Oral History of Mark Himelstein

Interviewed by:
Douglas Fairbairn
Uday Kapoor

Recorded March 28, 2023

CHM Reference number: 2023.0048

Kapoor: My name is Uday Kapoor. I'm a volunteer at the Museum, and I am involved with the [oral] history program. We capture the [oral] histories of all the luminaries in the industry. So, welcome, Mark Himelstein. Got introduced to you by Rob Mains, who I worked with very closely at Sun. And I was at Sun for almost 20 years. I also was the guy that did the first SPARC chip. The Cypress Sun chip, so have a little bit of history from Sun. So, welcome.

Himelstein: Thank you.

Kapoor: So, what we like to do is-- and working with me is my colleague, Doug Fairbairn. We have done many interviews together. So we'd like to start with your early history. Where you were born, your early life. So, if you could maybe start with that.

Himelstein: Sure, I was born in Wilkes-Barre, Pennsylvania, Northeast Pennsylvania. Probably most known now for being 20 miles away from Scranton, where "The Office" was filmed. But it was a very sort of depressed area because it was a coal-mining town and they mined too close to the river and that got shut down in the '30s. I was born in 1960, and by that time it was a pretty depressed area. So, I grew up in this kind of this quaint little town that now sort of the biggest industry are colleges. They have universities and colleges locally. But not much else. There's not much industry.

Kapoor: Can you talk about your family a little bit? Your dad, your siblings.

Himelstein: Sure, It's not the first thing I share in general but I'm glad to share it. I'm not in any way embarrassed by this, but it was a tough upbringing. My father was schizoaffective, so he was mentally ill. My parents got divorced when I was seven. And my mom was a teacher in a Hebrew day school, and not much money coming in. And we lived fairly simply. But we were happy. We played games and laughed and had fun. And that really dominated a lot of my youth. My mother got remarried and then we had a flood. Agnes, Hurricane Agnes, you might remember, there was 18 feet of water in my house. And this was when I was 12 years old. I didn't really get along with my stepfather. I went away actually to a yeshiva, a Jewish school in my teen years. And my mom eventually got divorced again, and I came home for undergraduate. And so, I went to Wilkes, then, College, now University. And I have one brother who's older. My parents are passed at this point. My brother lives close-by in San Carlos. I got an undergraduate degree in Math and Computer Science. I went into the Dean of Admissions Office and I was thinking about Psychology or Accounting, and he said, "Well, we have this new program in Computer Science, why don't you try it out? We're a liberal arts school, you have to take that other stuff anyway. After a year you can decide." And you know, my first Fortran class I was hooked. You know? And kind of never stopped from there. Ended up getting the ability to work at the Institute for Defense Analysis in Princeton, New Jersey.

Fairbairn: Before you leap ahead there.

Himelstein: Yeah, sure.

Fairbairn: So, what year did you enter university or college at the time?

Himelstein: I had skipped a year in kindergarten, so it was 1977. And I finished in about three-and-a-half years, so I ended up going to graduate school in 1980.

Fairbairn: So, before you entered college, you were not especially inclined to math or science or whatever before that.

Himelstein: Well, my SATs were pretty skewed. It's really kind of funny. <laughs> My math scores were really good; my English scores were mediocre. And so, that's why the Dean said, "Hey, why don't you try this new program? And they had the guy who solved the four-color problem, John Koch, helped solve that. He was one of the professors they had hired. And another guy was from-- I think he was like undergrad MIT and graduate school from, yeah, somewhere in Southern California, and they had gotten these two young professors to come work there. And boy, I was really lucky. I mean, it was a very intense Computer Science program at a time where that stuff wasn't happening. And they bought good computers. We had a Hazeltine 1200. So, I actually got to type on screens instead of doing cards.

Fairbairn: Punch cards, yeah.

Himelstein: Yeah. And we had to do punch cards for some IBM class, but besides that, it was all typing.

Fairbairn: So, you got hooked on Fortran and that's-- and you got a degree in Computer Science? They actually had a Computer Science degree?

Himelstein: I actually ended up with two degrees. I <laughs> it's kind of funny, the Math Department was on the same floor as the Computer Science Department and I didn't want to leave, so I just signed up for as many math classes as I could, and the semester before I was supposed to graduate, the head of the Math Department came to me and said, "Hey, you're one class away from <laughs> a degree in Math. Please go sign up today." So, I did, and so I have a degree in both Math and Computer Science.

Kapoor: Yeah, so you did a BS in Math and Computer Science.

Himelstein: Yep.

Kapoor: Okay. That was in 1980.

Himelstein: Yeah, and you know, again, I ended up being lucky because Dr. Koch, who is the four-color guy, used to work in summers at the Institute for Defense Analysis, and he dragged me down there and I got to work on a Cray-1, number six in 1979. So, that was quite amazing. And so, I was very grateful.

Fairbairn: Have you seen the Cray-1 we have downstairs?

Himelstein: No, it could probably bring back some interesting good memories. <laughter> You know, these guys had CDC-6600s, and you know, all that stuff, you know? But it was fun. I mean, you know, I wrote a diagnostic program that survived me and went off to Los Alamos and Livermore and stuff like that. And you know, it was--

Kapoor: So, you had quite a traumatic childhood.

Himelstein: Absolutely. Absolutely.

Kapoor: And in spite of that, you know, you did well.

Himelstein: I blame my mother. My mother was just great. She was very supportive. In the worst of times, she would encourage us to laugh, and you know, she said, "If you can't laugh during hard times, then life's not worth living." So, I'm very grateful. She was just-- I mean, I couldn't have asked for a better mom.

Kapoor: We can talk about mental illness later. <laughter> My son has schizoaffective disorder as well.

Himelstein: Yeah, I'm sorry. Yeah, I mean, there's more people in my family with it, too. So, it's an insidious very hard disease, and as you see later in my story, there are some people who have had some physical illnesses, too, and I would take a physical illness any day of the week over a mental illness.

Kapoor: So, then after your BS in Pennsylvania, looks like you moved Westward to UC-Davis, how did that happen?

Himelstein: Yeah, so the people at Princeton encouraged me to go to graduate school. And I got the ACM book, you know, with all the different schools, and the people with the best machines and the best stipend was this branch of UC-Davis at Lawrence Livermore Labs. And so, I applied there. And because I was finishing school-- I finished school in December of 1980-- because of that, I really didn't get accepted into this program, because they wanted to fill all the slots. And so, they put me on a waitlist. And the only other place I applied to was SUNY at Stony Brook, and I got in there as well. And so, I was kind of waiting that fall to decide where I was going to go, and then somebody had dropped out and they said, "If you want it, it's yours." And so, out I came.

Fairbairn: So, you were actually attending a school at Lawrence Livermore Labs as opposed to--

Himelstein: Correct, they had a branch, they called it Teller Tech, so Ed Teller set it up for sort of the frustrated professors in the lab who wanted to teach. And they only had two degrees. One was Applied Math, basically physics. And the other was Computing Science as they called it. And so, I was in the Computing Science program. It's like a very small building on sort of the east side of the lab, and we got to go ahead and have great professors. You know, like the guy who wrote a bunch of the OSI stack for Springer Verlag, and the very famous Oktoberfest book from 1977 or something, was one of our professors, Dick Watson. So, we had some just incredible luminaries who were teaching us, and

<laughs> it was quite something. And then because I had had already a security clearance for working at Institute for Defense Analysis, I got my clearance fairly quickly to work in the lab, and I worked on a wide array of things there as well, and that was fun.

Fairbairn: So, in your first two jobs you had access to the world's most powerful computers. <laughs>

Himelstein: Absolutely. And you know, I talk to guys now working on RISC-V about vector, and they go, "Oh, what was it like, you know, working on a machine that had Vector?" And you know, they had-- you know, there's things about staging data and now all that stuff is hidden behind caches and streaming and stuff like that. You know, back then, you know, Seymour had S-registers and you had to fill those darn things in order to fill the Vector registers, right? So, it was quite astonishing. I mean, they had, you know, Seymour had variable length instructions. He had 16-bit instructions called parcels. And we have variable length instructions now, RISC-V, 16-bit, 32, maybe 48 for sure 64. So, I was just very lucky. I was very grateful.

Kapoor: So, you went to Sandia Labs.

