Oral History of Jim Keller

Interviewed by:
Douglas Fairbairn

Recorded July 8, 2024

CHM Reference number: 2024.0071

Fairbairn: Okay. So, we're ready to begin. My name is Doug Fairbairn. We're here to interview Jim Keller, famous CPU architect and contributor to many processor designs over the last several decades, among other contributions and achievements and so, Jim, welcome and we're delighted to have you.

Keller: All right. Thanks. Great to be here.

Fairbairn: So, as I said, let's just start at the beginning. Tell me where you were born, when you were born, what your early family life was like, what were your parents doing. What was the family environment as you were growing up?

Keller: Yeah. So, I was born in New Jersey, and my father worked at GE Aerospace and they had a program where they'd move them around. So, he started in Indiana and then Lynn, Massachusetts, and I think Philadelphia and then they built a big plant in Valley Forge, Pennsylvania. We literally lived three-minute drive from the Valley Forge GE Aerospace office. I was one of six kids, the second, and my father was a mechanical engineer. He designed satellites. My mother was a valedictorian of her college. She had trained to be a teacher. She was very smart, but she raised six children, and she was not working when I was a kid, for the most part. Later, she became a therapist and was successful at that. But as a kid, she was a stay-at-home mom, and I grew up in a house full of kids. There were six of us, relatively close together.

Fairbairn: So, the fact that your father had an engineering background, was that an influence? Did he work with you? Did you do projects together? Did you do anything like that?

Keller: Yeah. My father was very good at describing and drawing anything. So, I grew up in a house where he'd come home and say "Oh, I saw this bridge, let me sketch it out for you," and he would quickly sketch out the whole bridge and then tell me why it worked and didn't work. I sort of grew up thinking that's how things were and my parents were both members of a book club. There were books all over the house and they believed in intellectual conversations. They were pretty smart people, generally speaking, and they had a family friend who talked to me on a family trip. The kids swapped cars and apparently, I talked to them for hours.

My father's friend told him that I was one of the smartest kids he'd ever talked to. The reason I mention this, because then I was dyslexic apparently, later diagnosis, and I didn't really learn to read until third or fourth grade and my parents never worried about me. So, they thought "Oh, he's fine, he's just figuring it out his own way." I think in the modern world, that would have led to panic attacks. Your son in second grade still can't read. But it never impacted me. I noticed in second grade that I was in the group with the books with the really big letters instead of the smaller letters and it just didn't impact me very much. Then periodically, I would learn. I remember in the end of third grade, I learned to read and suddenly, I could read books and books were fun and I read a whole bunch of books.

Fairbairn: So, it slowed you down in terms of learning to read, but you became an avid reader.

Keller: I became a reader, never a writer. I struggle with writing. I can talk mostly a blue streak. But if I can see the whole page of words, I can write them down and I think maybe in a world of text editing and writing and rewriting, it would be better. But as a kid, writing was always very painful. But math was also-- they call it episodic learning. I remember we were learning long division. I didn't get it and then one day it was like "Oh, they're guessing," and it was like "Bang, I got it," and then the same thing happened in algebra, like "F of X equals something." I didn't know what they were talking about. I think I went the whole year of algebra just thinking "I don't know what they're talking about." It was like "Oh, it's symbols," and then I just got it. So, I'd have these moments. But I was doing my own thing as a kid and school was time off from doing other things.

Fairbairn: So, you have five other siblings. Did any of them follow a technical track?

Keller: My older sister was a mechanical engineer. My younger brother is a computer scientist and very, very good and my siblings were all pretty smart. But there's a therapist, a sociologist, a philosophy professor-- we did fairly different things.

Fairbairn: Interesting family. So, you're progressing through high school. Anybody in high school, particular teachers or whatever, that steered you in a particular direction or got you interested in things?

Keller: Yeah. I had a couple of really good teachers. I went to a fairly large suburban high school in the Philadelphia area. We had a math teacher who taught calculus and for special presentations, he'd put on his good suit and his brightly colored tie and make a really big deal out of it. So, the fundamental theorem of integral calculus was an event. There was multicolored chalk and he was a character and some teachers in our school were basically terrible and we didn't do anything. In our school, we sort of ran amok. Attendance was optional and my friend was a student class president and our class made lots of money on concessions for football games. We kind of saw ourselves as operating outside of the school. It was kind of a wild place for a kid. The principal subsequently was fired and my brother said "I went to a way different high school than you did."

Fairbairn: Had to rein these students in, huh?

Keller: Yeah. So, the end of the 60s was pretty tumultuous in this country and I grew up watching television to find out if my uncles had died in the Vietnam War and it was pretty traumatic, I'd say. Then the 70s were kind of the aftermath and is when I went to high school and it was both good and bad. I had, like I said, some spectacular-- my geometry teacher was great. Two of my chemistry teachers-- Mrs. Anna Furco, I still remember teaching us stoichiometry. She was a fantastic teacher and then other ones were like-- they didn't care. We didn't show up. Nobody seemed to notice. I graduated high school with B's with literally no plan except for a girl I liked up the street was going to Penn State. So, I applied.

Fairbairn: Ah, that was the path to Penn State, huh?

Keller: Well, yeah, there was a funny event. I got called by the student counselor in my senior year. Back then, somewhere at the end of your senior year, somebody might notice whether you're going to college and nowadays they start prepping in elementary school.

Fairbairn: Years ahead of time.

Keller: She said "You're a really smart kid. You should go to MIT. Have you ever heard of MIT?" I think I vaguely had heard of MIT. I thought "Nah, I don't want to move to Boston," and then she gave me a list of jobs sorted by salary. At the time, the top college graduate salary was chemical engineering, then petroleum engineering, then mechanical, and then electrical and I thought, the first two seemed dumb. So, I flipped a coin between mechanical and electrical engineering. But my grandfather was an electrician and as a kid, when we'd visit him, his house was full of electric motors and washing machines and all kinds of crap. I thought it was amazing so, I randomly picked electrical engineering because it was high paid.

I drew a line because below the halfway line was biology, physics, oceanography. If you'd asked me what I wanted to do in college, every single one of them was below the line. But I thought it would be dumb to go to college and not have a job that could pay any money, and my grandfather was an electrician, and I thought that would be funny. So, I signed up for Penn State. I got accepted to electrical engineering. I called my grandfather, and he said, "Watch your power factor," which is the angle between current and voltage in an electric motor. So, that was pretty much the whole story about going into engineering.

Fairbairn: How you get there. So, when you took EE, was it microelectronics focused? Was it motors and...

Keller: No, no. It's...

Fairbairn: What was the focus of the education?

Keller: I had one Fortran course. I had one logic design course. Penn State at the time-- this is 1976-- they had a very good EE department. It was like 600 or 700 students-ish.

Fairbairn: Big.

Keller: Yeah, big. They started the core curriculum. Your first year was math, physics, electives. I was also a dual major in philosophy, which was a funny story, and math was great. Mechanical engineering was-- those were the weed-out courses, and I remember learning to solve accelerating, rotating reference frames, and I thought that was amazing and I really liked college. I'd say the first year was, it took me a while to get up to speed. Because, in high school, we just messed around all the time. While I had good teachers, we didn't have to do any serious studying. And the first year I started out strong, but by the end of the year, my grades were in the pits and my father said if I wanted to fail college, I should really think hard about that in the summer. Because rather than waste money on school, I could go get a job and I thought "I'm not going to fail college." So, I did pretty well the rest of the...

Fairbairn: So, how did you get into philosophy?

Keller: I read a lot of books as a kid and I was really interested in ideas in general and so, I was a double major and then the funny story about that is I got a-- like in my second year, I got a letter, a note sent to the electrical engineering department from the head of the philosophy department that they wanted to meet me and I went in there. I was kind of excited. It was like "Wow, the head of the department called," and he said "We've never had a student like you, like literally. There are no EE's for the philosophy major and then I just wanted to show you something." He shows me a paper written by a philosophy student, which is like-- he said "This is a midterm exam." It was like five pages, nicely written, double spaced, both sides and then he pulled out my test result, which was literally a half a page with multiple scratchings out. He said "I can't even read it, Jim. I don't know what you said," and he said "We write a lot as philosophy students, and I check with your teachers. They think you're great. But I have to tell you, you'll never get a degree in philosophy from Penn State University. But you can take any course you want. You'll always get B's." So, I got kicked out of philosophy because I couldn't write a paragraph in a test environment. So, it was pretty funny. But I took lots of classes and it was really an interesting thing to do.

Fairbairn: So, that's a pretty interesting story. So, you finished up there. Do you have any idea what you wanted to do afterwards? How did you...?

Keller: Yeah. So, at Penn State I didn't really start electrical engineering until the second year because that's how the curriculum worked. I got straight A's in EE. I got a call, actually, after the philosophy professor, from the head of the department in EE and I thought "Oh, shit, I'm getting kicked out of EE as well. Like, this can't happen twice." But he said "I just wanted to meet you. We don't have that many people who get straight A's at Penn State." And I said "Well, I'm working really hard at it," and he goes "That's obvious." So, the thing I really liked was electromagnetic fields. But then my advisor was Dr. Joe Stack, who's head of the semiconductor group and IBM had donated to Penn State a two-inch wafer fab. So, my senior classes, I took a bunch-- yeah, little tiny things. Well, we laid out transistors on vellum with tape.

Fairbairn: Oh, yeah, I know.

Keller: It was like six-micron transistors. They were big and we made transistors in the lab. I still remember the safety briefing on HF (Hydrogen fluoride). So, my senior courses were semiconductor focused. The thing I thought was I was going to be an electrical like RF engineer. But that was really interesting and then I took a programming course, which I thought was really fun. It was punch-card Fortran, and I had a logic design course, which I thought was curious.

