SPARC Microprocessor Oral History Panel
Session One
Origin and Evolution

Anant Agrawal, Robert Garner, Bill Joy, and David Patterson

Moderated by: Linley Gwennap
Introduction by: Uday Kapoor and David House

Recorded: June 9, 2011
ORACLE Conference Center
Santa Clara, California

CHM Reference number: X6182.2011

Kapoor: Okay. Well, welcome everyone. I'm very excited about this event. It's taken us a while to get everybody together. Unfortunately, one of the panel members, Wayne Rosing, will be late. He missed his flight this morning and hopefully he'll be here later. So we'll see how we could work around it. He sent some talking points, which I have given to Linley. So anyway, so welcome on behalf of Oracle and the Computer History Museum for this very special occasion to capture the oral histories of the SPARC development, one of the very special architectures, a very, very great architecture I believe. The Computer History Museum has made a commitment to capture, to record, and to preserve the computing paradigm and the computing revolution. And I think it's very important that-- we are so fortunate that this is happening. And later on, I would ask Dave House to say a few words about the context. Because he is on the board of the Computer History Museum, and also chairs the semiconductor special interest group, of which I am part of. And we have been working to do this for some time. We have of course a very distinguished panel here. These are the folks that made SPARC happen. And I think we have an opportunity to hear about the very early parts of how RISC happened and why the Berkeley architecture for SPARC, for example. Again, I was actually quite privileged, it so happened, to have worked on SPARC really early on. I was working at Intel under Dave House, and I was approached by my first boss to start a microprocessor team at Cypress Semiconductor, at that time had the fastest CMOS process. And they had a deal being worked on with Sun. So T.J. Rodgers spent some time with me to try to convince me to join Cypress to do this project. And so ultimately, I took the "RISCy" decision to move from a CISC (Complex Instruction Set Computer) to a RISC architecture. But I was very happy to do that, and I think I really was very, very happy to have support from the Sun folks. I had an office at Sun and Anant Agrawal and Wayne Rosing were my hosts. And so we put a team together and delivered the first custom integer unit in the CMOS process, fast CMOS process. That was right after the gate array that was done with Fujitsu. So again, I was very privileged to be part of that. So with that, I would like Dave House to say a few words about the Computer History Museum and the context.

House: Thank you, Uday. The semiconductor special interest group is one of four special interest groups at the Computer History Museum and focuses on capturing the history of the semiconductor industry and the ways that it relates to the computer industry. Obviously, advances in silicon have been critical to the advancements in computing. We decided to take a unique approach at the Museum. The Museum has hundreds, probably thousands of histories, individual histories of leaders, pioneers in the computer industry. We decided, particularly in the microprocessor area that we would do team histories. When you talk to various members of the team one on one, you get different views of the elephant. You get them all together and you get an aggregated view. We have done literally dozens of these, from Intel, from National, From Texas Instrument, MIPS, a number of other companies. And SPARC has been one of the holes in our collection of team oral video histories. But we've got a persistent group of people that don't give up, and Uday took the charge there in working with Rosemary and the other members. We want to thank this panel from assembling from around the world and United Sates here. So I think we should start the activity. Let's get going. Welcome.

W1: Everybody's cell phones off.

House: Cell phones.

Kapoor: So I wanted to say a few words about the protocol. First of all, normally these recordings have happened at the Computer History Museum. And it just so happened for our real privilege, it was not available because it was booked solid by Intel. And so we are here, and it's just appropriate that this is being done on Sun campus. So the typical environment in the recording is like a studio recording and there is no audience. But since we had this opportunity, I wanted to invite some very special people. This is not sent out to a broad public invitation. So what is requested is for you to please listen and not interrupt, not ask any questions. This is really a studio recording, that's the protocol. I'd like to introduce the moderator, who will then start the proceedings. So we are again very privileged to have Linley Gwennap, very well recognized technical publisher in the industry, and especially in the microprocessor field. So I'll say a few words about Linley. He graduated cum laude from Yale University with a BSEE and spent eight years working on RISC systems at HP, was the program manager for HP's 810 and 815 UNIX systems. Later, he did product marketing for HP's PA 7000 family RISC processors. He then joined MicroDesign Resources as an analyst and served as publisher and editorial director of Microprocessor Report, which as probably all of you know, we have read all these years with avid interest and a lot of, you know, it's an excellent report. Under his leadership, the publication won the Computer Press award for the best industry newsletter four times in six years. He then founded his company, the Linley Group, to provide technology and market analysis to a broad group of clients. Linley provides high-level consulting to industry on product positioning, strategic analysis and competitive assessment, and has assisted Freescale, Applied Materials, HP, IBM, Intel, and several smaller companies and investment firms. He has published hundreds of articles in publications such as EE-Times, Upside Magazine, Nikkei Electronics, Electronic Business, Electronic Design, and the San Jose Mercury News. He is often quoted in the Wall Street Journal, The New York Times, Business Week, and the CNET website. So again, we are happy to have him here to moderate this event. With that, I will ask Linley to start the proceedings.

Gwennap: Thanks Uday. Thanks for that introduction. Before we get started with the panel, I'd like to introduce the members of the panel to you. And I'm afraid I can't do as good a job as Uday, so I think I'm going to ask everyone to introduce themselves. So why don't we start with Robert Garner, and just tell us about yourself.

Garner: All right. So I started really a lot of my training at a place called Xerox, Xerox System Development Division, where we had worked on the Xerox Star, the first commercial workstation. We learned that workstations actually had user interfaces with bitmap screens and Ethernets, embedded Ethernets. I was twisted by Bill and Eric Schmidt to basically join this little dirtball company [SUN] doing a 68000-based workstation. But they were inspired by the Xerox vision of Ethernet and networking and whatnot. So I joined Sun in [March] 1984, was the lead architecture with these guys on SPARC, and worked on the Ultra-SPARC program here. And then I left Sun in '98 after working on some Java projects. At that time, I joined Brocade Communications, that were doing fibre channel switches. And at this time right now, I'm at IBM working in Almaden research in San Jose.

Gwennap: Great. And we also have with us Anant Agrawal. And Anant, can you give us a brief introduction?

Agrawal: Yeah. So my name is Anant Agrawal, I joined Sun in April of 1984. I had just finished a stint at a failed startup designing an IBM-compatible mainframe. And there I left a floating point unit implementation, which was about 16 chips. I stayed at Sun for 18 years. For 17 of those years, I led the SPARC processor group. So probably most of the processors except I would say two, Robert?

Garner: Uh-hm [Yes].

Agrawal: The Super SPARC and the Cypress chip came out of my groups. Thoroughly enjoyed it. Left Sun in 2002. And since then, I have been either CEO for startups or doing consulting.

Gwennap: Great. And we also have with us Bill Joy. Bill, can you tell us a little bit about yourself?

Joy: Yeah. I was a graduate student at Berkeley where I met Dave and learned about RISC architecture for the first time, and was working on UNIX software under a DARPA contract, where I was actually a university employee, a programmer, being paid while a graduate student to write and improve UNIX for the research programs at DARPA. And then in 1982, the other cofounders of Sun attracted me to come and be a founder. And shortly thereafter, I hired Dave as a consultant to help us to try to do better than the Motorola 68000, which wasn't meeting the performance needs of our customers, especially in floating point. And so thereby helped instigate the SPARC program at Sun. And these days-- I was at Sun until I think around 2003. And since 2005, I've been at the venture firm Kleiner Perkins Caufield & Byers doing investing in renewable energy and materials companies.

Gwennap: Great. And rounding out our panelists here is Professor David Patterson. Dave?

Patterson: Yes. I just figured this out the other day. I've only had four jobs in my whole life. I've been at Berkeley since 1977. I was consulting at Xerox PARC, I was interested in Smalltalk as a programming language and in architecture support for that. I was kind of impressed at the difficulty of getting that company to turn ideas into products. Bill Joy called me up and said he wanted me to drop by to talk. And I came to Sun's Building One, and because I'd visited Bill before and they were kind of in a garage, so I didn't believe Sun was real. But Building One was a whole building with two stories and it felt like a real company. And he said this company wanted to do a RISC architecture, which was bold, typical of Bill and Sun. And so I became a consultant for Sun. And I stuck out longer than all of you guys. I made it right up until the Oracle acquisition. I was a consultant. So I must have been a consultant for 23 years to Sun or something like that.

Gwennap: Great. And as Uday mentioned, we will be joined later by Wayne Rosing. So we'll work him in when he gets here. But to get started with the discussion, I think we should begin at the beginning. So if we look back to the early days of RISC, there was the IBM 801 project, the Stanford MIPS project and then the Berkeley RISC project as well. So Dave, maybe you can start by telling us a little bit about what was going on in these very early days of RISC and how some of those ideas evolved into what eventually became SPARC.

Patterson: Yeah, so my story is that I graduated from U.C.L.A., I had actually worked at Hughes Aircraft while a graduate student supporting a wife and two kids. As I think back, that's where I learned computer architecture, I just took software classes. But I ended up doing hardware at Hughes Aircraft. And they actually had a wafer-scale computer, that was one of their claims to fame for their radar division. We did micro-programmed designs. And we did a design study while I was a graduate student, I was just thinking about that this morning, in 1974 and 1975, where we kind of invented a language and basically we could program the microcode directly. And it seemed really fast, just a little hard to program. I remember we called it the Hughes Processor 234, or the HUP 234, if you remember military stuff. But we did this paper design that didn't go anywhere. My dissertation was about microprogramming and verification of micro programs. So that plays into a story is that Sam Fuller, who was a Carnegie Mellon professor but then joined DEC as I guess a vice president of engineering, knew about my microcode research and invited me there on sabbatical. And so I took the Fall of 1979 off with my family and lived in Boston. And we worked on microprogramming tools for the VAX. And at the time, with Moore’s Law and the Mead-Conway [design approach], it was pretty clear from the Berkeley perspective that microprocessors were going to be the most important technology for processors of the future, even though the VAX was many cabinets worth of things and it was hard for DEC people to believe one chip could replace many cabinets. But since microprocessor designers weren't that good at computer architecture, they just imitated the minicomputer companies. And so whatever the minicomputer companies would do was a good predictor of what the microprocessor guys were going to do. So I was kind of stunned at the complexity of the VAX, how difficult it was to get these 500,000 bits correct for the microcode. And in fact in my visit there, I knew that microprocessors were the future, but that nobody was going to come up with the microcode right the first time. The VAX, which was a very successful computer, had 50 bugs that they had to fix in the field. And they had FPGA-like materials and stuff to fix it. When I came back, I wrote this paper, which I just found. It was called “The Case for a New Control Store Architecture for the Next Generation of VLSI Computers”. And in it, it talks about how microprocessor guys are going to create minicomputers, instruction sets are getting more complicated, they're going to have bugs. And so I had kind of a microcode cache on board. Well, I submitted this to IEEE Computer and it was rejected. And so I was convinced that if they were going to imitate the minicomputer guys and build these complicated instruction sets and use microcode and do it in VLSI, it wasn't going to work. You couldn't repair all the chips in the field with microcode. So it was like this can’t work and it's a bad idea to build this way. So that kind of set the stage for what we did at Berkeley. The IBM 801 project was ahead of us. I don't know when it-- Robert might know, '75, '76?