Himelstein: For a short time, I went to Sandia Labs because I didn't really like the jobs at Lawrence Livermore. And so, I got a job at Sandia and continued on to my PhD. I had passed my quals, and I started in on the PhD, and did that for a year. Tried to bring timesharing to their Cray-1, which already had been at Lawrence Livermore Labs, but the Sandia Labs Cray-1 was still running batch. And so, I spent a lot of time down at Los Alamos and in Albuquerque and I got frustrated because I wasn't doing-- you know, I wasn't doing a whole lot of technical work. And so, I lasted about a year. I also got bored with classes at that point. I could read the books as well as my professors could, and so I just went down to Silicon Valley and never left.

Kapoor: Yeah, it's interesting that you went to Megatest.

Himelstein: I did, you know, Richard Swan from CM Star was running their engineering and so that's what attracted me. I mean, I've always been attracted by rational, incredibly brilliant people. And Richard was incredible to work with. And they had a tester per pin, which was unheard of in those days. And they had nobody who understood compilers. And I did. And so, I came in. And because a lot of their stuff had to do with compiling down some high-level language down to something the tester could understand and stuff like that. As well as they needed. I mean, we were running UNIX in those days, and they needed support with C-compilers and assemblers and all that. And so--

Fairbairn: And the CEO, the Founder was Steve--

Himelstein: Bisset.

Fairbairn: Bisset.

Himelstein: Bisset. Yeah, Steve Bisset.

Fairbairn: We did an oral history with him many years ago.

Himelstein: Yeah. <laughter>

Fairbairn: That company had quite a reputation for its lifestyle. It must have been quite a shock moving from defense-related work to Megatest. Seems like the environments might be quite different.

Himelstein: Less different than you might think. <laughter> And I would say enough said on that. <laughter>

Fairbairn: All right, well, learn something along the way.

Himelstein: Yes.

Fairbairn: So, was there any particular accomplishment, breakthrough, important happenings at Megatest that you'd like to talk about?

Himelstein: Well, for Megatest, the important thing really was a tester per pin technology. And they had a real symmetric multiprocessor, you know, operating system and piece of hardware running on 68Ks, multiple 68 Ks. And that was pretty unique in those days. The only other people who were doing that were people like Sun. Right? And so, it was quite amazing. And again, I helped out heavily in the compilers, but I also helped out in the operating systems, and got involved in operating systems at that point. So, you get a remote file system, when everybody else was doing NFS and this and that, I did one at sort of user level, and made that available before the days really of open source. So I got to do a bunch of just fun stuff, and just helped the company get product out there.

I think part of it was understanding regimentation. And I think some of my early upbringing of like we didn't have weekends off when I was in high school. It was just part of the whole thing. I really wasn't like everybody else. They looked at weekends as sacrosanct. Well, that was really good training for Silicon Valley, right? Because--

Kapoor: Yeah, that was the work ethic at that time.

Himelstein: Exactly, in those days, you know, you were there 16/18 hours a day. You ate dinner and other meals with your colleagues. Even when I was sick, I was on kind of full-time, you know?

Kapoor: Well, those are folks that were designing chips and we had large teams, and so that was the work ethic. You know?

Himelstein: Yep.

Kapoor: There was no weekend. You know?

Himelstein: Yeah.

Kapoor: And I remember buying the Megatest and various companies and listening to the sales folks and all that. So, I think you were the provider of all that background for tester per pin and all that.

Himelstein: And also, I think in those days the VLSI designers and other folks didn't have a whole concept about things like “build” systems and stuff like that. And there's a systems' gentleman, a guy named Ken Matusow -- one of the few people I think in the world who has a degree in Systems. He got it from Wisconsin. I can't remember which school it was there. And so, he looked at things more broadly and he helped me understand things more broadly. And we helped the rest of the crew understand things more broadly, so you could actually have one button installs of work that was being written by, I don't know, 50 people. And that was pretty incredible as well. When you start taking a look at complex systems, the integration's tough. I mean, you take a look at Boeing when they're building an airplane, it's hard to make it all fit together. And the same thing was true for the MegaOne.

Kapoor: So, then you moved on to MIPS, I guess.

Himelstein: Yeah, a couple of my colleagues went there and so went on to MIPS and again just an incredible crew of people. I ended up working for a gentleman named Larry Weber and John Mashey, who you guys know. And again, got to do incredible things. Another gentleman, Earl Killian who is an architect there, I've now dragged into RISC-V, so he's now the Vice-Chair of our unprivileged committee, mind like a steel trap. And he enabled us to go ahead and run MIPS instructions on a VAX at about 600K per second, which was astonishing in those days. And I created the debugger that allowed you to single step, either the MIPS or the VAX instructions and be able to see, right?

I worked with Fred Chow, who every famous in the optimization stuff. Worked for John-- well, was a student of John Hennessy's at Stanford, and then was part of MIPS, and we did cross-module optimization. One of the first implementations where you can compile things separately and bring them together and then do the optimization so you can do cross module register allocation and code hoisting and all that kind of stuff. So, just able to do an amazing amount of things. And then another thing was that in those days we had 8K direct map i-caches. And you know, silly benchmarks like N-ROTH didn't fit in an 8K i-cache. So, I wrote code that took profiling data and reorganized the code in order to make it fit in the cache.

I think the point of my saying these things is just there was an unending list of opportunities to excel and do something interesting. And if we actually did software patents in those days, I probably have 30 patents from that period of time. As it is, I have one patent from that period of time, and a few follow-ons, because I created the fastest algorithm for bite swapping using lower [?] lefts and lower rights and so I have bi-Endian patents. But it was a great group of people. And those people in the end brought me into RISC-V three years ago. They just said, "Hey, would you be willing to do this?" And so, they're at a company called Aril now. The Aril which is the piece of the pomegranate, the little seed. And those are relationships I built in the '80s, so.

Kapoor: So, would you say that with your background in math and computer science are you a computer architect or a micro- architect or a software developer? How would you classify your expertise?

Himelstein: So, I think that my biggest expertise is synthesis. I can take 100 facts together and then put a plan together that's got a 95 percent probability of working. I can gather stuff from a lot of people, and you know, you develop that more as a consultant over the years. And probably a third to a half of my career is consulting. And so, I honed that. I think I have expertise from the application layer all the way down to the ISA. I am not a VLSI designer. I'm not a micro- architect. But I have passions about pieces of those things. And we can talk about those in a little bit.

But my last job before RISC-V was I was helping FICO with anti-money laundering software. Right? And they had bigger customers than they've ever had before. They had to be able to do something that was fail over and you know, fail back and just reliable. And they didn't know how to do it because they were applications developers and scoring developers, right? Data Scientists. So, I think I have an eclectic set of skills and I find a lot of things interesting. I don't feel like I have to work on something in particular. I usually find something very interesting no matter what I'm working on.

Fairbairn: So, getting back to the Megatest to MIPS transition. Was there anything that drove you away from Megatest or was it just new opportunities at MIPS? What was the background of the change?

Himelstein: I think all companies have ups and downs. And we know that the radical ups and downs of various pieces of the industry are they're steeper, they're steeper up/steeper down, and I think we were in a steep down at that point. And it was affecting Megatest. These buddies of mine went over to MIPS and said, "Why don't you come over and talk to us?" and I went over there, I was really excited.

Fairbairn: What was the size of MIPS at the time?

Himelstein: I was Employee 45.

Fairbairn: Okay, so it was pretty early days.

Himelstein: Very early, yeah.

Fairbairn: And you were there for quite a number of years, six years?

Himelstein: Yeah, yeah, absolutely. So, got to do an amazing set of things. And I eventually ended up running the operating system groups for our desk size, our big servers, etcetera. Did all the position independent code for dynamic shared objects. So, just, you know, a wide variety of things. So, very, very cool, very exciting.

Kapoor: So, the next step after MIPS, could you explain yourself? It says Apple, so anything--

Himelstein: Well, actually I ended up doing consulting in-between and so that's when I started my love for consulting. Consulting for me was really wonderful in that I felt like I had impunity. I can go in, they're paying me good money, I can go in and tell them the truth. <laughs> And it's like, "If you want to accept it, that's fine. If you don't, that's okay, too. You're paying me either way." So, I just start-- I started initially consulting for DEC, and I did the floating-point completion routines for Alpha, because you know, I don't know if you know Dick Sites. So, Dick says, "You can blame me for this," he didn't want to put the d-norms support into the chip and so I did d-norms in software. They also had some of the status bits for floating point that were adhered to, and I dealt with those in software. And I did it on the Ultrix side because there was a whole bunch of people doing it on the VMS side as well. But I worked for them as a consultant probably for about two years. And so, that was exciting, too. I got to do some great things. Got to help them a lot.