Fairbairn: Curious?

Keller: It didn't go very deep. But it was amazing that you could build anything out of something so simple. So electromagnetic field theory is complicated and almost everything becomes almost incomputable, right? Even with relatively simple antennas, with relatively simple driving things, it becomes very difficult and nowadays, we build really elaborate simulators for that kind of stuff. So, the simple equations are cool. But the application quickly runs into the computational wall and material science was worse. It was my favorite class at Penn State because the professor was great, and the descriptive language of material science was amazing. But in the books, all the problems became curve fits to simple equations because everything outside hydrogen atom is insolvable and it was really an interesting-- again, as a philosophical basis for science. It's a very curious thing how big the leap is from the foundations of physics or something and the application of it.

And then here's logic design, which is the leap from logic design to the application of it is a Karnaugh map, which is trivial and then it's really kind of wild. So, I wasn't really sure. So, I interviewed for a whole bunch of jobs, mostly in semiconductor companies-- Fairchild, Motorola. But I randomly did an interview at Harris in Florida and I thought I could literally live on the beach and go surfing. And they had a really cool computer project that was doing fiber optics combined with writing data on film at high data rates and they needed a logic designer, basically, and so, I took that job.

Fairbairn: So, between the environment and the job, it seemed like a good place to go.

Keller: Yeah. Then the boss was a reader and he had a bookshelf full of books I had read as a kid and we had a great discussion. Then it turns out he was subsequently fired after they hired me and when I showed up, they literally had nothing to do with me. They were like "Why are you here?" It's like I had a letter that says, "Show up August 15th," and they go "Well, the college students all started in June." It was like "Yeah, but I drove across country and back." I negotiated that with the recruiter and they were like "We don't know." They gave me a badge and a desk and then after a couple days, somebody wandered around and said, "We have a whole bunch of test equipment to fix." So, then I spent about a year and a half fixing stuff and it turned out to be great.

Fairbairn: So, you never got close to computer design there?

Keller: Well, they had some boards with microprocessors that didn't work. They had a power controller for Florida Power Company that would remotely control that that didn't work. They had a whole bunch of random projects, and I was a busy kid.

Fairbairn: So, were these projects that somebody had designed but still didn't work? They couldn't figure it out?

Keller: Yeah. I got put in a group where they had multiple basically logic design problems to do and I did a little bit of logic design, but mostly I fixed stuff. I did one high-frequency board at the time, 40-megahertz STTL (Schottky TTL) board that encoded data for a fiber optic transmitter. That didn't work. So, that was like my first electrical engineering job, where we had to fix the ground plane and a whole bunch of stuff to make it work. That job is why I got hired at Digital Equipment.

Fairbairn: So, how did you find your way to Digital Equipment? You were down there surfing in Florida and...

Keller: There was an ad in the EE Times. I wanted to move back north. My family's in Philadelphia. Boston seemed close enough. So, I read the back of the EE Times and there was an advertisement for a clock designer at a computer company I'd never heard of, Digital Equipment. So, I got an interview at Raytheon the day before to practice because I hadn't interviewed for a while and they gave me an offer on the spot, which I turned down. They were pissed about it because they paid for my plane ticket.

Fairbairn: They paid for your trip.

Keller: Yeah. It was pretty funny and then they were like "Did you have any intention of working here?" I said "Yeah, if you guys weren't being like you were." So, it was pretty funny and then I went to Digital, where they needed a high-frequency clock designer and they quickly determined I didn’t know that much about it, but they hired me anyway and that was great.

Fairbairn: So, what group did they stick you in?

Keller: It was the VAX 8800 group. So, the lead architect was Bob Stewart and he was the architect on the [PDP] 11/44, 11/70, VAX 780, and 8800. So, he was one of their best computer architects. I worked for a guy doing the cache subsystem for that computer doing logic design and yeah, I told...

Fairbairn: So, you didn't know anything really about computer architecture or caches or anything like that?

Keller: No. So a friend of mine gave me the 11/70 and 780 manuals the day before the plane flight and I read them on the plane. I basically went in and said-- every single person I talked to, I said "I had a lot of questions," and I basically had the manuals and I said "Why did you do this and what's this for?" I was, let's say, fairly confident as a kid and they basically said, "We don't know what to do with this kid," but they'd never seen anything like it, apparently. So, I got a job there, and it was great. I worked for that group for about seven years.

Fairbairn: On the [VAX]8800?

Keller: The 8800 and then there was two follow-ons, which unfortunately got canceled, one for political reasons and one because the technology didn't work out. But I did quite well doing logic design and then board design. They were ECL computers. So, there was electrical design, board design, logic design, and somewhere in there I started thinking about computer architecture and CAD tools and I wrote a logic simulator, a timing verifier, as fairly large software projects kind of at night, literally, because it wasn't my job.

Fairbairn: Did they have anything like that?

Keller: Yeah. We bought-- there was a startup in Silicon Valley called Valid. They did a drawing system that had associated with a timing verifier. It worked okay, but it was difficult to use. Then Digital had a simulator group that was very thorough but very slow and at some point along the way-- so, I worked for--

Jay Grady was the manager, but Bob Stewart was the architect and Doug Clark was another architect. These guys knew all the-- Digital had a fairly big research community, and they introduced me to people like Forest Baskett and Butler Lampson. We're explaining to Butler how the logic simulator worked and he said "That's crazy. Here's how you should do it," and he stood up and wrote down on the board "Here's what you should do. If you do this, it will be way faster." So, I wrote a logic simulator that did exactly what he said and called him up and told him how much faster it was. He said "That sounds about right. Good work."

Fairbairn: That's Butler.

Keller: Yeah. He was-- yeah, I don't think he did a lot of implementation, but he could think really fast.

Fairbairn: No, he told everybody how it should be done, and everybody went off and did it like he said.

Keller: Yeah. It was literally exactly what happened. I also got to meet Chuck Thacker, who was another great architect and they kind of kept an eye on me because I was a little bit special. But I got to meet those guys when they were...

Fairbairn: Yeah. They were at Xerox PARC when I was there working on the Alto.

Keller: Yeah. So, Chuck and Butler were at System Research Lab, and then Jeremy Dion and Alan Eustace and a bunch of guys I met were at the Western Research Lab, which are literally across the street from each other. But they were like the two factions of research at Digital and I somehow got to know both of them.

Fairbairn: Did you work with Dan Dobberpuhl?

Keller: Yes.

Fairbairn: Was he involved directly or how was-- what was your relationship?

Keller: So, I was at Digital about 15 years. The first half was in a mid-range systems group, which was ECL design, big computers and then I joined the semiconductor group in about '89 or '90. So, I spent six months at Western Research Lab, where I met Norm Jouppi and then worked with Alan Eustace and a couple people on the BIPS computer, which was...

Fairbairn: Out here, out in Silicon Valley?

Keller: Yeah, in Palo Alto and then when I went back, Pete Bannon took a job in the semiconductor group and he said, "You should come here," and we became co-architects of EV5, which was the second Alpha chip. So, Dan did the first Alpha chip, EV4, and we did the follow-on and then Dan, at some point, moved to California and started the ARM team for Digital, which was a StrongARM team. So, I knew him pretty well. We talked a lot. But I didn't work with him directly at Digital.

Fairbairn: He was one that we tried diligently to do an oral history with and never succeeded before he passed away. So, I was curious about your remembrance of him and what his...

Keller: Yeah. He was kind of amazing. So, when the first Alpha chip came out, it was 166 and then 200 megahertz, which was literally three to four times faster than any other clock frequency of a processor ever built before. Dan's desk was always about this big and it was covered with schematics about this high and it was like "How do you find anything?" He said, "If you don't touch it, I can find anything," and he would pull out a schematic from one year ago and say "Here's how you do this, here's how you do this," and yeah.

Fairbairn: So, getting back to your projects initially on the ECL-8800, what was the biggest project that you were responsible for? Did you lead a certain area or what was your...

Keller: Yeah. So, on the ECL-8800, I was literally a board designer, logic designer. So, I did the bus interface, and I built logic to specs that other people did and then I started working with one of the teams that did the performance model architecture because Pete and I became the guys who debugged the computer.

Fairbairn: Pete...

Keller: Pete Bannon. So, he was a microcoder. I was a logic designer. When we powered the machines on in the lab, lots of people were there, but we basically lived in the lab for a year and debugged everything and ran all the software and as the software started coming up, the people that did performance modeling would want to try things and run it and then some things worked like we thought and some things definitely did not. Pete rewrote a lot of firmware and then in that process, I learned a lot about how computers were actually built end-to-end and the guy who was doing one of the IO boards quit and they needed somebody. So, I just went through the whole schematic set and became the guy who debugged that and then we delivered machines inside the company, like the first ten machines that were installed in the operating system group. As soon as they brought up that software and they started loading them, they started to crash and we debugged that.

Fairbairn: So, what was your learning-- what did you take away from those few years on the 8800 project in terms of later directions? You became pretty familiar with the whole architecture of the machine, what worked, what didn't. Did you come away with any opinions about "This is the right way to do it. This is the wrong way to do it. Do we need better tools?" What were the takeaways from that project?

Keller: Yeah. So, at first, it was just a whole bunch of parts. Like, "Here's a piece, here's a piece, here's a piece." But then it started-- like Bob Stewart built-- he built basically a RISC pipeline in a CISC machine early. So, we had a decode stage, instruction fetch, decode, register, read, execute, data cache lookup write-back. So, the VAX-8800 had a pretty clean RISC pipeline with an extra stage or two in the beginning to decode the CISC instructions but executed them as micro-ops. So, first, like he architected it. I didn't know what that meant. But by the time we finished the project, I knew what pipelining was. I knew what the cost of branch mispredicts were, because back then we didn't have a branch predictor. So, when it went the wrong way, you just flushed the pipe. What a cache was. So, basically, it was like architecture 101 for three years. By the time I was done, I knew how caches worked, I knew how branches worked, instruction decoders. VAXs were kind of complicated because we had multiple levels of privilege architecture. They took interrupts. We had microcode doing some of the heavy instructions. So, that all worked.

