the young professional and time to get a little bit more serious. But yeah, I took the job before I knew I was going to be a dad.

(Note to reader: at this point, Doug Fairbairn departs the interview)

Jenkins: Okay, this is Jess Jenkins. And I'll be covering the Xilinx years with Steve Trimberger. And as we left off, Doug had just transitioned Steve from VLSI, following his canary in the coal mine, Scott Nance, into the programmable logic world. Steve landed at Xilinx. And, as I understand it, you initially reported directly to Bill Carter? Or was that--

Trimberger: No, I-- oh, god-- so, I initially reported to Cathy Priebe. Cathy Priebe was head of software at Xilinx. And the easiest way to get fired from Xilinx was to be responsible for software. I'm not sure exactly how long she lasted after I was there. But, so, it gets back to this: semiconductor companies didn't appreciate software.

Jenkins: When you're saying software, you mean the CAD software used to design the ICs or the software used by the end user to design the FPGA?

Trimberger: This is the software used by the end user to put their design into the FPGA.

Jenkins: Okay.

Trimberger: This was what most customers saw as Xilinx. What do they do with Xilinx? Oh, well, I run this tool. And I do something. And then a bit of magic happens. And then I can put-- I can load this program into the control bits in the FPGA. And now, that FPGA device now functions as if it had been programmed with those circuits. So--

Jenkins: I'd like to throw in a Bill Carter quote.

Trimberger: Throw in a Bill Carter quote.

Jenkins: Bill Carter commented that "The software is the 'lens' through which the end user sees the FPGA". I thought it was simple.

Trimberger: Software is the lens through which the end user sees the FPGA. That's much more poetic than I did. That's Bill Carter. So, yeah, so that's-- I got to Xilinx, and my first job was "make the Place and Route tool run faster". So, this was-- <pause> I'm pausing here partly because it was so painful and partly because-- I mean did we even know what we were doing either tactically or strategically? So, I'm going to tell a bit of a story that maybe you were planning to get to later but an FPGA and PLD story. Let me tell the FPGA and PLD story.

Jenkins: Sure.

Trimberger: This is partly observation after the fact and partly as it was going on. FPGA was a new thing. But programmable logic was not. There were companies making PLDs in the 1980s. The most famous one, even to this day, is Altera. But there were a dozen or more companies making PLDs. And so, what was going on? Well, these dozen or more companies, basically, they're guys that had fabs, and they wanted to fill the fab. And they got some programmable logic device. And oh, yeah, they're making ROMs. And these PLDs weren't that much different from ROMs anyway. And so, they'd make these PLDs. They didn't do any software. The way they got away with that is there were these companies, Minc and Data IO, and they did the software. You typed in logic equations. And it was a real simple mapping from the logic equations to two-level logic in the PLD. And then it would spit out a file, the JEDEC standard file for that. And so, if you wanted to make a 22V10 for example, sure, you would-- just as long as it would be able to read the JEDEC file, you would just publish the specs about how fast it ran and how much it cost. Done. And so, —you could-- any semiconductor vendor can be in this business. They had zero software. And the software's actually the same for everybody. So, the competition was very level. The competition was very level. That meant the 22V10 business was a commodity business. Everybody was making them. Nobody could make any money. If you wanted to make something bigger, a 23V11 or something like that, well you just couldn't because there wasn't a JEDEC model for it. Well, you could go to one of these companies and say, "Well, why don't you do a new device model for me?" And the first response would be, "No." And the second response would be a seven-digit figure, how much you would have to pay them to do that. So, fundamentally, you couldn't innovate architectures in the PLD business unless you had your own software. And there was only one company who did: Altera.

So, this lesson wasn't lost at Xilinx. Basically, the people at Xilinx said, "Look, in order to innovate on our architecture, we have to own the software for the physical back end." So, why did Xilinx get into the software business? Because they wanted to be able to innovate the architecture. They weren't going to go out and pay somebody a whole big pile of money to do the tools for it who would then hold them hostage and create a commodity business, so they can sell more software more easily. So, that was strategically what was going on. Also, —the device also-- PLDs, it was this two-level rigid structure. So, as you added inputs and outputs, it grew quadratically. But the FPGA had this array of cells. It could expand to any size. It didn't grow quadratically. It grew semi-- sort of linearly.

Jenkins: Yeah.

Trimberger: But that meant you had to do Place and Route, so very time consuming. So, what happened to Xilinx is, Xilinx — first had a tool for manually pip poking , saying okay, I want to put this logic in this block, and that logic in that block, and connect a wire from here to there. A very graphical kind of thing. But then we wanted to get Place and Route to work as the size got pretty big and it would get—this is getting pretty complicated. And so, Xilinx was taking on, not only the ASIC business, but the EDA business as well. Both of them at the same time. It's amazing that company could survive! They weren't taking advantage of either of those industries. What really made them survive was that same point I made before: What are these FPGAs? They're slow. They're small capacity. And they're expensive. Nobody in the ASIC business wanted to be in that business. In fact, the first year, year and a half, at Xilinx, I was worried that VLSI would just come and clean -- take the market away because they had all the pieces. They had the software guys. They had the design centers. They could just make one. They're already in the programmable logic business. They could just make one of these things if they choose to do it. And they had a fab. They could just-- in three months, they could have driven Xilinx out of that business. And so, for me, I was working hard because I knew it. Any given Monday could be the last day. And as much as I liked that product and thought the company wouldn't be around-- at first, I thought it wouldn't be around because the product was worthless. But then, I thought it wouldn't be around because somebody else would just take the business away. But those people who would have taken the business away thought the product was worthless so they didn't bother doing it.

Jenkins: Interesting.

Trimberger: So, yeah. That was my job, making Place and Route run faster. The guys who had done the first Place and Route system really didn't have the background in EDA, in some of these complex data structures, and so on. They were really good at making software run on the underpowered PCs of the day. And they were very good at that! But it was very difficult to make that run. And the algorithms they had were-- some of them were very, very simple.

Jenkins: Were you calling on background you'd gained at Caltech and at VLSI to--

Trimberger: To be fair, no. In some ways no, In some ways, yes. Yes, in the fact I was very comfortable with complex data structures and the EDA kinds of data structures, network, net lists, and things, and knew some of those algorithms. But it was more from reading. That was the other thing. I read all the stuff. And I was still involved with conferences and publications. And I was going to DAC every year and reading the papers. So, even though I wasn't personally involved at VLSI in the Place and Routing stuff, I knew all the stuff. And so, I got to Xilinx and said okay, well let's proceed. And partly, I-- one thing I wanted to do was project an image of greater expertise than I think I actually had. I was coming in. Oh, this PhD guy is coming in with five years of experience in the EDA business and is coming into the brand-new company. And I said okay, I definitely want to calm this situation down because these guys were getting totally beat up. It's a semiconductor company. As I said, they don't appreciate software. It's a customer's vision of the company, the lens. And when the customer says, "This chip is too slow. I can't get enough logic into it," right, these classic ASIC problems, the semiconductor company is not going to blame the chip. They're going to blame the software.

Jenkins: Yeah.

Trimberger: So, software got blamed for everything. A lot of it was justified. It was put together very quickly, understaffed group doing things that they didn't have experience doing, didn't really understand. And so, that was very difficult.

Jenkins: So, let me ask you a question. I didn't arrive for another several years to land at Xilinx. But I do recall that when I did, Xilinx had engaged with an outside company named Exemplar who was somehow or other providing a capability their tools offered. So, Xilinx had still their own basic framework that they were doing the software with. But they were open to outside contracting or something like that in order to solve the current problem that industry needed. Is that--

Trimberger: Yeah so, there was certainly a recognition that you couldn't do everything. You didn't have enough people to do everything and all that. One of the things that Xilinx was doing was we had people doing code, —we also had some outside contractors do stuff. So, they'd bring in contractors to write stuff and link it in. And Exemplar was one of these, kind of a funny relationships. The agreement was they had a synthesis tool. And they would link that in. It's just a circuit optimizer. Synthesis is too grandiose from 2017 to think back. But it's a little circuit optimizer. And so, the idea would be to link that in and put that under the Xilinx umbrella, so nobody would even kind of see that.

Jenkins: Wouldn't be aware, yeah.

Trimberger: Ewald Detjens, at Exemplar didn't want to give his code to one place. He wanted to sell it everywhere. So, he had an arrangement where Xilinx would get it, and he would still be able to sell it and so on. And so, he kept the company name and sold that as well. That was bringing in this code under the Xilinx umbrella. Xilinx also did not do a schematic entry. So, it was links to FutureNet. It was links to-- god, I forget, Data IO at one point. That company came back.

Jenkins: We both remember FutureNet. That was the schematic, yeah.

Trimberger: There are also links to the third-party suppliers with netlist tools that Xilinx was not doing. And so-- go ahead.

Jenkins: I think the point I was trying to get, which I think is in agreement with you, is that Xilinx recognized they had a software problem that needed to be solved. But they also recognized they need to do it quickly because they needed to get into market so they could be selling chips. And sometimes, the way to do that is to use outside help.

Trimberger: Yes, and so, then the question is what part of it must we own. In order to preserve that ability to innovate architecturally, we must own the physical design part.

Jenkins: Yeah, back end.

Trimberger: The schematic entry, anybody can do schematic entry. Read XNF. Oh, what about this optimizer? You know what? We can link an optimizer, and do some optimization, and connect it to this physical design. Those decisions changed over time. We can talk about that, too, but also I should say, I got to Xilinx in ’88. The company was founded in ’84.

Jenkins: Right.

Trimberger: So, Xilinx has been through this a couple times. Xilinx had the original 2000 series. And it was primarily manual entry. 3000 series, some automation. There was one thing they had not done was pay a lot of attention to how much interconnect you need for the amount of logic you’re putting in. There’s good and bad on that. If you really look at some of the classic work, the Rent’s Rule stuff that came out of IBM in the ‘60s and ‘70s, if you looked at that, you’d need a lot of wiring to connect these things up. That would have made the chip even bigger and less efficient because customers look at how many gates you can fill, not how many wires you’ve got. And so, there was this really powerful tension that you wanted to produce something with as much logic as you can. And you really want to bound the amount of wire. And actually, it was semi-automatic because you could build this one cell, which is a couple thousand transistors, and make copies of it.

Jenkins: Yeah, step and repeat.

Trimberger: And that’s very efficient as far as design, as far as verification. You can make these really high capacity chips. But the amount of interconnect you need grows more than linearly with the number of blocks you put in. So, at some point, you make this array big enough, and you can’t route the thing. So, that’s what Xilinx was really running into. Fundamentally, what was Xilinx running into when I got there? They’d made this array big enough that you just couldn’t route it. Now, humans would do some optimizations that the computer couldn’t. Oh yeah, know what? When you get to this point, I’ll take this. You don’t really need this data. I’ll optimize that away. I’ll take that out. And they’ll restructure something. Humans would do things that a computer couldn’t. So, you live within the bounds you’ve got. That’s engineering, right. You’re given a constraint. You’re given constraints. You do it.

