



Oral History of Keith Diefendorff

Interviewed by:
Doug Fairbairn

Recorded August 24, 2022

CHM Reference number: 2022.0125

© 2022 Computer History Museum

Fairbairn: Okay. It's August 24th, 2022. I'm Doug Fairbairn and I'm here to interview Keith Diefendorff, who has kindly agreed to join us and tell us about his background and history in computer and microprocessor architecture and related fields. So, Keith, we're delighted to have you, and thank you very much for coming.

Diefendorff: Well, thank you for having me. I'm delighted to be here. I support the Computer History Museum and I like what you do here, so I'm glad to be part of it.

Fairbairn: Great. So, we're very appreciative. Thank you. So, in our oral histories we always like to start at the very beginning and talk about where you were born, where you grew up, a little bit about your family history, what your father and mother were doing at the time and how that might have impacted your future growth and career choices. So let's do that. Let's go back to the beginning. Where were you born, when, and tell me a little bit about your early family life.

Diefendorff: Well, I had an uneventful childhood, I guess. I was born in small town in Ohio, outside of Akron called Barberton, a small industrial community. Akron at that time was sort of the tire, rubber capital of the world and a lot of rubber industry, so a lot of industry in general in support of that, and--

Fairbairn: What year were you born?

Diefendorff: I was born in 1949, so <laughs> longer ago than I...

Fairbairn: <laughs>

Diefendorff: Care to remember, but so yeah. I was born there. I grew up there. Went to all levels of high school there and my father was in industry at Babcock & Wilcox company. My mother was pretty much a stay-at-home mom.

Fairbairn: Was your father an engineer or did he have a--

Diefendorff: He was not an engineer. Neither of them really went to college, which was good for me in a way because they never allowed me to even think about not going to college. <laughs> That was simply, "Okay," that was--

Fairbairn: Expectations.

Diefendorff: Expectation was, "You're going to college, and we don't want to talk about it anymore. That's not really an option," and so that's pretty much it. I didn't really have any particularly inspiring events or anything. It was just a normal childhood. I spent--

Fairbairn: What about siblings?

Diefendorff: I have one sibling, a sister, and I spent more time, you know, sports and that kind of thing than I really did academics in high school. I wasn't--

Fairbairn: Do you have any hobbies?

Diefendorff: Mostly sports kind of stuff. I mean, back then. I have lots of hobbies now <laughs> sort of, things I do for hobbies that are-- can turn out to be more than that, but I didn't have many hobbies other than sports, no.

Fairbairn: So electronics had not yet appeared on your personal horizon?

Diefendorff: No. Electronics had not. You know, I had some passing interests. My dad did a lot of projects at home, construction, you know, and I helped him there, so I was sort of a hands-on kid, but electronics really in-- throughout high school really, electronics wasn't any kind of a passion or a major focus or anything like that. It was more being outdoors in sports and that kind of thing for me as a child.

Fairbairn: So which sports did you pursue?

Diefendorff: Well, I really wasn't that great at any of them. <laughs> But I did, I played most of them. Football, track, gymnastics, golf, you name it.

Fairbairn: Just did it.

Diefendorff: That's where I just spent all my time, doing that from place to place.

Fairbairn: And was there any teacher in high school that steered you in one direction? I mean, were you-- find yourself good at physics and math or any of those things or was it just making your way through?

Diefendorff: You know, there were a couple of teachers that inspired me. I had a geometry teacher I recall. I worked harder for her; I don't know why. She just inspired me the way she taught and so it was-- it happened to be geometry and so I was pretty good at that back then, but it still was, I mean, secondary <laughs> on my mind, so I had a few high school teachers that I appreciated. Mostly in math and physics.

Fairbairn: So then your senior year, you're going off to college. What was your goal? Did you just pick a place and go or what did you--

Diefendorff: Yeah. You know, in high school you take these. They-- well, back then. I don't know if they do today, but they used to give you a test to figure out what you wanted to do. I mean, <laughs> and they sort of try to ask you a bunch of unrelated questions to infer what you really would like to do. And of course they told me I wanted to be a forest ranger and I have no idea where that came from because that's not really what I wanted to do.

So, I guess when I first went to college I sort of knew that somehow math and physics were, you know, that's the sector that I gravitated to more than others, and so I knew it was going to be something there. And so I went to Kent State in physics. I was-- said, "Okay. I'll be a physicist." <laughs> You know, not out of particular passion or anything but sort of the path of least resistance for me, I think, yeah.

Fairbairn: And how did you choose Kent State? Was it just the local college or what's--

Diefendorff: Well, actually, first I had chosen Ohio University and I went down there and, you know, frankly I just-- I just didn't like the, I don't know, atmosphere. Something about the university and I and/or maybe that it was the late '60s. It was just a culture and an environment that I wasn't too fond of, so it changed to Kent State, just nice, nice campus. It was far away enough from home that I lived on campus, at the school. I always thought that was important, to be sort of away from home to do college. So it was far enough away from where I lived that it-- that I actually lived there, but it wasn't so far from home that I wasn't-- it wasn't a long trip. And my high school girlfriend, now wife ever since, also went there and so that probably had some attraction.

Fairbairn: Great story.

<laughter>

Fairbairn: High school sweethearts, huh?

Diefendorff: Yeah, yeah, high school sweethearts.

Fairbairn: What a great story.

Diefendorff: Yeah.

Fairbairn: Well, speaking of the late '60s and Kent State, Kent State was not a calm place for long time.

Diefendorff: No. Well, it was when I went there, or, you know, when I started there. But it-- you know, it's kind of funny because in reality other than the one famous event it really, I don't think, was any more radical than other universities.

Fairbairn: Was hard to escape it anywhere at that time.

Diefendorff: Yeah. It was kind of hard to escape it anywhere, and I don't think Kent was particularly worse. Maybe, fact, almost the opposite, I would think. But there, you know, there was that one event that has just-- it made it famous and put it on the map and nobody will ever forget it, that just happened to occur at Kent. I'm not exactly sure what--

Fairbairn: And that was during the time when you were there?

Diefendorff: It was during the time I was there, yes, uh-huh. So, yeah, we had to finish that semester <laughs> remotely and everything and they shut it down and it was not a good time particularly. But just statistically if I'd have chosen another university, I don't know if there's any-- would any-- been less or more probability that that kind of thing would've happened there. I don't know.

Fairbairn: No, it's just that Kent State was such an iconic, you know, was that event and--

Diefendorff: Yes, it's that-- it's--

Fairbairn: --you'd think it-- no. You didn't hear about Kent State before I'm sure, but that was a--

Diefendorff: No. Right. Right.

Fairbairn: That was the thing that made the--

Diefendorff: Right. And so, at Kent I did physics and math most of my time. As time went on there-- as a matter fact, pretty late I guess in my degree there, I got invited to Rutgers to go into astronomy, as postgraduate kind of thing. But I just, I couldn't-- physics was somehow not quite right for me because I wanted a little more hands on than physics was. Physics was a little too-- I don't know.

Fairbairn: Too theoretical, too--

Diefendorff: Too theoretical--

Diefendorff: Yeah, exactly, so said, "Well, you know, that's not quite-- maybe engineering is what I really want to do," and Kent didn't have a EE program, so I pretty much tried to steer my degree towards the more hands-on stuff and I-- math. I took just about all of the math courses that Kent had, and I figured, Okay. I'll just finish this degree. Then I'll go get a master's degree in engineering at another school and so--

Fairbairn: So then you went on to University of Akron?

Diefendorff: That's right. Well, there was-- the army intervened there. Of course, this was--

Fairbairn: Oh, okay.

Diefendorff: --the time when, you know, my-- <laughs> my draft number was double digit. It was very low draft number and so I just went and joined as soon as I graduated from Kent. Spent some times there. For some reason the National Guard is the one they sent me to, but strangely enough with given the Kent <laughs> experience, but the Army National Guard is where they sent me, and so I spent some time doing that and then I went to University of Akron.

Fairbairn: So how long were you actually in active service there, <inaudible 00:12:40>?

Diefendorff: I don't recall, but active duty was 8 or 10 months-ish kind of thing.

Fairbairn: Oh, very-- you weren't shipped off to Vietnam or anything like that?

Diefendorff: No, and I don't understand that either, because it just worked out. I don't know. They didn't. They didn't send me there.

Fairbairn: Did you-- were you assigned to any-- I mean, here you were, you had graduated magna cum laude, right, from Kent State, and you're clearly a smart guy and did they try to make use of that in any way or did they just--

Diefendorff: No, they did not.

Fairbairn: Just put you... <laughs>

Diefendorff: No. Nope. I think at that time it was bodies.

Fairbairn: You were a warm body.

Diefendorff: Yeah, I was a warm body and they needed numbers and-- I don't know. I have no idea how that all worked out. It just sort of did, but the good news is that it was-- after my active duty it was mostly reserve duty so I could go to college at the same time, and so I went--

Diefendorff: Yeah. So that actually worked out pretty well.

Fairbairn: So you only had like a year break between--

Diefendorff: Yeah, about a year or something like that, right. Exactly.

Fairbairn: --bachelor's and then you-- and how did you choose University of Akron? Was that-- was there a particular thing that brought you there?

Diefendorff: You know, I went to several universities to check them out and I liked a couple of the professors that I had met with and talked to at University of Akron. Computers at that time weren't big things but they were coming things, and a couple of professors there seemed to know that and understand that and were motivated in that direction and encouraged me to come and work on computers there at--

Fairbairn: So did you actually get to work on microprocessors or were you using computers or--

Diefendorff: Well, it wasn't-- you know, microprocessors <laughs> at that time were, well, I-- there were a few little 4004. I mean, not much in the line of microprocessors. I worked on minicomputers mostly. Of course, we had mainframes for teaching.

Fairbairn: Yeah. Programming.

Diefendorff: Programming languages and that kind of stuff, but the department, EE department had minicomputers. Texas Instruments minicomputers.

Fairbairn: Oh, so they actually had TI minicomputers there, huh?

Diefendorff: They actually had TI minicomputers. Well, they were starting to get them when I started to go there, and so I actually being there sort of helped them pick equipment and get equipment in. I was a graduate assistant and helped out, did a lot of computer work back when punch tapes and punch cards and mag tapes, the big mag tapes, were the modes of input and teletypes and, ah, it was primitive, <laughs> to say the least.

Fairbairn: But did that light a spark? You said-- is it, you know, is computers something you wanted to do or was that just--

Diefendorff: Yeah, sure. Yeah. So I think with the professors' encouragement and that kind of thing it was pretty clear that of-- my degree was just electrical engineering in general but it was pretty clear that, to me, after a little bit of work there, that computers were the way to go. It was really much more interesting than motors and power transmission and all of that other kind of stuff, so I didn't have anything to-- I stayed away from that and focused on computers and logic design and programming and that's...