Then the following was Argonaut, which was take the basic architecture and double the clock rate. From that, I learned we pushed the clock rate further than the technology wanted it to go. It made it really hard to build and also, that the tools to do the analysis were really terrible. So, that's when I got involved in writing CAD tools. At first, it was a timing verifier and then the simulator, which we then used on multiple projects. By the end, I hadn't written a performance model, but we did this performance project, where we took one of the boards of the VAX-8800, modified it to record all the micro-PC addresses and some other information. Then Doug Clark wrote a paper, which was "Here is how a VAX computer spends its time," and I contributed. I built the hardware for that. Pete wrote the microcode. Doug did the high-level analysis and that was sort of like translated, all these moving parts into a computer with pipeline features to a computer that has performance because of how it spends its time. So, when I joined the Alpha team, Pete and I wrote the performance model for EV5 from scratch by ourselves and that was our first experience on doing that as a computer design from an architecture spec and a performance model first that translated into a chip.

Fairbairn: During this time, was there much interaction with the compiler group?

Keller: Some. So, the compiler group was very focused on high-quality compilation. So, there's an interesting journey. So, during the CISC era, at conferences, people would show up and say "My computer's better because I have more instructions," right? Then the compiler era was "Our compiler is better because it's reliable. It's solid. It doesn't have memory leak."

Then the RISC guys showed up and they built Titan, which was a RISC machine at SRC, at System Research Lab, sort of at the end of the VAX-8800. It was half the boards, a quarter the volume, and faster than the VAX-8800 per processor. It was also simpler. It didn't have the privileged architecture. It wasn't a multiprocessor and it was like "This computer is so simple. It's easier to build compilers for it," and the compilers are contributing to performance. That was a really big transitional moment, because the compiler, up until then, weren't the guys delivering performance. They were delivering functionality and quality. When we transitioned to Alpha, the Alpha team had an internal research group that did compiler performance work and then the compiler group at some point created a pipeline from performance ideas into production compilers. But that took years to do at Digital and ultimately very successful.

Fairbairn: So, your observations of the Titan and the results that came from that later influenced your own direction and being able to see both?

Keller: Yeah. It's kind of hard to say. So, Bob Stewart believed in making hardware as simple as possible and Titan was definitely up that alley. But when we got to EV-- like, EV4 was dual-issue, we'd call it super pipeline today.

Fairbairn: EV4?

Keller: EV4 was the first Alpha chip.

Fairbairn: Okay.

Keller: And then EV5 was four-wide, a little bit more pipelining, had a multi-level on-chip cache, which was probably a mistake. But to fill up a four-wide in-order machine, you really needed the compiler to work and then this is where the debate started about was compiler going to generate more and more performance or was the hardware going to do it?

So, EV6 was the third Alpha chip. Dirk Meyer and I architected that and that was our first out-of-order machine, and the premise was software could take you, let's say, a baseline performance of 1x. With compiler improvements, you get to 1.5 and some people said "Well, we'll just keep improving that." But the search space inside the compiler blew up. The VLIW projects, by and large, did not succeed in terms of delivering performance without unbelievable investment and out-of-order became the way to do it. But when we built EV6, there was still debate about whether you could build a high-performance out-of-order computer and make it work. Some simpler ones had been built, but EV6 was four-wide, 20 instruction window, 100 instructions in flight. It was way past our capability to think simply about. Nowadays, that would be considered fairly simple. But that was a big transition and by then, I would say I was-- this was back in '96. So, I'd been working for 16 years.

Fairbairn: Yeah. So, you were deeply embedded in...

Keller: And I would say by-- EV5 was my first big swing at an architecture that Pete and I did. We learned a lot. When it was finished, it was both the fastest computer ever built, or microprocessor ever built, and I was, let's say, so embarrassed about the mistakes I made in it, I could barely talk about it. It was really a funny personal experience, because I knew every single thing wrong with it, despite the fact that it was actually pretty good and the team that built it was a very good team, and it was interesting.

So, for EV6, we rewrote the performance model infrastructure from scratch. Alan Eustace-- I described the performance model we wrote. It was 30,000 lines of code, all kinds of complicated stuff and he said "Jim, I wrote a performance model with 1,000 lines of code." "What? Why is there just so much code? Literally, what did you do wrong?" and it was like a gut punch, like "How did I screw this up so badly?" So, I wrote the EV6 model. The first version of it was 1,100 lines of code. I basically stayed up for a week getting it to 1,000 lines where I could actually parse the alpha architecture, all the traces, run the out-of-order model inside the core, not the memory system, and made it work in 1,000 lines of code.

So, I called up Alan and said "I did it, 1,000 lines of code." I said, "I still don't know how you did it." He goes "Look, Jim, I never did that. But I knew how competitive you were. I thought if I gave you a better goal," he said, "30,000 lines seem crazy to me." That was pretty funny. But it made me really rethink exactly how you build performance models, really think deeply about it. Which before, it was sort of you wrote a model to explore ideas, as opposed to you wrote a model as a thing that has value, that is understandable and clear and usable and then Dirk and I went back and forth. He'd take the model. He'd say, I'm going to go add this and this and this and he'd just go rewrite that stuff and he was a much better programmer than me.

Fairbairn: So, the EV6 was the end of the road there at DEC?

Keller: Yeah. So, we...

Fairbairn: For both you and the company, I presume.

Keller: So, when we taped out the chip, Dirk went to AMD, where he ultimately led K7 and then became CEO. I stayed until we had Linux-- or not Linux, Unix at the time-- running on multiple chips and then I joined him to work on K8. Then they did a follow-on called EV7, which was the EV6 core with some tweaks, but with basically on-chip memory controllers using Rambus. And they built a 128-processor machine out of the EV7. I think that might have been finished after Compaq bought the company. So, yeah, about three months after I joined AMD, Compaq bought Digital Equipment. But Pete built EV7 and stayed and finished that computer.

Fairbairn: So, you went to AMD. What was the-- and your friend was already there and recruited you to come and work on the next...

Keller: Yeah. I worked for Atiq Raza, who was COO. So, AMD a couple years before had bought NextGen and their processor, the 586, became K6. So, AMD-- basically, 286, 386, 486 were mask copies of Intel processors. K5 was the first “designed by AMD” processor and they made it work, but it struggled. They bought NextGen because they already had the 586 in production and then they tweaked that to become K6. Dirk led the K7 project, which I spent a lot of time on and then I started the K8 project.

Fairbairn: So, what was the goal of K7 and K8, just the next-generation x86?

Keller: So, I'd say K7 was-- architecturally, it looked somewhat like EV6 that Dirk and I had built. But it had macro instruction architecture for cracking x86 ops and at the time, we thought you had to-- so, in a RISC machine, you have a load instruction, a store instruction, an add, a branch instruction. They're all separate instructions. But in x86, you could have an instruction with a load operand, a register operand, maybe a store operand and then some instructions dived down in the microcode. There was an argument at the time of should you fetch a macro instruction and take it apart or should you take it apart and issue micro ops? K7, on the integer side, was a macro instruction architecture. But the cache system and the bus interface were literally licensed from Digital. It was the EV6 bus.

Then on K8, the design we started that I was involved with, we were going to make it much more of a micro op machine all the way through and then we also decided to do 64-bit. So, my big contribution to that architecture was I was one of the principal authors of the x86-64 spec, the HyperTransport spec, which is the glueless multiprocessor spec, and a big update to the chip architecture that ultimately became Opteron. Now, I left to join startups during startup craze in 1999.

Fairbairn: Yeah. So, were you working for Atiq during this time at...

Keller: Yeah. So, I was working for Atiq and then Atiq left and then I left to start a company with a friend being funded by Atiq and then it turned out he couldn't do that project. So, then I joined Dan Dobberpuhl at SiByte.

Fairbairn: So, what was that company that Atiq started?

Keller: I don't even know if we got a name. So, Atiq was in the mode of funding startups and we were going to go build a network processor and build a CPU. So, AMD, I was there for three years, and we did do the 64-bit spec, we did HyperTransport spec, and we laid the seeds for what they call multi-core chips way back then and that mostly went pretty good.

Fairbairn: So, SiByte, you went to SiByte and Dan Dobberpuhl was there, right?

Keller: He was the CEO.

Fairbairn: So, he recruited you there.

Keller: Yeah, and then I was...

Fairbairn: That was a different kind of processor, right?

Keller: Yeah. So, there was a MIPS processor and then we built a networking SoC. So, it had gigabit Ethernet controllers, memory controller processors, and some accelerator functions and we talked to literally hundreds of network companies at the time. That's where I met Andy Bechtolsheim and a couple other people at Cisco and they basically laid out, like literally the napkin drawing, "You build this processor, Cisco will buy it," and we built that chip. It was super fun. It was like a really small team and that company was acquired by Broadcom.

Fairbairn: And were there other DEC alumni and so forth that staffed that?

Keller: Yeah. It was a mix of a DEC team. There were some Sun people and then a whole bunch of random contributors. We had people from Intel all over the place.

Fairbairn: How big a team was that?

Keller: I think at peak 125, 130 and then I was there for-- we were there about a year and a half at SiByte and then between that and Broadcom, I was probably there for four or five years.

Fairbairn: So, getting back to AMD, what was-- you've now worked on a very different sort of instruction set, but you said you copied some of the architectural learning from DEC. What were your takeaways after AMD? You talked about all the mistakes you'd made in the previous design….