I did exception handling over the years as well. What I didn't say was when I was at MIPS I did it as part of ADA and we crushed the benchmarks out of CMU. So I helped DEC with their exception handling and moving to 64-bit. You know, and all those things. Plus, you had to put fences in because if you wanted to do any kind of exception handling, if you wanted to do any kind of debugging, you know, their stuff was out of order and you had to put fences in to tell where things were at. So, all that stuff, again, I got to work on really interesting technology and got to help them be successful with that.

Kapoor: So, you were a consultant for four years, roughly.

Himelstein: I've been a consultant on and off through my career for probably, I would have to imagine 20 years. But under different auspices. So, Himelsoft, initially, and then Heavenstone. Heavenstone is the English for Himelstein <laughter>. So, Heavenstone, Incorporated. And so, I've had Heavenstone since 2004. But after doing consulting for a while, I consulted for Apple. And then in there we did MAE, which was a Mac on a like HP or Sun or other box, doing it with simulator/emulator depending on how you look at it. And you know, got some patents there as well. But I ran that group for some period of time, and then they asked me to come in as a director and not only run that group but to run their Tools Group, too. So, I ran their Tools Group and the MAE group for some period of time. But that was kind of like the worst period of time in Apple's history. It was like Spindler imploding, you know, the follow-on regimes that were just didn't understand their business. And that's where I met Ike Nassi, who obviously is very ensconced in the Computer History Museum. And he's still a friend to this day.

But it was a very difficult time, and so I stayed maybe a year as an employee and probably two years as a consultant there. And again, had a lot of fun…how do you go ahead and test a 68K from a software perspective? How do you test Mac OS from a software perspective automatically? And you know, we were doing it. And you know, we'd single step with P-trace simulation and compare register sets and you know, we'd have compression and blah, blah, blah. So, we did a lot of interesting things.

Kapoor: So, I wonder how the, you know, the security protocol at Apple is unequaled. I mean, there's-- they-- so I wonder in those days how they handled a consultant relative to--

Himelstein: You know, I was off in the Unix OS land. And that was really not in Mac OS land. And so, I think there was a little less reign over that. But you know, as long as I did my work, that's what they cared about. Right? And that's the great thing about being a consultant. It's like, it's really clear. You agree on what you're going to do. You do it. They pay you. They're happy; you're happy. Right? And it's a great thing. And along the way, I just-- I got better and better at helping teams out do things. I was a manager at MIPS. At the end, I went back and forth. Every time I'd go back and forth, people would say, "Oh, it's such a shame that you're not doing individuals contributor work, and now you're doing management work." You know, it's, "What a loss," right? And then I go from being a manager to being an individual contributor, and they, <sighs>, you know, "It's such a shame you're not being a manager anymore." But I went back and forth a number of times over the years.

And then, I think as a consultant, you slowly hone your skills in that in a way that you can't any other way. Because you change your language, right? It's more about, "Okay, I suggest," as opposed to "I say," right? It's, "What do you think?" And then you have to sandwich everything in a compliment, right? It's like, "Wow, you did really great," like, "What about this?" You know, and then <laughs>, "You know, that really was great work." So, you learn those things and that helps you to be a manager. And I think that, the times when I've I think stayed in as a manager in a company too long, I'd forget those things. So, I eventually wrote a book, and that was partially--

Fairbairn: Well, getting back to your MIPS. Since you'd been there for several years, what caused you to move on and was there--

Himelstein: Yeah, I think we, too, had a number of ups and downs. There's plenty of people have written about MIPS and almost picked as the replacement processor for the 68K at Apple, and that got killed at the last moment because of a leak. I mean, there was just a ton of those kind of things. They kind of initiated the first real licensee agreements, right? So, they licensed the actual design of MIPS to places like DEC and IDC, and you know, so on and so forth. It was a different model and that was all very exciting. But financially they never did really well, you know? You know, sort of the high point was when they went public. And after that it was just brutal. And--

Kapoor: Very competitive pressures like from Sun.

Himelstein: Yeah, competitive but also, I think there was-- in hindsight, I think, it's always hindsight. Mistakes were made and we could have done better. But the people there were brilliant. The technology is brilliant. And you know, those people have gone on to seed many of the leading places in the world.

Fairbairn: What would you say is one or two of the most critical decisions which led it down the wrong path versus the right path? I mean, yes-- history is always good-- easier, but it's also a good lesson sometimes.

Himelstein: Pardon me, let's first say, you know, some of the things they did really well, right? Is my sandwiching stuff with a compliment here. <laughter>

Fairbairn: Let's start with the positive.

Himelstein: It changed how the world looked at compilers. Right? The hardware/software trade-off that Hennessey talked about all the time was evident every single day of lives. And we changed the investment that not only we put into compilers but places like Sun Microsystems and IBM. Compilers got respect because it was a lot cheaper to go ahead and improve your compiler than go create a new chip with a longer pipe or more stages or more dual issue whatever kind of stuff. And so, I believe we, along with people like Sun, along with people like IBM, really changed everybody's viewpoint on that.

There's a couple of jobs that I've had in my life that I think changed the world, and that was one of them. And that was why. It really changed that, you know, balance between hardware and software. When we were designing new chips like the R4K, unless people could prove that a) there was functionality that was just totally missing like, you know, I don't know, watchpoints or something for debugging; or you got double-digit performance improvement, they weren't going to do it. Right? So, it also had an incredible means test. It had this reasonability about them. And I was in the design meetings for the R4K and it was like, "Yeah, go run it on the simulator and if you don't see improvement in the benchmarks we care about, we're not doing this." That was pretty amazing. But it's also why they fixed things that were broken. I mean, everybody believes, "Okay. SPARC, register windows, bad." Right? "MIPS, delay slots. Bad," right? <laughs> You know, and we understood that and we admitted it and we fixed it with like branch likelies, and stuff like that.

So it really did change how the world looked at instruction sets. It looked differently, you know, at how you do compilers, and that was pretty amazing. There had been a lot of great ideas in the past with bad businessmen, right. <laughs> Just because you have an idea and a good product doesn't mean necessarily that you know how to sell it. It doesn't mean you know how to make a market and so on and so forth. In the end, they really, in my mind, didn't play to a market that was going to go ahead and make them a lot of money. It was simple business, and eventually they sold off to SGI, who did have a marketplace, and, you know, they had their own problems later on. But they sold off to SGI, who's really towards graphics and computing at that level.

So I would attribute it to just bad business. I don't think from a technology perspective, at least, you know, in my rearview mirror, I don't think we failed in any way, shape or form. I think we did, we hit it out of the park. But that doesn't necessarily-- I mean, look at how many products in the past didn't do well because of business as opposed to technology, and, frankly, MIPS instruction set and compilers have affected the world since then. If you take a look at the RISC-V instruction set, I believe it's the most elegant instruction set since early MIPS. They added a bunch of stuff later on to MIPS, which I think, you know, made it a little less pretty, but, you know, I mean, still incredibly useful, and lots of companies made huge amounts of product, Cavium, et cetera, that made lot of product on MIPS, and it still has an impact today. I mean, there're still a lot of people using the MIPS instruction set, but MIPS themselves have gone over to RISC-V, and it just feels familiar to me now, feels at home, so...

Fairbairn: So, things changed. You moved on to DEC. Was there any particular thing that you wanted to highlight about the work at DEC?

Himelstein: Well, I think I talked about it already…. doing the floating-point completion routines, doing exception handling, so I, you know, I just helped them out. There was a gentleman named Ayub Khan who ran the compiler group for Ultrix, and once he and I got to know each other, he just wanted me to go do more and more stuff for him, so...

Kapoor: But that was a consultant, right?

Himelstein: As a consultant, yeah.

Kapoor: Okay.

Fairbairn: So that ran its course and you moved on to Apple?

Himelstein: Well, I probably did a couple things in between but yes, I ended up at Apple, and again, that was a consultant who worked for me at MIPS brought me in to Apple, to help him.

Fairbairn: So which group in DEC were you working for or with?

Himelstein: It was the Ultrix folks working on Alpha.

Fairbairn: So they're based in Texas or where was-- back East?

Himelstein: No. They're in Nashua.

Fairbairn: Nashua? Okay. They were--

Himelstein: Yeah.

Fairbairn: So were you working remotely or were you...?

Himelstein: I was. I actually kept my machine in DEC WRL (Western Research Lab)] <laughs> in Palo Alto, and just logged in through a modem, did all my work that way.

Fairbairn: Early remote worker.

Himelstein: I'd been a remote worker for so long as part of my career and very happy about it. My dogs are happy about it, I'm happy about it.

<laughter>

Himelstein: It's really good.

Kapoor: So your residence was always here.