Fairbairn: So you actually got some experience on TI minicomputers. Did that somehow lead you to--

Diefendorff: I did. Well, I guess what it-- yeah. I got to experience there and my thesis was in multiprocessor communication -- kind of between multiple minicomputers trying to--

Fairbairn: You're already on a leading edge, huh? <laughs>

Diefendorff: Ah. Yeah, well, <laughs> as it turns out. Of course, I didn't know that then. It was just something that I, you know, it would just seem to me, "Well, you know, if these two computers could work together on this problem then..." kind of thing, so that's sort of the niche that I went into but it was all done on TI960 and minicomputers, and so that certainly at graduation made me send a resume to TI as one of the companies that I was looking to go to and so--

Fairbairn: So who else did you apply to or what was your...?

Diefendorff: Oh, heavens. I don't even recall but would've been IBM and--

Fairbairn: Digital Equipment?

Diefendorff: Digital Equipment. Well, Data General and Digital Equipment back then, so I sent resumes to several places and went and interviewed several places but I really liked TI. I don't know. Could've

been my experience with their minicomputers. It could've just been that they were located in Texas, <laughs> not Ohio, so yeah. Wasn't particularly-- my wife, neither one of us, didn't really need to stay in Ohio. Moving away was--

Fairbairn: Happy to go somewhere else, huh?

Diefendorff: Well, I don't know. Not happy. I mean, you-- because you have friends and family and that's hard to leave and so there's always that, but at least in northern industrial Ohio, you know, wasn't hard to--

Fairbairn: So I have to ask. When did you finally get married? Was this after Kent State or after the army or before?

Diefendorff: Yeah. So it was right before the army, so after Kent State, was right in that time frame, and...

Fairbairn: You both graduated together and...?

Diefendorff: Yeah. We both graduated from Kent and moved to Austin, Texas, for-- with TI. Yeah. That was a big move. I think my wife cried the whole way but... But we're glad we did. It was a--

Fairbairn: <laughs> Adapted, huh?

Diefendorff: It was-- ended up being a good move for both of us.

Fairbairn: So you arrived at TI. What did they-- what kind of program did they put you in?

Diefendorff: <laughs> You know, back at those days and I just, I can't believe it. I walked in and they said, "Okay. You're designing this piece of equipment." Happened to be a CRU expander or what was a-- it was the I/O for the 990 minicomputer line, was the CRU, and they needed an expansion board and they said, "Design this."

Fairbairn: "Go do it."

Diefendorff: And I said-- <laughs> I'm out of college, you know. I have no idea how to actually design a project. I mean, I certainly know logic design and computer programming, but actually-- and it wasn't anybody-- was just me. They just-- Kurt Boenke came into my office and said, <laughs> "Here's your job," and I said, "Well, but--" and he said, "Don't worry. You'll figure it out." <laughs>

Fairbairn: And you were using TI's TTL MSI to--

Diefendorff: Yeah, TTL-- yeah, yep, exactly right. That's all we used for-- in those first days is TTL, so I figured it out. I mean, yes, there were sort of mentors. There were guys that I made friends with that

certainly advised me and helped me out. Daren Appelt and a few of the TI's architects there that helped me out, so certainly didn't figure it all out on my own but they just threw me into the thing. They didn't-- no training, no anything. I mean, it was-- <laughs> first day, "Here's your schedule."

<laughter>

Fairbairn: "Want this done in this time," huh?

Diefendorff: "We want this done in this time. Figure it out." So it was good in a way because they sort of just tossed you into the wolves and you had to fight your way through it, but... So that was a learning experience, I would say, and--

Fairbairn: Were there people that you met in those very early days that turned out to be sort of long-term mentors or friends or...?

Diefendorff: Yeah. Well, TI was-- of the companies that I've-- well, I don't know. I want to say that TI was a great culture and I almost want to say it was, of the companies I've worked for, one of the greatest, but that's hard to say because cultures are so different between-- they're just-- it's just amazing to me how cultures are different from one company to another. So, TI was different in a way that I really liked. I did have mentors that would take pity on me or feel <laughs> sorry for me or something as a newbie in this, but Daren Appelt was a big one. Joe Watson was a mentor. Several people like that that I looked up to. There are people that I knew from the early days of TI that I sort of had met and that would later in TI become real mentors to me. People like Harvey Cragon, the architect of the ASC. These guys, I just looked up to them. They were just heroes sort of, and lot of them are gone today, or have gone on to other things and I've lost contact with them but stayed in contact with them for most of my career. Really super people at TI. I don't have anything I could even come up with bad to say about TI or the people there.

Fairbairn: Oh, that's great.

Diefendorff: it was an awesome environment and I got along very well there.

Fairbairn: So you subsequently did a wide variety of projects, from color display controller to graphics display to--

Diefendorff: Oh, yeah.

Fairbairn: --Ethernet controller to bit-slice processor and so forth. Without going through each one of those, was there one or two particular ones that really opened your eyes, set you in a new direction or were critical factors in terms of influencing your future direction?

Diefendorff: Sure. After the first project that they put me in, they-- I did the TILINE coupler, which was sort of like the UNIBUS coupler. It was 990s-- the TILINE was TI's equivalent to that, and so I did the

multiprocessor interface between minicomputers for that, and so I got a lot of experience there. I mean, practical, real-life experience with multiprocessors and that kind of stuff. The logic for the 990, which was mostly done by Daren, was asynchronous logic, which blew my mind. I never was as good at it as Daren or would ever claim to be. It was just a strange kind of thing. Was all asynchronous.

Then I went on to microcode in the next generation of the 990, the 990/12. I did-- a big portion of the microcode -- all of the self-test for the computer. Not many computers had much self-test in them, so that was sort of a challenge. I had to do a lot of design in the CPU and in the microcode to make the machine testable, which was a big thing. Then I did quite a bit of the instruction microcode, but it was a full-blown CISC machine, so it <laughs> kind of one of my first experiences with that, and so I did those kinds of things.

And then we built a version of the 990 that was based on the TMS 99000 microprocessor kind of thing, but the 99000 microprocessor was done more for simpler computers. It didn't have-- well, or they couldn't afford the transistor budget or something for memory mapping, which we had to have for the minicomputers, of course. So I went on from there to do a map chip that allowed us to take the 99000 and put a mapping facility on the outside of it. So the first chip that I did was this map chip, and that was an interesting project because back at that time, it really wasn't clear what the long-term semiconductor process technology was. People debated, "Well, you know, NMOS, but maybe not."

Fairbairn: CMOS?

Diefendorff: "CMOS, ah, that's really-- that's too hard," or something. I don't-- I don't even remember all the arguments. But TI was, of course, big bipolar at that time and they had come up with this new logic called STL, Schottky transistor logic. So they said, "Oh, you're going to use STL for this map chip," and it had never been used for any chip. I mean, this was the first chip, and the map chip was a pretty sizable logic design kind of thing. "You're going to do this." Oh, by the way, we have absolutely no tools.

Fairbairn: I was about to say, what tool--

Diefendorff: <laughs>

Fairbairn: Was this by hand? What was the--

Diefendorff: No tool. No tools. No, I mean, nothing. The--

Fairbairn: What year was this?

Diefendorff: Let's see. What year would this be? This would've been in the early '80s. Yeah, early-- I don't recall, '80ish, I--

Fairbairn: So this is when the Daisy, Mentor, Valid were just beginning to appear on the scene.

Diefendorff: Oh, yeah, there was no-- there was no--

Fairbairn: You didn't have anything...

Diefendorff: There were no...

Fairbairn: So was all hand-drawn logic.

Diefendorff: Well, except for I said, I can't do a chip of this complexity and rubylith-- <laughs> It's just not going to work out, and so I just said, "Okay. Before I do this, I'm going to go--" So I designed a high-level language and a circuit analysis software package, so that I could design something in this STL logic, because it was kind of weird logic. Open collector logic, and it was odd. I mean, fan-out rules and everything were rules. So, I didn't have any sort of automated synthesis or placement, nothing fancy like we have today, of course, but at least something that let me simulate the chip and have good confidence that it was, in fact, going to work. And in fact, it did work on first silicon, and so the tools worked out and so I <laughs> at least got some experience with design tools before design tools were--

Fairbairn: So did that attract attention within TI? Did other people adopt that methodology or just--

Diefendorff: You know, I don't know whether that-- hard to say what effect this effort had on that versus that it was about the time that other things were starting to get talked about in industry, so I think there was a lot of inputs. I think one of the problems here was this was brand-new technology and back then the tools were tied to the technology. So, for example, TI in Houston had some tools for doing the 9900, 99000 microprocessors, but they weren't applicable...they were unusable for what we had to do with the map chip.

But it was about that time when tools started becoming a thing, maybe coincidentally. Maybe that's how I got the idea to-- that I had to write some or something. I don't know. Just the time that it happened. So then I went on to... of course, TI needed a next generation minicomputer architecture and we got-- because of a lot of the O/S software problems we got-- some of the guys got enamored with capability based architectures, and we got drift of things like the 432 at Intel and, you know, they had given a couple of papers. So we started a project to do a capability-based machine. I worked on that project for a while. But ultimately-- I also got very enamored at that time of personal computing as opposed to minicomputers with a time-sliced...

Fairbairn: Timesharing.

Diefendorff: Yeah, timesharing. I just-- I don't know. I just had a bad feeling about it. There were just so many problems and issues and performance limitations. So I got really enamored with personal computing, and started working on that, and we ultimately ended up going into LISP machines, so we acquired the rights of the-- from LMI [Lisp Machines, Inc.] for the CADR, and--

Fairbairn: So that's a pretty radical move from--

Diefendorff: Yeah, it certainly was and so--

Fairbairn: There's software and other things that go with that. How did...?

Diefendorff: Yeah, so we got all-- I mean, because we decided that our first effort, the Explorer, was going to pretty much be sort of an industrial strength rather than a university strength implementation of the CADR. So that it was software compatible... so that we got all of the LISP machine software, windowing system, O/S, garbage collection. All of that stuff we got from MIT. So I spent a lot of time going back and forth to MIT working with the lab there and with LMI, and we built the TI Explorer based on bit-sliced implementation of--

Fairbairn: And that was targeted towards the general industrial market or what was that...?