Garner: Yeah, about '74, '75.

Patterson: Yeah. And it was secretive. The Datamation article that came out was the most information about [it]. John Cocke of IBM visited us sometime. He spent a day at Berkeley and talked about ideas. I was trying to send an email to people to find out when. But the thing that set me up was the VAX, the kind of negative experience on the sabbatical at DEC, you know. Is VAX doing the right thing? Cocke came by sometime and spent a day and inspired us. I found the slide, back when we made slides. So to document what happened, we started a graduate course, which we often do at Berkeley to explore new ideas. And that spring, we were still on the quarter system then. So March, it was kind of time to investigate the architecture ideas and then we started building a chip. And the first chip is right here on the slide. But over, let's see, one, two, three, four chips and we taped out I guess from the first course to chip coming back was like 18 months. And that was RISC I. Subsequently, MIPS started. I just got an email from John Hennessy. I think they had their kind of architectural discussion in I think 1981. So the MIPS project at Stanford, and I think those were the first three RISC projects. I think the things that happened after that were all commercial events. I forgot the name of the guy from the 801 project who came to HP.

Gwennap: Joel Birnbaum?

Patterson: Joel Birnbaum, when he left he was frustrated I think that the 801 project wasn't let out. And so when he came west to run HP Labs, he started a RISC project. There were a bunch of start-up companies.

Garner: Pyramid was one.

Patterson: Pyramid. There were a whole bunch of them that once the ideas were out there, a bunch of people tried things. But I was kind of astounded that this little tiny company was going to build their own microprocessor. And it was such a contrast to Xerox PARC that these guys were going to build a product and there was nothing that those guys at Xerox PARC could do to get a product out the door, that it was extremely tempting.

Gwennap: Now is it true you coined the term RISC?

Patterson: It was actually Carlos Sequin, who was more senior than I. I was still assistant professor at Berkeley, we would drive back and forth to these, I think they were dinners for COMPCON Conference. It was kind of like how Hot Chips still is done today, is you give people a free meal and they drive 50 miles to get the free meal, and then they would invite people to make a different one. And we were going to that dinner. And he in the car proposed that we call this idea RISC. I actually invented CISC. That summer, I think when it looked like-- I think it was that summer that I wrote-- kind of this was called The Case for a New Control Store, I think that summer it was The Case for the Reduced Instruction Set Computer. And then I sent it to Dave Ditzel, who was my student, and he and I coauthored that paper. So it went pretty fast. DARPA played a role there too. We didn't have DARPA funding. Bob Brodersen, who was a colleague, I kept pitching multiprocessor projects. And Brodersen finally said, "What do you really want to do?" And I said, "I'd really like to build a computer." He says, "Why don’t you propose that?" And so that led to the Berkeley stuff. Lots of graduate students and then Carlos Sequin as a faculty member, closely. Bill was around. Bill was a force in the department. He was a graduate student who was a-- people wanted to know what Bill wanted to know. I remember when he took the prelim exam, the faculty member said, "He either gets an eight or a ten. There's a question we disagree about. “If I'm right, it's an eight, and if he's right it's a ten." But he was kind of an intellectual force in the department. We just, you know, the future was obviously C and Berkeley UNIX. The other RISC projects, the 801, was PL.8. it was some kind of ECL minicomputer, and they had some funny operating system. The Stanford stuff was Pascal and--

Garner: And Bell Labs had their C stuff.

Patterson: Yeah, that was [David] Ditzel.

Garner: Ditzel, yeah.

Patterson: And that was later.

Garner: Yeah.

Gwennap: Mm-hm.

Patterson: But I think that was it.

Gwennap: You mentioned the Berkeley UNIX. I think Bill played a big role in that.

Patterson: Yeah, I would say Bill played a big role in Berkeley. One of my favorite stories is, as a Berkeley professor, Stanford's always getting the credit. So Bill Joy wins some kind of student award competition for Berkeley UNIX and the Communications of the ACM, Bill Joy, Stanford Student, wins the student award for Berkeley UNIX. And nobody saw any conflict in that.

Gwennap: So Bill, you want to talk a little bit about the Berkeley UNIX and the impact that had on the RISC projects in development?

Joy: Yeah. I should say that, let me just first react to what Dave said before I forget all about the narrative arc that he had. At Berkeley first we did software on a PDP-11. Basically, Ken Thompson came back, --he had been a student at Berkeley before going to Bell Labs--came back for a sabbatical from Bell Labs. And Ken and Chuck Haley and I, while Ken was there and after Ken left, tried to make UNIX not crash so much. Because UNIX was really a research operating system. And the UNIX code didn't, as many programs written by people who were just learning to program, worry about corner cases, but an operating system has to handle such things gracefully So it's very tedious work. Just every time it crashed, we'd rewrite part of it so that it would be more robust. And then Berkeley had never had a large computer other than-- well, we had the--

Patterson: CDC 6600.

Joy: CDC 6600 in the basement. But we really never had a PDP-10, a departmental-level computer like the major computer research universities--Stanford and Carnegie and MIT--did. These were 36 bit machines. So when the 32-bit VAX 11/780 we, with a smaller budget, got an 11/780 and an experimental version of UNIX for the VAX.

Patterson: That was actually a departmental policy issue. There were these people on the DARPA gravy train who had PDP-10s, and Richard Fateman said, "We're going to try this VAX."

Joy: Right.

Patterson: Right? This 32-bit address space VAX. And we were strange in that we picked kind of the wrong computer.

Joy: So what happened, this is where this ties into Dave's story, is so that we ported UNIX to the VAX 11/780. But then that version of UNIX which we had, which I put virtual memory in, we put TCP/IP in to hook to the internet and to be able to run Prof. Fateman's Macsyma program. I then started sending it to other people and they ran it on other VAXes. Then I went back east to DEC to see some other computers that they were developing. And in particular, they invited me back to someplace in New Hampshire and I ported, by staying up all night for two nights, I ported UNIX it to a VAX 11/750, which was a gate-array machine. Which was kind of nice, it was almost as fast as the 780. But it anticipated what we did later with SPARC. It showed we can build a computer with a gate array. But then there was the 730, and that was a piece of junk, right? I mean, because it had all the problems that Dave had seen. It was all microcode, and it was incredibly slow. And it was full of bugs. I couldn't get the operating system to work because a lot of the privileged instructions didn't work. And it wasn't the same-- they got it to work for VMS, but the instruction set really didn't work. And they just didn't happen to use the instructions in exactly the same way that we did. And so this is exactly-- Dave had this thing about provable microcode. I mean, which anticipated some of the things we'd have to do later when we had multiprocessors. We'd have to prove that the protocols between the chips were correct, because you could never debug them afterwards. So I shared Dave's frustration.

Gwennap: Yeah.

Joy: And then time stood still. It's kind of like what happened with the semiconductors we got to one volt and time stood still, right? Nobody could get past it. So I was at Berkeley-the VAX 11/780 was about 1 MIP and VAXes didn’t get any faster for a long time. Meanwhile I saw the RISC ideas develop. And I went to Sun. And the other thing that happened is that at Sun, our big competitor was Apollo Computer. And we both had the same instruction set. Apollo's idea of how to make a faster computer was to basically build a gate-array version of the 68000.

Patterson: Oh, I didn't know that.

Joy: That would run in two or three MIPS. And it was really, really hard and expensive. This was like your $50,000 workstation, which no one could afford. But I never believed that-- I thought computers ought to be single chips. It didn't make any sense to me to build a whole board because of the wire delay. So when it became clear that Motorola was doing with their microprocessor line roughly the same mistakes that DEC had done with their microprocessor line, in other words, 68010 68020 68040, were getting more and more complicated. And they were slipping and they weren't getting faster anywhere near the rate that the underlying transistors were getting faster. Having a philosophical connection to Dave, I said, "Let's just do what they DEC did with the VAX 11/750, build a gate array version, but as a single chip." As long as the processor fits in a single chip, then the single chips can get better. But if you don't make that philosophical decision to have a single chip and instead go and do what Apollo did or what DEC did with the VAX 11/750 and the VAX 11/730, those are roads to ruin, because then the systems will never be cheap. You're never going to have single board computers. And the other thing was, we were at a point in time when there were reasonably dense fast SRAMS. And Andy Bechtohsleim had already built a very simple two-level MMU with just like 4K SRAMs or 1K SRAMs cascaded, a very primitive but serviceable memory management unit. And so the same technology basically could build a cache. I mean, the 16 or 64K versions of those. So we now could build, like the 750, we could build a gate array CPU, like Andy's MMU, we could build a cache. And you could work out just very quickly on a sheet of paper with just a pipeline drawing that we could build a reasonable clock rate thing that if we hooked it up to a floating point unit that wasn't micro coded in 80-bit registers and all the stuff that was clogging Intel's performance, that the thing could perform reasonably well. When I was an undergraduate at Michigan, been a program for the early supercomputers, the CDC Star, the TI ASC, and the Cray 1. And my view of floating point was that you have a multiplier and adder and you've got to keep them busy. The rest of the computer is just there to feed the floating-point unit. And the floating-point unit, like the Cray, ought to do one result per clock. Ideally, you had a multiplier and an adder and you'd get two results per clock. And that was so far from where the microprocessor guys were going with the way they organized their floating point [units]. And I didn't think we could ever do real scientific computation. So I wanted a microprocessor that would run at SRAM speed and would kind of produce a floating-point result for every clock. And that really was the point of departure for getting Dave to come to Berkeley. Let's build something simple but that has, you know, that really can move 32 and 64-bit words around.