Himelstein: My residence was always here. I came into California into Livermore and then two years later ended up in Silicon Valley. I've been in one place or another in Silicon Valley since.

Fairbairn: Did you-- between MIPS, DEC, Apple, DEC, it sounds like you perhaps were not really sort of immersed in the organization, but between MIPS and Apple were you-- was the style of the organization or the company shockingly different or were you-- so didn't matter at the level you were operating at or...?

Himelstein: Well, again, I consider myself a very honest person, and when somebody comes in and says, "Hey, can you do that in three months?" and you say, "No," and they don't like it, like, what happens? Well, if you're a consultant, it doesn't matter, right. You go in and say the truth and it goes on, but in companies like, you know, that are big, it doesn't make a difference how big, they're trying to meet some goal, and sometimes they don't really honor the truth, and I think that there are different styles too. I mean, you know, Apple-- I think there're a lot of people who had agendas there and that's okay, I mean, but it wasn't for me. It just really wasn't for me.

The best corporate environment that I ever lived in was the early days when I was at Sun because I always used to say at Sun you didn't have to worry about somebody stabbing you in the back because they're going to come right at your chest, and that I can handle. It's honest, at least, right. You know, that you know they're coming. You can have an honest discussion, and the leadership at that point, Ed Zander, liked those discussions. He let them happen. I was in many meetings where he let them happen, and you can come out with a really good solution if you're just honest, right, and you're straightforward about things and you're realistic, and so I, from a corporate environment, I probably got along better at Sun, and for a lot of people it didn't work because they couldn't handle people coming at them full frontal, right. But for me it was just fine because I didn't take anything personally. I--

Kapoor: You probably enjoyed it.

Fairbairn: <laughs>

Himelstein: Well, I don't know if I enjoyed it, but I will tell you, I mean, I have an attitude I don't take things personally. If you come to me and say, "Hey, Mark, I need you do something different because you did this and you pissed this person off," or whatever, and I go, "That's okay. All right. Give me the details. Let's talk about how it can be better."

I think I learned that early on through some of the mental health stuff and family therapy and all that kind of stuff. So I learned not to take things personally, and for the most part I can avoid being defensive. Yeah, there's once in a while. I'm human, just like anybody else, but that makes it so much easier. At Apple, it was still-- there was a lot of people telling other people what they wanted to hear as opposed to what they should've heard, and I think that changed later on when the whole Steve Jobs crew came in. I mean, it was a different dynamic that I'm not familiar with, but it certainly wasn't the dynamic that--

Kapoor: But you didn't overlap with Steve Jobs.

Himelstein: No, no. I left be-- like right at the beginning of Gil Amelio. Yeah.

Fairbairn: So it's-- move on to Sun.

Himelstein: Yeah. So I ended up doing a little consulting in between for a place called Magic Circle Media that was tracking objects frame per frame in a video and stuff like that, and then my buddy Larry from MIPS came after me. Again, Larry Weber was my manager at MIPS. He came after me at Sun. He was running the Tools group and so I became the Director of Architecture for Tools, and, you know, I did my thing. I came in and was honest, <laughs> and being a director of architecture it's easier to be honest, I think, than if you're a product-oriented manager. We accomplished some things and very quickly they were looking for somebody to go run Solaris, and I worked for Rich Green and he hired me to run Solaris. That was, I would say, another one of the earth-breaking, you know, ground-breaking jobs in my career. We did things like ZFS, DTrace, Zones, which are, I think looked on universally, as just an incredible set of things.

When I came in, my boss's boss, you know, I met with him and I said, "Okay. What do you want me to do?" and "What's success in a year?" and he said, "Two things. One is I want them to innovate," because they had just gone from SunOS to Solaris, and it took them 10 years to really swallow that. They had really turned all the innovators into bug fixers. Then the second thing was that they wanted some respect because the team wasn't very nice. <laughs> I remember one of my first meetings with the kernel guys and they said, "What do you want from us?" It's like, "Same passion; more respect. That's what I want." And so, I enabled them to go do a bunch of stuff. I remember one guy came in and I wanted him to work on reliability because there was a lack of an infrastructure around reliability inside of Solaris. I said, "I need you to be the architect for this," and he goes, "Well, if I'm the architect how am I going to fix all the bugs that those hardware-- software guys in the hardware group are putting in?" I said, "Well, if you want to do that, that's all you're ever going to do," and you just saw a light go on in his eyes, and so we just innovated. We were at that time competing against Windows 2000 and we had to tell a story and we had to have compelling stuff. It was the middle-- still in the middle of the dot-com era, and we were coordinating the work of 2500 people around Sun, 500 of which reported to me, and was everything from Java to the work to support domains and the E10Ks and so on and so forth, and it was fun. I mean, we--

Kapoor: So was there any discussion of Solaris on x86?

Himelstein: So I owned that too. I owned Solaris and x86. I also owned Secure Solaris, which was used by the FBI and folks like that.

Kapoor: But that didn't happen, right?

Himelstein: No, x86 Solaris happened, and we supported it. We had bought a company before I showed up there in L.A. that did a bunch of stuff, and in fact, we signed up for doing Itanium with Intel. We had red books and all that kind of stuff, and people used to ask me. It's like, "Hey, Itanium seems like amazing. Is anything going to beat it?" I go, "Yeah, there's one chip that's really going to kill it." "What? What chip is that?" "It's called x86," and <laughs> that was the truth. But just like every other company, hit 9/11, ups and downs. What are you going to do with your budget? And I basically said, “Look. If you really want to save Solaris for SPARC, you got to kill Solaris on x86.” So, they killed Solaris at my behest on x86, and I told all my guys, “Put it somewhere safe, because it’s coming out again. Don’t worry.” Sure enough, like--

Kapoor: And then Linux happened.

Himelstein: Well, not only that, but nine months later or something we resurrected Solaris on x86. I’m the person who forced the Solaris guys to put all the Linux libraries and tools into Solaris, and the guy who was running that stuff came to me and said, “I don’t want to do this.” I said, “It’s fine. If you don’t want to do it, I will find somebody who does,” and he went back and did it. So, we were one of the first ones. That was a proprietary operating system. We had a slash-- you know, user/Linux, and it had the whole Linux tree underneath it, so we just did a huge amount of stuff. We were leaders in things like protocol, so IPv6. We were leaders in security. We just did tons of stuff. You know, there was Secure Solaris, and it was very popular, and it knew how to do isolation that, many OSs still don’t know how to do today, and so it was a grand time. We just accomplished a lot of stuff. Just I did it for a long time. I did it for four years, and evidently, I was the one who lasted the longest doing that job. Then I decided I wanted to go do something else and I went and did something else and then all the regimes changed. So, Jonathan Schwartz took over and stuff like that and he and I never saw eye to eye, even before I was running--

Kapoor: So you overlapped with him?

Himelstein: Yeah, I did. I was involved in evaluating Lighthouse, which was how he came into Sun. So, he and I never saw eye to eye, and it was clear that when he took over, he wanted people who were going to say “Yes” to him, and that’s what he got, and I left.

Kapoor: I see.

Kapoor: So, lot of people later blamed Sun for not supporting x86. You know, Solaris on x86 properly or whatever. Lot of the users were very unhappy.

Himelstein: Yeah.

Kapoor: And they said that the Linux, rise of Linux with Intel and all that happened because Sun didn’t follow through very well. So, I don’t know what’s your comment.

Himelstein: Yeah, I don’t really agree with that.

Kapoor: I see.

Himelstein: You know, I talked to the CEO of Red Hat in ’99 and it was clear that there was something going on with Linux that was beyond technology. It still is not the best operating system out there, but people feel like it's their house, right. It's going to get better over time. If they really need to fix something they can, and they're not beholden and locked into a vendor who's going to charge some huge amount of money. And that mentality of, "It's our house. It belongs to us" is one of the fundamental reasons why RISC-V is successful as well. They feel the same way about RISC-V.

Linux grew up in a community and it built this following because people could play. They could do what they wanted to do and it also made money. Like it actually ran applications and did all these things and it constantly got better. I mean, I bought a Tesla in 2019. What do I love about the Tesla the most? I wake up and there's a new software release and there's new stuff in there. Same thing's true with Linux. I mean, new stuff comes out every year. It gets better and better and better. People have variations to address issues that the main Linux group doesn't want to address. That's okay. They don't mind. It's open source. They can do that. So, I think that Linux was going to happen no matter what. You know, at that point Gerstner had already started painting penguins on corners in San Francisco. It was a sociological movement as much as a technological movement, and it still is.

Kapoor: You know, this thing of support for x86 on Solaris, that debate went on after Oracle took over Sun.