Diefendorff: Yeah. Well, so--

Fairbairn: I didn't even realize TI had done a LISP machine. <laughs>

Diefendorff: Oh, yeah. Oh, yeah, yeah, we did it for several years. It was targeted for general purpose but the big thing at that time, it was-- was AI, <laughs> strangely enough. This was sort of AI before AI was cool.

Fairbairn: Yeah, this is the other AI.

Diefendorff: Yeah, this is the other AI. Right. Exactly.

Fairbairn: Symbolic processing.

Diefendorff: Yeah. Symbolic processing, and so that kind of processing we saw a lot of application for. Expert systems. A lot of companies...you know, the airline companies. A lot of these companies saw a lot of value in symbolic-- this kind of thing for their applications and so we built the machine targeting mostly the research organizations in a lot of the big companies that were looking into that sort of computing, which was quite a--

Fairbairn: This is the early '80s still, or mid-'80s?

Diefendorff: Yes, this was early '80s still. I don't-- you know, the years exact escape me, but early '80s, yes, kind of thing.

Fairbairn: So were you involved in the decision-making in terms of what things to target or what to build? Were you--

Diefendorff: Only what to build. I was the system architect of Explorer and I did-- I was the system architect but you know how these projects go. You get your fingers in everything. So, I did microcoding

for it. It was an interesting machine in that, well, it was like the ultimate CISC machine. They don't get any CISC-ier than LISP machines. And it was a writable control store, so most, or at least the critical portions of the operating system, were all in microcode kind of thing. So I did a lot of the microcoding for that and then I did the color graphics display system for it. I did a fiberoptic link for the-- so you end up getting into other thing--

Fairbairn: So all those things were part of the LISP machine business.

Diefendorff: All of those kinds of things. We built the machine from scratch. Industrial design. Everything. So I had my fingers in all of that kind of thing, It was a really great project because there were some really great people on it. Pat Bosshart in Dallas did the LISP chip. We put the whole CADR on a chip. There were lots of great engineers, and the engineers working on that project, probably the most of any project I've been on, loved it. I mean, the software was fantastic. It was just unbelievable compared to anything that any of us had worked on in minicomputers or anything like that. It was just so easy to do very complicated things in comparison to FORTRAN. it was just fabulous, and so the people loved this project. I mean, the--

Fairbairn: The designers. Yeah. Yeah.

Diefendorff: --architects and the designers in this project just loved it, and you could just-- just for the longest time and through multiple designs ... you know, there was Explorer, Explorer II, MicroExplorer-- through all of those projects, you could come in at any time of the night and half the team would be there. I mean, all night long. People didn't sleep. They didn't eat. They worked on this project. It was, in terms of, I don't know, fun, it was the ultimate project in that.

It was, in the end, too expensive. At the end of that project, I had become very down on the performance that we could get out of these complex machines. They were so complex-- the instruction set, the writable control store. They were so complex that you didn't have any complexity left to put into performance and that kind of stuff. And I think that was ultimately the downfall of LISP machines is the complexity was super high at the time, and C running on regular Sun workstations or whatever they were, just could run circles around it whenever you really wanted to make something that performed right-- the RISC workstations.

So I became convinced that CISC was not the way to build CPUs, and that RISC probably was the future, and probably not for some of the reasons that a lot of other people favored RISC. It's just that I wanted to spend the complexity on things other than just getting it to work, just getting the instructions to execute. I wanted to spend it in deep pipelines and issuing multiple instructions and caches and those kinds of things, and there was just no budget.

Fairbairn: So one of the things that led to the RISC thing was the issue of debugging and being able to fully verify these complex machines. Is that a factor?

Diefendorff: Yeah. I mean, I'm not denying that that was a factor. It wasn't a driving factor for me. For me, the driving factor was performance, performance and then performance. Because it seemed like, to the extent I was able to provide more performance for somebody, the software people would figure out how to do the clever thing. And so to me, the key thing was performance. And as a matter of fact, at the end of the Lisp-- when I had finished the MicroExplorer, which was an interesting project -- We put the entire Lisp machine, which was a box this big and \$50,000 workstation, we put it on a board in a Macintosh computer. And so I did the multiprocessor interface software, between the Macintosh, between the 68000 and the Lisp machine.

After that, we started looking at other architectures that we could get more performance from. And we actually met with Sun several times. We were thinking about adopting SPARC and possibly enhancing the architecture with tags, which at that time we thought were essential to some of the key good things about Lisp machines, like garbage collection, safety-- pointer safety. Some of those kinds of things were very important, and tags seemed like the way to do it. And so we were thinking about adopting SPARC. But the main thing was I got convinced through that effort and performance analysis and comparison and modeling tag architectures on SPARC and that kind of thing, that RISC was, in fact, the right way to go in terms of building these machines for the long-term, just because it put the complexity more in the right places. That's kind of the main thing.

Fairbairn: So TI headed off this whole 9900, 99000, 99X family. Then it got into Lisp. Obviously never became a major factor in the minicomputer or whatever business.

Diefendorff: No.

Fairbairn: So when did you see that? When did that happen? How did that happen? Were you still there as they wound that down? How did that happen? Tell me about sort of the higher-level.

Diefendorff: Well, so TI continued in the minicomputer business. There were still people working on the minicomputer. So, they continued that business. We did the Lisp machines. I don't recall when TI actually finally got out of the Lisp machine business. A lot of things came out of the Lisp machine that carried on from there. We learned a lot in doing that in terms of cooling big processors, in terms of display systems, in terms of windowing systems, especially -- MIT and Xerox PARC, all of this Windows system, personal computer kind of thing.

So all of those concepts sort of evolved from different places all at that time. It came from there. And so the big Lisp machine that we built was just for research, so it was a small market. So eventually TI didn't stay in that business. But I think a lot of the things from it survived. And as a matter of fact, that's how I went on to my next endeavor. My manager, Chuck Corley, at TI during the Lisp machine development, moved to Motorola. And it wasn't long before I got a call and said..

Fairbairn: Come join me.

Diefendorff: ... "Hey, Keith, how would you like to come and be the chief architect of our next-generation 88000, which we were going to call the 88110. We don't know what it is yet, but it's next-generation. What do you think?" And so that was hard to turn down, because I would really work on a mainstream RISC microprocessor. At that time, we knew MIPS and other companies. So that '88-ish sort of timeframe and the other people were starting to develop RISC microprocessors and I said, "Okay, that.."

Fairbairn: So when you went into it, was it already decided that this was going to be a RISC machine? Was that sort of a starting point?

Diefendorff: The 88110 that they hired me to be chief architect of was the second generation of Motorola's 88000. And that was a RISC machine architecture. And so this was going to be the next generation, which at the time they hired me, they didn't know what that meant.

Fairbairn: So you were already on that path. Yeah.

Diefendorff: It was just the next generation. I mean, that's what you do in a Moore's Law world is use more transistors, put more-- I mean, it was a two-chip RISC. It had the CPU and the memory management on separate chips. The 88000, the original was, and this was going to be all one-chip. They knew that. So I went over and started to work on that.

Fairbairn: Now, you had spent 13 years or whatever at TI.

Diefendorff: At TI.

Fairbairn: Was this a big-- was this a hard change? Was this..

Diefendorff: Oh god. I almost don't like to think about it. It was embarrassing. I just felt so bad. Well you know, I still get choked up.

Fairbairn: Walk in and tell your boss that you were leaving?

Diefendorff: Yeah. I mean, and some of these guys that I just thought the world of. I mean, people like Harvey Cragon. At the time I worked with them, they were older gentlemen. And you'd think, "Ah, this old guy. What's he know about computers and microprocessors and that kind of stuff?" Yet, you'd sit there and talk with him, and he would just come up with these jewels of wisdom. And I would say, "Where did that come from?" I mean, he doesn't know anything about this. Yet he did know something about that. I don't know whether he hid it or he didn't know it or it was just in wisdom, but some of these people, Willis Adcock and-- you'd just follow them off a cliff, because they knew. I mean, they were so charismatic or wise in ways that I didn't relate to at that time. And so, I knew I wouldn't have them anymore. So, this was a big thing for me, and emotionally it was really terrible. I just don't even like to think about it.

Fairbairn: It still hurts.

Diefendorff: It still hurts. It still does.

Fairbairn: What did your wife think? Do what you want?

Diefendorff: Yeah. She's not a techie, so she didn't understand RISC or the next-generation CPU or designing microprocessors or-- she didn't have a lot of advice. I did put a lot of stock in my boss at TI, who went to Motorola. He was a person that I thought the world of, and knew that if he was doing this, he needed me to do it. And so that was ultimately the reason that I went. And it was a new RISC microprocessor.

And so I went and did that, and when I got there, it turns out they really didn't know what it was. My job was to figure it out. What I knew was that for Motorola, for many years, Apple had been the biggest customer for microprocessor-related items. They didn't want to lose Apple. They, at that time, I found out after going, were in sort of the mode of-- well, the 68040 was late. They were having trouble with it, they were having performance issues with it. They wanted more compute power, more floating-point performance, especially.

Fairbairn: They, Apple?

Diefendorff: They, Apple. And they were looking for a RISC chip. Their architect at the time was one of the founders of Ridge, John Sell, and they, Apple, I think through his influence-- much of his influence-- were looking into RISC. So they were evaluating the RISC chips of the time, MIPS, AMD, 29000, the Intel 960. They were evaluating all of this stuff. And of course, Motorola did not want to lose them as a long-term customer. And Motorola sort of knew what they were-- that they were doing this snooping around in the next-generation CPU technology. So, it became my job to avoid Apple going to a different company. And so it turns out-- I went to Motorola to work, but didn't really work for Motorola as much as I worked for Apple. So, for Apple and, it turns out, NeXT. At the same time, Steve [Jobs] was off doing NeXT. And NeXT was using 68000s and they were in the same ballpark.

Fairbairn: Yeah. Had the same issues.

Diefendorff: Same issues. Craig Hansen was, of course, the NeXT architect at that time, their chief architect, and he was one of the founders of MIPS. And so they were going in that direction. And so it became-- and I don't know if Motorola knew or what, or thought maybe that Steve would eventually go back to Apple and it would all be one big, happy family again or anything like that. But they also didn't want to lose NeXT. Probably because of Steve. Steve was the most amazing driving force on the planet. It was just phenomenal. And so my job was to come up with a spec and a design and a micro architecture and the process technology and the whole ball of wax that would convince Apple and NeXT that the 88000 was the way to go...well, the next generation.

And so I did that, or worked on that. That was my job, and I wrote all the high-level architecture and microarchitecture specs for the 88110, and we started building it. And lo and behold, NeXT and Apple both-- they wanted to stay with Motorola. They had a long experience with Motorola, but they were

satisfied that our chip would be competitive. Maybe even more than competitive because we...I mean, I say "I" because it was me mostly on an architecture basis working with them. But then I would have to convince Motorola executives that we needed this transistor budget or that we could make these modifications to the architecture and still stay software compliant.