Jenkins: But there was an outside third-party consulting infrastructure that had learned to do some of that and were actually helping customers do designs by doing it manually.

Trimberger: Xilinx had a really excellent field application engineer force. And these guys could do wonders with these tools and get around the problems with the tools. This was good. This was bad. The good part was it kept the company afloat. Basically, these-- the way you got to design to work at Xilinx is you'd start it. And then if you were important enough, you could convince the field sales guys to come out. And they would--

Jenkins: Make sure it happened.

Trimberger: They would do something amazing and do something with it. And there was a joke while I was there that no software strategy lasts more than four months. And the reason was we had a FAE meeting three times a year. These guys would come in and say, "All the software, it's all junk. What I really need is tick, tick, tick, tick, tick. Oh, I need something, constrain a wire to a vertical long line. Oh, I need to be able to block out a whole row of blocks. Oh, I need—" and so, these were the priorities for the next few months. And so, look, we know we really need to put in a more efficient database for our Place and Route tool, so it could run faster. And once it runs faster, now we can try more things. And we'd be able to optimize it and do some of the stuff they're doing manually, we could do it automatically. But it's going to take us six months to do it. If it's outside the four-month window, there's no chance. So, we started sneaking some of this through. But yeah--

Jenkins: The thing is that I remember when I arrived, I was really impressed with how much influence Scott Brown, the VP of Sales, and his field applications team had in terms of driving technical direction. And Bernie Vonderschmitt seemed to be completely behind that.

Trimberger: Yeah.

Jenkins: And that support.

Trimberger: And it was the whole company. If I sounded like I was complaining about it, I apologize. So, I mean on one side, I was-- I wanted to do technically the best stuff that we could do. And I knew what had to happen. Or at least the next couple of steps.

Jenkins: Sure.

Trimberger: But I also understood that this is driven by money. And if you don't have the money, then you go away. And I had come from VLSI Technology. One of the things that-- I should say-- I should step back to my job interview. I got some time with Bernie Vonderschmitt. And one of the things that he showed me in that job interview was the only one that mattered. It was the money. And he was showing revenue growth. I'm looking at this curve, and this is consistent revenue growth. I didn't know how he did it. But this thing that I thought was not going to be very long lived was going to be longer lived. And it was behaving a lot better than the ASIC industry I came from.

While I'm on that, I should talk a little longer about this job interview with Bernie. So, one of the things-- when I was looking to leave VLSI, I interviewed at various places. And I did talk to a few CEOs. And I talked to a few managers and some managers who I would not have hired. And when I talked to a CEO, I told him that. But anyway, I got to Xilinx. I got a half hour with Bernie Vonderschmitt. He's CEO of the company, founder. And one of the things he showed me was this graph. Here's our revenue. And he could have stopped-- this is an adage in sales, right, when you've sold, stop selling. When you've made the deal, stop selling. But he kept going. And he said, "Look," maybe he didn't know-- maybe I had my poker face on. But then he said, "You know, gee, well we will never have a fab. Why? Because everybody-- even Japanese steel companies -- have fabs. There's no reason to have a fab. Everybody, there are all people out there who will turn sand into transistors for you. The trick is not that you can do that. That was last decade. This decade is you've got be making the right thing. You've got to focus and make sure that you're doing the part that adds the value." And then he pulled out some numbers about the total investment of the entire ASIC industry and how the entire ASIC industry had not paid off their initial investors. And he said, "Guess what? We will have paid off our investors by next year." <laughs> I said oh, god, okay.

Jenkins: I think he was an Eagle Scout, wasn't he, Bernie?

Trimberger: That I don't know. But he deserves it. So, the company had been in business. The curves were behaving very nicely. They're starting to make money. They're not public yet.

Jenkins: Yeah.

Trimberger: Okay so, I think this is a pretty safe bet, if we can just get the software thing resolved. But then the other things they we're doing right, whatever they're doing right, focus on sales. Make sure the sales guys are happy. Make sure those guys can function and a lot of other problems will resolve themselves. You get money flowing in. You can start hiring more people. That could only be good. And so, as much as it was aggravating to see that, I believed it. I believed in the process. One of the things I had done at VLSI, I had gone out to customer sites when they were having trouble with software and held hands with customers. And I did gain an appreciation of how difficult it is to be a customer and have a piece of software doing something. And you have no idea why it's doing it. But all you can do is control what you do.

Jenkins: Right.

Trimberger: And so, I had a real empathy for customers and sales/sales support. So, for a while, we were sort of splitting our effort. We had-- well, for a long time, we had wanted these core improvement groups. But we also had the shock troops in the software organization. So, you know what? When those application engineers come in, and they start demanding something, there's a dedicated resource that-- heck, they don't even have to come in for that. You get a phone call. You send it here. He's going to work on that. That was a job that Philip Freidin did for a while.

Jenkins: Right.

Trimberger: He had a really good feel--

Jenkins: Joe Maloney did that kind of thing for awhile.

Trimberger: Joe Maloney did. So, that was what I was brought in for, to work on the software. That first week-- okay, here's this theme that came back in-- gee, this guy Bill Carter. I went down to see this guy Bill Carter because he was the IC design guy. And I said, "Bill, please explain how it all works. I did a bunch of IC design. Don't dumb it down, really how does it work?" So, he took me through the basics. I said, "Wow, that's," I said, "Well, kind of neat. It's interesting." I hadn't realized it could be so,… that brute force.

Jenkins: Yeah.

Trimberger: And he said, "You know, we're going to start work on the four thousand series as soon as Jim Hsieh gets back from his vacation."

Jenkins: Okay.

Trimberger: So, what do I do? Gee, that sounds really interesting. I want to be involved in that.

Jenkins: There you go.

Trimberger: Here I am. So, he said, "You know--" After that first one, he told me about FPGA, about hardware. He said, "I want to understand how software works." So, I said, "Okay." The first meeting we had for the 4000, I said, “Look, here’s how software works. Here’s what Place and Route is trying to do. Here’s its vision. Here’s how it’s trying to do things.” And that had a big impact on that 4000 architecture.

Jenkins: That’s good.

Trimberger: The way it interconnects. It was good.

Jenkins: It always seemed to me that there was frequently a “here’s the spec” to the hardware guys. And they go off and do their thing to the spec. And the software guys are basically given the message, “Here’s what happened in hardware. Go make something that will work with this.” As opposed to let’s both look at the target together and see what’s going to nail us.

Trimberger: Yeah, a lot of that happens. A lot of that happens despite-- no matter how much you want to co-develop this stuff. Somebody has to start driving. And at that time, it was all you could do to deliver a few thousand gates of logic.

Jenkins: That’s right.

Trimberger: You just didn’t get that many transistors. And you had to-- if you’re doing a chip, you had to be pretty sensitive to that. If you put in ten percent more, this becomes absolutely unusable because it’s way too expensive, or you can’t even fit enough inside the reticle. You can’t even make them.

Jenkins: Yeah.

Trimberger: And so, that was really important to do something efficient from the silicon side. What we got to was I had done enough with VLSI design, drawn transistors, did all my optimization stuff, and things. And then I did EDA. And so, I was able to look at this and say this part-- this is on the silicon side, this decision is unalterable. This is a big lever and you can’t change that. On the software side, this is really important. And if you can accept them both, then you can have something really effective. So, the Place and Route, at the software side was fewer, more general decisions. That’s the way I expressed it to the IC design people because they had a lot of cases where all this wire can go either here or there. I’ve got two choices here. Then down here, I’ve got two more choices. Down here, I have two more choices. It’s like you’re walking a maze, and you’ve got two choices. And then each one of them is a resistance. So, each one of them was going to cost you. I said, “No, no. I have one place. I’ve got four choices or eight choices. And I pay it once, but then I’m there.” And on the design part, oh yeah, one eight to one MUX, or two four to one MUXs, or four two to one MUXs. It’s almost the same, but from a performance and from a software efficiency point of view it was real different.

Jenkins: Big difference.

Trimberger: So, that was kind of-- you know these sort of lessons because I was able to cross that boundary. And that's where we went with the 4000, some-- a lot of other really great stuff going on the 4000 that--

Jenkins: So, the 4000 had a lot of varieties in it. It seemed to me like there-- the speed varieties were the A, B, C, etc. And I think there was an H version. And was there an HL or something like that?

Trimberger: So, there-- so, everybody in the semiconductor business did speed grades, right? So, you build them, you bin them. And the fast ones you sell for extra money. Yes!

Jenkins: Right.

Trimberger: You didn't do any extra work. And then the other thing we observed at that point was we had this problem with big arrays needed more wire. And so, we said okay. We will envision three ranges. We're going to have the-- we're going to have a low, a middle, and a high. And we're going to build the middle first. Because that was going to be sort of the high end of the 3000 series and getting a little bit bigger. So, we'll build the middle first. And then we'll go back. And for the low-cost version, we'll tear out some of the wires to make-- and we'll make a smaller array.

Jenkins: Right.

Trimberger: The good part about that is very similar architecturally, leverage a lot of expertise. Bad part is it's new stuff. You had to do new layouts and everything. In fact, everything was pitch matched. So, when you shrunk it, you also had to shrink the IOs too because they were pitch matched. So, it was actually a lot of work from that perspective. Similarly, on the high end, gee, to avoid the problem we had with the 3090 and the 3095, which was you just couldn't route the dang thing unless you had an extremely benign architecture, or you wound up in the software knocking out whole rows and columns of blocks. No, you can't put anything here or here. But still you had the extra wire. So, on the high end, we would add more wire. And so, yeah, we built the original 4000 series and then later did the-- took out some of the interconnect and made the 4000A. And then there was the 4000H was a higher-- a little- had some changes into IOs, as I recall, for high speed.

Jenkins: Was it high speed or high voltage?

Trimberger: No, no.

Jenkins: It's always five volts was the peak, right?

Trimberger: Yeah, well yeah, and it kept coming down. And then there was a-- what was interesting was there was an N version, which was made to compete with Altera's devices that did not have memory.

Jenkins: Non-volatile.

Trimberger: No, no, not non-volatile, no memory.

Jenkins: No memory, okay.

Trimberger: So, the logic block in the 4000 could be structured as an adder, two-bit adder or instead of just a look up table for the logic, it was a RAM. You could write into it.

Jenkins: Okay.

Trimberger: So, you had all these distributed memories. And so, when Altera came out with their-- with a competing product, it didn't have the memory capability. Iit was lower cost. So, Xilinx created a marketecture, they called it, that basically it was the same device. But just like we would speed- bin for speed, just bin these for memory capability. And so we just didn't guarantee the memory and sold them for much less. And so, marketing stepped in on that. Then there was a version done… In one of those things where companies do something stupid. A big customer at the time was Quickturn. They did logic emulators. And one of the problems with logic emulators was they had to partition their design into a hundred chips. And so, they used a lot of IO.