Keller: Oh, yeah. Yeah. There's so many. So, I gave a...

Fairbairn: We always learn from our mistakes, right?

Keller: Yeah. So, I gave a talk at the Microprocessor Forum about EV6, which was super fun, because talking before me was the Power4 guy, one of the PA RISC guys from HP, and one of the Sun guys and the IBM guy was especially funny. He goes "Man, I have to talk after you guys. I can't believe it. Our processor's terrible." So, Power4 was really cool architecture, but the frequency was low and it was built on multiple chips.

Fairbairn: Power4 was whose?

Keller: IBM.

Fairbairn: Okay.

Keller: And then the HP chip was really high power and it was really interesting architecture. So, I went out and I started my talk. I said "I came here to talk about the world's hottest, biggest, and fastest chip and I just found it's only the fastest." Because the IBM was unbelievably big and the HP one was unbelievably high power. But at Digital, we literally designed down at the transistor level. The design itself was 150 schematics in almost every one of them. We had something called macro blocks, which could be a collection of transistors that you would stamp out multiple times. We had a small cell library, but we figured out one time there was 26 different flip-flops in the chip, but a lot of them were just hand-built by people and we used to joke that we had both things in our library, N devices, NP devices.

Fairbairn: So, optimized all the way down to the transistor level.

Keller: Yeah, and it turns out that's completely not scalable. Right. At AMD, they had a really good standard cell methodology, where they could write RTL and then translate that to gates, but lay the gates out in a way that you didn't lose that much performance, but the gates were characterized and that was sort of amazing to think about.

Let's say, well, how many abstractions levels are there in a computer? Because at the bottom, you have atoms, right? Then you have transistors, which are doped devices, and then you have metal stacks and at Digital, we were designing at that level. But then you have standard cells, and then slowly, you have CAD tools that place cells and then you have RTL with synthesis. And understanding that computer design was a layered set of abstractions that are fairly complicated and each one of them matters and at some level, each one of them trades off something, but lets you do something bigger. Yeah, that was a big event, I'd say.

Fairbairn: So, you found the methodology. Did AMD have the tools?

Keller: Some tools were pretty good. Some tools were less good. They wrote a lot of their own placers. A friend of mine said he felt like he was like a human synthesizer. So, you take the RTL, and then literally have to draw the gates for all the critical pieces. Nowadays, you always let the synthesizers do the work. Yeah, it was quite the journey on that stuff.

Fairbairn: So, did your-- getting now moving ahead to SiByte, did you, was all this directly applicable? Did you have to learn new stuff to kind of go into a network-oriented processor?

Keller: At SiByte, we still-- so, the DEC guys still really believed in custom design. It was probably too custom designed, but very friendly with the transistors. Like, we literally made our own...

Fairbairn: First name basis with the...

Keller: Yeah. We made our own PCI Express PHYs. Like, the idea of buying libraries and PHYs for people was just kind of-- like, it was considered crazy. But we delivered the product and the thing I think I learned at SiByte is I talked to so many customers and saw so many network devices and a lot of them had secrets. Like, they're doing this new special network box and I would literally talk to them for two hours and reverse engineer from what they said everything they were doing. That gave me this kind of, let's say meta-understanding of how that kind of system design goes together and then at some point I realized how smart-- like, the chip we were building was actually partly direct instructions from some senior people and partly talking to lots of different people about how to make a part they all wanted.

The embedded business is curious because if you put everything in that everybody wants, it'll be too expensive, and nobody can afford it. But if you miss important things, nobody can buy it because they can't use it. The trick in the embedded world is hitting that sweet spot between enough stuff that makes it a great part but not so much stuff that makes it stupidly expensive and that was like a really curious thing to do. It was almost like a sideshow from the-- we think going from digital to AMD, like in processor design, that was one path, and this was like a different kind of system engineering, but it turned out to be really interesting to me. So, it was quite a lot of fun.

Fairbairn: So, you learned a lot about applications.

Keller: Yeah, and then at Broadcom I met Henry Samueli. He was another genius guy and I traveled around with him one time and we'd go to talk to new customers and I would do my pitch on SiByte and I also knew about two other products and he would listen and about the third day, he said "Let me take this one," and he gave a better talk about our part after three hearings that I had done. He was kind of a wild character.

Fairbairn: So, you worked at big companies, DEC, AMD, now SiByte's a startup, basically, but...

Keller: Yeah, and then P.A. Semi was a startup.

Fairbairn: Yeah, we'll get to that in a minute. Did it matter to you, sort of your...

Keller: Well, so the VAX-8800 project and the Alpha projects were both small teams. So, the companies were big, but the teams were 100 people and so, there's a certain dynamic about how teams worked. For a while, I had a theory, and I had some data, which was like it takes 400 people four years to build a computer, and 300 people three years, and 200 people two years, and 100 people a year and a half. I wasn't sure a team bigger than 400 people could ever finish one. But then Itanium was 800 people and they finished the computer. So, I thought "Oh, that's weird." But it turns out, a certain group of people inside the team literally detached themselves from the management layer and built the computer. I talked to a whole bunch of people who worked on it, and they sort of finished the computer despite the fact the team was too big to actually build something.

Fairbairn: At AMD you had a big team, right? Or was that later, the second round of AMD that you had a big team?

Keller: Well, yeah, when I went back to AMD, the CPU team was 500 people, but they were doing multiple projects. They had their Bulldozer core, the Jaguar core, and then a couple of teams to do shrinks and then put the processors into the console parts and then there was a fabric team.

Fairbairn: Yeah. But that's the second-generation AMD, right? So, let's get back to Broadcom. What was the-- you got a brand new education in terms of different application space for these processors, right?

Keller: Yeah. So, Henry Nicholas believed in really owning the platform. So, you're going to make a network computer. There's a data plane part, a control plane part, there's NICs. There's all these different pieces that go together and one by one, they don't add up. They're not real products, but the whole system is a product and then the trick was how do you build that actually makes multiple people happy?

Yeah, it was really interesting how much detail there was and then networking at the time was split between the Ethernet world and the SONET world. One was cheap transport, one was guaranteed bandwidth allocation and guaranteed latency, and then those features over time kind of fused together and networks were also built on abstraction layers, the famous seven-layer network stack. Yeah, it was a great engineering tour de force to be with some of these guys and work on it all and then there was this kind of curious thing about the physics. So, when I started, 20 megahertz, 100 megahertz was a really high frequency and I still remember seeing the first 10-gigahertz scope and I was like "What the hell happened? How did we go from 100 megahertz was hard to 10 gigahertz is possible even?" and now, we have 112 gigahertz SARD1s and it's, again, there's so many layers of what makes that happen and I can kind of see them.

I know how packages work and wire bonds and inductors. I used to be an electromagnetic field person and it's just a miracle and to literally watch it evolve over time. I was talking to Henry Samueli, like, "When are we going to have gigabit wireless?" and he basically did the math. He literally just sat there and did the math. "Well, here's the slope of the curve for transistor performance and here's the size of devices and geometries of packages and here's the investment. If you want to do it here, it'll cost a trillion dollars. But if you wait five years, it'll cost $100 billion and here's how the curve's going to go and I predict 5G in 2018," and that's actually when it happened. To him, it was like "Well, these are shapes and the push bandwidth-- there's the analog signal bandwidth, but then there's the encoded symbol bandwidth," and that was a space. That was a design space and then there's parameters underneath it moving, like "Here's how semiconductors are moving. Here's how packages are moving. Here's how encoding works." and he had a plan. It was like "You're kidding."

Fairbairn: "This is the future, and this is how we'll do it and these are the steps to be done."

Keller: Well, you know, the famous Moore's Law. So, when we built the VAX-8800 in the fall, we didn't know about Moore's Law. Inside Digital, like maybe Gordon Bell knew or somebody. But we made the VAX-8800 faster than the VAX-11/780 because the customers wanted more performance. But there wasn't like "Here's the curve predicted by Moore's Law and if you're above the curve, you're going to make money and if you're below the curve, you're going to lose money." That wasn't the thing in the 80s and then the RISCs were...

Fairbairn: It was for the semiconductor people, but not for the system people.

Keller: Yes. Well, and you could design a computer. Like, the VAX-8800 was designed by computer architects using gate array chips and a couple of custom chips as components to achieve our goal to build this computer. Whereas in the Alpha team, it was like they built transistors and literally needed to do something with them and the mission of transistors by then was to make the transistor smaller, faster, put more on a chip. Then Dan's genius observation was if you build the pipeline, the circuit technology, and the clock networks properly, you can make the transistors run essentially full speed and then you don't push it too far on pipelining because then the logic design breaks. But if you get the clocks and the flops and the latches and all the tools right, you can make 200-megahertz processors in literally half micron technology, which you look back and you go "What happened there? That's kind of amazing."

Fairbairn: So, on to P.A. Semi. That was a pretty big transition going from...

Keller: Well, it started literally as the reboot of SiByte. So, we got architectural license to PowerPC, which was a long process. And then...

Fairbairn: Who were the other founders? I mean, were you a founder? Was it...

Keller: There were the original three founders from SiByte who were the senior founders and then five more of us.

Fairbairn: Does that include Dan?

Keller: Yeah, Dan Dobberpuhl, Amarjit Gill, Leo Joseph were founders and then Puneet Kumar, Mark Hader, myself, Sribalan Santhanam were kind of founders, but I think we weren't really senior founders or like the docs were definitely...

Fairbairn: I understand that.