Fairbairn: Compatible and all that. Right.

Diefendorff: All that kind of stuff. So I had to figure out ways to do all that. So it was sort of this balancing act. But the one thing that I was willing to do, and was able to convince Motorola of, is that we would make the high-level changes to floating-point architecture, to floating-point performance, to graphics requirements, these kinds of things that NeXT and Apple wanted. It was much harder for them to convince Intel or MIPS or other people of that. And so we sort of had..

Fairbairn: You were more accommodating.

Diefendorff: ... A strong. Yes. And I certainly knew the importance of having a huge customer and hoped Apple would be a huge customer again. They weren't doing so well there for a while, but we always hoped that they would be, and that this might even do that. So did that design, the 88110. It did a lot of firsts on a single-chip computer. I mean, if you list the things that it did, there were no other single-chip computers that did it all like this. And it was a push inside of Motorola, because it required die sizes and process technology that were leading-edge to get it done. But we did. We made it.

It did work, but just about the time it got into some systems-- the 88110 came out and got debugged and went into systems. This is really a hard one to figure out exactly what the week-by-week chronology was here. A lot of it happened behind my-- but Apple came to the conclusion that just they and just Motorola and just whatever few customers were available for the 88000, like Data General kind of stuff, was not enough. It wasn't critical mass to compete with the resources that the x86 was able to generate. As great as the chip was, as well as it worked, the 88000, it didn't have anything to do with that. It was..

Fairbairn: It wasn't that chip. It was the future..

Diefendorff: It was. It was the, "Oh jeesh. How do we get critical mass?" Because it's not just the processor architecture or the processor microarchitecture or anything like that. It's process technology. Well, how do you fund next-generation process technology? Well, it means Motorola has to have a lot of customers. It's got to amortize the cost of that process development. And if other people are running circles around you from process technology development because they can afford to do it, then I don't care what your architecture is. It could be anything, and it doesn't matter. One generation in process technology, especially at that time-- almost even now today, but more so at that time, one generation of process technology was enough to make up for any architectural mistake or any architectural benefit.

Fairbairn: Cover up all the other sins?

Diefendorff: It didn't matter. If you could be a process technology generation ahead, that was the winner. And I think that's how the x86 won long term is the software compatibility and the large number of customers for a fab. Because the cost of a wafer is directly related and exponentially related to the volume through the fab. The more volume, the lower the cost. And that's that. And so, Apple, through that kind of high-level analysis, I think just concluded that they weren't going to be enough, it didn't look like at that time. Today maybe it's a different story, but at that time it didn't look like they themselves were enough critical mass to do that.

And so, no matter how much we tried, how great our chips were, it wasn't going to cut it. So somehow-- and there are lots of competing stories about exactly how this happened. And whoever you ask this is going to give you a different story, because they saw it from a different perspective. But somehow IBM got in the picture. And they, of course were computers. And they had, although not the highest-yield fab, they had the world's greatest process technology at that time. They had copper interconnects before anybody had copper interconnects. They had chemical mechanical polishing before-- I mean, they were far out. It was leading-edge process technology. Yield suffered a little bit because of it, but they just kept getting better at it. I mean, because a lot of these things, like CMP, in the long run once you learn to do it and get the cleanliness and the fabs and that kind of stuff that you need to pull it off, it increases yield. And so that kind of thing made IBM a very attractive partner in all this. And so regardless of who you give credit or blame for it, in mid-'91-ish, sort of. Maybe early '91-ish, Apple, Motorola, and IBM met and decided that maybe we'll look into all doing something together here. So that was the start of the PowerPC.

Fairbairn: So what happened to the 88110? Did that..

Diefendorff: Well, okay, so 110 is still going on. It's still a chip, it's still in manufacturing. DG was using it, Apple was using it for a lot of software development in their labs on prototype boards and stuff. And meantime, me and the architects at Apple and the architects at IBM and the executives at IBM and Apple and Motorola are meeting in supersecret. It was hilarious at some points. I mean, IBM would call supersecret meetings in the basement of buildings that have been abandoned, so no one would know. It was hilarious. But a huge team of people, anyway. It must've been 50 people, half of them lawyers, meeting to see if there was some way we could come to agreement on a common microprocessor architecture to go forward with.

And of course, IBM is the gorilla in the room. And they pretty much, in essence-- I don't know to what extent they came directly out and forced it, but in essence they said, "You know, if it's not based on one of our architectures, we're not interested." And so we said, "Yeah, but the RISC architecture that you have.." which was the Power Architecture at that time, "... It's got some problems for use at Apple," and that kind of thing. And so we convinced them that as it sits, the Power Architecture wasn't the best or ideal. If it's going to be a long-term thing, we want it to be a long-term architecture. And so they said, "Well, okay, I think we get that." And so we finally sort of agreed on that philosophy.

They signed a bunch of contracts, and they created this what's called the AIM alliance, Apple, IBM and Motorola alliance. And we all agreed that, okay, it was going to be a new architecture roughly based on

Power. And the thing that IBM was willing to bring to this game is-- another convincing factor was that they had a power chip called the 601 that was in development and at the sort of stage where we could still make the changes necessary to fix it for what Apple and Motorola needed. And so, they had a development underway, fully staffed, in progress that they were willing to sort of flip the switch on and say, "Okay, change this." And the other thing that they brought to the process is that they were willing to share with Motorola their process technology. So CMP, copper interconnects, those kinds of things. They were willing to teach Motorola how to do it. And Motorola would argue, but I would say that in fairness Motorola at that time was somewhat behind. Certainly behind Intel, and arguably behind some other companies in the state of the art.

And to be fair, Motorola had focused more on high yields for embedded controllers for high-volume memory chips and those kinds of things. And it sort of didn't put enough effort into the high-performance features like copper interconnects kind of thing. So to say they were behind is a little bit unfair, but they were certainly behind in the performance that you needed for high-end microprocessors. If you wanted to be a state-of-the-art microprocessor, IBM clearly-- and they were willing to bring that process technology to the table in the AIM alliance. And so that sort of convinced us that, okay, their architecture-- even though we all thought that the 88-- I think a lot of the IBM architects would have argued that the 88000 architecture was cleaner and better and more suited for Apple. Legally and contractually IBM wanted the PowerPC or the Power trademark. And that was their thing and they weren't going to give that up no matter what. So we went that way.

So, I had to go back. For example, at Motorola after the 110 was finished and working, I went off at Motorola and Ford was looking for a new microprocessor. Ford had made the decision in the past to-- they were sort of a central computer architecture in their cars. And they had chosen many, many years ago-- ten, twenty years ago-- or I don't know how long ago, but when they started. But they had chosen a chip and a chip architecture that was more than they needed at the time. And that served them very well, because it lasted a long time. It lasted them for ten years, that kind of thing.

And so, they knew that it was time to go to a new architecture for that. They wanted the same sort of system architecture, and that RISC was probably the way to go. And so, they came to Motorola and Motorola supplied a lot of the embedded processor chips to Ford. And so they came to us, and so I got the job of architecting an embedded control RISC chip for Ford. And they loved it. They wanted, strangely enough, super high floating-point performance. You'd think, "For automotive? I mean, where cost is everything?" And their argument was, "Look, we spend--" they had an exact number. I don't recall what it was, but it was 80 to 90 percent of our bugs are from scaling integers of all the-- it's a nightmare, because we do a lot of signal processing, a lot of analysis like that, and it's just too painful.

And so they wanted floating-point, and they wanted floating-point adds to run at the same speed as integer adds run at. So unlike most embedded-- I mean, I don't think Motorola had ever run into an embedded customer that cared about high-performance floating point, but Ford did. So anyway, I designed a version of the 88000 architecture for Ford, and that's the way they were going. They had adopted it. And so now, I'm off working with IBM and Motorola on PowerPC. So I had to go back and

convince Ford that we could switch that architecture and they wouldn't know it. It's a different instruction set, but..

Fairbairn: Oh, so it actually switch to PowerPC architecture?

Diefendorff: They switched to the PowerPC architecture, and that was some tough convincing, but they said, "Okay, we'll do it." Or, "We can live with it." And so at that point in time, Motorola had lost the two big customers, Apple and Ford, for the 88000. And so they let the 88000-- it wasn't that they let it die, they put all the resources on PowerPC processors, the development. Because that was the future. That's what we had agreed on and that's what we did. So as part of that-- the big negotiations went on, and as part of that what we all agreed to was an architecture committee that was one chief architect from Motorola, one chief architect from Apple and one chief architect for IBM. And so I was to be the chief architect from Motorola. John Sell was the chief architect from Apple and Rich Oehler was the chief architect from IBM. And of course, we all had other people that would assist, because it's a bigger job than a single person. But if the three of us agreed on a change..

Fairbairn: That was it.

Diefendorff: That was it. We were going to make the change. And we did. We meaning sort of Apple and Motorola fighting from one side of it didn't get everything we wanted in the PowerPC. Some things survived that we think shouldn't have, and we didn't get some floating-point changes that we think we should have and all that kind of stuff. But it turned out not terrible. Pretty good. And the advantages of the 601 being halfway done and IBM's process technology and all that sort of made up for it. So we kind of didn't skip a beat, really. The companies just switched to PowerPC.

Fairbairn: So how-- from a process technology point of view, outside of the details of how it was done, how would you characterize the difference in the performance and density improvements that you got for the first generation PowerPC versus the 88110 was that a..

Diefendorff: Okay. So the answer depends-- I think what you want is sort of a net answer to that are, but..

Fairbairn: Were you working in a, "Wow, we got a different world. We get twice as many transistors versus what we were aiming for."

Diefendorff: Yes, yes. It was earth shattering. It was monumental differences. And I don't think that Motorola fully appreciated even, at the time, how big the differences were. But it was bump technology rather than wire bond. We didn't have-- we just used wires-- wire bond technology. The copper interconnects and umpteen layers of metal and all of that, Motorola didn't have any of that. And it was just fabulous. I mean, SRAM cell size was a tiny fraction of what Motorola could build, a high-performance cache SRAM cell was just mind blowing. Now, like I say, IBM developed most of their high-end process technology—you've got to always remember that IBM, in its heart of hearts, is a server company. They're not a personal computer company. All that stuff in Florida, nope. IBM, in its heart of hearts, is a server

company. And they developed all of that kind of stuff just for high end. And so the yield, okay, so what if we get crappy yields? Run a few more wafers. It is only 100,000 mainframes. It's not a big deal. Which, of course, was a problem. So we did have yield issues going forward, and yield issues always-- not always, they often come back to the issue of, well, you can't quite get the frequency you want because something doesn't work. Something like that. And it was a struggle to get the IBM process technology.