Jenkins: Right.

Trimberger: In fact, they were using only about ten percent of the logic because they were IO bound. And so, what they asked for was more IO. They asked Xilinx to build a chip with more IO. So, Xilinx did. And it actually took the 4000-- took a device, the 4005, and basically brought all the signals, input and outputs out. Just brought them all out, and doubled the number of IOs for a specific customer. And so, as soon as that was available, the customer said, "Oh, that's very interesting. We notice the IOs are very similar to the 4013, the bigger chip. We want the bigger chip but sold to us at the lower price."

Jenkins: I don't want to pay for what I'm using.

Trimberger: And Xilinx said yes.

Jenkins: Oh my.

Trimberger: Should have said yes from the get-go; it should have been the first time they came and said twice as many IOs, we said, “Look, we’ve got one of those. It’s called a 4013. And we’ll sell it to you for a twenty percent discount,” or whatever. But they didn’t. Anyway, that was that. So, what made FPGAs successful? Amortizing the design cost across all the customer base. So, when you start making custom chips for somebody, you void the advantage. And so--

Jenkins: You only had one customer.

Trimberger: One customer before you’re back in the ASIC business. And what killed-- one of a number of things that killed the ASIC business-- too bad Doug’s gone. But he could argue. But I’ll tell you, one of them was only one out of three of the designs ever went to production. And those companies killed themselves on the front end. The tools weren’t that good, a lot of design effort. It was expensive. And so, they would-- two out of three would not go to production, you’d lose money.

Jenkins: Yeah.

Trimberger: And so, in the FPGA business, the ones that didn’t go anywhere didn’t cost you that much. Yeah, maybe you had some field support help. But you didn’t make masks. And you didn’t write new test programs. And you didn’t do all sorts of other things. And so, you did not want to get back into that ASIC business where you’re doing custom FPGAs for somebody who then will buy it or not buy it with a thirty percent probability. So, that was a lesson that got learned, and unlearned, and learned, and unlearned, throughout this industry by everybody who’s been in it.

Jenkins: Did you think-- well jumping ahead maybe five years or six years to the Virtex-II Pro, was that a similar thing because they had taken the subsequent architecture and worked a specific microprocessor into it? And as I understood it, it had all the support issues you have with that as well as not quite the acceptance that you’d want to have from the customers. And even the ones that wanted it didn’t really step up and buy it.

Trimberger: Man, there are so many lessons in that device, good and bad.

Jenkins: We jumped over a lot when I brought that up. I just thought it was a similar type thing is all.

Trimberger: Yeah, it's worth discussing, but the dropping of processor in there adds a whole new dimension of complexity in the software side.

Jenkins: Oh, yes.

Trimberger: And that just drove this whole issue through the roof. There was some very clever work done to encapsulate that. And gee, we're going to offer three different versions of a processor in FPGA. And they're going to come with the wrapper of programmable logic around them. And depending on what you're trying to do, you'd use one, two, or three, in an attempt to bound that. But it would just-- the complexity was so great. First of all, everybody wants to step outside the model. They're engineers. They're going to meddle with it. But yeah, that was-- it just got-- it got really hard to use. And that was just the tip of the iceberg. And you start looking to the 2000s what happened. But yeah, that was a good example.

Jenkins: Let's go back and try and get up into Virtex pretty quick. But I remember you said when Jim Hsieh came back from his vacation that you'd be starting in on the 4000 family. Was that--

Trimberger: We'd start some design meetings. We'd get together every week, and we'd do the design. So, we stared that, which is understanding all these other problems.

Jenkins: What was the date approximately? What year? Is that '89 or--

Trimberger: That was probably still '88. I mean I got there-- I got to Xilinx in October. And I believe that was by the end of 1988 we started talking. Yeah, it had to be.

Jenkins: Okay. And it seemed to me that the 4000 seemed to roll out over about, the varieties and so forth, a two to three-year period, wasn't it?

Trimberger: It was longer than that.

Jenkins: Okay.

Trimberger: So, the way this worked is-- well, yeah so, during the 4000 development, Ross Freeman got sicker and sicker. And this wasn't-- wasn't really able to--

Jenkins: Wouldn't be coming in--

Trimberger: To participate. He was executive at that point anyway, and Bill Carter was in charge of the design stuff. Xilinx had a kind of a nominal year development process. And then the first device you make- this was Bill's goal, it's sort of a moderate size, moderate large capacity, something big enough to be interesting, and useful and maybe made some headlines.

Jenkins: But learn about the main problems.

Trimberger: And would yield really well.

Jenkins: Yeah.

Trimberger: It's sort of the sweet spot kind of device I'd characterize it as. And then the second one would be kind of the stretch device. Make something big to really make some headlines. And the goal on that, I remember him saying was "Yeah, we want to be able to yield three per wafer." <laughs> That's how far you push it. And then after that, you start filling in around devices. And there was a lesson that came out of that. In retrospect, at one point, I looked back and said okay, which were the big sellers in different families. And it was that-- and they were basically in order of--

Jenkins: Delivery.

Trimberger: Delivery. And so, that's what impressed on me gee, those things we were talking about, performance, and capacity, and cost, ***availability*** trumped them all. If you had it and could hold it in your hand, you'd use that. It sort of made total sense, right?

Jenkins: Yeah.

Trimberger: If I'm building a-- if I could build a wall, and I've got bricks I can hold in my hand versus bricks I can order from the factory that might come here someday--

Jenkins: Yeah. Let's get started without tools yeah, what we've got.

Trimberger: So, yeah that was-- but anyway, so these would roll out. That initial 4000 family would roll out over the period of oh, about one a month. You get the first one out. You want to debug it. Any type of problems you want to take care of. And then they could roll out pretty quickly. And then we did-- started with the variants. And the A series, and so on. Somebody would say, "Oh, yield improved." Oh, well, we can make something even bigger. Oh yeah, now that one we thought was yielding three is yielding twelve. Look, we can make a bigger one.

Jenkins: Excellent, we can get to that three yet.

Trimberger: Yeah, we can keep pushing that up.

Jenkins: So, I recall-- I'm going to say I got to Xilinx around '92, and maybe even '93. But what I'm getting at is that the field sales guys were talking about the biggest part they'd ever dreamed of, something called Akebono, which was named after a giant sumo wrestler. It was an inside name at Xilinx. And I remember somebody in the field sales at one of these sales meetings or FAE meetings saying, "We're never going to find a customer who wants to buy one that big." And looking back on it, that would be like a spec in your eye now in terms of its size. And yet, it was, what, the size of a postage stamp, I think.

Trimberger: It was a very large die. And part of the reason for the complaint was not that he didn't think customers needed that much logic--

Jenkins: Bragging rights?

Trimberger: Well, the concern was the only way we could make that was we're just going to tile this thing out until-- gee, we're just going to fill the reticle. Okay, you know what? We're going to bet on-- we're going to bet that yield is going to improve, we're going to yield some of these things and we're just going to fill the reticle. And we're going to get, you know, we're going to get bragging rights and we've got this big die. But a few things broke when you do that. You get a thing and you get these long wires that are, you know, the configuration of memories, and in memory you get these really long bit lines, really long word lines. Oh my God, right? So some work had to get done there. And then there's also this place and route issue, right? You make this really big array. You're not going to be able to route it.

Jenkins: Don't have enough routing. Yeah.

Trimberger: But you know what? By that time we had sort of figured out some of these techniques. You know what? Knock out a whole row. If you knock out a whole column, then you have two wiring channels next to it, then you can think of it as a double wide - twice as much wire. And so you start doing some of these tricks and the tools can manage that and you start-- this is manageable.

Jenkins: Well, not to mention Xilinx, this is going to sound like it's down in the dirt, but Xilinx had figured out how to use logic cells for routing already.

Trimberger: Yeah.

Jenkins: In other words, if that's what I need then I'm going to burn some of these guys here, just sacrifice them.

Trimberger: Yeah. Yeah. That was another one of these little tricks you could play that, you know, that software didn't-- couldn't do it originally but application engineers would. And so they had an advantage they knew about. They knew about these sneak paths and back doors and shortcuts. And eventually the tools started to catch up. But the real answer is don't: you can't be relying on that. And that's really what happened in the '90s that we all recognized Moore's Law, double number of transistors.

Fairbairn: Yeah. Sure.

Trimberger: But 90s we went from two layers of metal to 12 layers of metal.

Jenkins: The 90s?

Trimberger: Yeah. And so suddenly you could put a lot more wire in there and that made a big difference for the architecture of FPGAs.

Jenkins: The routing didn't go wide. The routing went thick.

Trimberger: Right. And you could say like well, the routing went long. And so you could reach a lot of places. You weren't as stiff from a software point of view. You could put these two blocks further away and the timing is just about the same. So that was a big deal. And that was an architectural innovation we put in the 4000X. It's the high end of the range, we called it X.

Fairbairn: Right. Yeah.

Trimberger: So there's the EX family, the XL and XV and which voltage it was running at. And those actually have additional wiring channels. But, yeah, they were just huge. And then they put this Akebono to shame.

Jenkins Yeah.

Trimberger: Yeah. It was, you know, suddenly the whole lid blew off that we could produce and we produced these really huge devices. And they were routing, too. That was the other part.

Jenkins: So I also remember Xilinx had some other threads going on where they had a family, I believe it was called Albert? Was that the 5200 family?

Trimberger: Albert was, I think that was five, yeah, the 5000, 5200. Yeah. So that's really interesting, so hang onto that thought.

Jenkins: Okay. And I was going to say and Cat.

Trimberger: And Cat. Let's talk about Cat first.

Jenkins: Sure.

Trimberger: Cat was the Compacted Anti-fuse Technology. That was Erich Goetting's first project. Well, not his first project. He was an Akebono guy, too, I think. But anyway, he was doing Cat. One observation inside of Xilinx or coming from the customer base was "Gee, these antifuse devices, they're faster than the RAM devices and they got, you know, obviously gee, if you could have a memory that as small as a Contact or a Via, that's a lot better than memory that's six transistors and all that junk."

Jenkins: Sure. Yeah.

Trimberger: So clearly it's better and so we should do one of these too, because that's the wave of the future.

Jenkins: Right. It was unclear what the future was going to be in terms of things.

Trimberger: Keep in mind it's a different technology and this is what, you know, eventually led to its demise. Then, Xilinx did this and Erich was leading that and I was on it for sort of an advanced development kind of role. Did some interesting work in working around defective antifuses, which I think that was really interesting. Xilinx produced these devices and one of the goals was to actually make an FPGA. Another goal was to replace the entire TTL catalog.

Jenkins: Mm-hm.