Himelstein: Yeah.

Kapoor: And I was actually in some of those discussions because I was hearing from our users that, "Please support--" you know.

Himelstein: Yeah.

Kapoor: <laughs> And--

Himelstein: That's why it came back. After I killed it.

Kapoor: <laughs> That's right. <laughs> Okay.

Himelstein: Look, there's good technology out there. There's some things that Solaris did better than anybody else. They had basically parallelization all the way down into the drivers, right. If something stopped working well on Oracle, it was because somebody screwed up a driver inside of Solaris. And we hopped on those things really well.

The same thing was true on x86. The same technology was there. They had locks and so on and so forth, so you really weren't stuck with this almost sequentialness of execution that occurred before Solaris started doing its thing. And boy, that's just amazing! So the impact was better performance by far. Again, and we also did some amazing things, so we redid the file system. So, Veritas owned the file system business, and the day that we announced that we were fixing some of the weirdnesses in the files system inside of Solaris, Veritas's stock went down by a third. Because people understood that it was just, you know, Sun's not paying attention to it for a while that made that happen. And then on top of that, we went to ZFS, and ZFS is still a thing, right, open ZFS and so on and so forth. And boy, I mean, that sort of changed the whole game around those things.

Well, Sun did that in place after place after place, whether it was fiber channel or resource management or performance analysis. DTrace has become sort of the standard for how you do performance analysis, and everybody's got a version of it. Linux has got its version with STrace and a couple of other things; Apple's got its own DTrace. So, it's really, we changed the world in so many ways.

Kapoor: Yeah. Some of concepts that came from Sun.

Himelstein: When I came in there, they had no plans in different areas, so I created six different programs, some in discrete things, some in more attributes. So networking, liability, you know, data, blah, blah, blah, and I forced them to go ahead and actually have, you know, "Give me a picture of--" I said, "Don't give me these 30-page presentations. Give me a picture of what you see today and what I will see in three to five years if you actually accomplish what you want to accomplish." That also enabled us to go raise money internally in order to get the work done.

Kapoor: So since we're talking about Sun, do you have any views on the Java developers and also the impact, you know, the relationship with Google, for example?

Himelstein: Well, I owned my own Java group when I first took over Solaris because the main Java folks were all working on an x86. Eventually we handed our Solaris and SPARC Java people over to the main Java group. But look, the concept of having some kind of architecture-independent ability to execute has lived for a long time. I mean, Forest Baskett, right, p-code, right. There's tons of cases, and it was very clear that we're heading to a world that does that. So maybe today more of it's done with scripting languages. Either JavaScript or Python or Ruby on Rails or whatever it is. But those things don't need to be recompiled, and that's really important, and we've lost some of it over time. So, if you take a look at most of the stuff on Android it's actually compiled natively for Arm. It's like, "When did that happen?" So it's almost like we've fallen back a little bit. But, for a whole bunch of these applications you can get as good performance in something that's doing a JIT compiler and stuff like that, as you can for native stuff. Obviously if you're doing something fancily with rendering or data processing or inference engines, then yeah, you want to be native, but if you're doing something that's a user application, why would you ever go native, right? I've got to believe it just increases the maintenance and development cost, and I think Google feels similarly. Yeah, they're a very strong supporter of Java and other languages.

Kapoor: Okay. So beyond Sun then, can you talk about what happened?

Himelstein: Yeah. So I left Sun and I went to a Paul Allen company for a little bit and I think the same thing happened with me because I ended up in a company that really didn't want to hear the truth, and so I didn't last there too long. Then I went off to Infoblox and got them to the point where they had a product that was viable, got them to actually file patents, which was important. In the end, they couldn't have gone public if we had not filed the patents that-- and I told them in a board meeting. I said, "The most important thing that I probably will have ever done for the company is file these patents." But after a year they changed chief executive officers, and again, I didn't really click.

Fairbairn: What was the business of Infoblox?

Himelstein: They were doing DNS routers, DNS servers, and it was really great. I mean, we had failover, failback. The biggest problem with DNS is when things get out of sync and we had things all syncing up and we could-- we had farms of DNS servers where you can just constantly keep on pulling the power on them randomly and it still stayed up and served, and that's really what people want because you have to be able to resolve names.

Kapoor: So and then beyond that you went to, what, Quantum?

Himelstein: Yeah, I went to Quantum as a consultant, and then they tried to bring me in as a full-time and to take care of the "de-dup" group and I did that for a while and then turned into their chief technology officer. But they too were going through some very difficult financial times and so I ended up leaving them And for a while I joined one of the Hadoop startups and then eventually started my own company with Fred Carlson, Fred Carlson.

Fairbairn: Is a hardware-- you know, did Quantum think of itself as a hardware company and software as a secondary thing or is that--

Himelstein: Well--

Fairbairn: What was the nature of the--

Himelstein: They had this ongoing revenue stream from tape formats and tape and stuff like that, but I think they looked at the future. They also had a hierarchical file system. The guys in Minnesota, these were old Cray guys. And they were doing some interesting things, so lot of their stuff was used by major league baseball and other things because it enabled you to-- and by large media companies-- it enabled you to keep stuff out on tape but refer to it in the file system, be able to bring it back on the fly. So that's where I lived mostly, inside of Quantum.

Kapoor: Mm-hm. So that was a short stint, right?

Himelstein: That was a short stint.

Kapoor: Yeah.

Himelstein: Yeah, and then, again, I went off and helped a Hadoop company for a little bit, but that was not meant to be. So I ended up, again, starting a company with Rick Carlson called Graphite Systems doing very, very parallelized access to flash. So it was a big data machine, had about a hundred terabytes of flash, and so we had about 6,000 devices. Think of NVME on steroids. So the OS actually controlled all 6,000 devices, and the concept was if you wrote in parallel-- I'm sorry, if you wrote distributed you could read in parallel. So we did queries for Twitter where they had 200 machines doing a query that would normally take them about 30 minutes on 200 machines and we could do it in 8 minutes on one machine. So--

Fairbairn: There must've been far more applications than Twitter. What was the...?

Himelstein: Well, it was all about databases, so things like bongo and catch basin. You know, was all database. Oracle. We kind of redid a database from-- I redid a database from scratch that did everything very much parallelized, and we ran it on an AWS and be beat Oracle native on TPC, for example. So it was a combination, again, a hardware solution with a software solution. I think the technology there was really good, but it also was at a time before large capital-intensive investments were being made by VCs. So came to 2015, people consolidating portfolios. Sequoia said, you know, "Sell this thing because we can't--" I told them. I walked in the first day and said, "We need 50 million to do this," and they gave us 20.

<laughter>

Himelstein: And so they sold it off to DSSD, which was part of EMC at that point.

Fairbairn: What is DSSD?

Himelstein: That was the object-oriented thing by the guy from Sun. I can't remember his name.

Fairbairn: Anyway, part of EMC.

Himelstein: But it became part of EMC and they basically wanted my guys because we-- basically what we did was we knew how to use Ethernet as the transport for these 6,000 devices. And our guys knew how to do this, both in the ASICs, as well as in the software. So they basically bought the staff and paid Sequoia back for their investment.

Kapoor: Okay. So then EMC acquired Graphite.

Himelstein: EMC acquired Graphite and then Dell acquired EMC, and then Dell ended DSSD, which therefore ended Graphite.

Fairbairn: So did you stay through those acquisitions or part of them or...?

Himelstein: No. No, no, no.

<laughter>

Himelstein: I moved on.

Fairbairn: From the first one? <laughs>

Himelstein: Yeah.

Kapoor: So the Heavenstone started at that time or...?

Himelstein: Again. Yes.

Kapoor: Okay.

Himelstein: It started in 2004 after Infoblox, but continued through that time. But when I consulted for Quantum I was using Heavenstone, and so on, so forth.

Kapoor: So maybe we can transition to RISC-V then.

Himelstein: Yeah. So I finished with EMC and I was in a startup for a very short period of time but my wife was diagnosed with cancer, and so she was diagnosed in 2012. I sold the place to EMC in 2015, but by December of 2015, it was clear that she wasn’t going to make it, so I took the year off and took care of her until the end of 2016. We made the best of it we could. She was a psychotherapist. We processed a lot of grief. She had a short respite. I took her on a cruise around the British Isles and we saw Springsteen at Wembley, Vivaldi at St. Martin in the Fields. We made the best of it we could. But luckily, with the sale of Graphite I was able to actually take that year off pretty easily and go do that. And then I ended up starting consulting again, through Heavenstone for FICO and working on the stuff that I told you about before. So I’m working at FICO. There for about three years. You know, get a call from--

Fairbairn: That’s a long time for your... <laughs>

Himelstein: Yeah, it is. I usually don’t do that, but in November of 2019 I got a call from the MIPS guys saying, “Hey, would you throw your hat in the ring for this?” and I said, “Okay. Sure.”