The three main things were the packaging bumps, CMP, copper interconnects. To get that stuff into Motorola fabs, which were not designed for that, which did not have the equipment for that, for those kinds of things, took time. It took a lot of time. It took more time than we thought, but it, in the end, was certainly worth it in terms of die size and frequency and everything else you could think of. So we could build-- I don't think I ever did a real analysis of how much more dense it was. Because the things like cache cells were a much bigger gain than things like high-VT or low-VT transistor. So I don't know what the net was, because there were never really any place where we could really do an apples-to-apples, because it depended on how much cache, how much logic, how much <overlapping conversation>.

Fairbairn: But it was a big difference.

Diefendorff: It was a big difference.

Fairbairn: And in a sense, Apple got what it was looking for. That is, it got a much more advanced process technology, and more advanced processor.

Diefendorff: It most certainly did. It got that. And so there were a lot of things we didn't get in the PowerPC architecture on the first round of chips. So the other thing we did as a part of this is we-- I say we as a collective IBM, Motorola, because-- it was strange, because I had a badge that got me into any. It was like, "Okay, am I an employee of IBM? Am I an employee Motorola or what?" But we had this design center called Somerset, which was in Austin, and it was staffed by people from all three companies. Architects, logic designers, chip designers from all three companies. So that's where we did all the chip design. And Rich and I and John worked mostly on the high-level architecture. But of course, you only work on high-level architecture by working on the lower level architecture. Because we always, "Well, can you build this?" Can we design an ALU that can do this in a reasonable die footprint? And so we spent all our time trying to figure that stuff out.

The one thing that we didn't get in the PowerPC initially, which I was a proponent of but was impossible to get into the 601, which was the first thing, because it was just too big of an additional-- stuff is the SIMD instruction, the vector instruction set. We had some vector instructions in the 88110, some SIMD instructions in the 88110 that I designed. And it worked pretty well. And during that work and a lot of the-- so when I did that, the issue was well, how much does this die area accelerate something? And so you have to go off and you just have to code all these DSP algorithms, everything, and figure out what exactly here is worth the bang for the buck. That's what the job of an architect is is ultimately figure that out. In so doing that at Motorola for the 88110, I became convinced that fixed vector instruction set extensions were every bit as valuable as frequency or any of the other-- scalar instruction issue or pipeline length or any of

those other things you can think of. It was at least that big of a thing, if you could really take advantage of it. If you could vectorize the code that you are after, it was a huge win.

And so, I wanted that in PowerPC. We couldn't do with the first round because-- we didn't even contemplate putting it into the architecture because we knew the 601 couldn't do it. And the critical thing was getting the 601 to market. That was what we had to do. And so, we couldn't get it in the first round. So that was a shortcoming, I think, of the first PowerPC architecture. Ultimately, we did, but that's sort of a story down the road. But for then, we didn't. But still, a couple of the chips turned out very well out of Somerset. A couple of them didn't turn out so well. I don't know why. Well, maybe I sort of do, but I sort of wouldn't say.

Fairbairn: So how long did you stick with the PowerPC program with this consortium?

Diefendorff: Yeah, quite a while. I was at Motorola a total of seven years, I think, something like that. About the first two, I spent on the 88000, two and a half, and the rest of it was on PowerPC. And so I was PowerPC chief architect at Motorola for a long time.

Fairbairn: So you went through several-- two or three generations of that?

Diefendorff: Two. It took us-- like I say, it took us longer than we had hoped to get the generations of PowerPC beyond the 601 out because of difficulties in Motorola bringing up the process technology, because of other things. In some cases, I think we bit off a little bit more than we could chew in microarchitecture in some of the chips. In some cases, it was because the three companies didn't always see eye to eye, didn't always agree on exactly what it was. And like I say, IBM-- as much as they gave lip service to, "Oh, yeah, we want to support Apple and do workstations and minicomputers." No. They wanted to make sure that the chips that came out of Somerset were usable in AS/400s and in mainframes and in scientific workstations and those kinds of things. So we did have some agreements on-- not at the architecture level so much as at the microarchitecture levels and those kinds of things. So I stayed with Motorola for a long time. And then after-- pretty much I thought that the architecture of the PowerPC, as much as we were going to get from IBM for me there was kind of done. I mean, it wasn't much going on. And so I left there, convinced by one of the Apple architects who had moved to a company named NexGen.

Fairbairn: I know about NexGen.

Diefendorff: Yeah, you know about NexGen. And of course my best friend in life, Nick Tredennick-- who you know passed recently. Very sad-- was one of the founders of NexGen. For multiple reasons, anyway, they talked me into moving from Austin to California to join NexGen and help them with this x86 stuff. Which of course, for me, it was, "Oh, man. I don't know about that." But on the other hand, by that time it was becoming clear that volume and fabs and process were key, and x86 was going to be a winner no matter what. No matter how bad you thought their architecture was, it was going to win. So maybe I ought to think about reluctantly getting on the other side of it. And it's a weird thing, because Nick was not a RISC fan, but anyway, they talk to me into coming out to NexGen.

Fairbairn: Okay, so you've decided to make the move to California and to NexGen. Was Atiq Raza there?

Diefendorff: He was. Yeah, Atiq. Atiq and Vinod Dham and Dave Epstein and Mike Yamamura. And Mike was an architect for Apple during the PowerPC days. And so he was one of the reasons I sort of went to NexGen. And I got there, and so their chip that they were working on wasn't done, but the architecture was done. They were just trying to get it to work, sort of, and get it out the door kind of thing. And so, I didn't have much contribution to what became the K6, I guess. Except for the fact that when I got there and started looking at it and I started doing some performance analysis and looking at what they had kind of thing to figure it out. And I said, "You know, I don't know. If you are going to go into the business of competing with Intel in the x86 business, you can't afford to do the infrastructure and everything else that you're going to need to support this chip." So I told them, I said, "Why don't you put the Intel front-side bus on this chip? Because otherwise it's a nonstarter. It's hopeless."

Fairbairn: You mean to make it plug compatible?

Diefendorff: To make it plug compatible with Intel processors. Socket 7, I think it was. I can't remember. But at least electrically compatible, if not pin compatible. And so I spent most of my first six months there trying to convince Atiq and Vin and Dave and Dick Stiles that that was the right way to go. And they eventually did, and it's a good thing they did, because it wasn't long after that that AMD acquired them. And I don't know what it was for me, but I have never really been a fan of AMD. I don't know why.

Fairbairn: Yeah, certainly speaking of culture, it has its own culture, I suspect.

Diefendorff: Yeah. It does have its own culture, which I had been exposed to a couple of times. I liked some of the people there. Even Jerry Sanders, because we met several times in negotiating then.

Fairbairn: He was still there at the time. I had forgotten what the sequence was.

Diefendorff: He was still there. Yeah, and he was still there when AMD bought NexGen. So I had interactions with him. I didn't mind him much, but I don't know. I don't know if I just thought at that time that it was suicidal to compete with Intel on x86's, which begs the question why did I go to NexGen? But I don't know why. For some reason, I didn't want to work at AMD. I just didn't want to work at AMD. So just coincidentally, Apple had called me up. I hadn't been in California for-- I hadn't been there a year yet. I mean, it was less than that. Apple called me up and said, "Look, we got to have a PowerPC architect back. We can't get IBM and Motorola to do what we want without somebody that is going to push them." And so, I agreed to do that and I went over to Apple. And Apple made that transition pretty easy. They were willing to accommodate me.

Fairbairn: So did they send you back to Texas?

Diefendorff: No. Nope. They said, "Here's your office in Cupertino." And so all I had to do is go a different direction on 101. And so I went to Apple. And so I did two things there. I was there for, I forget, two or

three years at Apple. But I wanted to get done two things for sure. I sort of had a list when I went in. One is I wanted to get a SIMD instruction set into PowerPC. So that was..

Fairbairn: You knew everything going on. You knew all the people, you knew everything.

Diefendorff: Yeah, that's right. I mean, I knew everybody. So it was kind of easy to do from that perspective. I wasn't new to the game. Everybody knew who I was.

Fairbairn: You knew the history.

Diefendorff: I knew the history, I knew everything. So I went down that path. I called a big meeting at Apple of all the graphics architects and all the floating-point architects or software people. Mostly that's what they were. They didn't have much hardware in these at the low level. And I went through, "Okay, here's everything everybody else is doing, has done. MMX and Intel's working on SSE and you've got these instruction sets, these from the 88k. We have these from here. This is what everybody's done. And I think Intel was only rumored to be working on the MMX update at the time I started this. And I said, "Okay." And I had several proposals. I was like, "Okay, would this kind of an instruction set? This kind of vector." Vector instructions for the RISC machines are great, because it's not like a CISC instruction. It's not like a vector machine instruction. It's one instruction that has two or three operand and a result operand and that's it. And so it fits perfectly into a RISC architecture and microarchitecture kind of environment. So I was wanting to stick with that sort of a notion, because we were running out of frequency at that time. we weren't getting, in the process nodes, the frequency bumps that we were getting.

Fairbairn: Is that because you were running out of power?

Diefendorff: Mostly because we were running out of power. And they hadn't yet figured out the leakage problem. There were several contributing factors here that, to some extent, the architectures-- the microarchitectures had just-- given the complexity budget we had were just not capable of much higher frequency. So anyway, I was wanting to push people toward some kind of a parallel sort of solution-- vector, data parallelism solution was one avenue we hadn't gone down. And so, I showed them all these other things and said, "Okay, what of this could be useful? I mean, what do you see here in this universe of SIMD and vector stuff that could be useful." And they all went off and came back and said, "None of it."

They said, "You know, we just-- give us higher frequency" kind of thing. And I said, "Well, is there anything you can imagine that would be? Any variation on these themes that would get to some kind of a data parallel solution. And so they thought about that and they said, "Yeah, one thing." And it was floating-point vector parallelism, which we hadn't had before. Most of the MMX, they were mostly all integer stuff. They said, "That's where we need the parallelism. That's where it is. That's what we do." And so I said, "Gulp. How am I going to convince IBM and Motorola that they can afford the die area for big, parallel--" and they didn't want slow floating point. They wanted floating point that was as fast as everything else-- that we could do that. And there were other issues with that. When you end up looking at a lot of the algorithms for a lot of this stuff, one of the problems with the SIMD, the lane parallel or data

parallel approach is that you just end up with so many cases that the data doesn't quite align with the ALUs. And so you end up wasting a lot of things that you could vectorize, you can't vectorize just because of these details.