Trimberger: Said look, there are these companies and they're making these, you know, TTL Databook chips and they've got a couple hundred chips they've got to make. Well gee, we could make one chip and just burn it a couple hundred different ways and just ship it. And now with one piece of silicon we could replace the whole catalog, we could take all that business.

Jenkins: Well interestingly, in John Birkner's interview on PALs, that was what he was targeting to do with PALs. It was a goal that he saw.

Trimberger: The reason that it didn't play out that way was because engineers weren't building out of jellybeans anymore because they had—I could put a few thousand gates onto one chip. Why am I worrying about octal inverters or quad NAND gates or something like that No, they won't do it. Right? So--

Jenkins: The push for integration was there.

Trimberger: Yeah. So it was so obvious to the engineers doing it. But, you know, still not a bad idea. If you could replace the whole catalog, whatever, we thought that was nice. So anyway, the anti-fuse device caused problems for Xilinx because Xilinx had been arguing against antifuses for a long time. It was sort of like a politician saying, "You know what? We're, never going to raise taxes. Never going to raise taxes." Then they raise taxes.

Jenkins: Yeah.

Trimberger: So they can't, they just can't do it, right? But, you know, Xilinx had this going and actually had shipped some of these to some customers fairly quietly, but they shipped them, and then just pulled the plug. So a lot of this has to do with competitive pressure. So around that time Altera had come out next--

Jenkins: Did we also bring in Mr. Roelandts as president and he was kind of trimming things?

Trimberger: Yeah, nah. I don't recall that being a Wim issue. Wim was a lot more hands-off than Bernie. His stated position in several of these meetings is, "I hire all you lieutenants and I don't know this technology better than you do. If you can't agree, I have to make a decision and it's a coin toss. So don't come to me with an argument. I mean I'll just toss a coin and it'll be as good as anything. Come to me with an answer."

Jenkins: Yeah.

Trimberger: So, if the silicon guys and the sales guys were arguing with marketing - I can't think of how to spin it. He could just make a decision. And he would, but that was his decision to go. He really wanted to push that back onto his people. That's a sort of management style that I tried a few times but has not been all that successful for me.

But anyway, to get back to the Cat. What was happening was we were pushing out the large 4000 devices. That was in response to Altera. They had taken the lead on big devices, on speed and market share. And so what actually-- and that was what led the push to the 4000. So we had all these derivative 4000 parts we've talked about, but it really distracted the company for next bigger, next bigger, next bigger and we're doing the smaller ones and no memory ones and so on.

Jenkins: Right.

Trimberger: And so then this-- the Altera 10K device came out and we were hurting.

Jenkins: Yeah.

Trimberger: So that focused me. I had been working on is, this Prizm project. I said, "Oh," I went to Dennis Segers and I said, "Look, been doing that and I've also been doing a 4000 derivative. Look, we can complete the architecture definition really, really quickly." And he said, "All the pizza you can eat; get it done this weekend."

And I said, "Well, we don't have to do that, that fast because we had this whole process that we have to go through. But we can nail the important parts within a couple of weeks and get the next people working and then we will simultaneously progress on architecture definitions, circuit design, layout and completion." And so, we didn't actually have to get it all done in a weekend or even a week.

Jenkins: Right.

Trimberger: Actually, that wasn't Dennis who said that. Who was it that said that, anyway?

Jenkins: It wasn't Bill.

Trimberger: No, it definitely wasn't Bill.

Jenkins: Dennis was kind of leading architecture definition or something at that time, wasn't he or?

Trimberger: Dennis was--

Jenkins: Dennis Segers?

Trimberger: Dennis Segers. So he was, well, he was kind of advanced development, heir apparent to the head of engineering. So yeah, that's how that played out. Let's make some decisions very quickly. And that's when I, I actually got pretty good at forcing decisions at that point. because I contributed on the architecture definition 4000, I led the architecture definition for the 4000X. And so that was, "Okay Steve, make it so!" And so, you know, it was interesting because we didn't have any time. A lot of decisions that we would like to have data for, my rule was so here’s the thought experiment:..

Jenkins: Right.

Trimberger: Assume Moore’s Law is 2X in 24 months. Call it 25 months. Call it 20. Okay, well, that's like four or five percent per month.

Jenkins: Okay.

Trimberger: If you work at a decision that's going to make less than a four or five percent difference, and it takes you more than a month, it doesn't matter what your answer is. If you have a decision that's going to make one percent difference, if it's going to take you more than a week, it doesn't matter.

Jenkins: Yeah.

Trimberger: And so we had a lot of these things that came back said. "Gee this, you know this sounds like a five percent thing can we get that done?" And so I squeeze it all down. Said, "Look, we can come up with an answer for that but really, the right thing to do is pick one. So we're going to pick one. We'll pick this one because it's power, or simple or whatever it was." And boom. Nothing clears up decision-making like realizing you don't need any of the data. <laughter>

Trimberger: And yeah, oh yeah, we like to run all these test cases and so on and so forth. How long that take? Four months. Is it going to make 25 percent difference? No.

Jenkins: Yeah, interesting.

Trimberger: So we just flew through that, and it actually made a lot of really good decisions because we had a lot of people with expertise at that point, which we didn't have five years earlier because there just wasn't that much expertise in the FPGA business.

Jenkins: So you to said the ‘P word’. I'd like to hear you go down and talk a little bit about what Prizm is.

Trimberger: Prizm. So let's roll back a little bit.

Jenkins: Sure.

Trimberger: And so after getting the 4000 out there, I was still doing stuff with architecture, doing a lot with software. Something I had realized about software was that observation that the way to get fired out of Xilinx was to get in charge with software, and I had that opportunity to become in charge of software. And one of a few times in my life I declined. And I actually had a nice talk with Wes Patterson about that.

Jenkins: It was a land mine you didn't like standing on.

Trimberger: Yeah, I mean just and so anyway, I stepped out of that. And so I wound up in this role of advanced development and doing architectures. And then what I had characterized advanced development as is-- and I use the pizza model. So the pizza model is: you know when you go into a pizza parlor, they don't, you know, start chopping the mushrooms when you order mushrooms, right? Those are all chopped in advance. So they have all these bins of all these ingredients ready to go and you decide what you want in.

Jenkins: Mm-hm.

Trimberger: So I said, "Okay, my role is going to be I'm going to fill some bins for you." And so when the next architecture comes by and say, "You know what? I want to put in a large block of memory. I want to put in a microprocessor or I want to put in, you know, high speed. There's going to be a bin that says how to do this."Okay, so I did some of those. And then this opportunity-- this question came up. We were suffering under these issues of device capacity, how many gates do I have?

Jenkins: Right.

Trimberger: How much should it cost and so on? I said, "Well, gee, why don't we have a device where I can make gates just by having multiple copies of the configuration memory?" So, you know, memories are relatively small. Inside I have the gates programmed one way and then in a cycle I switch. It's like microprocessors, right?

Jenkins: Right.

Trimberger: I got a CPU and I got it in different instruction every nanosecond.

Jenkins: Right.

Trimberger: Well, of course, back then it was every, few tens of nanoseconds.

Jenkins: Yeah.

Trimberger: But I said, "Okay, well, that's a great idea, right?" Then if I have 10 copies of the configuration memory, I've got 10 times as much logic in the same thing. I solved this problem that we've had since day one in this business. This problem of "I don't have enough gates." Okay?

Jenkins: Right. You created 'virtual gates'.

Trimberger: Virtual gates. How do you do this? And actually, I knew how to do this because I had done-- I had studied years earlier, I looked at things like compiled code simulators and stuff and a lot of that stuff applied here.

Jenkins: Yeah.

Trimberger: I said, "Okay, gee, here's how you compile. It's a scheduling problem. I got all these gates that have to happen and I've got cycles it's got to happen in. I can schedule them out and gee, will they fit? And that was the first experiment and yeah, they'll fit. Actually, to be fair, roll it back. The reason I had looked down this road was because I had been looking at how do you compile? How do you include a microprocessor in the FPGA fabric-- or FPGA microprocessor. How do you stick them together?

Jenkins: Right.

Trimberger: So, that's how the compiled code simulator and stuff fit together. Then I wound up going through this thought experiment. If I was going to really, I was going to put instructions in the microprocessor to make acceleration happen fast. What would they look like? They looked like lookup tables and that's what I had, so okay, sequencing through the instructions. Okay, it was a very short period, a very small amount of work to say "You know what? There's cheese down this tunnel, I think!" So then when this question came up. We were just starting to get beat up, and it was similar to the Cat thing. Those antifuse guys, they got more gates than we do and are cheaper. How do we get cheaper? I said, "Okay, well, gee I, you know, there's the thing if we could reprogram the device." So, I said, "Okay, let's proceed. Let's go work on this for a while." So I did a little bit more and I started writing some place and route software and things like that. Took the architecture, and did this simple thing to the architecture. Took an existing device, multiple copies of the configuration memory, some clever design about pipelining the load so you can switch to configurations much more rapidly. Then we got a customer interested, Lockheed Martin interested. So they came in and said, "Oh yeah, this is really cool." So they were excited. They had a government contract they wanted to do, and actually wound up landing the contract and writing us into it, which caused problems a little bit later because a little bit later this was not working.

Jenkins: Right.

Trimberger: And the reason it was-- and so it was okay. I mean everything that we looked at was correct. Yes, you could schedule the design. Yes, you could do the place and route. After a little more analysis the problem came down to, the first part was the amount of-- the configuration memory in the FPGA- is not negligible. It's about 30 percent. I mean about half the transistors are about 30 percent of the area. So we were making something with eight copies. eight times point three is two point four plus your other point seven. So this is three times that area. Or said the other way - a third as much logic in the same frame. Hmm, okay, my 8X then went down to eight over three X. So, so that was that. Well, gee, then you are doing things like well it's three times the size, your wires are a square root three longer. So you're roughly giving away 70 percent of performance. Ooh. . so now my total amount of logic capacity in here is down to 8 over 6 kind of numbers.

Jenkins: Right.

Trimberger: And there's a little bit lost in framing and scheduling and things like that. And then this thing called Moore's Law said look, by the time we get this done, we could have just sat on our rear ends and got 2X. So yeah, there was the software complexity, it turned out to really be an issue too. So that's what was happening here.

Fairbairn: See eight over three is two and two and two-thirds X, right? Something like that. So, yeah.

Trimberger: That divided by 1.7, now you have some percentage points and now, you know, the arguments before said, "Okay, how long does it take to get to make up the 20 percent? Three months?"

Jenkins: So one of the things that a lot of people don't know about you with your warped personality and all that is you would carry around a little engineering notebook with you all the time. And as I have grown to actually witness ideas occur to you all the time because you're looking for ideas of things to do and so forth. And you'll just, excuse me, open your little notebook and make a little note of something, maybe today's date, and then proceed with the conversation and go on. And then 90 minutes later you'll do it two more times etcetera. And so this, of course, became little kernels of ideas that you would have. And later on, I mean things came out of this Prism project that we may not have ever realized in terms of-- or Xilinx may not have ever realized in terms of direct payback, but the ideas themselves resulted in a different form of payback, which was legal protection.