Gwennap: So at the time, I mean, this was a pretty radical decision for Sun to adopt this new architecture. How did that decision get made? Who was involved and how difficult was it to convince the people to make that choice?

Joy: Well, you don't have to get everybody to agree on why you're doing something. You just either have to get people to agree or get out of the way. So it was disorganized enough that it wasn't extremely hard And it's not like we also tried to do it alone. At various points, we talked to MIPS, we talked to Motorola, we tried to get a common architecture with other people but nobody would-- people just wouldn't do what we needed. So from my perspective, I don't think it was-- it wasn't all that difficult to get it started. To get it finished is a whole other story.

Patterson: If I could add to the story here. As the project's rolling along and then MIPS becomes visible. And I want RISC stuff to succeed, and so I feel like, "Boy, I'd sure like SPARC to be a winner, but this can't be the right thing to do. The right thing for me to do is to fold my ego and recommend that we rally around one instruction set. It'd be better for the RISC industry to have one. I don't think MIPS is going to adopt SPARC, how about-- I think I should, as a scholar, should recommend we withdraw SPARC. And I go to Bill with this idea, and fortunately, he had a long list of arguments why that was a bad idea. But it was a really tough night. Oh man, this is what I should do. But then SPARC will disappear from history and Hennessy will get all the glory. Oh well, but this is the right thing to do. Yeah, so why didn't we embrace MIPS? Sorry.

Joy: Well, they wanted to charge us a $1,000 a MIP, which was, my view was that these things should be cheap, that memory chips were cheap, why shouldn't processors be cheap.

Patterson: They did it value-based pricing. It's a 10-MIPS processor. That's got to be worth three times as much as--

Joy: That was $10,000.

Patterson: Yeah, $10,000.

Joy: Which, compared to the people building gate arrays like the Apollo, that was the kind of pricing you'd get. But it was on the wrong learning curve, so that just wasn't going to work.

Garner: There was one other interesting factor Bill, you might recall. We were among one of a dozen 68000-based workstation companies. And one of the challenges was how would we differentiate ourselves.

Joy: Right.

Garner: And Bill, you had the vision that we needed to have a fighting machine. I looked at my notes, and you called it that. And Eric Schmidt would always tell me, he was co-managing the engineering department, "I don't know if RISC is a good idea, but we're attracting a lot of good talent because we're doing our own RISC."

Patterson: That's interesting.

Garner: He wasn't sure it was the right thing to do, but we were bringing exceptional people to the company.

Gwennap: And where did the SPARC name come from?

Garner: We can do that history.

Joy: I don't know. Turn it over to these guys.

Agrawal: Very interesting story. So we had an engineer called Will Brown.

Patterson: Berkeley student.

Garner: Yeah, he had worked on SOAR (Smalltalk on a Risc).

Agrawal: Yeah. And we were struggling with the name. And at Sun, especially in the microprocessor group, naming a project is always a special thing.

Garner: It was called Sunrise initially.

Agrawal: Right.

Agrawal: Sunrise was the microprocessor name. But for the architecture, so Will Brown decided to write actually a program to give various combinations of four-digit words, right?

Gwennap: Acronyms.

Agrawal: And he came up with like, I don't know, five pages of names?

Garner: Yes. He randomly ordered all the buzzwords at the time and printed them out and we looked to see what acronym sounded nice.

Agrawal: And that's how we named it.

Garner: And it stood for “Sun's Processor Architecture for RISC Computers.

Agrawal: But we first chose the name SPARC and then tried to justify what it meant

Garner: I heard it went the other way around. That we first saw SPARC and then it stood for that. Remember, he had all the names together, so there was Sun's Processor Architecture for RISC Computers, that was one of the combinations in the five-page printout. Later Wayne changed that to Scalable Processor ARChitecuter.

Gwennap: So it was less vendor specific.

Garner: Yes, exactly. That was the problem.

Gwennap: So Sun was committing to SPARC, what was now SPARC.

Garner: Can I make some more comments about that?

Gwennap: Oh yeah, sure. Sorry.

Garner: Before we commit to it. These guys pointed out quickly that there were many companies doing RISC machines at the time. I was looking through my list here. Even places like Signetics, you would never even have thought of. I mean, every, even the memory companies were doing it. They were all coming to us during this early time, in mid-'84 and saying, "We really want you to adopt our RISC." And we would kind of put a smirky smile on our face saying, "Hey, we're turning into a real systems company, we're not going to adopt your RISC." Motorola actually, I saw on my notes here, as early as mid-'84, August '84, started their own RISC effort. Mitch Alsop, I don't know if you remember him, started their effort. There were other – do you remember any of the others?

Agrawal: Actually, Motorola started it after we had disclosed to them what we were doing.

Garner: Yes.

Agrawal: Because we wanted them to adopt it.

Garner: There was a coincidence that they started after us.

Agrawal: Yeah.

Garner: Yeah, they recruited a guy [Mitch Aslup] from Denelcor who started a [RISC] project [what became the 88100]. So for us though, it was also very frustrating. One more comment on this. Because a high standard had been set. We had read the IBM 801 paper, there was another one in the [IBM] Journal of R&D that carefully described the 801.; every instruction was one cycle -- the whole mantra was single-cycle execution. But we were like a survival game, where we were in this warehouse with just crude tools to work with. We weren't full-custom chip designers. It wasn't obvious how we could achieve single-cycle execution right out of the gate. We'll talk more about that. But we had to work with the tools we had and be very simple. As a result, that really drove our decision making process to keep things as simple as possible and to say no to the other companies.

Gwennap: So let's talk about the initial project then. So the idea was to take this new architecture and put it into a gate array. Why did you want to use the gate array?

Garner: Actually, it didn't start with that. We started with options. Remember that we were talking to many companies. Would we do full custom? The problem was, we knew there were full-custom chip designers at MIPS. So we talked to Cypress.

Agrawal: But you know, the decision to do the gate array was probably the most crucial decision we made. Had we taken a different approach--

Garner: We would have failed.

Agrawal: --it would have taken us much longer and people like Eric would have run out of patience. It would have been expensive.

Garner: We would have failed.

Agrawal: Yeah. We would have failed. So I think that was one of the best decisions we made. Now Bill, you didn't say a very interesting story. When Sun came to recruit you in Berkeley, it was I think without Scott, I don't know, Andy.

Joy: Yeah.

Agrawal: And you sat there on your terminal for a long time making them wait.

Joy: Right.

Agrawal: And they later asked you, "Bill, why did you do that?"

Joy: Right. Because I was waiting for the adults to arrive.

Agrawal: <laughs>

Joy: But also, the interesting thing, I think this impressed Andy, I had a room full of VAX 750s, with the gate array machines on the fourth floor in Evans, like a half a dozen that were all connected by three-megabit Ethernet cards which Dave Boggs and I had built in the middle of the night at PARC. He had a design that some summer student at PARC had done. So we had our own vampire tapped three-megabit Ethernet. But when those guys came, I took them into the room and I just turned off one of the machines. I didn't shut it down, I just turned it off and pulled the boards out and showed Andy all the gate arrays because I thought that this kind of cool. I had enough machines, I knew that there was nothing going on in that machine. And one of the things you always have to adjust in the operating system is power failure, so it didn't really-- but they were very impressed that a software guy knew how to pull board out. It was a big board. And I had the courage to pull a board out of the machine. I was a fan of gate arrays.

Agrawal: And that's how I ended up at Sun. So let me step back a little bit. So I went to interview at Sun, and Sun was my last choice because I came from mainframe world, Sun was a systems company. And one of the engineers I worked with, Bill van Loo, had recently joined Sun. And he said, "Yeah, I could come and talk to these guys at Sun." So I talked to Dave, Bill, Robert. I'm not sure if I talked to Andy. Anyway, so Dave, Bill, and Robert explained the division that they just talked about. And I distinctly remember I was driving back from the interview and a strange thought went through my mind. I said, "These guys are Looney Tunes. These guys have never done a piece of silicon before, and now they're looking at doing a microprocessor. This is crazy." Then as I was driving further, I said, "You know what? These guys are so crazy if there's any group that can pull this off, it's these guys." And that's when I decided I should seriously consider joining Sun. and it's really that interview with these guys that made me--

Patterson: I remember you had a better offer somehow.

Agrawal: Yes.

Patterson: You were clearly probably going to do. And I think we tried to appeal to the intellectual challenge of doing it.

Agrawal: So yeah. So we decided to do a gate array instead of taking a traditional approach so we could contain the program.

Garner: Can I say one more comment on that?

Agrawal: Yeah.

Garner: So one of the companies we talked with, Bill, you'll remember this, is John Doerr was trying to connect us with two of his investments, Silicon Compilers. So we were talking with Cypress, which just had raw technology, but we had no way to build a chip. And so John was hauling us to his favorite company. And Silicon Compilers was Mead-Conway. And at Xerox, I had been working with Lynn Conway, where all you need is a red, green, and blue pencil to design a chip. But we knew that couldn't work, and Silicon Compilers, the chips they were making were all too slow, as Anant will remember.

Agrawal: Yeah.

Garner: So we also looked at wafer scale integration, when I looked through my notes here.

Patterson: We did?

Garner: We did. We talked to them, but they weren't really in the merchant business. We also talked to a company called CTI or something. I don't know if you remember them.

Agrawal: I don't remember.

Garner: And then so finally Anant really said, "Look, guys, stop goofing around talking with these slow things. I'm going to show you Fujitsu.

Agrawal: Well, that was an interesting problem, because we estimated that to do the first processor we'd need about 20,000 gates. And there was no gate array around that had that capacity. They were mostly four or five, maybe 10,000 gates. But then I discovered through a friend of mine that Fujitsu Labs was working on a 20,000-gate gate array, but it wouldn't be available for two years. So I come to Bill, Bill says, "Why don't we go to Japan?" So Bernie, Bill, and I get on a plane and go to Japan to meet with Fujitsu. So we land there, next morning, we eat for breakfast, and both Bernie and I are dressed in our black suits, ties. Bill walks in with his funky clothes, you know. And Bernie said "Oh my God, we are going and talking to this conservative Fujitsu company." But fortunately, it actually worked out great. Because they all knew Bill from BSD, so they had tremendous respect for him. And we were able to convince in that meeting, convince Fujitsu management, that they should let us be an early adopter. And they actually decided to customize part of the gate array for us, put a register file in there. And that was really the start of the implementation phase. And then of course at that time Robert and I used to share an office. And Bill's office was across. We had a huge open space in the middle. That's where we used to have our beer busts. So I started designing the pipeline, and I used to do diagrams like this, Bill if you recall, pipeline diagrams. And, you know, every day I thought I would be done. I mapped every instruction and I would share them with Bill, and every day Bill would come back and make changes and make changes and continue to make changes. And finally I told Robert, "You know, we should kind of close our door and not let Bill in." But at that time, I really got to appreciate Bill. I always thought of Bill as a software guy, but he was right in the middle of designing pipelines and making great changes. I don't know Dave, if you recall those, but it was just an amazing experience.