Fairbairn: So what were the applications that they came back that drove wanting to do this floating-point? Was it for graphics?

Diefendorff: Graphics.

Fairbairn: Graphics operation.

Diefendorff: Well, more than that. It was-- I would call it media in general, because it's all mixed up. A lot of the algorithms are the same. It's just for video rather than audio. So it was for digital. Because Apple, which I have agreed with forever, they had tried many times in the past to do little accelerators. Okay, we got a DSP over here on this board that we're going to use for audio. We have this over here. GPUs weren't a big thing back then. They were starting to be, but they had all these different things that they did. And they had tried these little accelerator things, things that set outside of the CPU and did a specific job. And while they worked, they were incredibly difficult to use software-wise. How do you use them? And you get this DSP from this vendor and this-- and they're not the same. And so we can't write software that optimizes for this one, because they may go out of-- it was like-- it wasn't in the architecture. And so compilers-- and so people couldn't depend on it. And so they were sort of in line with the idea of, "Okay, let's make it so the CPU can do these things in a power-efficient sort of manner." And so they were-- their response was not strictly graphics. It was, "Okay, well, we can use that in this and this and this and this," but almost all related to media processing of some kind.

Fairbairn: And that was important to Apple.

Diefendorff: FFTs and all the standard algorithms. And so, another architect, Ron Hochsprung, and I at Apple at that time set off to design an instruction set around what we had heard from the Apple software people. And we were able to get the PowerPC architecture committee back together and explain that here's what we wanted, here's about what we want. Here's the kind of thing we want. We want it as a first-class piece of the architecture. We want it to be in every chip that gets built. There either can't be a chip that doesn't have this feature, because if there is, then our software guys won't use it.

Fairbairn: People won't use it. Yeah.

Diefendorff: And so we had some of these goals, but to solve the lots of floating-point multiply adders, and multiply add was a big thing. Not just multiply and not just add kind of thing, because of the precision error accumulation properties of it. And the other problem of solving the lane permutation issue. Those were two issues we had, both of which were going to cost a lot of silicon. And we as architects knew that, but we went to the Somerset and said, "Okay, here's what we want." And surprisingly, the micro-architects there didn't..

Fairbairn: Didn't throw you out, huh?

Diefendorff: They didn't-- yeah. They said, "<makes mumbling noise> Okay." So it was a lot of work to convince them that the die area just-- or I mean that the performance increases would justify the die area and all of that kind of stuff. But long story short, we worked through all that. We got it adopted. It officially was called AltiVec. It was called VMX when we started. It was called-- stood for Vector something. Vector Machine Instructions or something. But it was later named AltiVec and we got all that stuff, the floating point multiplier. And we got this instruction which turns out to have been, I don't know, one of the best things-- one of the best and single instructions I'd ever come up with. It was a permute instruction that allows you to put any data lane on any other data lane in one instruction. And that really made it fly. So it worked out very well, and I think is the reason that PowerPC was able to stay competitive with the Intel chips for several years after that, after we got that into all the chips.

Fairbairn: Did your one-year-plus diversion to the world of x86 give you any different or unique perspective that you brought back to the PowerPC world? Or it was like a lost year.

Diefendorff: Let me think about that. I don't want to say no, but I can't think of anything.

Fairbairn: Subtle.

Diefendorff: Yeah. You know, I don't think so. I mean, it was good from the point of view that I met some smart people. I got more of an appreciation from where they were coming from. I met a lot of the sort of CISC proponents in the other company. Well, the architect community is not a big one. I knew most of these people to start with, but I had interactions with a lot more people than I otherwise would have, I think. And so that was good. But the other thing that I wanted to do in my two years, or the time that I was at Apple-- I didn't know I was only going to be there for two or three years. But the other thing I wanted to do was, after the data parallelism thing, I wanted to attack the thread level parallelism kind of thing, because I was on the parallelism kick because I just saw frequency slowing down. In the long term, frequency slowing down. And the gains, that kind of thing.

So the other thing that I was doing, all the Apple designers that were at Somerset reported to me when I was at Apple. And so what I wanted to do is, at a high level, design a chip that had multiple-- was CMP...was Chip Multiprocessor kind of thing. And so I came up with something that I thought that they could implement at Somerset, which was composed of four small PowerPC cores and one big PowerPC core, and an on-chip DRAM controller and on-chip L2 caches, and the kind of things-- It's not unlike what you see today. But it would have been one of the first, had they done that. And so, I pushed pretty hard at Somerset to get a project like that going with this footprint kind of thing.

And that one, no one at the executive level at that time related to multiprocessor. I don't know, it was just too strange kind of-- and by the way, there is a point to that. And the point is-- and one we were aware of when working on this project, that the software is not written for multiple cores. It's written for a single core. And so, you can put all the processor cores you want in there, and it ain't going to do anything. But at that time, the Apple OS was pretty advanced, and they had a lot of internal thread scheduling structure

in the OS that in my dealings with the OS guys thought that they could utilize and build on to make some of this software we're working on. So, in all that big project going on, I tried to get that going. I could never sell it as, I don't know if it was because everybody knew it was too far ahead of its time, or too many mindsets of, "Oh, just make the processor go faster." Wider superscalar..

Fairbairn: Been doing that for a long time, let's keep doing it.

Diefendorff: Yeah. Just keep doing it. We know how to do that. Oh, this multiprocessor, how do you do this and DRAM interface and mumble, mumble. So I didn't get that done. I mean, they didn't do it. They went on and did the G5 PowerPC processor, which was a big monster power eater. It was high-performance. It had AltiVec and all that kind of stuff, but it was very power-- I mean, it was hard to use for Apple in low-power laptops and that kind of stuff. Impossible to use. And so Apple did get the SIMD stuff in. They got it in all their machines. It was very successful, and I think the primary reason, because Apple's kind of claim to fame in a lot of their stuff was graphics windowing systems, media, that kind of stuff. And so the PowerPCs with the AltiVec were much faster than the Intel machines on those things. On the regular things, they were not so fast. Intel was running circles around them on single-threaded, nonmedia kind of stuff. On net, it looked like they were competitive.

Fairbairn: Yeah. The ones they were really selling to were the ones they could <overlapping conversation>.

Diefendorff: Yeah, exactly. So it worked for a while. But it didn't work forever, and they knew it wouldn't work forever, because Intel was bound to do the same thing, and they have. And after having done that, they were going ahead in process technology kind of thing. So it was very frustrating in time that I couldn't get the chip multiprocessor done. I tried. And at about that time, Apple was really, really going downhill. There were rumors of bankruptcy the next day. It was not pretty kind of stuff, and Steve..

Fairbairn: What was it? This was '96 was it?

Diefendorff: This would have been '98-ish.

Fairbairn: '98-ish, yeah.

Diefendorff: '98-ish. So I forget exactly what Steve's status was, but he wasn't back as full CEO, and I just-- I don't know. It just didn't seem to me like there was a future. And Michael Slater calls me up and says, "You want to write for Microprocessor Report?" He said, "I'll hire you as editor-in-chief." I said, "I'm an engineer. What do I know about writing?"

Fairbairn: Why do I want to write? Yeah.

Diefendorff: Kind of stuff, but I had done a few presentations at Microprocessor Forum and those kinds of things, and for some reason he thought he could teach me or train me to do this. And since I was an architect, I was at least technically competent in all of the subjects his newsletter was about. And he

thought I could do that. So I reluctantly went off and did that. That was a tough job. Oh god. I mean, I spent a lot of hours-- late hours and all-nighters as an architect on many projects in my life, but this was a new level of difficulty. I had-- as editor-in-chief, the whole staff of writers reported to me. I had to oversee all of them and the articles they wrote, and check them. And I had to write most of the pages of Microprocessor Report. And the good news is I met a lot of people doing that, because the companies all liked Microprocessor Report back in those days because they trusted them to do what they said they were going to do in keeping things confidential when they were supposed to be and not. And so they would disclose a lot of stuff to us that were not disclosed publicly to anybody else. And so, I learned a lot from-- talk about jobs where you learn a lot about what you're going to do next. Well, I learned a lot from that. I mean, I interviewed and talked to and got technical dumps from all of the key people in the industry. So it was a good job from that point of view, but it was super hard work. I mean, it was just super hard work. Because I'm not a writer. Michael and Linley and those kind of guys, they write effortlessly. I mean, they'd write an article like they're falling off a log. It doesn't-- I don't know.

Fairbairn: It just flows from their fingertips.

Diefendorff: It just flows from their fingertips. It's magic. But that was never me. I mean, for me, it was tough. And maybe it was because I was more technical than I was an editor. And for me, all of the details in getting all of the detail into the article was-- I put a lot of weight on that. And that's sort of the hard thing to do. So it was a hard job, and I-- so I did that. And then I tried-- did several other..

Fairbairn: Yeah, so how did that lead-- you went to ARC Cores and then to MIPS.

Diefendorff: Yeah, Jim Turley, who reported to me at MDR, went off to this company called ARC Cores. And I didn't know anything about ARC Cores other than they kind of were ARM wannabes. Unfortunately, they had very few people in the company that knew anything about computers. And they had this idea that-- so anyway, Jim talked me into coming over. They had this idea that they wanted to build a next-generation microprocessor, and so they hired me as VP of research to figure that out. And so, I hired a small team, but it turns out they didn't even know what that meant. They didn't know anything. I mean, they just were clueless of how much that cost, what kind of people you had to hire, what kind of time it took. They were..

Fairbairn: Now they were sold to Synopsys. Was that sometime later?

Diefendorff: Yeah, that was after I was-- yeah. And along the way, John Stockton promoted me to CTO and he became CEO. And so they wanted me to stay and he was trying to save the business and that kind of stuff. But I said-- well, it's just clear that what I do is design performance microprocessors. It's not something they were going to be able to do as a company. It just wasn't in them.

Fairbairn: And was it to build standalone chips? Or they were going to do IP, or was that..

Diefendorff: No, they were going to do IP. But it doesn't matter. I mean, if you're going to do a synthesis of all synthesizable, high-end, configurable sort of thing, it's a different level of stuff than they were into, you know they just weren't into that, and they didn't get it.

Fairbairn: Then you moved on to MIPS.