Trimberger: Yeah.

Jenkins: And I'm not sure that Prizm itself was the source of those, the main ideas, but there were plenty of them there and you had that pattern.

Trimberger: Yeah. You know, I started carrying a notebook because my memory wasn't good enough to remember all of that stuff, to be fair. <laughter> And by the time I left Xilinx I had a couple of big boxes of notebooks that I had gone through. And it was not at all uncommon for, me to go back and look and remember thinking about that:" I think that was back in November and so well at least back to the previous November and find something.

Jenkins: So how many ideas-- like, this is a site thing. How many times would you have an idea and then say to yourself, "I think I'll write this up as a disclosure?" Because I remember you had very terse disclosures. You could get to the kernel of the idea and turn it into legal, and they would generally say go for it.

Trimberger: That came from experience knowing what was essential and what was not, and dealing with legal directly was different than dealing with the patent committee.

Jenkins: The two. Yes.

Trimberger: Yeah. When you had to describe it for that group, you had to have a little more meat on it. And the composition of that group changed, it had to be different as well. But yeah, I've got a bunch of patents. But I think probably only one in three or one in five, or something like that, of those snippet ideas actually were even considered for patent. It's kind of an interesting thing to do and I'll write it up and I'll note it. But no, that doesn't sort of pass the bar as something that I really want to chase down.

Jenkins: I just figured that as a role model for somebody who's an inventor that might be watching this they might want know that part of your secret process is to just write down everything you could, you know, you thought might hold water and--

Trimberger: Yeah. Well, that's what I think it's Thomas Edison, "To have a lot of good ideas, have a lot of ideas." And so, part of it was that it's just too hard to remember it. But, you get these flashes of inspiration and you've got to preserve them. I remember, this may happen to everybody, but I remember when I was younger a lot I would wake up or partially wake up in the middle of the night say, "Oh, I've got this great idea." And it's in the middle of the night. And I, you know, just know, I'll get go right back to sleep. In the morning I wake up and, oh yeah, I had this really idea-- good idea. But I cannot remember what the idea was.

Jenkins: Yup.

Trimberger: So I started sleeping with a notepad by my bed. And so one night I woke up, I've got this great idea and a notepad is right here! And I reached over and I grab the pen and I wrote, because it's dark, right? And you, so that's another thing you learn. You write big in the dark, because it's more of a chance that you're going to be able to decipher it. So I wrote big, big letters and I got it on a notepad and I set it down and I fell asleep. I slept the sleep of the virtuous that night knowing that I had captured this absolutely brilliant idea. So that's that story.

Jenkins: That's fun.

Trimberger: Now that I've got you in suspense, do you want to know what the idea was?

Jenkins: What?

Trimberger: So I got up in the morning and I looked at the-- I looked at the pad of paper. Very clearly written in big, big script, "Turn it around to the other way."

Jenkins: Okay. What was "It?"

Trimberger: I have no idea what "It" was or why you should turn it. But sort of the lesson from that was I think--

Jenkins: Maybe turn the lights on and give it a--

Trimberger: I think, yeah, it's sort of like what's the answer to the question of life, the universe and everything is: 42! I mean, but “turn it around the other way”. And the lesson I actually took from that was; it is possible to be certain without having something to be certain about. And so yeah, I've tried to be a lot more specific.

Jenkins: Yeah.

Trimberger: But yeah, I actually did a lot of work. One of the things I all my life I, going to bed I think about what's the next problem or this talk I have to give, what am I going to say. Or this problem I'm wrestling with, what I'm going to work on. I go to sleep and I let my mind work on it while I'm asleep.

Jenkins: I've experienced the same thing.

Trimberger: And, you know, you just have to. So, you know, why waste that time? Unless it starts interfering with your ability to sleep. That's you have to worry about.

Jenkins: Interesting though. We kind of talked a lot about the 4000 and a bit about Prizm. What about the time that Xilinx was making the transition into doing the Virtex family. You sort of changed your role, I think, about and that you'd been working in Bill Carter's design group for quite some time. I remember that the guys that did the original Virtex were like Steve Young and some architecture guys like that. How much contribution did you have in that family? Do you remember?

Trimberger: Yeah. It's a lot less. So what happened was I was the 4000 Series guy. And there was 5200-- 5000 Series group which Steve Young and that-- those guys worked on. And they went back and said, "How do we do a lot of the stuff much more efficiently?" Because that was intended to be a low-cost architecture, and it was. A lot of interesting things happened on that. They rethought a lot of the decisions, you know, good work from them - Scott Nance and these guys.

Jenkins: Yup.

Trimberger: But I was a 4000 Series guy. So I was working on 4000. So what happened was the 5200 came out and then we got clobbered by the 10K from Altera. So clobbered in every dimension, technically and business. So actually, Xilinx lost leadership in the FPGA business to Altera.

Jenkins: Yeah.

Trimberger: And so I stood up as the 4000 guy. "Look, we have this architecture that's going. I can very quickly get us back into building a bigger device and make it routable and higher performance." Because of what I had done. I had all these little pizza bins and "Gee, if you do it like this you're going to unload the wires. It's going to run faster. If you do it like this, you can to save area or you're going to do this." I had several techniques that I was ready to drop in. I actually had architectural changes to the logic block as well. So I had the tools background, as I was still working in tools, this whole time.

Jenkins: Sure.

Trimberger: But so I have the tools that verified that we could improve it like 20 percent or 30 percent - the logic capacity of each logic block. It'd be a big deal. So anyway look, we need to catch up real quick. So I got on the 4000X project and charged ahead on that. Meanwhile, Steve Young and those guys started working on this next generation, larger capacity, low-cost architecture-- remove some of the performance issues that they had and that was what became Virtex. So it was a separate group. I was on 4K extensions and all that. We did 4K extensions and then Cat got terminated, so Erich and that entire design team moved on. They started working on Virtex. To launch that, I had been talking with Dennis and said, "Okay, well, I did my evaluation in what's happening with this thing we were going to call Virtex. Forgot now what the internal name was, but it'll come to me, I think. It was very good. And I told Dennis it was very good.

Jenkins: Right.

Trimberger: “So I can keep pushing the 4000 but, you know, the stuff that they can really deliver on what they're saying. That's a great thing”. It was a different architectural style. The 4K was still in this grid style where I've got a logic block with connections on the sides. And then Virtex had a logic block with a connection into the mesh grid. Connections only logic-- logically connection on one side. That let them take that principle of fewer, more general decisions, take it one step further. All the decisions wound up in that logic block-- in the interconnect block

Jenkins: Right.

Trimberger: It was much more efficient. They could actually play with the efficiency right there and it was much more effective. So that's-- that was that decision. So then I am the four thousand guy-- I am the old guy, right?

Fairbairn: Yeah.

Trimberger: So these guys went on. So now what do I do? Well, I can join that project. You know, honestly, there's some tension there, and they're successful. Why would I want to step in and mess that up? So I said, "Okay, they're successful. Gee, now what do I do?" So that's when I took this advanced development role, which I sort of had been doing advanced development for the product, to be more advanced development for the company.

Jenkins: Yeah.

Trimberger: And so that included-- well, now Wim came in and Wim came out of HP. Wim's vision of research was, "I have product people on my product. Research, I want you to do something new for me, like HP. Invent me ink jet printers or something. Do something new." Well, so I became the most practical side of that whole thing. I didn't really want to do anything new. There's plenty to do with what we were doing. So I worked on low power. I worked on the--.

Jenkins: Security.

Trimberger: Yeah, security. But, you know, low power at that point was right when it started to be interesting. People started talking about it. And that was funny because, you know, Erich Goetting, now manager, head of the very successful Virtex project and a rising star. Erich did not believe in low power. He said, I remember one quote, I maybe shouldn't repeat it, but yeah, "Our customers will water cool our parts if they want to." <laughter> And we laugh about that now, but the value of FPGA was so high and that availability was so valuable!

Jenkins: Right.

Trimberger: But when we all kind of . . . You know, even I didn't appreciate just how important this was. I thought it was just one more thing to do. But you could save a lot. You could do a lot. And, you know, in the circuits, in the architecture and in software, a lot of that stuff got done. So low power is one and security is one. I have been working on the-- I had been interested in security since I was in grade school. I loved encryption stuff and puzzles. Because it was a puzzles thing, right?

Jenkins: Did you have like skeleton keys and things like that as a kid?

Trimberger: Yes, I did.

Jenkins I did too.

Trimberger: But I didn't have anything to open with them, but I had the keys. I had all the decoder rings and all the different kind of substitution ciphers and things like that. I was familiar with cracking substitution ciphers and things like that. Did the puzzles in the newspaper and all those things. Those were all a lot of fun. But the other thing that I followed was encryption algorithms. And I was really too young to really understand, but I did know a little bit. What I did observe was people were coming up with encryption algorithms and then shortly after somebody would find a flaw in it. And then a new one, a new flaw, a new flaw, and so until DES, and that-- I didn't understand the algorithm. I didn't even know the algorithm.

Jenkins: Right.

Trimberger: But what I knew--

Jenkins: The 56 bit key, that one?

Trimberger: Yeah. What I did know was nobody is arguing anymore that I broke it and somebody patched it and I broke it and patched it. And so this kind of, the arms race kind of was over. Now okay, it's only 56 bit and whine, complain, whatever. You know, at that time the 56 was okay. But it was also absolutely clear to me that if it was not okay the fix was really simple. Right?

Jenkins: Jack up the new bits.

Trimberger: Bigger, more bits in the key. You know, and have a nice day, right? So that game was done. And so that parked in the back of my mind. Then we come out to around the year 2000 and I'm looking at FPGAs and say, "Look, these devices are so big now that there is valuable intellectual property stored in here. And it can be stolen really easily just by observing the bits flying by." We could encrypt this bit stream. Encryption's pretty straightforward. There's this data encryption standard algorithm, been a standard for a long time. We could just use that. And--

Jenkins: And it's been checked out by all the experts and so forth.

Trimberger: Yeah. And because I mean I'm not going to invent it, and again, some of these concerns at that time was encryption was a *controlled* technology. You couldn't export it. You're going to hit legal issues. Remember what the '80s were like, right? And so this was a concern. So, what did I have to do to get encryption in? The first thing was I had to understand the law enough to know that you could do this. Nobody else cared. And I took this idea of "Gee, we can encrypt our bit stream." So I took that to our IC design group, "Look we can encrypt bit streams." And their response was, "No. Because the software guys. We're not going to do a decryptor because the software guys have to do an encryptor and they're so busy just making place and route run effectively. They don't have time to do any of that stuff. They're just not going to do it." So, you know, fair enough. So I went over to the software group and I talked to the software guys. I talked to Rajeev and said, "Look, you know, I think —we can encrypt the bit stream. We can put it in the ROM encrypted and decrypt it on the chip and protect this."