Patterson: Bill was, I assume still is, exceptionally well read.

Agrawal: Yeah.

Patterson: And so one of the things that convinced-- I believe what convinced Bill was Manolis Katevenis' dissertation. It actually won the ACM Award for the best dissertation. But in a nice, short dissertation, he makes the case for RISC and talks about how you would build it and implement it. And his design, RISC II, really got people's attention because it cycled in four hundred nanoseconds, which is two and a half megahertz. Which, this was a student design, two-student designed chip. And it got presented at ISSCC, which is the leading solid state circuit conference. And we actually did a shrink to the next technology and it ran at three megahertz at the time. I remember when Manolis presented the result, you know, audience of, I don't know, 1,500, some giant audience. And then he, "And here's our results," and how many transistors, how long it took, and the clock rate it ran on. And here's it running programs, and here it's running programs compared to Intel processors. And literally at that time, graduate students could build a microprocessor that was faster than what Intel could build. And just at the end of the talk, there was this kind of <crowd noise>. Everybody was talking for five minutes, there was a hubbub going on. There was electricity in the room. But I think that chip and Katevenis' dissertation was a pretty powerful set of arguments.

Agrawal: And then the simplicity is what made this happen. I remember the team was very small. I started the implementation, then we hired a great guy from Hewlett Packard called Masood Jabbar and Don Jackson to do the floating point controller. So it was really a team of four or five people that did the entire processor. We decided to use Daisy workstations at that time.

Patterson: I remember that.

Agrawal: And of course, I knew nothing about Masood really helped us out. So I recall us sitting in a room in the first floor of building one, in the corner. And at that time, I was dating Carmen, who is now my wife. And she was going to school and she started complaining, "You don't see me anymore." So we had a solution. We had her bring her homework here into our lab. And she would sit there all day working on weekends. And we would have lunch or go out for dinner together. But that was the solution. Anyway, we worked on it for about a year, then we did the tape out. Now, the tape out occurred a little before, a few weeks before their Golden Week. So Lee Quach who at that time was leading the effort from Fujitsu, and I flew to Japan because we had to convert the Daisy database into the mainframe. That's what they used for their backend checking. So the first day we land there, the second day we go to the Fujitsu office and they said, "Okay, here is today. Here is when the Golden Week starts. If you don't tape out before the golden week, it will be delayed by a month." So we said, "Holy mackerel." So the conversion process started on the mainframes, okay. Now, I don't know if you recall, but mainframes used to have something called JCL, Job Control Language, that controlled the priority of the jobs. So we go in and check with the data center manager, and our jobs were in pretty low priority. So I recall Lee Quach going out to dinner. He had some beer. And he says, "I have an idea. Why don't go into the data center tonight and change the JCL?" I said, "Are you sure?" He says, "That's the only way they'll tape out." Fortunately, he was a

Fujitsu employee, so he was able to get access to the data center. He changed the JCL. And of course, our jobs were running at very high priority. Two days later, we get called in, "What in the hell are you guys doing?" Fortunately by that time most of our jobs were done. The problem was, he found a bug. And it was a minor bug, but it was a bug. Unfortunately, we didn't have time to fly back to the US and make changes on the Daisy workstation, send everything back. So I think we called Howard Lee at that time and told him about this. And he says, "You know, you guys do what's right." So we actually went into the mainframe, the binary--"

Patterson: The file, right?

Agrawal: --of the design, and made changes to that.

Joy: Manually.

Agrawal: Manually.

Joy: You patched it.

Agrawal: Patched it. And we were, I would say less than 50 percent sure this would work. But this is what we had to do to get it done in time. Fortunately the chip came back and worked.

Patterson: So how many spins were done to the gate array before it was a product?

Agrawal: I think two.

Garner: My notes show two. Well one spin, just two versions.

Agrawal: Yeah.

Patterson: One spin, two what?

Garner: One first version, then one spin. But integer unit, floating point I think also something. By the way, I'm not sure, before we get too much into future implementations, could we talk about the architecture a little bit?

Gwennap: Sure.

Garner: Okay. So having all these high-powered people at Sun, you could imagine how charged it was inside. And if you've ever been around people designing computers, it's very religious. Everyone wants the kitchen sink, they want their own function in it. So one of my roles, and Dave helped, assisted me in this, was to help the prime generator of crazy ideas sitting between Dave and I to calm down. Bill had amazing negotiating skills that showed at an early age. He would say, "If you take this instruction, I'll stop asking for these two." We were actually swapping instructions to make Bill happy.

Joy: I wanted a double-precision floating point.

Garner: Bill was beyond irrational. We'd have these meetings where he would just be screaming, "I want this, I want this, I want that." At one point, right, I would get fired for this today. But at one point, I don't know if you remember Dave, I brought a water pistol and I shot it at Bill and hit him right in the face.

Gwennap: Bill?

Garner: Bill. Just to get him to calm down. At one point, Dave, we called Bill into a side room. And as your former teacher, he had--

Patterson: Had some respect.

Garner: Yeah.

Joy: Well I knew that if the thing was a success it would be almost impossible to change.

Garner: To change.

Patterson: That was the right instinct.

Joy: And so if you don't get the stuff in the first time, it's almost-- you know, the same thing happened with Java. We tried to put support for late binding into the virtual machine. And I didn't get it in the first-- I missed it by like a week getting two more instructions added. And it was 12 years before they got in there. It was a sense of much of history-- history is unkind to innovation.

Garner: So one of the things you really wanted badly was something to support the LISP machine people.

Joy: Yeah.

Garner: And so tagged add was one of the things. We allegedly put it in--

Joy: It didn’t work.

Garner: --in exchange for something else. And it's the only instruction on SPARC that had a bug. And it was probably my fault.

Joy: It was probably deliberate.

Garner: But we didn’t look at it closely enough. Yeah, it was probably deliberate. I'll never forget going to TI to talk about SPARC, because they had a LISP group there. Everyone was so solemn. I walked into the room, everyone was quiet. And I was like, "What's wrong?" And the first thing they said is, "You know, you have a bug in one of your instructions. It's the tagged add we can't use it." And I was thinking, "Oh great." One of the other instructions we debated putting in the architecture was population count.

Agrawal: Mm-hm.

Garner: And I'll never forget getting a call, "Meet me at a restaurant at somewhere," people I had never met. Then flying to the No-Such Agency.

Gwennap: No business cards, huh?

Garner: No business cards. No identification. Flying to the No-Such Agency [NSA]. That was fascinating. I saw Xerox Stars installed in the No-Such Agency. Which was kind of nice that someone could afford them. But we just said, "No."

Joy: Population count and sheep and goats were two instructions that would have given us a lot of business.

Garner: We just said yes to it the next version. So we did--

Patterson: Yeah, I just wanted to say that I think one of the best pieces of advice I've ever given out, well, as a consultant, was when we were trying about instruction sets. And I said, "When it doesn't matter, do it exactly the same way as the 68000. Don't argue, just copy the 68000." And I believe the--

Garner: I had my own version of condition codes and you--

Patterson: Corrected that. I believe the SPARC had virtually no compatibility problems, right, software compatibility. Almost none. And it completely changed instruction sets, right? And address size.

Joy: I wish we hadn't had condition codes.

Patterson: Yeah.

Garner: That was the area that I tried to make a change

Patterson: Well, we had condition codes.

Joy: But they're hard to pipeline.

Garner: So one of the other things that happened at that time architecturally is we had— we were a systems company. I mean, when I first joined, Eric [Schmidt] drew me an org chart for the entire company on one sheet of paper for engineering. We had a languages group, Steve Muchnick was head of it. We had people like Chris Aoki and a bunch of other people. We had a windows group, a graphics group, we had a systems group with the two Lyons, Bob Lyon and Tom Lyon, and Bill Shannon, the best OS person you could imagine. Dave Goldberg, who was a very free thinker. Dave said, "Go RISC all the way. Don't bother with 68000-based stuff." John Gilmore. So we had these people to draw from. So for instance, they could say decouple the call instruction from the register window change instruction. Then we had the whole thing of do we do registered windows. So I'll never forget Dave and Steve in a room together. I asked a simple question. We all knew about register coloring. Steve said it required two to three months--

Patterson: So register coloring was an IBM breakthrough that was well known. Hennessy worked on it too.

Garner: Extensively.

Patterson: We didn't do that.

Garner: MIPS had no interlocks. Steve Muchnick said, "I don't have two more people." So we were done.

Patterson: I think I asked, "Should we do register windows?" "Well, can you do graph coloring? And he said, "Are you going to do graph coloring?" "No, there's no way we're going to do graph coloring."

Garner: Didn't have enough people.

Patterson: Well, if you're going to do a RISC machine and you're not going to do graph coloring, you'd better do registered windows.

Garner: And it saved our butt. Because the compiler was straightforward, we could use the portable C compiler initially, simple, high performance, no advanced optimizations. Our compiler actually ran, produced code that worked, whereas MIPS, if you used -04 <laugh>.

Joy: There was a long history of this. It's like on the IBM 360 when I was at Michigan. You know, Fortran G worked fine, Fortran H, which was the moral equivalent of the register coloring compiler, did not. You could try compiling parts of your code with it, if they didn't work, you'd use G. So you'd have to use the optimizer only when it worked. It never completely worked.

Garner: Yep.

Joy: So I was pretty much convinced that register windows-- well, register coloring was great in theory, in practice do register windows

Garner: Now one more architectural thing. So one other architectural thing bothered me when I first got there was Sun had this bizarre MMU, this two-level segment page thing. And so the size of the tables was proportional to the size of memory, as opposed to a reverse page table. So we spent some time trying to figure out whether we should do a real TLB, Fujitsu did one later. But Andy's breakthrough of a very simple SRAM-based TLB prevented us from going down the wrong path and using a complex TLB.