Fairbairn: And that specifically was for the CTO job or for the...?

Himelstein: For the CTO job at RISC-V, and then COVID hit. <laughs> So every couple months I get a call from them, “Hey, will you talk to couple more people?” “Sure.” And yeah, at least in my past, when I end up talking to 20 people, like, something doesn’t go right, right?

Fairbairn: Right.

<laughter>

Himelstein: But things kept on going well.

Fairbairn: Something’s wrong there.

Himelstein: They'd call me and say, "Oh," because this started with 30 people applying, and again, you know, wasn't something that I really cared one way or the other about it. Didn't really know whole lot about it, and they call and they say, "Well, we're down to 10. Will you talk to some more people?" "Sure." "Hey, we're down to the final three. Would you talk to some more people?" "Sure," and then finally in May they called me and said, "Hey, we're making you an offer today." Go, "What?"

<laughter>

Himelstein: So I said, "Yeah," so--

Fairbairn: So people don't normally go out and search for CTOs. They're normally part of the founding group or they come up inside or whatever. Tell me a little bit about the RISC-V organization.

Himelstein: Sure.

Fairbairn: Tell me a lot about it. Just--

Himelstein: So--

Fairbairn: And how this evolved and I-- I know the early roots, but tell me how that's--

Himelstein: Yeah, so--

Fairbairn: --come about.

Himelstein: So RISC-V started at Berkeley. They started doing a bunch of seminars on it, telling people about it. There grew interest; 2015, 2016 they started this foundation inside of Linux Foundation. And they moved along without a CTO and they had a president and went on for a number of years and they switched presidents. And they got this president who's, I think, more on the marketing side and she wanted a partner on the technical side, and--

Fairbairn: Who is this?

Himelstein: This was Calista Redmond.

Fairbairn: Okay.

Himelstein: And she's an amazing boss. She's just done amazing things for RISC-V. And most of the people who are like micro-architects or ISA architects or whatever, even compiler or operating system architects, they're working for startup companies putting out RISC-V products. So I think it was not something that was obvious to a lot of people. And then the second thing is we have 3,000 technical members -- some of them are corporate members and they may have a bunch of technical people. We probably have about 500 or 600 active technical members, and we have probably something on the order of 55 groups, half of them doing specs, half of them doing strategy, gap analysis and prioritization. When I started there were 15 groups.

Fairbairn: So how many people within the RISC-V Foundation? Is this the appropriate term to--

Himelstein: RISC-V International is the term now. They changed it. They moved to Switzerland because it was more acceptable to all parts of the world, so Switzerland has now become the Switzerland for technology as well.

Fairbairn: <laughs>

Himelstein: Right now we maybe have 12, 13 employees. Most of them are program managers. Right now there's two technical people. There's myself and I have somebody working on a technology called Sail out of Cambridge, which is a formal modeling language for ISAs, so not RTL, not Chisel. It's really specifically for specifying ISAs. But everybody else are really technical program managers or marketing program managers or marketing managers or marketing directors.

So when they were looking for somebody I think the things that resonated with them were what I did with Solaris, because I had 2500 people around Solaris. Only 500 reported to me and I had to make this thing work. And I was an early employee of MIPS, and I really understood a bunch of pieces of the ISA. They knew I was not a hardware guy. I was a software guy. They also knew that most of what they were doing was hardware and most of what they needed to do was software.

So, when I started, I'd say out of the 15 groups, 14 of them were hardware, 1 was software. Now, two-thirds of what we do really is around software, and the software could be on the software/hardware boundary -- things like IOMMU [Input-Output Memory Management Unit]. So how you do memory mapping for I/O or it could be on how you're going to deal with microarchitecture site channel attacks, or control flow integrity or things like that. So, I did the same thing here that I did at Solaris. I created these organizations that were responsible for major areas. So, there's one that's responsible for SOC, so that's the hardware-software boundary, quality of service, RAS, IOMMU, debug, trace, all that kind of stuff. One's on security and it's doing memory isolation, micro-architecture side-channel, cryptography acceleration, et cetera. Applications and tools. So runtime libraries, tool chains, and then privileged software. So operating systems and there's things like SPIs and UEFI and all that kind of stuff, and then there's two ISA committees, one for un-priv and one for priv, and those are the six major committees. And under each of them are a bunch of task groups and a bunch of special interest groups that <inaudible 01:09:50>.

Fairbairn: So how does this all work? What-- I mean who decides what the next group is? Who decides how it functions? Who decides what the membership is?

Himelstein: This--

Fairbairn: Who decides the final thing? What's the--

Himelstein: So--

Kapoor: Yeah, and could I add to the question does it-- is like competitive to Arm? I mean does Arm have a similar kind of an operation?

Himelstein: So no. Both Arm and x86 are proprietary commercial architectures. So if you want to modify x86 you can't. If you want to modify Arm, you can but you got to pay them a lot of money to do so. And they've changed their licensing over time. But they've been around since the '90s and they have a lot of legacy things and baggage. We stand on the shoulders of giants. We benefit both from x86, Arm, POWER, HPPA, SPARC, et cetera. We benefit from seeing all that they've done, and we also benefit from Linux and watching this community grow up. So Linux is open source. We're an open standard.

You got to think of us more like Bluetooth or wi-fi; however, we're more pervasive. You know, Bluetooth or wi-fi, you work on your protocol, and you make sure the apps that use those protocols work and you're done. With us, it's general-purpose computing. It's everything, right. It's basically the equivalent of a Turing machine, <laughs> or like this is it, right? And so, we aren't like them. When I came in also because I'd done governance before with Solaris, I came in and my first question was, "Okay. How do you guys know you're done with a spec?" and they go, "Uh.. I don't know." I go, "All right. You're going to have a way to do that. I'm not telling you what it is, but I'm telling you, you have to have one." And so now we have governance. We have a life cycle, and we have major milestones.

We also have a process around group creation. The board of directors … you get to be a member of RISC-V by membership dues, because that's what pays my salary and does our conferences and all that kind of stuff. The board of directors delegates all technical things to the technical steering committee. So if you have a board seat you also get a technical steering committee seat. Other people have technical steering committee seats. There is a fee you can pay to be on the technical steering committee, but also if you're a committee chair. So, if you're running security, you get a seat and a vote on that group as well. One vote, one company, so it's very egalitarian. So, if there's duplicate representation, only one gets a vote.

If you want to create a new spec, let's say, for example, you want to do matrix ops, you put a proposal together. It has to be sponsored by a committee. So it'd be sponsored in that case by the unprivileged instruction committee, and they would come to the technical steering committee and say, "We want to start this," and they have to have it be pretty small because in the past we did things that took forever. So vector took us six years. Bit manipulation shouldn't have. It took us four years, and we don't let that happen anymore. It's divide and conquer, smaller things, get them out. There are different types of matrices. You can do something that's tacked onto vector. You can do something that's a whole different instruction set and blah, blah, blah.

So, you'd go into the technical steering committee and say, "We want to create this," and at that point there would be a preliminary charter and there would be-- we had already done a call for candidates, and we will have picked a couple of candidates for chair and vice chair, and then the TSC would vote. They get to interview, they get to do whatever they want, and then they go ahead and start the group, and then the group has to go through the life cycle. There's a plan milestone where you say what your proof of concept's going to be, you say what your timeline's going to be, you say what your strategy's going to be, et cetera, and it goes on from there. Just sort of normal stuff, except the end product isn't an implementation. We don't do any implementations. It's only a specification. That's why we're a standard, open standard, as opposed to open source.

We have some sister organizations, CHIPS Alliance, OpenHW Group, lowRISC. They're all doing implementations that are open source that people can use. They have RTL, DV, blah, blah, blah, all that kind of stuff. So that's sort of the basics of how it works. We run it with a set of meetings, and I go to more meetings than anybody else. Part of my job is to make sure the left hand knows what the right hand's doing because we don't want duplication in various areas. We also want to make sure that we foster the software ecosystem, so we have great relationships with open-source projects like GCC, LLVM, Linux, et cetera, because that's the only way this works, right. If you have an ISA without software, <laughs> it's pretty useless. So there's been just a great amount of work in that arena.