Jenkins: Right.

Trimberger: And he said, "You know, Steve, nice idea but we don't have that process technology to store the key." Right, what we haveis this SRAM device.

Jenkins: Right.

Trimberger: "So no problem. We could do that. But you'd never get the process. You don't have a processing technology. You can never get one." So I went to the process technology people. Said, "Look, is there a process technology that we have? Can we store a key somehow in a piece of non-volatile--even if it's not very good, I only need, you know, 64 bits, right? Fifty-six bits, right?"

Jenkins: Right.

Trimberger: And they said, "You know, there's something we could probably do there for that small amount of bits, but the IC design guys would never put this into the chip."

Jenkins: Yeah.

Trimberger: So everybody's pointing to somebody else for the excuse. So I wrote the HDL for the decryptor. I wrote the software for the encryptor. I devised a gate breakdown anti-fuse. Actually the stuff is similar to Cat stuff for process technology. And, had investigated the law. Then, I took this back around to everybody. And as soon as somebody said "No" for some reason, I'd say. "Yes, because here. Oh, you know, software? Oh, here's the software."

Jenkins: Yeah, that part's solved.

Trimberger: Yeah, the process isn't going to let you? Well, you know, if you put four and a half volts across this gate, you're going to blow it up and, it's not going to come back. And so anyway I went around and got everybody sort of nodded and together. And at that time this was Erich Goetting starting the 'Virtex Vortex'. So I said, "Okay, we want to put more and more different things in there, microprocessors in there. Yes, it makes total sense to be able to encrypt this stuff. There's a lot of stuff in here. We're trying to pull all these pieces together." And so that became--.

Jenkins: There was also, I mean that was the timeframe I think, that would've been about the late Clinton presidency when the DRM (digital rights management) was a sort of thing.

Trimberger: Oh yeah, about. It was 2000 when I was kind of working on this, 2002, about the time when I got deployed in VIrtex too. So that's how it goes.

Jenkins: Okay, but I remember two things that you'd shared with me. One was that the FIPS 140, which is one of the NIST standards, had a requirement of having the key be erasable. And you'd said, "Duck soup, we pop it in RAM bits and power it down and it's gone." And the other one was that I think in the same timeframe, cryptography resurged and had started doing power analysis and differential power analysis and showing that they could knowing what the algorithm was, i.e., is it DES, AES, whatever. I could tell you what the key is going by, by looking at the power supply with a set of experimental trials. And, you know, you were saying, "This has got the crypto boys really scared."

Trimberger: Yeah.

Jenkins: And so that got you thinking along these lines, even more so I think.

Trimberger: Well, you had to get on to the next problem, right? So I said, "Okay, DES said the algorithm isn't the issue." Of course, and by this time is coming out that the wheels were coming off DES because of brute force attacks on 56 bits. So okay, we put in triple DES.

Jenkins: Right.

Trimberger: Oh the thing about erasable keys, that's if you want to do military or whatever. And then the other part was “Gee, if you're just doing a device that's decrypting something, then a lot of these other laws don't apply.” About the time that Cryptography Research was doing their first side channel attacks stuff. Yeah, it's got the cryptography establishment worried. But we had bigger —problems. We were using DES because it was approved. AES existed, but it hadn't been approved yet. So I said, “Okay, we got to go with what we got." So that was a bigger issue in most people’s minds than, "Oh yeah, you're using an old algorithm. We don't like it."

Jenkins: Right.

Trimberger: But, the thing with the side channel attacks, yeah, we see them happening. We see them, these guys talking about it, but this is when marketing steps in pretty boldly and said, "Iif customers aren't complaining about it, we're not worried about it." -

Jenkins: Yeah, solve real problems is what they're saying..

Trimberger: Yeah. And then so just getting the very first one done, and this really didn't turn into be an issue until, you know, 2008 or nine or something like that when that started. And, the problem with that was we had a late start on getting solutions done .I didn't properly fill that bin for the pizza. Didn't get them all. So yeah, that was an interesting transition. So there's another talk. I believe it's online in a couple of places where I talk about different phases of the life in the FPGAs. There's early years - you've got to invent stuff and you're just so squeezed for transistors. All you can do is what you can do, and you don't don't a lot of interconnect, and there were some very "efficient" in their use of transistors. Things like the 6200 from Xilinx was obviously one of these mesh architectures, grid architectures that looked really good on paper. But then, when you tried to place and route, you get into real trouble. Moore's Law really took hold in the '90s and you got a lot of gates. And you got a lot of wire and we got automation in there, and that was the golden age of being an FPGA architect because you really didn't have to do any work.

Jenkins: I was going to say you also had this exploding communication market where the price of things was not so important because people would pay for capability.

Trimberger: There was this market. So FPGA also suffered from, because it was more expensive than a custom alternative. The obvious markets you would look for in the semiconductor industry just weren't there for FPGA. So oh yeah, we ought to be in a PC. You know what? They make a lot of PCs, back in the '80s and '90s. And so, they make so many of them they can afford to do custom chips.

Jenkins: Yes.

Trimberger: So oh yeah, we're in these emulators and things where you need reprogramability. Later you look at-- so then when the internet infrastructure was being built out and actually communications infrastructure more generally, internet, cell phone, all this stuff, they had this infrastructure, those cell phone-- they've got to build a lot of towers, got to build them fast.

Jenkins: That's right.

Trimberger: They're going from gen three to gen four or whatever. They're changing. Yeah, they're making money out of the data flowing through it, not out of "I bought it. I sold it for a 50% margin of what I bought it for."

Jenkins: Right.

Trimberger: That's really what made the FPGA a business. And so, by the year 2000, 2002 FPGA were no longer a field programmable gate arrays. We didn't make field programmable gate arrays. Half the business, more than half the business was in communications infrastructure. Those devices are "field programmable communication infrastructure chips."

Jenkins: Yeah.

Trimberger: Now the fact that they could do custom logic for anybody was somewhat incidental.

And so, that's what drove the whole thing through. What happened? Gee, now instead of just being a bunch of arrays of blocks, I've got to have memories because I'm staging the data flowing through it. I have high speed transceivers. Oh, suddenly I've got DSP units because I've got all the stuff coming off a cell phone tower. It's this mass of stuff and no longer is it this a nice model where I can just say "Okay, I'm Boolean Algebra. Do some optimization. Throw it into the bucket. Place it, route it, optimize.." No, I've got one of these blocks. How many logic stripes per DSP stripe? All these questions that made the architecture really complicated, and the answers actually, to be fair, weren't that hard to come to. You just go talk to the customer and they say "Oh, we think we need this much of this and that much of that." Xilinx makes an array about the right size. They get it wrong, they make another family member with a little bit of extra whatever they had left out. That was painful but yeah, that's what was happening. And so, that was the transition. There's also-- these devices were at huge capacity at this point.

Jenkins: Yeah.

Trimberger: Moore's Law had really worked well.

Jenkins: Right.

Trimberger: And so, in the '90s there was this graph of the average ASIC design start versus Moore's Law. The EDA companies said "Well this is the design gap. We can't design as much as we can build." And I looked. This is the average ASIC design start. In 2000 FPGAs passed the hump of that curve. In 2002, a $10 FPGA passed the hump of that curve and in 2004 the hump of that curve wasn't there anymore because FPGAs had eaten the whole bottom half of it. And so, a dramatic change and a change in what you had to produce. So now you've got to produce - you're building whole systems. People are putting multiple subsystems into the FPGA. It's just too big and roomy.

Jenkins: Yeah.

Trimberger: And so, now the question is "Oh, it's not capacity. How fast does it run?" Whole power, yeah we were worried about power. Now it's a prime mover. You used to be able to make money in the FPGA business because you could sell something a little bit bigger than the guy up the street. Now it's got to have the right collection of stuff, at the right power and trying to meet memory interface standards. It turned into a really different business.

Jenkins: So I was going to say around 2002 wasn't it, that Xilinx purchased the CoolRunner 2 team from Philips?

Trimberger: Yeah. I was so involved in the FPGA stuff.

Jenkins: Yeah. It was CPLD and I remember that Ron Cline at that time who was the lead design manager, had made a comment that we found out that people like having low power on the parts that we make, but they won't pay extra for it. And it's just something that I kind of wonder. I haven't been paying that much attention to FPGAs in the last few years, because I'm retired. But the thing is, are they now willing to pay more for a lower power product or can you tell?

Trimberger: Oh, I can't tell.

Jenkins: Or it's just expected.

Trimberger: Yeah, there's a lot of that and the semiconductor vendors conditioned customers to demand more for the same price. And so, the FPGA in the business, was pretty much flat for 10 - 15 years. Flat in revenue. But skyrocketing in delivered number of devices and capacity and stuff. So just delivering more for the same amount of money and willing to accept that.

Jenkins: Is that the dark side of Moore's Law?

Trimberger: It gets right back to what Bernie Vonderschmitt said: “You have to do what adds the value.” And if you're not adding value, they aren't going to pay for it.

Jenkins: Agreed.

Trimberger: And so, there's a hard question for marketing in that industry. Okay, how do you make more money? A few things we know don't work. We know there's no price elasticity. Why? Because every two years we've got twice as many transistors, so it costs half as much and the sales don’t double. We know there's not performance or capacity elasticity because every two years you get twice as many transistors and they run a little bit faster, but sales don't boom. What is it?

Jenkins: Yeah.

Trimberger: And in fact, as you noted, what kicked FPGAs up in the late '90s or early '00 is this market for communications infrastructure. If there's another one coming up then maybe there's another boom. Otherwise the same old customers buying the same old stuff for the same old applications is probably not going to make a big difference.

Jenkins: I often wondered, people always talk about Moore's Law which I always think of as 'Moore's Observation', but I can't think of another industry where there was a yardstick that an investment company could use to predict where you're going to be in two years. You know what I mean?

Trimberger: So I have opened many talks with a graph that-- and it looks— it's exponential and I says: here's this exponential, here's this curve. This is *not* Moore's Law. This is miles of railroad track in the United States from 1820 to 1890. In 70 years it grew exponentially. And we know that didn't continue, because if it did, you couldn't take three steps in this country without tripping over a piece of steel.

Jenkins: Pretty much. Yeah.

Trimberger: And so, there was an industry and you know what? A lot of people made a lot of money in that industry, and all parts of it, right? From the steel makers to track and railroad companies, everything. And it petered out and actually started to peter out before the airplane and the automobile. So, it was a bit more fundamental.

Jenkins: They'd solved the problem basically of availability of transportation as needed, at the time.

Trimberger: Yeah. And so, that was very painful. So there's another example, but it says exponentials don't go on forever.

Jenkins: Yeah. They have to..