Joy: Because that would have been a lot of micro coding…

Garner: And the other thing, I intentionally decided we'll put nothing in the architecture which is OS specific. Which is not the approach that was taken at HP on Spectrum. In Spectrum, all OS-specific stuff was encoded in the instruction set. We knew that we would be changing the operating system all the time, why would we want to pin anything down in an instruction set definition. So we intentionally left everything out that was OS specific, I/O specific, and made it not part of the architecture.

Joy: Another thing that bugged me was when we looked at the pipeline drawings that I always believed that traps should be the same as mis-predicted branches.

Garner: Mm-hm. I remember that.

Joy: And if you think about now when people go to threaded processes, going from one thread to another is basically like a branch, right? And when we had register windows, and I knew about people thinking about-- I didn't understand why we would want to debug the hardware twice, because there's a change of control instruction that's a branch or a call, it's basically the same thing on SPARC. One's PC relative and the other is absolute. So if you could unify all of them into one, then the whole design process would have been a lot simpler. I don't know if that's ever been done explicitly. It may have been done implicitly. But that's the kind of thing I was pushing you guys for. Because I knew if we could get this in from the beginning, our life would be better. But then we had to back off for a reason. We only had a gate array and Daisy workstation.

Garner: And just one more comment on the instruction set. One thing that I did was to make sure I did not leave a lot of room for extra op codes. Dave might remember this. I found in my book where I encoded the instructions. I actually took a quarter of the entire op code space for the branch instruction.

Patterson: Yeah.

Garner: Yeah. Well that way we could have a branch, always put branch in 32-bit--

Patterson: Well that was partly to support languages like Smalltalk.

Garner: Right.

Patterson: Right.

Garner: We could always do a branch in a 32-bit address space. Also the encoding for the other instructions. I intentionally made it as very few openings as possible so people would be not inclined to want to add anything. We did not think about 64 bits at all when we designed 32-bits apart. But when we went to 64 bits, it was a simple, painless extension. Bill, you added stuff for trapping, which was important.

Joy: Finally tried to fix some traps that were a little slow.

Garner: But we didn't think of 64 bit when we designed it, but it was so simple to extend it, it was almost brainless.

Agrawal: So let me continue down the product space.

Gwennap: Just watch the mike.

Garner: Are you going to leave the gate array though?

Agrawal: Yes.

Garner: Okay, before you leave the gate array, we've got to show the Sun 4 board. Sun became a $2 billion company based on the gate arrays <pointing>. When we said single chip, he's talking about the integer unit. In addition, there was the floating-point control unit which Don Jackson worked on which controlled two Weitek 1164/1165 chips.

Agrawal: right.

Garner: So this is not quite a single chip.

Agrawal: It's better than 17 chips.

Patterson: Floating-point units were separate then too.

Garner: Yeah. Floating-points were separate. Control chip. Only 20,000 gates in the integer unit. There's more gates in an I/O cell today, I mean in an I/O ring today.

Joy: There were only 15,000 gates, because 5,000 [gates] were the register.

Garner: Exactly. Right, were the register file. This is Andy's MMU using the same chips that the cache did. So 64K bytes of MMU.

Joy: But to be precise, there are 15,000 gross gates, but net routing you only get about 12,000 gates, right?

Garner: It was only about 12,000, exactly.

Joy: You're overstating it to say 20.

Garner: Absolutely.

Joy: It's is actually around 12 [used].

Garner: Yeah. And we have the picture of the gate array down there if you want to see...

Joy: That's a small number.

Garner: Very tiny number.

Joy: For a processor.

Garner: And in fact, I actually saw <pointing to my journal> where the kernel was only 300 kilobytes, by the way, the Sun kernel at that time. So everything was small. In fact, I found a place where Tom Lyons said “We don't need more than 27 bits of virtual address space. I can't imagine an application needing more than 27 bits. Okay, so here was the MMU, and here is the 64-kilobyte direct map cache. So one of the things that we debated endlessly over was whether loads should be one cycle or two cycles. For one-cycle loads, you need [both] an instruction cache and a data cache. So our load instructions were two cycle, our store instruction took three cycles. But what I realized is that cache misses far dominated whether it was a one-cycle load or a three or a two-cycle load. So we realized from a system perspective that cache performance reigned, we could do two-cycle loads and three-cycle stores and have this kind of-- so this <pointing at SUN-4/200 circuit board> is actually the whole size of the first SPARC implementation.

Agrawal: It was just one cache?

Garner: One cache. One combined cache, right here, one 64 kilobyte cache. So two-cycle load, three-cycle stores. But we knew the cache system effects would be more important. We had a write-back cache, you could plug lots of boards in. In parallel to this was a 68000 board, never co-existed in the system but we used the same chassis.

Joy: So can I say something?

Garner: Yes.

Joy: If you look at this board, I mean this is an antique now. So these guys designed a 12,000-gate thing here with what? One bug in it or something?

Agrawal: Yep.

Joy: Look at the number of wires on the rest of the board.

Garner: Including blue wires on the back.

Joy: Why is it that when you're designing this chip you can get so many things right, but when you design this board, which is about the same complexity as what's inside there, we make so many mistakes. I could never figure that out.

Patterson: Because it's repairable.

Garner: Well the main reason is because we're talking to an asynchronous system here, right?

Joy: You think that's what it is?

Garner: Yeah, yeah. This is very constrained.

Joy: I think it's more of a mindset.

Patterson: I think it's because you can repair it.

Joy: I think because you have to. If you believed you had to get this right, you would. You believed you had to get that right.

Garner: One of the problems is if you look at the total number of different chips that are involved. They're from different vendors.

Joy: There's a lot of wires inside of that--

Garner: This is one vendor, one chip, one single unit.

Agrawal: Bill, I agree with you.

Joy: Doesn't that surprise you?

Agrawal: It's the design philosophy also. You can make a change here, you cannot make a change there. Right?

Joy: Well I think you should pretend you can't make a change there.

Garner: Well this has all been sucked into one chip today.

Patterson: This is from a software guy, right?

Joy: I believe software should be finished too, because no one else believes it

Patterson: No one else will believe that.

Joy: No.

Agrawal: So Linley, I'm going to continue taking us further into the journey.

Gwennap: So after the gate array, where do we go?

Agrawal: After the gate array, Sun started, took that design and I think went to LSI Logic to have them do another version of that.

Garner: Twenty megahertz.

Agrawal: Yeah, we did a chip at Cypress. But Bill and I were thinking a little differently in that time. So we found a company in Portland that was designing so-called low-power ECL technology. So again, Bill and I took a trip down there.

Garner: Up there.

Agrawal: Up there. Met with them and convinced them that they should be working with us. So again, an ECL gate array implementing a SPARC processor. So we started that program, we staffed it, we actually had an entire system group working on a workstation version and a server version. And in that server version, we put so many testability features and things that we sought in mainframe that didn't appear in traditional servers until much later. Anyway, to make a long story short, Sun audited that company and decided that this company could not meet our volume requirements. So we decided to cancel that program. However, we did finish it. And the reason we had to finish and productize it because Cray decided to use that processor. And I don't know if it's known to many people, but years later, Sun decided to buy Cray. And that was probably one of Sun's best acquisitions. And that all happened because we productized the ECL processor and Cray was hooked on that.

Joy: Actually, yes, it was SGI bought Cray.

Patterson: SGI bought Cray.

Joy: SGI bought Cray, but they didn't want that product [SPARC] and they didn't want to lay the people off because they didn't want to take the write off.

Garner: It was the evolution of the SunDragon system bus too.

Joy: And then they gave us--

Agrawal: E10K.

Joy: They gave it to us for a dollar basically to avoid having to take the write off charges. And it turned into one of the biggest, E10K, it turned into the most profitable.

Garner: It was a follow on to the [Xerox and Sun jointly designed] SunDragon.

Joy: And they didn't want it because it wasn't the MIPS instruction set.

Garner: Right.

Agrawal: So after that, again, Bill and I we always were looking at different things. And we looked at gallium arsenide Technology and decided to sell the program with Vitesse Semiconductor.

Garner: And you put me in charge of it. Thank you Anant.

Patterson: Yeah, I visited that company [Vitesse].

Garner: You gave us consulting.

Patterson: That was an under-architected company. I visited, in one day I shaved three stages out of their pipeline. And I'm not that good. So that was venture capital at work.

Garner: To us, it was a silver bullet. It actually came later didn't it, really? In some sense, Viking was--

Agrawal: Yeah, gallium arsenide came after.

Garner: Came after Viking. Well.

Agrawal: Well, so gallium arsenide actually started a program called Tsunami which is an integrated version of a processor. It had I/O bus controller in it, SBus controller, the memory controller. And that became a huge success. In fact, when the first chip came out, it ran at 60 megahertz right? And at that time, our other processers were running at about 33, 35 megahertz. Anyway, so we started this gallium arsenide program and we were in the middle program and one day Scott calls me into his office and says, "Anant, our CMOS strategy isn't going very well for our mainline processor," and so it's Super SPARC. And he says, "I'd like you to start another group doing a mainline CMOS SPARC processor." I said, "You know, I'll have to get a new team in place and it's going to take a while." He says, "Sun can't wait. And if you don't pull this off, we won't have a future." And so he says, "Why don't you cancel, BRUTE, which was a gallium arsenide program and move them over to this new CMOS product?" So I said, "Yeah, I can do that on one condition, that you come and tell the team what you just told me. That if you guys don't pull it off, we won't have a future." Well, Scott of course, sent Wayne Rosing over to make that announcement. You remember that?

Garner: Yeah, Bill, Wayne and you.

Agrawal: But I think that was probably one of the best decisions Sun made, because UltraSPARC indeed took Sun to a very different level of playing field.

Garner: Yeah, a little background on that. Anant mentioned SuperSPARC. And we were so accustomed to doing very simple chips. The gate array or the LSI version, the Cypress version, similar pipelines, very simple. But in order to Sun to be perceived, [as a leader], as a real chip company, we needed a real single chip with all the caches on the chip, kind of an Intel 386-type architecture. We recruited those people from Intel. Well, we didn't recruit them from. They came to Sun is the right way to say it, I guess, and started the SuperSPARC program. And we were observing it, Bill I know you were observing it.

Joy: It was driving me crazy.