Keller: Let's say hierarchical. So, the original idea was "Let's go do a similar part, but upgrade everything." So, one gig went to 10 gig, processor performance went up. It was an out of order PowerPC core and it was a much more aggressive design in the architecture so that we could stream. To make the performance work, you had to stream data in from the 10 gig ports, go into a on-chip cache or memory, process it without it having to go to memory and then be able to do interesting things and send it back out. So, it was an architecturally much more aggressive part and then at some point, it was also architected in a way that you had a processor area, you had a slice ahead, memory controllers, PCI Express and network stuff that was pretty generic and then you had the network acceleration stuff. We built a part so you could tape it out two ways. One as a network processor for the Cisco business and all those guys. But you could also build it as a standalone part, which could have been a personal computer part for Apple. And we talked to Apple quite a bit about...

Fairbairn: So, I thought that a major goal of P.A. Semi was lower power. Was that-- did I make that up or?

Keller: Dan was in the low power because he did the StrongARM project and then all the network guys, they had lots of boxes with limited airflow. For 25 watts, you could put a part on the board and just blow air over it. For 50 watts, you had to put a heat sink, put it in the right spot. For power above that, things got exciting pretty fast and then there was some low power circuit designs and tricks and the team was pretty good at clock gating and low power, but very accurate clock distribution, which I think was better than industry standards. By today's standards, it wasn't really a low power design, but for the time, it was actually pretty good.

Fairbairn: So, the goal was performance, but with minimizing the power along the way?

Keller: Yeah. In the network processor world, it was sort of performance, but hitting certain power points so you could deploy it properly, like as a system engineering problem. The PC world is much more fine tuning power performance by then. So, desktop parts are 95 watts because it's easy to put a fan on them. Like desktop is go fast as possible. As soon as you go into mobile, now you're trading off hotspots on the package and battery life and at the time, the mobile parts were starting to get plausible. They were still 25, 35 watts.

Fairbairn: So, did P.A. Semi actually end up selling parts, or you...

Keller: The company was acquired by Apple.

Fairbairn: Before it actually offered parts?

Keller: We taped out a part that worked, but it was purchased.

Fairbairn: And you left Semi before or you went to Apple first and then...

Keller: Yep. We finished the part, taped it out, and then I joined Apple and then Apple-- so, then there was a funny situation where Samsung was partnering with Apple to build a one-gigahertz ARM processor and a guy I'd work with-- so, Samsung partnered with Digital Equipment and they built what's called the EV56 and 68 parts, which were shrinks of the Apple chips on the Samsung foundry. The guy who ran that project called me up for my help to get the one-gigahertz processor to work and I said "I already work at Apple, but you should call Dan and Dan will help you get this," because they needed a big client, let's say, and then Dan figured out what was going on. He called up Steve Jobs and said "Why should I help Samsung when you can just buy my company and we'll build it directly?" and Apple acquired P.A. Semi.

Fairbairn: So, how did you wind up at Apple?

Keller: I had some friends there in a research group that said, "You should come do something fun," and I said "What do I work on?" They said "We can't tell you," and then for a little while, I think I went through-- at one point, I walked through three locked doors to get to my office because Apple loved secrecy, partly for the drama, I think, and partly for-- Steve didn't want this project messing with this one and like "You do your stuff, you do your stuff, you do your stuff."

Fairbairn: So, were the people that got you to Apple, were these from Digital or...

Keller: Yeah, Dave Conroy, I worked with. He was at the [DEC] System Research Lab working with Chuck Thacker and he had joined Apple to be one of their guru computer guys and then I met a couple other people there. I'm trying to think of who I knew besides Dave. I don't think I knew-- oh, Steve Polzin was another guy from the digital system group. He worked in Mac engineering. So, I knew like literally three people when I joined Apple.

Fairbairn: So, what was the state of processor design at Apple at the time? I forget exactly what the-- this was in...

Keller: Yeah. So, I joined-- so, the first iPhone chip.

Fairbairn: 2009?

Keller: Yeah, something like that. Is it 2009?

Fairbairn: 2008?

Keller: Yeah. So, the first iPhone was shipping. They were literally weeks from taping out the second chip, which was called H2 internally and that...

Fairbairn: And that chip was-- the chip of the first one was an ARM-based...

Keller: Yeah, ARM-based, ARM IP. The first chip was literally-- the person who said he was the architect, he showed me the architecture document, which was an Excel spreadsheet with all the IPs he needed. So, he said "I need this CPU, this memory controller, this bus, this..." It was just a list of IPs.

Fairbairn: So, he put a bunch of IPs together and...

Keller: And so, they gave that to Samsung and Samsung put them together on a chip and it worked. The memory latency was terrible and performance was weak and they could barely get the software to work on it, but the software guys figured out how to make it work. They did an unbelievable job because they took basically OS X and shrunk it down and got it to run there and by doing that, they optimized and minimized the software. So, later on when we gave them faster chips, the software could really grow. Like somebody said, the biggest difference between iOS and Android was Android got ported to a chip when the phone chips were already pretty good, and it never really got cut down in the way iOS did and that may be true.

So, I got involved on H2 and then Tim Millet and I were co-architects of H3, which we really optimized a lot and H4 is the first chip where we put a really big graphics unit and then H5 was the first time the P.A. Semi team delivered the first processor called Swift and then H6 is when Gerard Williams and I designed essentially the big out of order computer that Apple's been building on ever since.

Fairbairn: So, before that one, were there-- was it a matter of executing on sort of known architectures and so forth and just getting the power down and so forth to make it work in this environment?

Keller: Well, the original phone SSEs were pretty straightforward. It's sort of like you have a radio interface, you have a display interface, you have a keyboard interface, you have a processor that runs the operating system. There's a little GPU to render the screen. Like the diagrams are, a piece of paper this big with a bunch of blocks and interconnections. So, the architectures of phone chips have mostly been straightforward since the beginning. The thing that makes them, let's say magical, is really hitting performance per watt and power and then Apple was very aggressive on new things.

So, we put 64 bits in before anybody. We did high resolution display before anybody. We put a lot of graphics performance in H4. Like, it was just completely outside anything that anybody ever done. If you looked at the chip, the GPU was literally five times bigger than the GPU in every other phone chip and some of that came from Steve Jobs and Mike Culbert and Steve had this idea to make the best technology, you need the best components-- to make the best product, you need the best technology, the best components. And then the Apple guys, when they started really getting into touchscreens and visual apps, they went hard on stuff like high resolution, lots of graphics performance. Graphics performance in service of the computer interface, as opposed to graphics performance in service of just a game. Like, Apple's never a gaming company.

Fairbairn: Were there unique aspects of the architecture to achieve those goals or was it a matter of optimizing architectures and sort of...

Keller: Yeah. There was some. Like we had hardware widgets for all kinds of stuff. Like, the way Apple UI went together was multiple layers. We had our display pipe did really sexy compositing. At a pretty low level, the way the GPU, the CPU, the camera processor worked together was pretty, I think, novel at the time. But a lot of it was just like really looking at the target and aiming at it and then Apple, by designing their own chip, they didn't pay the margin stack. So, we bought wafers directly from Samsung and then later TSMC, rather than buying them from Qualcomm or Nvidia. So, those guys bought wafers from TSMC and then they had to mark them up. So, Apple could turn the extra cost into way more transistors and we did.

Fairbairn: So, that actually translated not to necessarily better margins, but maybe also in terms of the size of the chip you could build, that it can be cost effective...

Keller: For the same cost of the chip in an Apple phone, we had twice the transistors. Some of our competitors, which-- and then the software applications and the processor kind of co-evolved together and so, we made optimization choices that let us do stuff like-- if you said "Hey everybody, do you want this great camera processor?" and one person says yes, and one person says no, then you put an average camera processor in there because you can't monetize it. Whereas Apple was like we knew what we wanted all the time, and it was dedicated design.

Fairbairn: Right. So, you could really optimize the design. You had more transistors to work with. What about tools, design methodology, or was there anything unique that Apple had that wasn't available?

Keller: This is one of those things where new teams often do things that old teams can't do. So, as the Apple design team grew, we had people from all over the place-- NVIDIA, Intel, AMD, small companies-- and then at some point when we were working on methodology stuff, it was sort of, instead of iterating on the design methodology that everybody had been working on for five years, it was like "What are the best features from all the different methodologies anybody'd worked on?" and at one point I thought we probably had the best design methodology in the industry and that's because it had people from all different places with different ideas and it was like-- I like to say hybrid vigor, which was pretty good.

Fairbairn: Was Steve Jobs a constant presence in your activity, or was he...

Keller: No. He's a looming presence. I only talked to him a couple of times. Like, everybody knew what Steve wanted like the next day. It was amazing. My boss talked to him all the time.

Fairbairn: Who was your boss?

Keller: Mike Culbert and he was sort of the unofficial CTO of Apple. We thought of him as that way and then we also worked from Bob-- so, he was the technical guy in Bob Mansfield was the senior vice president of Mac engineering and later phone engineering and chip engineering. He was a great senior vice president, like the best I've ever worked with and it was really interesting to watch those guys translate what Steve wanted and they talked to him a lot and to get what he wanted and it was-- yeah, it was pretty much a vision driven company with, let's say, lots of direction, but not-- like, Steve's not a micromanager.

Fairbairn: Not a how to do it or whatever, but just...

Keller: Yeah, he's not a how to do it. He's "Here's what I want and what can you do?" and "Tell me what you can do and then we'll see what we make out of that." So, yeah, it was a really wild experience.

Fairbairn: So, you left. Why did you leave?

Keller: Steve Jobs passed away and then Mike also passed away. Yeah. It was sort of like I knew it was going to be a way different place and I partly went there to learn stuff. Like, Apple's an amazing company and I remember after being there-- after about a year, I thought "I still don't really know how they get anything done." Because it wasn't the same bottoms up engineering culture that I was used to from Digital Equipment and then when I was at AMD and then the startups with Dan, they were all the same culture companies. Apple was much more of a "I want to do that and we're going to do everything to do it," as opposed to "We're great engineers who know how to build computers."