Diefendorff: Yeah, I moved on to MIPS because the person I'd worked with back at Motorola, Kevin Meyer, was at MIPS as VP. So he called me up and I said, well, MIPS is a more credible company, I have a better chance of working on microprocessors at MIPS than I do that, and so I went mostly because of Kevin's influence and he wanted me to do a specific project when I got there. And that project was a big project. He wanted me, so at the time Microsoft was looking for their next generation Xbox CPU and he wanted me to win that contract. Well, he didn't want me to win it, he wanted me to contribute to the architecture of a chip that would ultimately win that stuff.

So, of course, I did, I met with Microsoft many times, I met a lot of really smart guys there, and so did what I had done at Apple only a little bit different scale is try to figure out what the customer, what it would take in a microprocessor to win the customer. So I spent, I don't know what kind of timeline, I guess I spent a year working on something for them and we were finally ready to go propose it to them and we did. They liked the chip but there were two problems that I want to blame somebody, but I don't want to blame somebody, because everybody works under constraints and so I can't blame people that are constrained by things above them, and you know they do the best they can, and anyway there were two issues. The first was that MIPS management was never convinced that they could actually deliver that chip. It was sort of bigger than they thought that they could swallow.

Fairbairn: From a design and complexity point of view or from a manufacturing sort of deal?

Diefendorff: Yes, both, but the manufacturing was sort of the be--

Fairbairn: That was off because it was IP, right?

Diefendorff: That's right, exactly. So this one may probably not have ended up IP, it might have ended up specific for them and in a hard chip, and MIPS did hard chips so they knew how to do that. But they just didn't have the teams and the depth of talent necessary to pull off that chip, and I don't think any less of a chip would have had a chance at Microsoft because their performance demands were high.

Fairbairn: So this incorporates some of your vector instructions?

Diefendorff: It would definitely do that, which I think is one of the reasons Kevin brought me there.

Fairbairn: Right.

Diefendorff: So that was sort of the first problem. And I think that Microsoft by looking into the eyes of MIPS execs could see they didn't believe that they could-- You know they wanted to believe, but they

didn't believe, and so I think that sort of-- The other issue is that the competition was PowerPC with AltiVec.

Fairbairn: You knew what you were competing against, right?

Diefendorff: Yeah, so I knew what I was competing against, but and the architect of the chip that ultimately won that, it was PowerPC that won that, with Jim Kahle one of the architects at Somerset, and I think the sort of thing that put a lid on it was IBM could build the chips, they had the process technology, they knew how to do that. It was captive, I mean, IBM could promise anything, and they could deliver anything compared to MIPS. MIPS couldn't say, oh, yeah, we have this fab process and it's going to be--

Fairbairn: These are volumes that IBM would not normally be shipping, right?

Diefendorff: Yeah, it's true, but, I mean, IBM over those years had actually come a long way in terms of bringing the yield of their processes up. So it is stuff that they could, and I don't know, you know who knows what they ended up promising MIPS. Hey could have said, okay, we'll build you a fab, or, I mean, promising Microsoft, you know for that kind of a win they would have built a fab, and IBM could do it, but MIPS couldn't do that, and so I think for those two reasons we didn't win that contract, and so my job at MIPS was pretty much done, and MIPS really struggled after that. I mean, they went near, well, they eventually sold out after, a couple of years after I left and so they didn't really know what, they didn't really have a niche. They'd lost their, they had one of the biggest wins they'd ever had was, I think was Nintendo where they had a chip from MIPS and it was such a big winner for them. But without anything like that--

Fairbairn: Sony also?

Diefendorff: Yeah, Sony did the Cell architecture, right, which is basically the AltiVec instruction set. Yeah, because IBM designed the Cell architecture for Sony kind of thing, and they just took the vector instruction set, that core vector execution unit, from the PowerPC AltiVec units and built the Cell processor based on that. It was all the same kind of stuff. So PowerPC you know with the SIMD Extensions ultimately ended up winning Sony and Xbox and Nintendo I think, I forget exactly, I don't recall.

Fairbairn: I don't remember either, yeah.

Diefendorff: I was I think gone by that time or something, but I don't remember. So MIPS was out, and so then I hooked up with Peter Song who was the architect of the PowerPC 604 at Somerset, and his idea, which I thought was a good idea, it was something I'd always thought was necessary and nobody was doing and I didn't understand why, is he was building at Memory Logix which was the name of his company.

He was building a very low power x86 for embedded control, synthesizable. So it would have been IP, it would have been low power, but it would have been x86 compatible. Now that may seem like an

aspiration because x86 isn't known, I mean, they've always struggled, they're struggling today with low power for x86s kind of thing. But it turns out that it can be done with the right microarchitecture and with a smart enough team that you can handle the complexity of it, and at the time the modern technologies and tools, I thought we could actually do that. So, I hooked up with him for a while, mostly in search of VC funding for the thing, so I didn't contribute to that project much in the way of CPU design.

Fairbairn: You contributed your resume.

Diefendorff: Yeah, I contributed my resume and a lot of hard work working with VCs, which you know is not something I want to do again, that wasn't a good time. Well, that aspect of it wasn't a good time. But I was convinced that Peter could in fact do what he was saying he would do, there was enough evidence that that would work, that he would do it, and so we worked a lot on VC funding, got a couple of other people involved in that, in getting VC funding effort kind of stuff. So that went on a while, and finally got Vinod Dham interested, from New Path Ventures, and he was interested, and he was willing to put up money for it.

But Vin because of his history, I mean, he was Pentium architect, NexGen, because of that history he always wanted to compete with Intel, and I kind of joined up with Peter because he was doing x86 not competing with Intel. So I have always thought, well, not always, since you could see the light sort of that competing with Intel was just suicidal. You could try it if you want, but you're wasting your time and money, and so I never wanted to do that. But Vin did, and so he gave Peter the money but he hooked him up with the NexGen architects, Greg Favor, Mike Yamamura, those kind of people, and I wasn't inside at that time because the minute that became clear that that's what Vin wanted I said I'm just not a participant, I don't believe in it, I'm not going to go there. It just so happened, and these things, so many of these things just, they just happened, you know I don't know how things happened at the time they happened, but Apple calls me.

Fairbairn: Timing is everything.

Diefendorff: Apple calls me back and says we still need a PowerPC architect again, because they had come, they were on their way back up, yeah, now they're coming back up. So I was gone the time they were like this and now they want me back, and so I said no thanks to Vin and Peter, and then that became Montalvo Systems, and that was a big effort and I don't know the history there. It was after, it was completely after I was gone because once I figured out what Vin wanted I didn't want that.

Fairbairn: Yeah, I don't remember the name Memory Logix but I do remember the name Montalvo Systems.

Diefendorff: Yeah, so that was the root of it is Peter and the NexGen architects Greg Favor and Mike Yamamura.

Fairbairn: So back to Apple, they were still PowerPC.

Diefendorff: It was still PowerPC, and it was G5 by this time, and so the job was the same, you know work with IBM to make sure that these chips have what we need, what we, Apple, need for the future. So Jon Rubinstein who was at Apple, he was the one that called me up to see if I wanted to come back because he was sort of Steve's right-hand man at NeXT and followed Steve to Apple. So he called me up and to see if I might be interested in this kind of thing and then he handed me off to Bob Mansfield who was running all of Macintosh Hardware at the time, so they hired me back there at Apple primarily to work with IBM as the Apple architect team.

Fairbairn: Rejoin the team again.

Diefendorff: Rejoin the team, and now at this point there's a lot less architecture work to do. You know we have the SIMD instruction set, the instruction set is what the instruction set is, so mostly it would have been microarchitecture, process technology, chip issues, this kind of thing, but it was to work with IBM to get the right chips. The thing that IBM wasn't getting that Apple wanted that was sort of the hard sell is Apple wanted low power, they couldn't deal with things like G5 kind of thing. And there were other chips, but even those other PowerPC chips, but they were either manufactured by IBM or by Freescale at that time, Motorola had sold its semiconductor business to Freescale, and they just really never got how low power Apple wanted.

When Apple said low power the vision at IBM especially, but even Freescale, was a different level. They weren't talking, it's like Apple wants power down here and they're thinking low power is up here, and so they just didn't get that. But the other thing that Apple said but we don't want to sacrifice any performance for this. We still need performance, and we need more performance than we're getting from the chips that you're giving us and stuff. So that was always Apple's nature is to push the edge just a little bit more than it could be pushed. In this case though I think Apple was right that there was just a huge amount that could be done in power, both in process technology and architecture and microarchitecture to substantially lower power. I think Apple is kind of proving that today with their M1 and that kind of stuff, that just is a whole different level of power, and IBM and PowerPC didn't get it. And the other thing is that at this time Intel had in fact added all the SIMD stuff, SSE and even next generations beyond that, into their chips, and so sort of the media performance gain that we had was it was still high because AltiVec was a better part, but it was a better vector instruction set, more targetable by compilers, more easier to use, all that kind of stuff. But that gain had been reduced by [Intel], and [Intel] was just kicking butt in single-thread performance kind of thing, and they had moved on to multiple CMP, to multiple cores.

Fairbairn: Apple or Intel?

Diefendorff: Intel.

Fairbairn: Intel was kicking butt.

Diefendorff: They had moved on to multiple cores and all of this kind of stuff. So now, I mean, here my story has to, I can't, because under threat of being, of summary execution I can't talk much about what I did at Apple or why. Maybe I can talk about a few of the high-level things, but--

Fairbairn: Ten years on it's still a--

Diefendorff: Ten years on, partially because I told them I wouldn't, and partially because I wouldn't do it. Apple was the world's gre--, TI was tough, but Apple was the world's greatest company then. I think that company is just super well managed.

Fairbairn: I was going to ask you sort of the, you know we started with the culture story at TI and we didn't really touch on Motorola, but--

Diefendorff: No, Motorola's culture was less appealing to me. It was much more confrontational. You know I can deal with, and certainly have dealt with confrontation, but it was a culture of it. It was a culture where there would be more people trying to undermine other people, you know? I hate to blame--

Fairbairn: Political confrontation.

Diefendorff: Yeah, political, that's exactly the right thing, political, that's exactly the right word, that I didn't have at TI, and Apple, I don't know, it may have a little bit of con--, but it's a totally different thing. At Apple, in the first place they were super well managed. I just have absolutely nothing but the ultimate respect for Steve and his vision of where he wanted that company to go and what people wanted and all of that kind of stuff, and I think he instilled that culture, I mean, Apple has had several other CEOs than Steve, you know Sculley and Amelio and some of these guys, and I don't care how good they are as CEOs, they never could have managed Apple. Apple was Steve Jobs. He made it in his image. It was a culture that he started, he fostered, and it was what it was, and it was highly innovative; it was engineer everything absolutely to pieces; no fault is acceptable, no bug, the inside of the computer has to look as good as the outside of the computer. There was no detail that was acceptable--

Fairbairn: Overlooked.