Garner: The goals were very ambitious, to design a single, what, one million transistor, fully integrated custom chip with a band new pipeline, not BiCMOS, not just CMOS, so some new ideas. A lot of latches were involved, a lot of CAD issues were involved. but the original goal was a one-year development, and that stretched and stretched and stretched. Now, luckily for Sun, there was a recession at about the same time. But in the face of that stretching schedule, what was Sun going to do? So Anant was doing the ECL implementation and we started the gallium arsenide project as a silver bullet, because we were falling behind now in the market because SuperSPARC was not coming out. And Anant was in the, I'll call it the penalty box. I mean, ECL's great, but you aren't going to get high volumes. The gallium arsenide chip proposal was a wafer. Proposal was 24-die, 16 in the ECL die. Gallium arsenide is going to be a great silver bullet, but you're not going to get high volumes. The gallium arsenide chip proposal was a [multi-chip module] Proposal with 24 gallium arsenide die plus 16 ECL die since you couldn't do memories in gallium arsenide, a whole bunch of gallium arsenide chips, a true silver bullet.

Joy: Really fast for the time.

Garner: For the time.

Joy: Very fast.

Garner: It was going to be really fast, like 200 megahertz was our target. Extremely, blazingly fast. And Dave would say, "What are you guys doing?"

Joy: It beat the mainframes at the time.

Garner: Well absolutely. But the point is, is that it was a wafer-scale type system.

Joy: Yeah, very aggressive.

Garner: PMOS gallium arsenide devices were slow. Vitesse wasn't really up to the manufacturing capability. We were behind in the market with SuperSPARC, we were falling behind in the charts that you guys Microprocessor Report guys published <Pointing to Gwennap>, there's this gap of two years where Cypress chips and then SuperSPARC comes out. That gap is very obvious in your charts. So in this environment, Anant basically says, "We need two parallel CMOS efforts and I want to take charge of the other CMOS effort." And so we started, we took the whole BRUTE team, we said, "Guys, it's been great. You've worked hard, but we need to do the right thing for Sun." And we started the UltraSPARC project, two years behind, basically, the market. How do you hire when you're two years behind?

Gwennap: So Anant was hoping we'd just skip right over SuperSPARC, right?

Garner: No, it made us realize that you can't have a program that slips in schedule. When I looked at the schedules for the gate array, we were basically on schedule. You know, we've looked at the LSI project when Andy did his chipset for the SPARCStation 1, he did five full ASICs on schedule, very small design teams. We had a very small design team. We were used to that. And all of a sudden to have to have a team of 30 or 40 people and then have the schedule slip and slip and slip, that was a shock to all of us.

Gwennap: But as you said, I mean, SuperSPARC was very complex for the time.

Garner: Extremely complex.

Agrawal: Yeah. And of course, in this other group, we took a different approach. I mean, BRUTE was a simpler design. ECL was a simple design. Tsunami, the MicroSPARC-I, was a simple design. Anyway, so I told you how UltraSPARC--

Joy: So UltraSPARC was the one Les and I worked on.

Garner: Yeah.

Agrawal: Yes, yes.

Joy: I did the basic pipeline and then Les took over.

Agrawal: Correct.

Joy: Because I remember doing it, flying back and forth to Colorado

Garner: Les Kohn you're referring to.

Joy: Yeah. Because I mean, it was very simple. Because it was all that stuff that was in SuperSPARC was unnecessary, in fact getting in the way. The [UltraSPARC] pipeline was much shorter.

Garner: By the way, I looked at my old notes and I--

Joy: It was four stage with one extra stage for floating-point

Garner: Les Kohn is one of the best architects I know in the valley. I actually looked in my notebook and I found we actually interviewed him in early 1984. He said that Sun would never make it with SPARC.

Joy: He wanted to a build a million-transistor [microprocessor].

Gwennap: And he went to National, right?

Garner: Intel he did the...

Joy: Yeah, i860.

Garner: Eight sixty.

Joy: Or 960. He worked on the 32032, didn't he?

Agrawal: Yes.

Joy: And then he worked on Ultra, we hired him finally to do UltraSPARC.

Garner: Yes.

Agrawal: Yes. So we got UltraSPARC-I started. And of course, as I mentioned earlier, naming was a big tradition at Sun. So we had the team choose a few names. And the finalists were B-O-B, Bob, which is Best of Breed.

Garner: Which I did not like.

Agrawal: And Spitfire, okay? And so I called an all-hands meeting to decide on the name.

Garner: I remember that. I was glowering; it looked like I was going to lose.

Agrawal: And people voted, and it was dead even, okay? And then there's a guy who used to run our CAD group called Nigel Ross. And he was always, always late to meetings. So he walks in and casts his final vote, Spitfire. And that was the internal name. And after it was chosen, I said, "Nigel, if you go to Germany to sell this product, you going to call it Spitfire?"

Garner: They loved it, it turned out in the end. Yeah.

Agrawal: Yeah. So that was probably one of the best executed programs at Sun. It was on time.

Garner: You want to talk about the organization?

Agrawal: Yeah. So I actually took a very different approach in managing Spitfire, because it had become a large team. We hired some great people and I should thank Intel for getting us some great talent.

Joy: They trained them for us.

Agrawal: What?

Joy: They trained them for us.

Agrawal: I should thank them. So I had a group of three directors. It was Robert, it was Hem Hingarh, and Nigel [Ross] who was running the CAD group. And then of course I had two distinguished engineers, Ron Melanson and Mike Splaine. So this was a team of--

Garner: And Les Kohn.

Agrawal: And Les Kohn, of course. A team of six people. But what I did was on a weekly basis I would flatten the organization. The status meeting was with, what do you call them, SF leads?

Garner: Mm-hm.

M1: SFADM.

Agrawal: SFADM. And this was a team of all the technical leaders who met with me and the management team and we discussed the status and how things are going. And I think that was very, very effective in getting the complexity, how to manage the complexity. This was I think the key. And of course, the results were stellar.

Garner: We maintained the small company spirit, basically.

Agrawal: Mm-hm.

Garner: There were about 300 people, 250 people total on UltraSPARC, 230?

Agrawal: No, it's about 150.

Garner: Hundred fifty.

Agrawal: Yeah. Anyway, there were some very, very interesting events that occurred during the Spitfire days. It's too long to go into that. Some very talented people. In fact, once I mandated towards the tape out a six days work week. And of course, immediately the chairperson comes running to me and saying, "You can't do that." I said, "Yes, I'm going to do it." And we did it. And I started doing SFADM on Saturdays.

Garner: And we had the meals, the catered meals.

Agrawal: Yeah. And at that time, my older daughter was four years old, so she would come to the office with me because she loved the meals. And when I had the meeting she would sit through the entire meeting. She would sit on a chair like this. And in those days, we used to pass our status on paper. So she would pick one up too. And as we were going through this, she would go through that as well, which was pretty interesting.

Joy: Should be working at Intel now.

Agrawal: And then I remember we were getting close to tape out. It was Thanksgiving and I said, "Team, you've got to work on Thanksgiving weekend." And sure enough, they did come in. And I talked to my HR person and we decided to do something very interesting. We went to every engineer's office with a menu saying, "What would you like to eat today?" We took orders for every engineer. Two of us drove in two cars, picked up all the food and delivered all of the meals to every engineer, customized. And that went a long--

Garner: So you might think that Anant drives people hard, but the reality is, this is the culture we had at Sun. From the early days, we worked around the clock. I mean, Bill would never go home, from what I could tell. Andy would never go home. I remember during the development of the gate array project working six months in a row every night, every day, Saturday through Sunday, never going home before three a.m. And I would be upset if I arrived home after the morning newspaper arrived. I mean, that was the culture we had back then. People were working around the clock with their schedules shifting. Their normal sleep time maybe is 25 hours. So people would start shifting through the day. One team would come in, one team would leave. You remember that. So this culture, a true startup culture, most of us were not married. You were odd in that you had a small family at the time. But most of us were single. And we truly worked ourselves to death. And you were just continuing that culture.

Agrawal: Yeah, even in Cheetah, which was a much larger program, I don't recall how many people we had on Cheetah, but it was much bigger than Spitfire. Again, I mandated a six days work week. And this one, I think it lasted for six months. And of course, we always had dinner served because I didn't expect people to leave before nine p.m.

Garner: That was normal.

Agrawal: Yeah.

Garner: That was normal.

Agrawal: And interesting statistics, Dave. We served 40,000 meals to tape out.

Patterson: Forty thousand?

Agrawal: Forty thousand dinners to tape out Cheetah.

Garner: Wow.

Agrawal: But you know, that's the reality of microprocessor designs. I mean, you can't just add more people and have people work a shorter amount of time. It's that core set of people, that core team, that just has to work like hell to get these designs done.

Joy: They're the most complicated things that people build I think. People don't really appreciate it. And that goes back to I think the comment about why there are wires on the rest of the board. There are not allowed to be wires in the chip. The level of the CAD software is just barely keeping up, right?

Garner: It's mind blowing.

Joy: There's a lot of Band-aids on that CAD software.

Garner: The analogy I like to make to chip design is imagine like in the city of San Jose, everybody has to get up, they have to walk out their door at exactly 7:01 a.m., and they have to arrive all at their office by exactly eight, right? You have all the freeways, which are the data paths. So the CAD software or the discipline required to make the timing. Functionality is the freebie, you know, the timing is the hard thing.

Agrawal: So we were talking about CAD software. I mean, as we all know, CAD is a very important element of getting these microprocessors done. So when we started Spitfire, there were a lot of complaints. We have this data center to do simulations and everything. A lot of complaints about uptime and resource allocation and licenses and all that. And that's when that team of six got together. And Nigel came up with a program that he would implement called DREAM Distributed REsource Allocation Manager. And to make a long story short, before DREAM was implemented, our average utilization was maybe 50 percent in the data center. And by the time DREAM was done, and I used to get a monthly report, if the utilization was less than 97 percent it was unacceptable, okay? And eventually, Sunil Joshi [and a person working for him, I forget the name. Jim Gately. They did a fantastic job of creating a world class data center. When I left, there were about the equivalent of 10,000 CPUs in it. And all of those 10,000 CPUs were utilized at 97-plus percent utilization, which is totally unheard of.

Gwennap: Well the nice thing about designing chips at Sun was you were able to design them using Sun processors as your data center, right?

Agrawal: Yeah.

Garner: Seemed like that's all our systems were good for is designing the next chip.