Fairbairn: "We'll tell you what you get."

Keller: "And we'll tell you what you get." Yeah. I wanted to work a place that created stuff out of nothing by intention and it was a wild place to work, and it was great. But I thought I learned a lot and I kind of wanted to try it out. So, I went to AMD, which was a failing company.

Fairbairn: So, who brought you back to AMD?

Keller: Rory Read.

Fairbairn: He was the CEO at the time?

Keller: Yeah. So, I knew Mark Papermaster. I had met Mark during the PowerPC negotiations at P.A. Semi and then we overlapped for probably a year or so when he kind of came in to run the phone group and then Steve fired him over the antenna gate stuff. But Mark was a really good guy and so, when I was looking around, I called Mark and then I talked to Rory a bunch. I quite like Rory and then I worked for Mark. Rory said, "I do the business, you do the technology. I'll make sure we don't run out of money. I need the chip."

Fairbairn: And the goal at that-- AMD was not doing well at the time, right?

Keller: It was doing poorly, yes.

Fairbairn: And so, it was to design a chip. Was it designed from scratch?

Keller: Yeah, yeah.

Fairbairn: Nothing is quite from scratch in that business, but...

Keller: Well, so Rory showed me this graph. So, AMD Server, starting with Opteron went from zero to 35% market share and right back to like 4%. He said "Jim, what happened?" I said, "Well, the product sucks and so, nobody's buying it." He goes "It looks like nobody even cares," and it turns out Bulldozer wasn't a good architecture and then it was still being built.

Fairbairn: Bulldozer was the architecture for Opteron?

Keller: Yeah. Well, no, Bulldozer came later. So, the K6 guys came up with this great cell-based methodology, which we built K7 and then K8 on, which was Opteron. Then they wanted to take a step further to raise frequency and do the next level and they basically brought back in a lot of custom design and let's say some questionable architectural features. Those two things, they didn't achieve their frequency goals and then the architecture features were not great. So, that created a product that wasn't competitive and then AMD also was struggling with foundry and they ultimately spun it off to become Global Foundries. So, then they were saddled with a tier two Foundry and a tier two architecture and so, I was brought in to be the architect of Zen and then to rebuild that.

Fairbairn: Was it clear what needed to be done?

Keller: Oh, yeah. Yeah. It was super clear. The methodology was bad. The team organization was bad. The architecture was bad. Then the team would tell me stuff like "Jim, we can't compete with Intel because our process is behind," and I would say "Well, actually the IPC is literally half of Intel's and I'll tell you what-- if we match their IPC, I'll get us a process that matches their process. But you can't complain about the process if the IPC is no good."

Fairbairn: IPC is the...

Keller: Instructions per clock.

Fairbairn: Per clock.

Keller: And so, there's an architectural component and a process component. At some level, process floats all boats. So, IPC is like how good your architecture is and then architecture times process performance gives you your product performance, and they were missing on both. Now, I had no idea how to get the process that was good. Mark was a key player on that. They negotiated a cross license from Samsung 14, the Global Foundries, which worked. And then later on, Rory's contract said something like AMD will build at Global Foundries as long as Global Foundries has a competitive leading-edge process. Then Global Foundries announced they were not going to do their seven-nanometer process.

Fairbairn: Yeah. They stepped back from that, didn't they?

Keller: That set AMD free to go to TSMC. But yeah, it was a pretty big lift. So, I called Rory up and said "Do you mind if I cancel all our current projects?" This was the thing that set Zen free. They had a Bulldozer roadmap and then a Jaguar roadmap. They had a big core and a small core and neither of them were competitive. The Jaguar was actually pretty good, but people weren't buying that. That zone of processors and Bulldozer were not competitive for a number of reasons and Rory said "Jim, nobody cares if we're 50% or 53% of the competitor. They're just buying this for pity." So, he says basically "Do whatever you want." Then he would call me up "How long is it going to take?" I said, "Four years." He goes "I don't have that long," and the board timed out on him and he left.

But yeah, it was super fun and it was partly an architecture thing. Like, architecturally, big strokes-- there is a spreadsheet that says "Well, you need this many instructions per clock, this kind of branch predictor, this kind of-- you need this frequency." We set this bounding box, which the team was endlessly trying to renegotiate. I said 10 square millimeters, five watts, three and a half, three and a half gigahertz and they would come back and say "Well, we can do 10 millimeters, but it's only going to be three gigahertz," and I was like "No way."

The mission was to turn the gaps in the problems to be solved as opposed to things we can't do that we renegotiate, and that mostly worked out. Then we had an ARM processor that was like an architectural companion because there was a belief that ARM servers would be big and I think that could have been true, but it was a little early for ARM in the server space. Then we delivered a couple of other products. We did an Xbox chip, a Sony game console chip.

Fairbairn: So, how long did it take? You said four years was too long. How long did you?

Keller: Yeah. It took four years.

Fairbairn: Four years. Lisa Su came in as CEO somewhere along the way?

Keller: Yeah. When I was there, she ran one of the business units and it was ARM servers embedded in semicustom and so, for most of the time I was there, she was doing that. She was not part of Zen.

Fairbairn: So, at what point did you choose to leave? You got that project done or well on its way?

Keller: It was mostly done and then let's say there was some management difficulties. I didn't really want to leave, but...

Fairbairn: You what?

Keller: I didn't really want to leave, but like not everything goes your way. I'm kind of a bristly character and I like things done-- well, it's hard to say. I was going through a phase of like I knew how to build everything, and it was pretty clear.

Fairbairn: So, then onto Intel?

Keller: Tesla.

Fairbairn: Tesla, yeah.

Keller: So, Doug Field worked at Apple in the Mac group, and he went to Tesla and I was talking to him. Then Raj Singh and another guy that worked for Doug, they said, "You should come here," and I was like "What do I do? You guys build cars." He goes "No, no. Cars are going to be like a computing platform. There's going to be computers in everything," and I was like "That's hilarious," and then I started researching-- like, Tesla had an autonomous driving plan and obviously Elon was already somewhat famous even way back then. So, I went in and talked to Doug and a bunch of people and it was like "Come in here. You can work on electronics as well."

So, we needed to upgrade the autopilot hardware and do the stuff, and I thought "Shit, we could literally build a chip for this." The chips they were using weren't very good and they weren't hitting the target, let's say. Then they organized an interview with Elon. So, I flew down to LA to meet him at SpaceX and I walked by a conference room. It was a Landsat conference room. So, my father was the chief engineer on Landsat years ago, which was a satellite and then there was a rocket taking off. I'd seen a Saturn V launch when I was a kid and so, Elon and I reminisced about the Saturn V rocket and Landsat and then we had a pretty good chat. I told him I could build a computer chip to drive a car in 18 months and he basically said "Okay."

Fairbairn: How long?

Keller: Eighteen months.

Fairbairn: Eighteen months?

Keller: And we did. It was like a world record.

Fairbairn: Is that for-- you were using TSMC?

Keller: The chip was at Samsung. So, I took the job. I didn't tell anybody. There was still some dust settling from leaving AMD. Then my friend, Pete Bannon, called me up like the day before Christmas or after Christmas and said "Hey, I heard you're going to Tesla," and I said "How'd you find that out?" and he said "Your wife put it in the Christmas letter that you sent to all your friends." So, I said, "I start like January 5th or something," and he said "Well, I want to talk to you about it." I said "Great. Do you want to come over then?" He says "How about tonight?" And he came over. We talked for like two hours, and he was still at Apple, and he was frustrated with their AI direction. So, I joined and like three days later, Pete interviewed with Elon and then Elon said, "Pete's great," and then I hired Keith Witek, who was a lawyer, because we're going to build a chip in 18 months, I need to execute like 35-- we ultimately did 35 IP contracts, supply agreements, all kinds of stuff. I called my friends at Samsung that I worked with at Apple and said, "I'm going to build this chip. We need the tape out in one year," and they're like "That's very hard."

So, it ultimately was fourteen months to tape out, four months in the fab, two months to bring up the software, and we drove a car. And Pete had this unbelievably cool idea about the AI engine. So, at the time, one of the popular frameworks was called CAFE and it spit out a list of op, data, data, data, and then people were taking that and building a compiler to write a vectorizing compiler on it and Pete said "What if we just execute the op and the data as written?" And we made a chip that basically executed the CAFE instructions as instructions. And the data might be three by three convolutional data source or something, which has a whole bunch of interesting ways to walk through it and we built a memory subsystem to swizzle the data around to match what CAFE thought it was executing, which made the compiler really simple. So, everybody in the world is hiring hundreds of people to write AI compilers and Pete wrote the compiler himself because the translation from CAFE output to instructional of output was, let's say, relatively simple. It was ultimately some more detail on it. That gave us an architecture definition and then I hired some friends from a variety of places to go build the engine and then Samsung built the chip.

Fairbairn: What was the key to getting it done in 14 months, getting the IP contracts done and just every step of the way, optimizing?

Keller: Every step of the way. It was like parallel execution. By then, I'd gotten really good at-- I tell people I'm a 100% person. So, you had a-- there's an AI engine, a compiler that goes with it, there's IP contracts. We had a really good SoC [System on Chip] architect, David Glasco. Dan Bailey joined me from AMD, where he's one of the best silicon guys on the planet. The execution team in Samsung, I knew those guys. We spun them up pretty fast. We made decisions really fast, like security architecture, safety architecture, bus architecture. We screwed around for a little while on camera processing. Like, AI images at the time were all processed by camera processors, but it wasn't obvious that was a good answer. We were unsure about some software stack components. We had to import a bunch of stuff. So, we used our own GPU.