I just went to Linux Plumbers and GCC Cauldron this past fall, and we did get a bit of an earful. We're in this, the supply chain issue, so there's a dearth of development boards out there but hopefully that will be freeing up and fixing this year. But we have those open lines of communications, and the greatest thing is I tell everybody we're a continuous improvement organization. If you don't like what you see, your members help us make it better. "Oh, you're doing too much." "Well, what don't you want to do?" <laughs> It's up to them. You want to do something? Well, are you willing to lead? Are you willing to be a chair or vice chair? Are you willing to be an author of a spec? Those things are really important. If you aren't willing to do those, and you just say you want it as a gap, it's not going to happen, right. So it's a very large community.

We've also created a thing called development partners which is unique because we didn't have enough software developers within the community and most of them were ISA developers and VLSI developers and stuff like that. So we go to folks like IIT-Madras and Chinese Academy of Sciences to take on whole pieces. There's a lab called RIOS lab which is a joint thing between Berkeley and Tsinghua University, and they've taken on all the formal model work and all the architecture tests for vector, and so we've done some innovative things in order to get this thing going. But the thing that's amazing is that we started with 15 groups, you know, two-and-three-quarter years ago and we're sitting at probably, again, 50 active groups at this point, and they're working in every corner that you could think about.

Kapoor: Yeah. So looking at the size of this organization, I mean, size of the community--

Fairbairn: Ecosystem.

Kapoor: --how do you keep conflicts from rising where the...?

Himelstein: So first of all, we don't allow encumbered information in any RISC-V venue. That's important. You know, we don't want to contaminate members. If somebody has something, they got to work with their council. If they want to have side conversation with other members they do that, so that's the first thing. We just don't allow that. There are other architectures where they've actually given open source to Linux and-- but the document explaining it is proprietary. We don't allow those things in. The other thing is we don't allow implementations in. So, if you take a look at something like bit manipulation there's a standard way to go ahead and do some stuff and it's encumbered. Well, we don't care about that. That's really up to the implementers to deal with; that's not up to us to deal with. We work with OIN, the Open Invention Network, and Unified Patents and they work to try to keep trolls at bay and other things and stuff like that.

So number one is around IP. We're really careful. Then the second thing is that we have a hierarchy and I try to build an organization that would scale, and I push power down but stuff has to roll up. It's trust and verify, right. So, people have to tell us what they're doing and there are some checks and balances in place. Every committee that exists has to sign off on every milestone for every spec, so if you're putting in vector and you don't talk to security and you're going to have a hole, it's going to be shut down at some point during the process. We also have a public review cycle, so 45-day or 30- to 45-day public review, so the world gets to comment at a certain point, and then we have to sort of resolve those comments. It could be like, "Hey, that doesn't make sense because of this and this," or they could say, "Oh, this needs clarification in the spec." We'll do that, so on and so forth. So, we manage the organization by distributed control and by as efficient communication as we possibly <laughs> can have. But, when you're dealing with 500 engineers, right, and 70 leaders, that sometimes is difficult. My boss has called me chief diplomatic officer and I hope I-- I aspire to live up to that name.

Fairbairn: Yeah, that's a...

Kapoor: <laughs>

Himelstein: But I'm kind of the ombudsman at times.

Kapoor: Yeah, this is amazing. Yeah.

Fairbairn: So there are companies that you have the spec; others have to actually implement this, so--

Himelstein: Yes.

Fairbairn: --there are companies, one I'm familiar with, Andes Technology.

Himelstein: Perfect example.

Fairbairn: So they take the spec and they say, "We're going to implement this portion of it." They may not implement everything; is that right? Is there minimum set? Is there-- how--

Himelstein: Yes. So that's a really good question, and it's a really good time to ask the question because just last Thursday the board of directors ratified the first profiles. You think of profiles as a generation of instructions that work together like RNB6, RNB7 or some Zeon family or whatever, so we have major profile groups which have a set of instructions with them and those are the things that we ask the distros and the toolchains to target, so they're not targeting--

Fairbairn: Distros are...?

Himelstein: Are Linux distributions.

Fairbairn: Right.

Himelstein: And the toolchains so that they're not chasing everything, and so you can say, "Hey--" so there's a-- we have technical names for them. Marketing's working on marketing names, so I just want to have that disclaimer. But so our first major release was RVA20, everything that was ratified in 2019 or earlier, and so the idea is that it's about portability. If you went ahead and compiled your application, you can run it on multiple implementations. That's what profiles are about. So, we do that, and we're working on platforms as well, because we need the same thing for operating systems so that an operating system can have one set of bits and be able to run it on multiple implementations with just some configuration kind of stuff.

Fairbairn: So getting back the question of Andes, for example, is there-- they have to implement a minimum set of things to be able to run the--

Himelstein: If they want a brand as RVA20 compatible, they have to do that. But Andes not only does that-- this one of the unique things around RISC-V-- we allow custom instructions. So Andes can have custom instructions that they know their customers need, and we're okay with it, right. It's just fine, because we think that things like that breed innovation, and a lot of things that may be custom today may be sedimented in the future and they may come to us and say, "Look. We really want this to be standard in the future," but Andes -- they provide IP. Their supply chain is quite long, so ASUS just announced a board based on Renesas. Renesas is based on Andes, and it's a board that's used for automotive. So somebody at some point's going to have some automobile company using that technology. In order to go ahead and do their next generation of whatever, you know, and a lot of times we see people, because— So look…the ramp for different products is different amounts of time, so IoT, embedded, all that kind of stuff, it's shorter. We have an earbud manufacturer in China called Bluetrum. They have, you know, since 2021, at least 600 million RISC-V cores per year for profit.

Fairbairn: That they developed themselves?

Himelstein: That they developed themselves. They don't care about us. Because they're embedded. They're running captive applications. It's not general-purpose computing. It's not multi-user. It's fine for them. They're happy. So for them it's a short period of time to go ahead and develop, but then you go up the chain and it takes longer. Disk drives are a year and a half. Automobiles are probably three years. Servers are probably five years. So it takes a different amount of time. So a lot of the stuff you're seeing that has come out in RISC-V has really been in the—embedded, in IoT space.

At our summit in December, Qualcomm announced quietly that they had already shipped 650 million RISC-V cores on their Snapdragons, and that they were going to add more cores, because what happens over time is you have it do security or some I/O piece and then you start expanding what you can do with it, and so you throw down more cores. EDA's experienced a renaissance in the last 15, 20 years, the modularity, things like chiplets, you know, make it just easier to plunk something like that down and go do something with it.

Fairbairn: So can somebody take, I mean, taking one of your examples like this earbud manufacturer, they could take any portion of yours and implement it. They don't have to be compatible with anybody, right, and they don't have to claim--

Himelstein: Right. They don't care.

Fairbairn: --that it's RISC-V or whatever.

Himelstein: They don't care. They don't brand for themselves, let along for RISC-V. <laughs>

Fairbairn: So they can take whatever piece works for them.

Himelstein: Yeah.

Fairbairn: And they have to implement certain amount of stuff so that the--

Himelstein: And no other architecture.

Fairbairn: --ecosystem works and...

Himelstein: No other architecture will let you do that for, you know, for free, right.

Kapoor: So do you care as an entity how many cores are developed versus, say, Arm cores?

Himelstein: Look. We care because we get asked by places like Semico, but we don't require people do that, right. I think what we care about more than anything is that we remain a community and the reason we're a community at all is because people get to share the effort. They get to share the effort to define the ISA and they get to share the effort, more importantly, in developing the ecosystem for the ISA. Even large companies, the Googles and Intels of the world, would not want to spend the money on implementing everything in the ecosystem needed for a new ISA. We share that burden, and that's actually-- that's the gravity. That's the thing that keeps us together as a community, because we share the work in GCC or LLVM or, you know, Chronos libraries -- the graphics libraries or whatever it is. We share that effort as a community. That's why profiles work. People want one target because they don't want things to be divergent to the point where they have to maintain things themselves. If somebody does something custom that's really good, but they've got to go ahead and maintain the tools themselves. We don't tell the toolchain or distros to go ahead and support those things. But sometimes things like time to market or differentiation are so important that people go ahead and do that, and we allow them to. We encourage them to. We think it's great. So, it's just an amazing ability and I think it gives people comfort. I mean, this again goes back to it's their house. They want to put some fancy, stained-glass window in the front, we're good with it.

Fairbairn: So people can do anything they want with it. If they want to label it as RISC-V whatever, they have to meet some benchmark standards, run certain tests or whatever.

Himelstein: Correct.

Fairbairn: But if they don't care about--

Himelstein: Branding.

Fairbairn: --branding they can do it if they damn well please.