Agrawal: And it was pretty interesting software that they used, where the jobs would be submitted as if the data center was a single machine. So imagine 1,000 engineers, 200 different applications and maybe 3,000 machines, how do you manage all of that? And that's what DREAM did and that's how we got this high utilization. So the jobs were submitted, most of the simulations ran at night. If there was a failure, the simulation would be rerun with traces captured around the failure before and after. So when the designers came back, they had the complete data on what happened and how to go and debug it. And by the way, if the utilization was going down for any reason, there were mechanisms in place to start running random diagnostics on the model, okay? And of course, random diagnostics are very important. So we were constantly, constantly verifying the design, whether designers were ready with their diagnostics or not.

Garner: We also did emulation, so we were going to try to boot the OS before we taped out. We also had a functional simulator for UltraSPARC, where we, in other words, “What is the architectural definition of the design, right? So we had a fully functional simulator that we could actually compare against the gate-level simulation. So we actually had multiple-- when I say functional, I don't mean a gate-level model, I mean a totally independent high-level model, which was very critical to finding bugs for UltraSPARC.

Agrawal: So Robert is right, I think we maintained a startup culture even through very late in the '90s. We were taskmasters, but at the same time, we balanced that with human gestures, with whatever we could do to make the team feel comfortable. I'll give an example. One of the engineers, when I mandated working late and weekends, he came back and said, "Look, I live with my older parents and I do all the gardening on the weekend because they can't do it." So we hired a gardener for that person. One of the engineers was working very long hours and was having problems at home. So we bought him, for both of them, a ticket to Hawaii, booked a nice hotel for the weekend, and had all kinds of goodies. So these little things really balanced us driving the team really hard with, you know, we care for you. And that really worked. And in a company of about, at that time, we had about 40,000 people. Having a team work until at least nine o'clock and come on Saturdays, it's very hard to do.

Gwennap: So we've talked about several generations of the processor. Can we put the processor then into the context of what was going on in Sun. How important was SPARC to Sun's success, and how did it help Sun compete in the market?

Agrawal: Bill, you want to take that?

Joy: Clearly, what happened for us was the 68000s which we were using were not scaling, and SPARC put us into a different place. And especially then when through the strange circuitous path the E10000 came back from Cray to SGI back to Sun, suddenly we had the equivalent of the mainframes, the multiprocessor mainframe-class computing. And that really put Sun on the map for doing more corporate computing. Because what really was happening is the capabilities of the PCs were getting better. And so the business we first had, which was the desktop scientific business was being surpassed by people adopting Macs and PCs to do that. So the robust nature of our software and then this really incredibly cost-effective machine that could scale up to, what was it?

Patterson: Sixty-four.

Garner: Bill, remember, it started with the SunDragon project, right, at Sun?

Joy: Yeah.

Garner: Yeah.

Joy: But I mean, this was at a level of capability that was unaffordable to be built using mainframes. It was possible because SPARC was simple enough to do these high performance implementations, and we had a robust software platform which had been developed over a very long time. And those things came together and really gave us a second business. Because Sun really never got any substantial percentage of its revenue from software. So that was really our second hardware business and really, really important to the company. It wouldn’t have been possible without the SPARC architecture.

Garner: One thing I'd like to say about SPARC is a lot of people maybe got confused and thought it was the instruction set. But the reality with SPARC was it enabled us to design our own systems. We could design our own memory interfaces, our own multiprocessor interface function. So that gave us the power to design systems of our choice. We didn't have to wait for Intel to come out with the next microprocessor coherency bus. We were designing coherency busses for the SunDragon project, which by the way, I'd like to comment a little bit about. Bill had mentioned earlier that Xerox couldn't commercialize their work. Well, they had designed an early multiprocessor called Dragon at Xerox PARC. They went searching for a partner to work with. They came to Sun, chose Sun. So one day, all the people I used to work with at Xerox PARC were all of a sudden walking around my building.

Joy: We licensed it

Garner: And I had been working on the I/O section of that chip at Xerox. All of a sudden I was put in charge of the I/O group, so it was like, "Finish your work, Robert." And Xerox had decided to adopt SPARC, which was nice. But now they wanted a whole system.

Gwennap: And this was about when?

Garner: This was Pradeep Sindhu, this was in '89. So Pradeep Sindhu, who is now at Juniper. Who was the other co-architect?

M2: John Marc [Frailong].

Garner: John Marc [and Pradeep] were the two lead architects. And we developed a very interesting shared memory machine which became the SunDragon, which became the SPARCcenter-2000

Joy: We had many projects to do coherency busses inside. But I don't think-- the one that we got from Xerox was the one that really made the difference. The other ones mostly failed in some way.

Garner: Right. But the point is, we had the flexibility to do that. I mean, having an instruction set was fine. It was good we didn't tie anything to the operating system or the I/O, we could [preserve] the instruction set. But the key power was in designing these new multiprocessor busses, MBus, the SunDragon Bus, other busses. That was the key differentiator for us.

Joy: That was really key getting it working to the mature design that we could productize

Patterson: Yeah, but I think a couple things I wanted to say. I remember one of the days, I was just a consultant, but I came there in the lab the day McNealy visited. And he said, "You guys are going to take us from a half-billion dollar company-- SPARC's going to take us from a half-billion dollar to a billion dollar company" in the early-- this was the gate array. I remember at the announcement, Sun was not a shy company. They announced SPARC as a new standard and actually had the standard-- it's an IEEE standard. I think it's 854, right? And they pushed it through. Hennessy actually took a position that you shouldn't standardize instruction sets. He wrote a rebuttal. I'm not sure what the intellectual argument for not standardizing the instruction sets is, but this was typical Sun, right? We're announcing a new instruction set standard, anybody can implement it, it's a family of things. And then what Bill talked about the E10K, that was really the dot com era, right? The dot com thing to do was to get an E10K as your Oracle server and then you would do the front ends with PCs. But Sun was just the machine you bought and that was many doublings of Sun price, right? The bastards at Sun showed me what my stock, 1984 stock would have been worth if I had kept it, 1984 stock that I got as a consultant. If I had kept it up until the dot com era. It was a very disheartening number to see.

Garner: Well like the bicycle you bought is a million dollar bicycle, if you think of it.

Patterson: Sun stock was flat for a decade, as far as I recall. And somewhere in that decade, I finally gave up and sold it. Had I kept it another six years, that would have--

Garner: Our stock went down at our IPO.

Patterson: Yeah, that was the other thing at Sun, every time Sun did something technically great, the stock went down. I never did understand that.

Garner: We were correlated with the same [stock] motions of Fruit of the Loom, is what we learned.

Joy: Yeah, it's true. It's true.

Gwennap: So can we talk about some of the partners that were involved with SPARC, some of the other companies that were adopting SPARC at the time? Cypress, LSI, TI were all involved.

Agrawal: Yeah, absolutely.

Garner: TI, yeah.

Agrawal: Yeah, we did work with LSI, Cypress. Then we decided to move to Fujitsu. But we started with Fujitsu then we did another program called Swift with Fujitsu. Fujitsu always had been a great partner, we always stretched the limit of the technology and they always delivered, always on time. We also did a program with NEC, but that was just one program, again, great success. But we spent most of the time with TI. And in most of the instances, again, we stressed the technology. So towards the end I recall we had a group of about 23 people that were in the technology group. And this group would always help define TI's next node in terms of what the metal pitch should be, in terms of what the SRAM cell should be, right? And in terms of what the i-drive of the transistor should be. And of course, TI was very happy with it because they had somebody driving their technology and then they could use that technology as it came down the cost curve for their other logic products that they were doing. But yeah, Sun actually defined the technology for TI.

Garner: I think Ron Melanson had a huge role I that.

Agrawal: Yeah, Ron Melanson was the key guy driving that.

Garner: Whenever we were late, and I would just say, "Ron, we need a faster cycle time." And he would go to TI and beat the crap out of [them] with a new process.

Gwennap: What was the first project where you started working with TI? And how did you get involved with TI?

Agrawal: So Tsunami, or MicroSPARC-I was the first program I managed, and that was done with TI. Of course, by that time, SuperSPARC was using TI as well, except they were using Bi-CMOS process. And their fab I think was in Japan somewhere. After doing ECL and flirting with gallium arsenide, I said "CMOS is it, that's the mainstream." And I decided to use CMOS for the MicroSPARC-I, MicroSPARC-I. And since then, we stayed with it. And as we needed more performance, we'd go work with TI in enhancing the process. We were also the drivers for making revisions to the transistors, so it was not common to have dot A, dot B, dot C transistors, right? While TI stayed with dot A, we went to more aggressive transistors for follow on products.

Gwennap: And so when you were working on some of these projects, I mean, were their chip yield issues or functional issues to deal with?

Agrawal: Of course. Sun fortunately had a very sophisticated, I wouldn't call it lab, I'd call it a test floor, where we had some state-of-the-art equipment to test and debug and look at the yield issues and all that. Very good test group to support that, and of course, we got a lot of support from TI. TI had I would say a couple of hundred people supporting us in our efforts. And yes, there were always yield issues, yield related to some violation in design rules or temperature or process variations. But again, this group at Sun working with TI and the equipment pad, we were able to go fix that. Sun was big time into also something called risk buys, where when we taped out a chip, we would actually start a whole bunch of wafers, costing millions of dollars. And that was done so if everything went right, they could build hundreds or thousands of machines and run them through the system-level tests. And I recall when we taped out UltraSPARC III, the risk buy was worth $56 million. And so--

Gwennap: Fifty-six.

Agrawal: Yes. So I went to Scott [McNealy]

Patterson: It's a different kind of risk.

Agrawal: Yeah. So of course Scott had to approve it, so I go to his office. And he says, "This is too low." He says, "If you can accelerate the deployment of the machines in the market by four months, okay, 56 million is too low because the amount of revenue we can-- the upside potential on the revenue generation is much higher." And at that point, I said, "No Scott, I can't take a high risk. And you know what he said? He said, "You have never been wrong. So go increase it." Fortunately, he didn't. But that's the mentality Sun had.

Joy: You didn't increase it?

Agrawal: Yeah. I told him I would but I didn't.

Joy: <laughs>

Agrawal: But that's the mentality Sun had is really, as Robert said, it was a system design and it was really systems that were bringing the revenue in. So how do you optimize not just the design but also manufacturing and putting things together to optimize the delivery of the system?