Fairbairn: So, does Tesla still use the same architecture?

Keller: So, Hardware 3 stopped shipping something like nine months ago. They shipped it for almost five years. It was the best autopilot chip on the planet for five years. Now, they're shipping Hardware 4, which is an upgrade. There's architectural similarities. We started that project when I was at Tesla, but I'm not privy to the details of how it came out.

Fairbairn: So, did you have a lot of interaction with Elon along the way, or not at all?

Keller: Yeah. He was there two days a week and we talked to him all the time. Then I had autopilot hardware and at some point, Doug asked me to run low-voltage electronics and then on and off, the autopilot software people reported to me depending on who'd been hired or let go recently. So, that was, let's say dramatic.

Fairbairn: Did Elon want to insert his own ideas about how things ought to be done or understand the architecture and...at

Keller: Yep. So, at some level, he wasn't a chip guy. He had written software. He's really good at physics, mechanical engineering, and visualizing things. On a lot of the chip stuff, I'd explain it to him. He'd be very interested, but he didn't have lots of ideas about that. But how the overall thing worked and why, yeah, he was really interested, and it was interesting to explain it to him. I learned-- so, in a lot of places, you start with "Here's the problem statement and here's the data," and you walk people through your journey. Elon's the solution-first guy. "We're going to do this. That's because we had this problem. Everything else is back up." One day we were walking him through it. "Here's the images we captured at night. Here's what's going on. Here's why they're hard to..." He just lost his mind, like "What are you guys talking about?" "Well, we're going to get to the solution." "Well, what's the solution?" "It's on page 18." "Well, start on page 18. What's wrong with you?" It's like "Here's the old picture. Here's the new one." He goes "That's great. The new one's way better. Why is it better?" "Oh, let's just go through the presentation backwards." His directness was great. Like, when he talked about first principles, I've said this before. It's like, you think you're a first-principles person and then you talk to Elon and it's like, no, no, he meant atoms. Like, first principles is like a whole other level.

Fairbairn: Right. Yeah. So, what got you out of there?

Keller: We finished Hardware 3, started Dojo, but it was clear to me it was going to be a while before... So, Elon's a big step guy and let's say chip engineers, there's a roadmap of cadence and continuous improvement and I thought-- I wasn't quite sure what I was going to do for two, three years before we taped out. So, Hardware 3 to Hardware 4, you can read about in the papers. You can read about it in the papers. It was like four to five years apart. As far as I was concerned, we were going to tape out Hardware 4 a year after Hardware 3 and make steady improvements. But like Elon's more of a "Take a big step. Use the shit out of it, take a big step," and there's something to be said for that. But I thought "Eh, I want some new excitement." Then I got offered SVP at Intel to run their silicon engineering group and I thought that would be wild because Intel had the best fab, the best CPU technology, the best-- they had cross-point memory, they had a whole bunch of stuff. I thought we'd go do the next generation of high-end servers. Not any-- none of that happened, but it was a really fun job.

Fairbairn: What made it fun?

Keller: Well, when I left AMD, the team was 2,500 people. Tesla was 400 or 500 people. I joined AMD and my team was 10,000.

Fairbairn: You joined Intel?

Keller: Yeah, Intel was 10,000 and the scope of it was epic-- client parts and server parts and networking parts and then I slowly figured out that a lot of Intel technology was great, but a lot of it was stuck on old CAD tools, and old design flows, and...

Fairbairn: Yeah. Intel really got in a rut with their design tools, didn't they?

Keller: Yeah. So, it was like "Well, how do we re-engineer an organization this big to be oriented properly?" and they had done some stuff like-- they used to make integrated client parts. So, the CPU team, the memory controller team, PCI Express team, they all worked for the same guy, they all worked together, and they made one part. Then Murthy [Renduchintala], who was EVP, said "We need to have IP teams and SOC teams and CPU teams," and he broke them up. But it never really got refactored. Like, the IP team wasn't really IP. The IP was just a design group that delivered stuff to the SoC, who tested it. So, they had all these things like half-transitioned and the challenge was "How do you re-engineer that and fix the CAD tools?" I got a call from Aart de Geus who said "The Israeli team is using our new CAD tools. How'd you do that?" I said, "A lot of fighting." But it was fun. Those guys are great. They built literally the best design flow on the planet and used it for 25 years. It was very difficult to say the world's moved on and change is going to come and like "How does that happen?" and some people were like "Hey, we're ready for it," and some people obviously, it was-- people are very emotional. Like, engineers spend their lives building stuff.

Fairbairn: Intel still had a big internal CAD group, right?

Keller: Yeah, 1,200 people and a lot of them were just building wrappers around tools. So, we introduced this thing we called the bare metal flow, which is how do you build low-level interfaces for all the CAD tools? How do we change how we think about CAD tools? Some of that worked out pretty good.

Fairbairn: Tenstorrent.

Keller: Yeah. So, Tenstorrent's an AI company.

Fairbairn: So, how did you-- did you burn out at Intel with 10,000 people, or what was the...

Keller: No. Like, at a high level. I thought the company and the engineering team needed, let's say, a much bigger reset than the CEO did and I quite like Bob Swan. We were colleagues, but his vision of what needed to be done and my vision were pretty far apart. I wasn't burned out. I was...

Fairbairn: You couldn't get done what you wanted to get done?

Keller: It was very difficult, and I'd learned you need to be in alignment with-- the CEO runs a company and you can't be going to a meeting and say "We need to do this," and they're like "The CEO said the opposite," and he had his reasons. He was financially managing the company, and I was looking at a whole bunch of key engineering technologies that needed a pretty big overhaul. That was more than you could do, say, just on the way. At AMD with Rory and I, it was like "We need to overhaul this from scratch," and he said "Great, go do it because what's happening is not working." So, that worked pretty good. At Intel, it wasn't working so well. So, I thought I wanted to go work on a startup. I thought of starting a company. I had been the angel investor at Tenstorrent. I liked the AI technology at Tesla. I got to see every AI...

Fairbairn: Tesla was your first introduction to really be inside AI technology?

Keller: Yeah. Even there, I wasn't the architect of the AI engine. Pete was. I architected the plan to build an autopilot chip and I'm, I'd say, very good at working across functional boundaries to make things come together. I thought I could help Tenstorrent launch the product and also build the next generation. Also, Tenstorrent AI technology is built on RISC-V processors, and we use multiple RISC-V processors to enable that, from very small ones to medium ones to big ones. So, we raised the investment to build a high-end CPU team to build a high-end basically tentpole RISC-V processor.

Fairbairn: To basically what?

Keller: The tentpole. So, RISC-V started in Berkeley. Amazing quality. The BOOM Core and the Rocket Core are open-source RISC-V processors. Then SiFive, Andes-- Ventana built, I would say, more product-oriented mid-range processors, but nobody built a wide, out-of-order, high-end RISC-V processor. In the CPU world, the people want to go from very low to high. The genius of ARM's IP plan is-- like a lot of CPU companies, every time they build a better one, they drop the old one and ARM is like they got M-class, R-class, A-class, server class. So, RISC-V needed to extend that line. So, I thought it would be really interesting to build A-5 based on RISC-V computers at the bottom, but also build the coprocessor in RISC-V to put that together.

Fairbairn: This all has to be done in the context of how you deal with ever-changing AI models and algorithms, right?

Keller: Yeah. So, it turns out the AI stack is really complicated from top to bottom. It's not like a person has an idea and they write a C program that runs through a compiler to a chip. There's about four or five levels of translation. Like, PyTorch programs run very slowly, like 100 kilohertz. So, it takes you a while to get your brain wrapped around that. So, you're going to put a petaflop of compute on a computer program that's running at 100 kilohertz because it's parsing ops, and the ops are really big and then there's translation from the PyTorch level, what the programmer thinks he's doing, down through the software stack. Then you need to build that in a way it's flexible and resilient to the new ideas they have at the high level, but translates into high-performance computing at the low level and that's the intellectual AI challenge that Tenstorrent's working on.

Fairbairn: So, did Tenstorrent start to build a processor, to build-- what was the original...

Keller: To build an AI engine.

Fairbairn: To build an AI engine for what application?

Keller: General purpose AI computing. Like, the basic engine is...

Fairbairn: Data center level or...

Keller: Yeah. The basic engine is a tensor processor. There are generically three kinds of computers-- scalar, vector, and tensor. So, AI is tensor operations on tensors, which are multidimensional arrays of data and the core of it is a tensor processor fed by RISC-V processors that interpret, let's say, one layer of the software stack and feed the operations to that processor and then move data between processors. So, yeah, our mission is to make really general-purpose computing that's accessible from top to bottom. So, every layer has a way for a human to be involved or the software can take care of it.

Fairbairn: So, how has the mission, direction, whatever, evolved over the time you've been there, first as CTO and now as CEO?

Keller: Yeah. So, startups go through phases. I realize this is-- it seems obvious. There's a research and development phase, "Here's the big idea." Well, there's the crackpot phase, which is "I have this big idea to have to raise money." But then you go do research and development to build something that's worth doing and then you transition to engineering, like make every piece of that really good and then you transition to product, "I have a product that somebody could buy," and then you work towards being customer-focused and then ultimately big semiconductor companies are production-focused, let's say. So, I joined, I guess, to help transition from research development phase, which the founding team was good at, to, let's say, engineering excellence phase end-to-end and this year, we're starting the ship product to be a product company and kind of move through that journey of each step properly.

Fairbairn: But who the customer is has evolved over time, right?