Diefendorff: --overlooked and no detail that could be tolerated if it wasn't as best as the technology would allow. And the people knew that, and the people knew if they had to go to Steve with something that was less than perfect they may not be there the next day. Steve was ruthless. But the management that he put in place, that company is super ethical, they, I don't know if it's, no, no, if it's conscious through their executive efforts or if it's just left over from Steve or if it's just the way it is, Apple's culture is a great one, I really like--

Fairbairn: So did you have confrontations with Steve?

Diefendorff: No, I didn't, you know I mostly didn't have confrontations with Steve, of course knew him and met him and that kind of stuff, and actually more at NeXT, while he was at NeXT, because it was a much smaller group. You know when I was back at Apple and Steve was there it was a giant company so I didn't have many interactions with Steve personally, no.

Fairbairn: So, I mean, but your work at Apple was on architectural design work?

Diefendorff: Yeah, chip in general, but architecture design work, and then you know one thing I think is public knowledge is that at all times in the history, back when we were 68000, back when we were PowerPC, back every place, Apple would always port their OS to other architectures. In the background in a lab somewhere OS, whatever it was, was running on something else, and because they always wanted to keep their options open. And at some point in time after, I don't know, when was that, I don't recall exactly but a couple, three years after I went back to Apple the execs had seen the light and figured that it was just impossible to compete with Intel.

IBM's yields just weren't good enough, their frequencies weren't high enough, it wasn't going to happen, and the x86 had just found so many ways to get around the performance limitations of their CISC architecture and now they had all the media stuff, or some of it, a lot of it, enough of it, and we could run our OS on that end, and a matter of fact we could emulate PowerPCs to a pretty good equivalency on that, and so it suddenly one day became my job to interface with Intel rather than IBM.

It was like a light switch, I mean, it was strange, funny in ways but really strange because instead of flying down to Austin I would fly to Oregon or fly to Israel to meet with Intel architects, but the job was the same. Of course we had less architectural influence. We still wanted what we wanted, we still wanted chips that had lower power than Intel was delivering at the time. Intel has always struggled with power. They were better than G5, but they weren't anything like what Apple needed. So my job suddenly became to sort of convince Intel to step up their game on power without sacrificing performance and all kinds of things. I mean, packaging technology, Apple always wants the world's thinnest package so that they can put it in ever thinner, I mean, Apple has--

Fairbairn: iPads and notebooks and other things, yeah.

Diefendorff: I mean, they have an overriding goal kind of has been smaller is better, and that's that, and so, well, thinner is especially better.

Fairbairn: It's all part of that, yeah.

Diefendorff: It's all part of that kind of stuff, so all of that kind of stuff. So I got involved with Intel from many fronts, not just processor architecture, but packaging and process and everything, but not much process because we kind of thought they had that nailed, and at that time they did. At that time Intel was two generations ahead of anybody else on the planet in terms of process technology. Of course, they've stumbled along the way and that's no longer true.

Fairbairn: So Apple eventually did adopt the Intel architecture.

Diefendorff: Yeah, they did.

Fairbairn: And then they switched again later on.

Diefendorff: Yeah, that was, well, I retired--

Fairbairn: That was after you left.

Diefendorff: Yeah, I retired from Apple, what, 11 years ago now, and so back at that time, at least to my knowledge, there was no real talk or even a perceived need for Apple to do their own processor.

Fairbairn: They wanted low power, but they were just going to get the best that they--

Diefendorff: But they thought that they could get that from Intel, and as Intel stumbled a little bit with process technology and they had big problems at some of their nodes and leakage and that kind of stuff, and you know some of their recent nodes have been delayed, and as we've seen that. So right now I'm talking about things I don't really know about, I'm retired, you know I still have friends at Apple but--

Fairbairn: But maybe that's what I wanted to know was whether, was there any sort of whiff or thing of when you did leave, they didn't have the infrastructure to go do their own processor.

Diefendorff: No, they did not.

Fairbairn: So they had to build that afterwards.

Diefendorff: Right, that's right.

Fairbairn: I mean, that was a major long-term undertaking.

Diefendorff: Yeah, and there may have been visions of it, I mean, I don't know what Tim or Bob, I don't have any idea what their long-term vision was. But I know just some conceptual things that are obvious from the outside is. When they came out with the phone, the first thing you look at and you say, well, that's a dang computer, that's a real operating system, and, of course, knowing what's in it I say, well, there's no reason that can't run Mac OS. So all of those kinds of things are sort of obvious to any onlooker. It doesn't take any insight from me or anything like that. But it's clear that from a development cost point of view making MacOS and iPad OS and the iPhone OS more and more similar, which might mean running on similar hardware kind of thing, is a good idea.

Fairbairn: What was the original iPhone, what was the core processor?

Diefendorff: It was an ARM processor but it was built by, I don't think I'm saying anything.

Fairbairn: StrongARM or whatever.

Diefendorff: No, it was built by Samsung.

Fairbairn: Oh, okay.

Diefendorff: I think, I don't even recall.

Fairbairn: It's not critical. I was just curious.

Diefendorff: It's not critical, but that's what I remember.

Fairbairn: All right, well, I won't push you anymore on your Apple time. So you retired from Apple, what made you decide that was the time to retire?

Diefendorff: Oh, I don't know, I sort of always wanted to retire a little early only from the point of view that I figured it would be nice to have some retirement while I still had health. The longer you stay the less, I mean, I'm so busy these days I'm over committed 10X, the time I have, I'm really busy, so there's a lot of things that I've always wanted to do and could never do kind of thing because I spent all my time working, and so I kind of wanted to do that while I had the health to do it.

Fairbairn: Your physical and mental faculties. So you'd been in Silicon Valley for quite some time, did you immediately want to return to Texas, was that a--

Diefendorff: No, no, no, we had no vision of that. California has great weather and I have a lot of friends here. All my friends for the last 25 years are here, and so I had never planned to leave. But for my son and his wife and now my grandchild the Valley is a very expensive place to live, I mean, killer expensive, and they're not techies; they don't work for Google, and so they had started thinking about other places to move, and taxes. You know I mean there's a cost, there's a tax to live in Silicon Valley.

Fairbairn: I tell my kid get out of here, you can't live here.

Diefendorff: So they thought that they might like to do that, and we said, well, you know we have property in Texas and all that kind of stuff, because my son was actually born in Texas so he's 36 now, so he was born in Texas when we lived there with TI and Motorola, and so he was all for moving back to Texas. So we all decided we'd move back to Texas and we'd buy a temporary house and build two new ones so we'd each have our own house so that's what I'm spending my time doing today.

Fairbairn: So you've got land, you've got Longhorn cattle you said?

Diefendorff: Yeah, we have a ranch.

Fairbairn: How big a ranch, how many cattle?

Diefendorff: It's a 110 acre ranch, so it's not big. But it's enough to raise some cattle and some goats and horses.

Fairbairn: Do you make money raising cattle or is it just a hobby?

Diefendorff: No, no, no, well, you know you might. In order to make money raising cattle I think you have to raise a lot of cattle, and to raise a lot of cattle you have to have the land too, a lot of land to grow hay.

You can't afford to feed a lot of cattle, and our ranch doesn't have the kind of water we would need to grow enough hay to feed enough cattle to make any money. But we still like the animals and ranching and all that kind of stuff, so we have tractors and bulldozers and that kind of stuff, so that's fun toys kind of.

Fairbairn: I know that you've also gone on to spend a fair amount of time doing some consulting in patent litigation and that sort of thing.

Diefendorff: Yeah, I do a little bit. You know it's one of those things you have mixed emotions about. I'm in a way not a fan of our patent system. It seems to me it doesn't always protect and reward the people it should, and I think it gets abused. I think there's a lot of patents that are just junk, I mean, why, and it takes a lot of bandwidth for lawyers to sort that all out. So I'm not a big fan, but lawyers do call me up from time to time wanting expert witness assistance, either technical or with expert reports or testifying or whatever. The case that they bring to me may be just particularly interesting. I mean, there's some that I've actually had IP that would relate to the case that they're kind of interesting and I have special knowledge about. There are some that are just interesting topics in technology. I still, you know I'm on the Hot Chips committees and steering committee and organizing committees, and so I still like to stay up on the technology, and so they come to me with an interesting thing. And if I see that, okay, I think there's a party here that's being wronged and the lawyer is on the side of right, sometimes I'll take a job. I try to avoid it because it can become a big full-time job and I've got other full-time work, so it's just a sideline thing that I'll entertain from time to time.

Fairbairn: Is there any particular case that you would, you sort of look back on and say, boy, I'm glad I participated, and I really made a difference in terms of the--

Diefendorff: Well, there is one big one I, fortunately all these things are sealed. I don't know what's public, what I could say. But there was one that I took. It was one I didn't think I had a lot of expertise in, but it turns out that I did just because sort of the name of the case didn't turn out to be exactly what the real underlying issues were. So I was able to contribute a lot to that case I think. It turned into a very big case, and I thought we were on the side of right in the case, and I learned a ton of stuff. It became a very big case and I learned a lot.

Fairbairn: Can you say at least what the case was? I mean, you know this is public knowledge, right?

Diefendorff: I think so. It was in Nvidia v. Samsung. I think it's, I hope it is, if not I'll have you edit it out. But that was a great experience actually. Because I did a lot of different things that I'd never done before, had to look into technology in areas that weren't my, you know what I would claim to be my main focus in my career but were nonetheless very interesting, very interesting issues. The lawyers in the case were great, on both sides, and so that in hindsight was a great experience. During it, it could be painful, you know because the other side is always trying to destroy your credibility as a witness and all. So it can be daunting from that point of view. You kind of have to have a thick skin. But it all went very well, and so it turned out correctly, in my opinion, and that was a good experience. The other ones have been much

smaller and so no real super learning I think from those, but worth contributing to. I guess the thing is can I contribute to this, and if I can't then I don't take it.

Fairbairn: Okay, is there anything that, we're sort of getting to the point of wrapping up, is there any other point that you want to make or thing that you want to talk about or a story or whatever?

Diefendorff: Oh, probably a million things that I will remember tomorrow.

Fairbairn: We can always have you back again.

Diefendorff: So not that I can think of.

Fairbairn: All right, well, thank you very much, very much appreciate the time you've given.

Diefendorff: Well, thanks for the opportunity.

Fairbairn: And, like I said, we've had you on our list for quite some time and I'm delighted that we were able to have you in.

Diefendorff: Okay, thank you very much.

Fairbairn: Thank you, Keith.

END OF THE INTERVIEW