Trimberger: But everyone who's predicted the demise of Moore's Law has been wrong so far. But the other point is Moore's Law is not a technological law. It's an economic law that says each generation of technology makes enough money to pay for the next one and to pay off employees, investors, and regulators.

Jenkins: Right.

Trimberger: So, if any one of those gets greedy then that could stop it. And so, what's happening today? Well gee, if there are only three places in the world that can make leading edge technology and they make-- their investors demand a lot more return on that. And if it's not sufficiently improved over what you did last time around then you're not going to get that much money and it may take you longer to make that much money. Even if you do have some new sort of technological advantage.

Jenkins: So we've kind of gotten your career up to the point where you went into advanced product development or whatever. At what point did-- I believe Wim is the one who wanted to use the Hewlett-Packard model and have an advance research labs. And so, Xilinx's Labs came into being, and you're one of the primary players in that area. I wondered when did that all transpire pretty much? When did it sort of become into being?

Trimberger: That's pretty much as soon as Wim arrived. He brought Ivo Bolsens in to run the lab. Actually I'm not sure. Right around that time, - 2000, 2002, somewhere in there.

Jenkins: Okay, and you were quick to transition into that lab as I recall.

Trimberger: Yeah, it was clear that's where I belonged. So, I was no longer on the next product. That was Virtex train. I could join that group, but it was getting pretty restricted, and one of the things you get spoiled with when you come to a startup is a startup doesn't have a lot of people, and so, it's up to you to do a lot of the stuff.

Jenkins: Right.

Trimberger: And so, that played perfectly to me. I loved that. Okay? “Gee, I come in. I was supposed to do software. Well let's do some IC design stuff. Let's do that. Oh, we're looking at-- we're trying to solve all these other problems, IP development.” It's..

Jenkins: It keeps things interesting for you.

Trimberger: And then, okay, security stuff. Nobody's doing that and I can get my head around the whole thing. So, and then by that time Xilinx is big enough that these were getting to be pretty compartmentalized and I had thought that was boring. Besides there was-- I had this list, at least conceptually, go looking back through the notebook of all these cool things to do. Okay, and actually I did. I wrote a document. I called it "Five Technologies That Will Revolutionize FPGAs," and I wrote that up and I presented that to Wim and I presented it to Ivo and they both never commented about it again. But I wound up doing a few of those.

Jenkins: Yeah.

Trimberger: And so, I wanted to do more of these. Some low power stuff with Tim Tuan. Got involved in die stacking. In 2002 Bernie New was doing some of that. I pulled him on and we worked on this together. He had done some other sort of technical analysis. And then we pushed that forward with Arif Rahman.

Jenkins: Right.

Trimberger: And that real early exposure to die stacking. The most interesting part about that was it was a technology that was being talked up a lot in the labs, in various labs around the world. And so, we went out to these guys. Okay, you say you can do this. What can you do? And we went sort of one pass around and collected data and did kind of an architecture analysis.

Jenkins: Right.

Trimberger: Another pass we were doing some collecting. What do people want? What do people think we could build? What would be really good? Oh, it'd be really good to kind of combine dissimilar technologies. That'd be really cool, because you can't do that monolithically. And you'd like to do something that Moore's Law wouldn't just clobber you. I had learned that lesson from Prizm.

Jenkins: Yeah, okay.

Trimberger: Prizm, not prison! Yeah. So we looked at that and we talked to these guys and said "What can you build?" They said "Oh yeah. What do you need?" We said "Well, what about can you do a five micron, through silicon vias with five-micron pitch?" "Yeah, yeah." "Twenty to one aspect ratio?" "Yeah, yeah, yeah." Okay. Great. Then go off and do something. We'd do an architecture and come back and then Wim did the most important part, he said "Okay, here's $700,000. You guys can make some silicon." Okay, great. So then we went back to these guys and said "Okay, five micron pitch, 20 to 1 aspect ratio. We want to fill the die with it." It's going to be the biggest die we could make because the whole reason here is to put more stuff together. There's no reason to put more stuff together if you can just do half of the area. I remember one of these. We laid it on the table and the guy couldn't touch it. He said "Well we're a lab. We can put two of them. We can't put 200,000 of them." Alright? And so, that was the next round. It was okay, what can you *really* do? What can you *really* make a lot of? And that process was like-- it was a phase change - crystallizing where you're freezing or whatever.

Jenkins: Sure.

Trimberger: Okay, we went around again. We did a new architecture and then "Yes, this is something. Oh yeah, we make this." Okay. "Can you thin the wafers?" "Well yes, but I can't thin 12 inch wafers. We've got to core them down." And so, yeah. So that first thing. just forcing this system through, just forcing the technologies to all work. Get the die made. Get the wafers thinned. Get them stacked, vias built, assembled, put together. A lot longer than expected. A lot more painful than expected, but it just forced everybody to actually build something. And yeah, we wound up with something that nothing worked completely, but it demonstrated all the pieces. And it demonstrated some failures. We were putting through silicon vias through active silicon and learned a whole bunch about why you don't want to do that.

Jenkins: Sure.

Trimberger: And a whole bunch-- Well there turned out to be a whole bunch of patents related to how you get around these problems, because you're the first one to find them.

Jenkins: Yep.

Trimberger: And so, that's why when Xilinx's stacked silicon technology came out, they didn't try to put these silicon vias through active silicon because it was too painful. It wasn't going to work. Even though-- And they would rather have the test problem, than the performance issues, variable performance issues. So that was the die stacking stuff. There's a thread that came through with that.

Jenkins: Which is interesting because ultimately it says anything's game, moving forward because if we can successfully stack different technologies and things like that, we don't have to have everything be in one fab and we can mix. We can go get the best of any fabs you want, and go marry them together in good combinations.

Trimberger: And there are people really still trying hard to make that happen and it's really hard because that's really a big die for all of the yield issues, they're cheaper than a bunch of little die. I mean it's as expensive to assemble these together. Now Xilinx wound up not mixing these dissimilar technologies but deploying the technology to make four little die instead of one enormous one, and the advantage of the yield improvement was so great that it was greater than the pretty expensive cost of the assembly and the yield loss in the assembly and so on. And so, there's a lesson in how to make this cost effective, but that was something. At that point, that wasn't in my domain anymore.

Jenkins: Right.

Trimberger: I said "Okay look, I've demonstrated all of these technologies including a couple you might not want to use." And that's kind of my-- I use it. I call it the Loony Tunes version of how to do technology development, not because it's loony but because remember the Road Runner would run off the cliff and quick step back before he fell. So that's technology. You run off the cliff and quick step back before you fall.

Jenkins: Don't look down.

Trimberger: Don't look down. So that was the style of project that I was taking at that point. Just how do we extend in kind of a new way? And so, Wim supported that and then when Wim was gone and we got a new CEO…

Jenkins: Moshe.

Trimberger: Moshe, yeah. Gee, I'm so bad. I've only been gone six weeks and I can't remember. So Moshe had a different vision. And he said "No. We have one product road map. We're not going to be wasting money and time doing all these other screwy things. One road map, and do something on the road map." And so, then I instantly switched from becoming the most product oriented guy in the lab to being the most vision oriented guy in the lab. Because everyone else said "Nope. We're going to jump on doing the next thing," and I kept doing sort of the stuff I was doing, kind of the direction I was going.

Jenkins: Most of the choices that you've made had to do with areas that focused on your interests and passions. In other words, you went all the way back to high school where these clowns were coming in and telling you "Let's go march on Sacramento," or whatever it is and finally you were listening to somebody who wanted to get something done. That's sort of like a common thread that I'm sort of perceiving in your career, if you will, and it's good to see some level of consistency. At least in my viewpoint, in that way.

We've probably been on this discussion quite a bit and I know the last 15 years or so you've spent doing a lot of very interesting things with the labs. You've maintained your outside contacts. Actually you've expanded your relationships with outside academia because I mean you've been going to all these different conferences and probably made sure that various professors got funding they needed to do something that was going to be beneficial. I suspect that you're one of the hidden hands behind the agenda on some of those.

Trimberger: That was actually one of the other roles I had earlier at Xilinx because when I got to Xilinx there was nobody doing any of that. And so, okay well, okay. I can do that. I was going to conferences and meeting the professors and talked up the problems that I think they ought to be solving for us. To be fair, if there’s interesting stuff they should be able to get funding for it. I actually managed the Xilinx's academic funding for awhile and funded some pretty important people, like Jason Cong. He told me later the money he got from Xilinx, that he got from me, was the very first grant he ever won.

Jenkins: Oh my. That's great.

Trimberger: So I feel really good about that. I know how to pick somebody's who's good. So, yeah. That's another thing. Okay, nobody else is doing that. I can do that. How much budget do I have for this?

Jenkins: Sure.

Trimberger: And seed some money in some good places and get some nice work done, and then through the years more and more FPGAs became a legitimate place to do work. Which was really nice in academia. And so, then it was a question of again just sort of "Gee, can we keep these guys focused on the right problem to work on without--" I mean maybe it's not fair for what I did but I didn't really want them to start competitors. I didn't want these guys to work in academia for three years and have a clever idea for a better FPGA and go off and build it. And so..

Jenkins: You'd rather hire them at Xilinx.

Trimberger: I'd rather hire them at Xilinx and have them build it for me, or better yet from the git-go, instead of investigating some other screwy thing they ought to be doing that communication network study. And so, compared to the best possible how good are the kinds of things we build today? The other thing that was happening and this happened all through this is FPGAs were invented in industry and industry always has had a lead over academia in architectural structures and it kind of surprises me but kind of not. I mean it takes a lot to be in the FPGA business. When FPGA started out, and we started doing software as well as silicon, okay we took on ASIC and EDA. If you want to compete you've got to do both. Oh, and you've got to do IP. You've got to do libraries. Oh yeah, you've got to-- there's a lot you've got to do. Well, if you're in academia and you want to get into the FPGA business you've got to do all that stuff.

Jenkins: Yep.

Trimberger: And so, it's a high bar for academia as well, especially since the design part is probably in the computer-- in electrical engineering and maybe computer engineering and the software side might be in computer science. Or, it might not. I'm not even sure. And so, that requires more collaboration than you might ordinarily get. So it's been very effective. It's been effective to keep the-- It's a hard business to get in. The bar is high to keep the explosion of competition out.

Jenkins: So, I think you've been fortunate but I also think you were the right guy at the right time and place for a bunch of this with the right attitude. In other words, that attitude of "Yes, let's try it." Saying no says it's over. You know what I mean?

Trimberger: Yeah.

Jenkins: At least you had a shot, and you had mentioned earlier that you were there for an IPO at VLSI but heaven's, you were there for an IPO at Xilinx, too.

Trimberger: That's right.

Jenkins: So I'm aware of the thing that exists that you called the Trimberger Foundation, your family foundation or whatever it is, and I know that you are involved in some community-type things. At one point I think you were doing something with a website and soccer or something like that. Are you still doing anything like with your family trust and all that?