Keller: It's interesting. The software, generally speaking, at the AI application level has gotten a little simpler. The original Inception ResNet were pretty complicated graphs of computing. The modern like Falcon, Llama, Mistral are simpler transformer models. They're still sophisticated in the ops they do. They've mostly grown larger, way bigger training sets, bigger training times. People want to be able to expand quickly from a single chip to many chips, depending on the application. I'd say it's still changing at a high level, especially compared to the traditional compute server client world. But it's changing in directions that are, let's say, understandable.

Fairbairn: You've described the company as a design company.

Keller: Yes. We design AI processor, AI software, and CPU and I want to be really good at all aspects of that, top to bottom. We design chips, we design systems. We are kind of an end-to-end company. Like, one or two of the startups have turned into "We're going to put our AI behind the cloud, or semiconductors are hard." But no, we're a design company. We are friendly with software design, chip design, architecture design, system design and I want to be really good at all of it.

Fairbairn: Do you have a sales force?

Keller: Yeah.

Fairbairn: Who are they calling on?

Keller: Well, it's all over the place. So, the mistake all the AI startups, they say "We're going to build this chip and we're going to go sell to AWS, right?" That has to run everything on day one. It turns out there's literally thousands of people who buy AI. They put it in equipment. They put it in small data centers. They put it in research labs. People want to buy AI and put it in their own chips. The diversity of customers is really high. Now, the Wall Street guys focus on the $100 billion of high-end computing for data centers. But there's literally billions of dollars of business all over the place and usually, new entrants. You can bang on the door of the cathedral of the incumbent, or you can go find new customers that are doing new, different things and the big guys can't focus on the little guys. NVIDIA's not looking for another one-million-dollar customer. But I sure am.

Fairbairn: The million-dollar customer can't afford a million-dollar system. So, you've got to have tiered systems, right?

Keller: Yeah. You have to have different price points and also, our sales guys would-- you get 10 $1 million customers, it's $10 million sales, it's real. Like, I'd be thrilled about that and I've done this before. You also can't pick and choose the winners. When we designed parts for SiByte, we had 100 design wins at one point and some of them became quite big for Broadcom and it wasn't obvious which ones were which. So, I think there's going to be another generation of AI going all over the place and I want to be one of the companies that has AI that works, software that-- we open source our software stack. It turned out that's a great recruiting tool because a whole bunch of people, they want to write software and publish it and they review our code before we hire them. It's really kind of a wild experience and because we're not the incumbent, we're not proprietary, we're open, we have access to all kinds of people that incumbents don't have access to.

Fairbairn: Could Apple be a customer at some point?

Keller: Yeah, maybe. But they have 5,000 people working on chips and they're experts and I'm sure they're doing their own thing.

Fairbairn: Other processor vendors?

Keller: It's possible. Yeah. I've been through multiple transitions like this. When we built Opteron and K8, nobody wanted it. Everybody knew servers have backplanes, right? We built a computer that didn't have a backplane. Now, 20 years, it's been the dominant server architecture.

Fairbairn: So, if somebody wanted to buy something today from you, what could they buy? What could you deliver today?

Keller: Yeah. We're selling our Grayskull chips and we're about to announce our next chip that's networked together and we'll be able to hook up like a single chip to 100 chips together.

Fairbairn: So, your Grayskull chip is a single...

Keller: Single chip. It goes on a PCI Express device and then Wormhole has 1,600 gigabit Ethernet ports and we hook those together and at the end of this year, our third chip, Blackhole, will be ready to ship and that's like a shrink with a whole bunch of performance optimizations.

Fairbairn: And people can license IP from you as well?

Keller: Yep. It's been really interesting to talk to people because some people, they want to buy the chips to try them out. But they really want to build their own product and they want to license IP and obviously, NVIDIA won't license their IP to you to build products that they would think competes with them.

Fairbairn: So, you'd like to try it out, verify performance and everything else in chip form and then, as the next generation, embed that IP in some of their own proprietary product?

Keller: And RISC-V is doing the same thing. People are using RISC-V processors for all kinds of little things, but we're going to tape a chip out at the end of the year that has our big RISC-V processor, so people can try that out and then either use that chip in their product or license the IP to put in their product.

Fairbairn: Could that RISC processor be used for something other than an AI function, or is it...

Keller: Yeah, of course.

Fairbairn: It isn't so specialized or...

Keller: No, it's-- so, I want the RISC-V technology, so when I build AI, I have all the pieces, and I can modify it the way I want. But it's also a pretty general purpose [processor] that we can do almost anything with it.

Fairbairn: So, do you enjoy being CEO or CTO or both?

Keller: Yeah. It's been really interesting to do. CEO's kind of wild, because you're sort of responsible for everything.

Fairbairn: Yeah, sort of.

Keller: And it turns out to be-- the company's scope is way smaller than Intel or AMD or Tesla. But like I say, the breadth of it is higher.

Fairbairn: How big are you now?

Keller: Five-hundred people.

Fairbairn: Do you work in the office or virtually or all of the above?

Keller: Both. Yeah. We have offices here and there. So, I'm in the office, I'm at home. A lot of times, the calls start early in the morning. So, I start doing calls from the kitchen table at seven o'clock and then...

Fairbairn: Do you care whether people are in the office?

Keller: A little bit. I'm really impressed with how much work people get done, whether it's being wherever they are. For some people, the flexibility is life changing. People have lots of needs, family, taking care of their parents. They live far away. There's all kinds of reasons for doing it. I think it's really healthy for everybody to get together in the office periodically. They like it. Every once in a while, the office is full and everybody talks nonstop. I think something like two or three days a week is really good for human beings. But the flexibility, like they drop their daughter off or they pick their mother up or they have a doctor appointment and it's just sort of scheduled in and nobody cares-- that seems also very healthy.

Fairbairn: So, let's spend a few minutes on Jim, the person. I heard somewhere along the way that you had a wife who sent out Christmas cards disclosing proprietary information.

Keller: Yes, she did. It was only proprietary to me.

Fairbairn: You're married. Do you have kids?

Keller: I've been married for 20 years. I have two children. They're going to be 19 and 20 this year.

Fairbairn: So, they're off to college?

Keller: One's in college for two years, and one starts at Cal Poly in the fall.

Fairbairn: Doing what, the two of them?

Keller: Whatever the hell they want to. But yeah, they seem to be both smart and alert and yeah, we'll see what happens.

Fairbairn: Where'd you meet your wife in your busy career?

Keller: She was a friend's sister. So, I was at a 40th year birthday party and I met Bonnie and somehow we hit it off.

Fairbairn: Twenty years in, huh?

Keller: Yeah, 20, 21, maybe, this year.

Fairbairn: Somewhere in there.

Keller: Yeah. It's been great.

Fairbairn: What do you do when you aren't hiring, raising money, inserting yourself in meetings?

Keller: I've always been into being physically active.

Fairbairn: Looks like you're in pretty good shape.

Keller: I'm in reasonable shape, yeah. I'm starting to get a little older. But I kite surf and I snowboard and lift weights. I had to stop running because one of my knees started bothering me and I talked to two 85-year-old guys and they both told me they wished they'd quit running sooner. I said "Well, when did you quit running?" The one guy said "Eighty. If I was doing it over again, I'd quit at your age," because the knees wear out and knee replacement is hard.

Fairbairn: My wife had her knee replaced and wished she'd done it sooner.

Keller: Yes, mostly it works out good. But like I said, I've been pretty active most of my life, so.

Fairbairn: Do you do a lot of reading?

Keller: On and off. Sometimes I feel like I read all the time. It used to be like a couple books a week and now, it's like there's blogs, there's the internet, there's articles, there's this. There's infinite sources of stuff and people really pour their-- they still pour their hearts out. I tell people to read a lot. The funny thing is, people always ask me "Well, which book should I read?" and I think "All of them." The problem is, even if this is a really good book, that book in the context of 100 other books is a much different book than a book standalone. As a young person, maybe you don't read that much, but I've literally met people in the last month that I thought "Christ, if they read 10 books, their lives would be completely different. There's literally problems solved," and it's not just the way the book-- I recommend books that I don't agree with all the time, but the thinking and the alternative-- like, there's so many things to know and to think about. So yeah, I still read quite a bit.

Fairbairn: Being that you're still relatively new in the CEO role, you've had the opportunity to observe lots of CEOs in the past. Is there anyone you work with now or that's a good mentor for you, anybody you rely on for advice or...

Keller: Well, I've worked with really different people. It's really good to work with different people and learn what you can. But then you have to go do it your way and then go figure it out. When I went to AMD, I thought "Well, I'm going to try this and this and this and this." I actually hired a consultant because I was going to scale from a very small team to a very large team. Some of the experiments were great and some of the management books-- I asked a few friends at AMD like "How many management books have you read?" and the average number was like zero and I'd read 20. I'd read literally 19 more books than the-- like, one person said one. And I was like "Wow, I literally know more about management from books," and it's not like books are perfect, but the top 20 management books in the world, you just go to Amazon, you start Googling for management books and you read the reviews and some of them are amazing. Some of those books are written by people with 40 years of experience, they're passionate about it, and they poured their heart out into it and you can literally read it and then you try it and it's like amazing how often it just works. Even if you don't even know why, but you learn from that.

So, I don't know if I have a favorite. I really liked the way a whole bunch of different people worked. Like, working with Doug Fields was great. I liked the way he managed people and how he cared about them, and he thought really carefully about it. But Bob Mansfield was crazy smart, and he always knew when something was wrong. He always knew when somebody was making something up. How did he do it? He had a very clear style, which was very different from them. I really liked working with Elon. I learned so much from that guy, it's amazing. But my management style is way different. I'm like a person enabler, a gap filler, and Elon's like a force of nature. So, I don't know what to learn from that, except that I learned a lot and I changed a lot.

END OF THE INTERVIEW