Himelstein: Right. And even if they do care about branding, we do have sort of a grandfather brand which has the basic instructions, and they can brand under that. But they'll say, you know, RISC-V compatible for RVI20 and the-- again, that's a very nerdy name, right, so...

Fairbairn: And are you constantly-- what's the pressing need at this point? What's the area that is not being served or that there's most demand for or now you have 50 groups working, it's like <laughs> "We've got it covered," or what's the...?

Himelstein: No. No, there's-- I mean, look. We, you know, for better, for worse, we started from ground zero, right, in 2015-ish, right, and, you know, these other architectures have had 30, 50 years behind them. So there are things that people have become accustomed to on other architectures that they want to see something similar or better in RISC-V, and we're all member driven. Again, if a member doesn't want to lead it, it doesn't happen. So members come to us and say, "Hey, I'm doing this debugging thing and I really could use a buffer that has the control transfer history of, you know, for some amount of entries of all the branches and jumps that have been taken," and then you start having the discussions and you need it for different layers of protection because you might have different domains. You m-- yeah, whatever. Those are the discussions that happen in a group, but it's driven by need. It's not like we arbitrarily come in and say, "We have to have this." It's not Krste, who's the godfather of RISC-V. It's not him coming and saying, "No, we have to have this." It's members coming and saying, "We have to have this." You know--

Fairbairn: And then they've got to step up to do it.

Himelstein: They do, and look. It's hard because a lot of the folks who come from the microarchitecture VLSI piece of the world aren't the folks who grew up with Linux, right. By 2014, supposedly something like 80 percent of the Linux developers were paid by their companies to work full-time on Linux. We don't have that situation. Our contributor culture is still evolving, and so yeah, the companies have to say, "Yes.

This is important enough for me to do--" to spend time, you know, have you spend time doing that, because I'm paying your salary, right. But I think everybody who's part of the RISC-V economy is learning there's great opportunities for them to sell product and make their customers happy.

Kapoor: Are folks from Berkeley involved, Dave Patterson involved?

Himelstein: Oh, yeah. Dave's the Vice Chair, Krste's the Chair. There's some cadre of those folks who went to SiFive, and SiFive is a very big contributor to RISC-V. A lot of universities are involved all around the world. It's every continent you can think of, except for maybe Antarctica, we've got people involved, and it's really exciting. We also have a lot of work in trying to train the next generation of engineers, so we give out single-board computers. We have a bunch of classes. We have mentorships. Because we know it's not just what you do today but it's when somebody goes to work and somebody's asked, "Hey, what should we do?" if they have experience with RISC-V they'll go, "Hey, you should consider RISC-V," right, and we know that that's an important piece of the pie.

Kapoor: So must be very satisfying for you as a...?

Himelstein: It's great. In some ways it's one of the most challenging jobs I've had and in some ways it's the easiest job I've had. I don't hold secrets. I work for a nonprofit. I can't sign an NDA, right. This is really cool. I mean, you don't realize how cool that is until you're living it.

Fairbairn: <laughs>

Himelstein: I don't have to tell somebody some story about when I'm going to get product out, because my product is specs, and when you're in community, somebody goes on vacation, what are you going to do? They don't work for me. It's not like I'm going to say, "You need to work over the holidays," or something. I always felt like every time I thought I knew patience the universe taught me how much I didn't, right, and this is one of those cases where you go, "Okay. You just have to be patient. If it doesn't happen today, it's going to happen tomorrow." Those are my boss's words, and, doing that, but being encouraging and still trying to get people to sign up for things, that's an interesting challenge. Making sure that, again, the left hand knows what the right hand's doing when you've got 50 groups going with people from different companies and different agendas. That's a challenge.

Fairbairn: Did you model this after anything? Did you have to create this from whole cloth?

Himelstein: Well, the basic pieces of pushing power down, finding good leaders to lead the committees, because everything lives under the committees, that's from Sun. That's what I developed at Sun and did for Solaris, and successfully, and so I knew I wanted to head in that direction. But again, we're in a community. It's not like I can dictate that that's what's going to happen. I spent many, many hours with different senior architects explaining what I was trying to accomplish and finally, I mean, I wanted it to be regular. Guys who are engineers don't think about regularity in organizations. They just don't, right, and so I wanted it to be regular.

Again, I wanted to push power down. I tried to not have everything bottlenecked by some committee, but I also tried to make sure that there was some process in place, right, and that's-- again, that's the synthesis where I go. Okay. I understand what they're trying to do technically. Now let's figure out how to make them successful. Let's make sure we do things that pass sort of the basic test of reasonableness," and so, yeah, and I wrote a book on "100 Questions to Ask Your Software Organization." Some of that stuff's in here. I wrote it years and years ago, and mostly informed by the experience at Sun. Just things like getting meeting times. When you got folks in India and Europe and L.A. and Boston and oh, my God, right. 7:00 A.M. is the most amazing time. That's what everybody wants. We tried to make things more friendly for China. They said, "Stop doing that. We work our day jobs during the day. We want to do our meetings at night for you guys."

Fairbairn: <laughs>

Himelstein: It's like, "Oh, okay." So you learn things like that, and again, we're continuous improvement. So, we take input from everybody and try to incorporate it so that we're doing a better job. I don't think there is a project that's this big that grew this quickly ever in the history of open standards. I don't think there is one that exists. Now, there are some that are this big but they grew over time, and this is just a stunning piece of work by the teams. These are--

Fairbairn: Is there any similarity between your organization and how things get done in Wikipedia?

Himelstein: I don't think so. I think Wikipedia is way different. I knew one of the principals over there and what they described to me isn't what we do. We're doing technical work, right, defining these things, and that's really not what they're doing.

Kapoor: You said in India it's IIT-Madras? What is that?

Himelstein: That's one of the groups. They also have the Shakti project, which is a government really, but the government of India's actually on our board. They're a board member. You know, we have just great participants from India. I mean, energetic, knowledgeable. They're part of-- other groups are par-- development partners, but they also are just doing an incredible amount of work and we're very grateful. Two things I heard was somebody's working on a-- in India they're working on a hearing aid with RISC-V in it, and somebody's working on a cane with RISC-V in it. It's like, "Okay, why do you want a cane?" So I guess you'd count the steps or something. I don't know what it is. So a lot of innovation, and that's one of the things that RISC-V does. It gives you the ability to do interesting new things. I heard that there was a 12-year-old or 13-year-old in India who actually created a RISC-V chip on his own, and that's the other thing. We have so much stuff that's out there. There are proprietary implementations, commercial implementations like Andes, and blah, blah, blah. But there's also so many designs out there for RISC-V that anybody can pick up and just go do and then you can throw it on to an FPGA and do something, right, you know, it's pretty exciting.

Kapoor: Thank you. Anything else?

Fairbairn: No, that, I think, covers it. Anything we've missed--

Kapoor: Yeah. <laughs>

Fairbairn: --that you'd like to say about your career or your observations of things going on these days or...?

Himelstein: Well, I mean, I think the biggest thing is gratitude. I am just so grateful. I often didn't chase the money jobs. I chased the jobs that I thought would make me happy, and this job makes me happy. I don't know if it always will, but I'm very grateful for every day I wake up and have a chance to do this because this is changing the world. This is not some small thing. In 2021 over 10 billion RISC-V cores were sold for profit.

Fairbairn: Was going to ask you, do you have-- what's your marketing number, whatever, of what's--

Himelstein: Well, I mean, it's anecdotal because only when people tell us. All of a sudden you hear from the Qualcomms, the NVIDIAs, the Bluetrums, et cetera, Alibaba. I mean, you just hear how much is being shipped and all you can do is kind of add it together and go from there. Sometimes I wish we did require people report how many cores so we knew, but, pardon me, I guarantee you that it's like all over and, you know--

Fairbairn: It's billions.

Himelstein: Yeah, it's billions. If in 2021, it was 10 billion, you can imagine what it is now. I think, you know, the other thing is our buzz word, our key, there's two of them. One is RISC-V everywhere and the other is RISC-V is inevitable. So we get shocked every day when I see some new product come out. It's exciting and I'm grateful to be part of it. I'm grateful that I'm working with such brilliant people, and I get a chance to help make them successful. How many times in your life do you get a job like this? Not many.

Fairbairn: That's great.

Kapoor: Yeah. I think it shows, <laughs> the joy shows.

Himelstein: Yeah. I definitely have joy. There's no question about it, and I'm grateful for you both asking me to talk, so...

Fairbairn: All right.

Kapoor: Thank you.

Fairbairn: Thanks very much, Mark.

Kapoor: Thank you.

Himelstein: Thank you.

END OF THE INTERVIEW