Trimberger: Yeah. So, I was real fortunate to be at some startups and Xilinx is an exceptional company. I mean even as startups go it was spectacular to ride that. So, by the late 1990s when my Xilinx options needed to be exercised, I started looking at that and said "Okay. Well this is more money than I'll ever need. What should I be doing?" And I simultaneously looked at estate planning. How do I keep it out of the hands of the government? So yeah, I ran into a few individuals who were saying "Look, here's a possibility. You could start your own charitable foundation." I said OK, I had sort of not thought about that too much.

Jenkins: Sure.

Trimberger: I said "Okay. Okay. Let's try. Yeah, let's do that." It wasn't quite that easy. Because what it says is you're taking a large amount of money and giving it away. Giving it away to an organization. You can control what it does with it, but it's still not your money. So yeah, in 1999 Laura and I founded the foundation. What we did was that was funded with a chunk of that Xilinx stock. And so, that went straight to the foundation, and then we looked at what do we give money to today? Gee, what would the foundation do? Well what do we do now? And so, we identified two major areas where we were giving our money donations and spending time.

Jenkins: Right.

Trimberger: One was amateur athletics and the other was appreciation and understanding of science and technology. So these are two areas I thought were really important.

Jenkins: I was I don't know, heartened, I guess, having recently interviewed with Don Faria to find out about his commitment to doing wheelchair soccer. Which-- that's what a thread I saw and he was telling how much it had changed the lives of people there. And still I think it's good that people who come into wealth and all that, do take the time and sit down and figure out how is the best way to manage this. I know that you have, what, one son and one daughter or is it?

Trimberger: One son and two daughters.

Jenkins: One son and two daughters. Your son, as I recollect, was an Eagle Scout.

Trimberger: Oh, and the way the Eagle Scout works, you're once an Eagle Scout, you're always an Eagle Scout. So the way that-- it's not past tense. He is an Eagle Scout.

Jenkins: He is an Eagle.

Trimberger: ...an Eagle Scout!

Jenkins: Okay, good. Were you in Scouts?

Trimberger: I was in Scouts. I didn't get nearly that far. I got to second class Scout.

Jenkins: Yeah. I think I got Tenderfoot.

Trimberger: And so, this is one of those amateur athletic kinds of things. I was involved. I got involved in Scouting. Got involved in the soccer league and things like that. And kind of under the umbrella of the foundation, did some of the science and technology stuff in Scouting too, but actually his involvement, my son's involvement in Scouting let me re-live a lot of that. So I got pretty active in Scouting and enjoyed that.

Jenkins: You relearn some of those knots that you'd forgotten, didn’t you?

Trimberger: And some of them I never even knew. I managed to fumble through it at one point and look at that. It still held together. But yeah, some of the first aid stuff and again looking back, a lot of this stuff it's that nonfiction. Right? So okay, a lot of Scouting stuff starts with first aid and there's camping. I always liked being able to do stuff - have the ability to do something. My dad took us camping. We all learned that and pitching a tent and where and why and how and all that stuff. So I really enjoyed that. So then to go through that in Scouting and they do a lot, right? So I got to do that. I got to do the wilderness survival, spend a night in the wilderness without a tent or sleeping bag or anything.

Jenkins: With a jack knife I hope?

Trimberger: No. Well it's whatever you're carrying, whatever you would be hiking with, right, but yeah. If you have to get into a fight with a bear don't count on that knife to get you out of it. But I got to do that because my son was doing it.

Jenkins: You're going to McGuyver it, as they say.

Trimberger: Yeah, so we had a lot of fun with that and again, learn some things that I didn't know about a lot of different subject areas.

Jenkins: Now as I recall while you were still at Xilinx you were I think active with science fairs or some of the robotic challenges and things, right?

Trimberger: Yeah, both actually.

Jenkins: Okay.

Trimberger: So, as well as science fairs for a few years. I got to admit some of these projects were just astounding and some of them were pretty boring, but some of these are just astounding and I got kind of cooled to it. Look, here's somebody who's sequenced the DNA of broccoli or something. I said "My God, she's working at a national lab. Of course she's got all the equipment there and she brings the broccoli one day and does it." But you know, certainly it is head and shoulders over the kid who swabbed the sample of the doorknob in the lady's room or whatever. But it struck me that there's a problem. I really got turned off. These top rated projects were so over the top that we're losing the middle.

Jenkins: Right.

Trimberger: Yeah. So that kind of frustrated me. There was an article once in the Caltech alumni newsletter and it said "are we training or are we selecting?" And so, with the Foundation I'm more interested in training and bringing more up than selecting the top off. Right? So yes, gee, we could give a scholarship, right? Scholarship fund, come to us, and you know what? The top two percent would get another few thousand bucks. Maybe that's not what I want to do, and that's something you can do when you have your own charity. You say "How do I make a difference?"

Jenkins: Yep. You were going back to that high school situation you had.

Trimberger: How do I make a difference? So yeah, one of the places we support most, the SETI Institute here in Mountain View has a radio broadcast podcast called "Big Picture Science" and it's a weekly thing and talk about big issues in science, and there's a particular segment on critical thinking and that I really do appreciate because the world needs more, better critical thinking. Too many people are suckered by Ponzi schemes or whatever it is. Or just don't know the difference between a million, a billion, and a trillion when talking about national debt, right?

Jenkins: Right.

Trimberger: It's really kind of important.

Jenkins: Yes it is.

Trimberger: So okay, can I put the money there as opposed to "Gee, do I give Caltech another few thousand dollars against their $2 billion endowment?" Not going to make much difference there.

Jenkins: Right.

Trimberger: Love the place. But how do I make a difference?

Jenkins: Yeah, so finding where that leverage is going to make a difference.

Trimberger: Yeah and then it's not just going out and spreading money around, which was another one of those things in high school. I said okay, what am I doing? How can I participate in this? So yeah, getting involved in The Computer History Museum is one of those things. Okay, well I'll put some sweat into this. I'll actually do something.

Jenkins: Yeah, sweat equity as they call it.

Trimberger: Yeah and just how committed are you to this? Are you committed enough to give some money? Okay, this is the usual Hollywood type commitment. Oh yeah. Or I'm actually going to do something? And actually do something. So that's appealing to me and it gets harder and harder to do every year. It's just there's a lot I'm trying to do. And it's just really hard, but there's this other-- I told people there's no such thing as lack of resource. It's priorities, it's only priority.

Yeah. Sorry mom. I didn't have time to call you. No. Mom, you were low enough on my priority I didn't call you. You never say it that way but that's what happened, and as long as you-- Don't fool yourself, right?

Jenkins: Can we get that edited out so his mom doesn't see that?

Trimberger: No. I called my mom. You should, too. It's one of those things that no one can do for you. And so, that's something that you must do.

Jenkins: And they won't be here forever.

<seems like some transition is missing here> <this belongs elsewhere>

Trimberger: Yeah. So anyway, for the teens, 2000 teens I did a few projects like that, looked at Moore's Law and then a variety of interesting things you can do inside the package. Looking ahead and saying "Gee, if and when Moore's Law ends, what would we like to have in the bin for pizza for that?" And so, okay. This die stacking stuff. What else could we be doing in here? How does it end? It's not a brick wall and suddenly you slam into it. In 1890 they didn't stop making railroad track. It just didn't end. They didn't run into a wall.

Jenkins: Yeah.

Trimberger: Something different happened. What was it and can we get started on that early? That was the other lesson of what is advanced development do? Identify long lead time items and get started early.

Jenkins: Good point.

Trimberger: And you can..

Jenkins: We've got to start training the children now so that they'll be ready to go to college in the right fields at the right time.

Trimberger: You know, training children to leave the home —-- Not a lot of parents realize that their job is to become superfluous. And that's something that Laura and I did. We did sort of consciously. It was a training exercise.

Jenkins: Oh yeah.

Trimberger: And not all of our kids' friends were able to make that transition to college and say "Okay. I happen to be at a new place and I'm not going crazy and not getting drunk everyday or whatever." Okay, here I am and these people are acting really strange, mom. I don't understand it. But that was another thing that's sort of the lesson of being a parent. You never have enough time to tell your children everything you want to tell them.

Jenkins: True.

Trimberger: And so, when my kids were in high school when I realized that I was pretty much out of time but there was still a lot I wanted to tell them. Gee, one way would be to lock them in a room and sit down and lecture them. But instead I wrote a book. I wrote a book and the title is "Things I Know" and I just started collecting these thoughts and things. And then I wound up collecting them in chapters or whatever and I wrote a book. And so, I gave it to them - made four copies. Kids each got one, I get one. And so, it's like 300 pages. I must know a lot. I used a big font. So yeah, all those little things that -- okay, well maybe someday they'll go back and read it.

Jenkins: Yeah. That's fun Steve.

Trimberger: Yeah. We'll see if that works.

Jenkins: I think we could go on forever talking about your career in FPGAs and so forth, but I also think that we've got a lot of time here on videotape and we need to wind it down at some point. So we'd like to say if you had some advice that you'd want to give to a person new and getting started, what would it be? Can you think of something?

Trimberger: You know, advice is so easy. It's unfair, but the one I say is "Say yes." So join in. Say yes. Try it.

Jenkins: Yeah. And like Nike, go for it.

Trimberger: It's go for it. Say yes and then learn as fast as you can and, yeah, sure. Make yourself prepared but give yourself opportunities. Whether it was oh yeah, I was out of college. I was looking for a job but I also applied to graduate school, and then multiple of both. I didn't know where I was going to land. And yeah, going to Caltech. Thought I was going to Stanford. Two weeks ahead of time I went someplace else. Okay. I had options, good options. Don't fret those.

Jenkins: Yeah.

Trimberger: So that would be the big pieces. Other pieces of advice, so as you pointed out, you know I had a lot of interests. I didn't pursue them all, right? So I didn't-- I really like this cryptography thing. I didn't become a cryptographer. But I didn't lose it. I hung on to that. I grew up in the '60s, '70s, space race. God, that was so spectacular. I loved that. You know, honestly I thought astronomy, that must be my thing. Went to Caltech. Why? Jet propulsion laboratory and Hale Observatories Palomar. I wanted to do that. I didn't become an astronomer.

Jenkins: No.

Trimberger: But I didn't lose it. I love that stuff. And so, what's happening now? Okay, SETI Institute. Still involved, right?

Jenkins: It's still part of your world.

Trimberger: Now, I actually did that astronomy stuff at Caltech. So yeah, it's okay to love something a lot and not make it your career. And surprise, surprise, you get another chance. The loves come back and it's still great. So I'd say those are-- Just say yes. Give yourself opportunities, and don't be afraid to turn your passion into a hobby while you're really working.

Jenkins: Thank you.

END OF THE INTERVIEW