Joy: Yeah, I think the big issue is, the first program had a small budget and it had a reasonable revenue impact. So the ratio worked out okay. I think the issue really was just as the business went from the desktop to being servers, the number of chips we could build; we were building for those was minor. So you take the cost of these programs and start dividing by the number of chips. And the R and D cost got to be-- and out of the model that I thought we were going to have. Because I thought we were going to be doing more desktop computing. And so I think that's the difficulty with an architecture that doesn't have a presence at the low end. If you took something like ARM today that's shipping such large volume, they don't need very much revenue per chip, and the whole thing squares up. To square up an architecture today for just a high end, it's very difficult to make the amount of R and D you have to do. And you divide that by the number of chips you're going to ship and it's a brutal ratio. So the value of a better architecture-- I'd like to see some better architecture for cloud computing, but try to get the economics to work, it's not easy. That's why they're built with chips that aren't customized and system architectures that are largely not customized. It's an economic problem.

Gwennap: All right. So the cost of doing a new architecture has just increased phenomenally since then.

Joy: Yeah, I think the-- maybe there can be a new way of doing an architecture for the cloud that doesn't cost $100 million. These programs, what was the one on the East Coast that was cancelled? Remember that one? I don't even remember the name of it, but--

Garner: Well, I don't know. But one thing I want to--

Joy: We had some really big chip development programs that, they got really expensive. And you're only going to ship 10,000 or something, 100,000.

Agrawal: Millennium [program].

Joy: Millennium, yeah, that was a couple hundred million dollars. And it failed, right?

Garner: Yeah. We also had the MAJC [Microprocessor Architecture for Java Computing] processor.

Joy: Yeah.

Garner: Which cost us $100 million, I think maybe, which was an attempt to have a new instruction set, break away from SPARC. Somewhat focused perhaps on a graphics backend. But Java, a target engine for Just In Time compilation for Java. We started the project builder. We had an offsite at [Asilomar] Monterey. We walked on the beach and we said, "What can we do? Okay, SPARC's had a nice run, what's our next play in terms of this kind of thing, hardware, systems."

Gwennap: When was this?

Garner: That offsite was in--

Joy: Ninety-six or '95.

Garner: Ninety-six. No, earlier, yeah, '95, '94. So we took a walk at the beach there in Monterey.

Joy: This was really at the beginning of multi-core.

Garner: Multi-core. Super--

Joy: And it was really simple.

Garner: Multi-issue.

Joy: These processors made SPARC look complicated.

Garner: Right.

Joy: They were really, really simple.

Gwennap: So why didn't that go anywhere?

Garner: I actually ran the program. And we actually had a very large team, about 100 people. The problem was, probably, we targeted it too much to graphics and we couldn't find customers. We would go to Japan to Nintendo and Nintendo would say, Java? Graphics? Please. And there was just a disconnect between-- so we chose the wrong marketplace initially. Now these ideas, Les Kohn took these ideas later and got Niagara.

Joy: We took it to Sony and Sony wanted it. But then Sony went to IBM and IBM built it for them.

Garner: That just as well.

Joy: The one that's in the PlayStation.

Garner: The Cell [processor].

Joy: That's the architecture that--

Garner: Well no, this [MAJC] was earlier than that [Cell].

Joy: This is the architecture that we-- basically what IBM did for them, they basically took our block diagram and asked IBM to build it for them.

Garner: I don't know. Oh wow. I didn't know that. That’s amazing.

Joy: You know, the idea worked, we just weren't chosen as the supplier, because...

Garner: I didn't know we had gone that far with Sony. That's good.

Joy: It was frustrating.

Garner: It was frustrating. It was a disappointment. The project had a great team and all the ideas were correct, i.e., that we would rely on Just-In-Time compilation, so we could be even more flexible in the hardware.

Joy: But this would have been a higher volume alternative, like what nVidia did.

Garner: Mm-hm.

Joy: These were basically the precursors to those kinds of processors, but done in a more architectural way. So not as funky, right?

Garner: Michael Deering was the graphics architect. And one thing about Michael is he dreams really big. And I think some of his ideas for graphics were beyond what the industry was capable of doing. So we were a bit ahead of the industry, unfortunately.

Joy: But these were simple. I mean, if you look at your board, you see separate caches, separate floating points. These were really simple. Very simple, what, 32 registers. Floating point, there was no distinction between the floating point and the integer register. Just really, really simple.

Garner: Yeah, we had a good design for MAJC.

Joy: Yeah.

Garner: What I'm talking about is Michael Deering's graphics backend rendering engines, which were triangle oriented

Joy: The graphics, yeah.

Garner: The triangle. Everything was going to be a triangle. Everything would render to a single triangle.

Joy: But that was an attempt to take this thing that Dave had done at Berkeley where he had a slide with a chip that could be done real quickly. Then we did it again with SPARC. MAJC was an attempt to say once again, now in the context of really high graphics performance we can build something with a small team that, since it doesn't cost very much, we can get a beachhead in the marketplace with a relatively modest number of units to start. But in the end, the industry went to nVidia and ATI-type graphics solutions in volume, rather than a programmable thing.

Garner: Yeah, we were barking up the wrong tree.

Joy: Although that programmable architecture was really only adopted in the IBM chips for Sony.

Garner: Now another tree we barked up, I'll never forget Bill, at one of Scott's meetings, you were sitting next to me and you asked me how big was the [Xerox] Mesa instruction set implemented in silicon. I said, oh, about two millimeters by two millimeters. Bill raises his hand and goes, "We're going to do picoJava, we're going to do Java in silicon."

Joy: Right.

Garner: And that was our next foray into different alternatives to SPARC.

Joy: It's just a question of how much do these programs cost, is there an uncertain market, and can you make the ratio of the engineering investment to the benefit work? And it doesn't seem to be possible very often.

Garner: And we did implement picoJava. There are now many startups out there still implementing Java accelerators. So that idea still has merit. But it's hard to produce revenue from picoJava chips.

Joy: Right.

Agrawal: But if I look at the size of teams over the years, it has grown exponentially. The complexity of these chips, as Bill said, it's become very, very expensive.

Gwennap: Yeah, things have changed a lot.

Garner: So we tried with MAJC.

Joy: There's no one left doing it.

Garner: Yeah. And Niagara I think was a step in the right direction.

Patterson: Yeah, I was going to say.

Garner: Les Kohn’s.

Patterson: Niagara was a different bet. For me as longtime consultant, Sun just kept thinking, "Well, if Intel can do it, we should be able to do it." And it just was obviously not Sun's strength. And so as I was consulting for Fowler, John Fowler, who's still here, right? He has survived, and I don't know, he's got some big title. But I said, "This doesn't make sense. If Sun's a server company what it should do is do simple cores and a lot of them on a die, not try and compete with Intel with a small number of heavyweight cores, right? This matches Sun's skill set." And I kind of made the philosophical argument. I had Fowler and the guy he used to work for, Jonathan Schwartz, was in the acquisitions part of Sun, which I wanted to go consult there just because students would start companies and wanted to see what it was like from a venture kind of investment. And so when he heard the pitch from the Niagara guys, which that's what that was. A lot of simple cores, let's do a lot of threads, kind of a bandwidth-oriented chip, rather than a latency-oriented chip. And it was SPARC, that he saw the value of it, acquired the company [Afara]. I think it was a pretty successful product for Sun. And I remember the guys from IBM, it was like, "Well that's cheating. Anybody could do that. Anybody can do eight simple cores on one die. Real people build heavyweight cores, they don't do these wimpy cores." That was obviously again a smallish team. I think that Niagara was pretty successful, that product line.

Joy: Yeah. Well, what happened was that we had that offsite in Monterey, and I was pushing for multiple cores.

Agrawal: Yep.

Joy: And it was clear that the SPARC people in Sun weren't ready.

Garner: We weren't doing that yet.

Joy: So roughly at that time, then Mark Tremblay and I decided, well, we'll just stay focused on floating point and graphics [i.e. MAJC] because that's something that SPARC isn't really as strong as what we can imagine. And Les simultaneously went and said, "I'll do one that's focused on integer and basically transaction processing or commercial." Niagara in fact, they had n-cores and one floating point unit.

Patterson: Right.

Joy: They had really weak floating point and we had really strong floating point. We didn't have instruction set compatibility. And that was kind of deliberate, because if it had been a SPARC, the antibodies in the organization, because the organization didn't want to do a multi-core SPARC.

Patterson: Niagara was a startup though, so.

Joy: So by making it have a different structure set.

Gwennap: It was a startup, what was the name of the startup?

Joy: It was outside, but inside I said, "If we just do a graphics-oriented instruction set, then maybe people will leave us alone." Because we could have always taken the architectural block diagram and just changed the instruction set. That wouldn't have been the hard part. It was the concept of going to four cores on a very small die with very high throughput that was necessary. That's why these graphics accelerators have so much parallelism in them.

Garner: Now Les's, I can't remember the name of the start up, maybe you do Dave, that Les was at, and Sheraton.

Patterson: Yeah, it was an African name because it was Kunle's company.

Garner: Kunle Olukotun's company, right?

Patterson: Yeah, Afara

Garner: Afara. So Afara's perspective was that now that the last mile would be solved and that homes could be receiving video over IP that we needed a processor for that market. And that's why it's so much throughput performance oriented. And with that system vision, it was nice they chose SPARC. But that was the system vision. Dave, I remembered, you also did a lot of consulting, telling people maybe they should think of SPARC. I recall you were at Thinking Machines.

Patterson: Oh yeah.

Garner: And made suggestions that they should switch from a serial, bit serial processor to something more.

Patterson: Well, Thinking Machines was a very innovative company. It shared a lot of similarities with Google. They had their own chef. They had an eccentric hiring practice where three people would do it. And Danny Hillis, who many of us know, is a real inventor, right? That's the way to think of him. There was this thing in the Boston Airport of a set of Tinker Toys that could play tic tac toe. Danny did that. So he had this idea of the successor to the Connection Machine. The original Connection Machine, which was a bit serial SIMD machine with thousands of processors, he wanted to do something much more aggressive. And they had their own design. And he asked me to come in and evaluate SPARC in that design. I said, "Well, I don't want to waste my time doing this." "No really, take a look at it." And so they came up with a SPARC-based design. And I kind of, I think basically I showed how if you write in assembly language you could get good floating-point performance out of SPARC, basically I think I was doing software pipelining basically is what we'd say today. The performance was there, so then they negotiated the deal with Sun at some time and announced this, I don't know, a 1,000-processor SPARC-based system that I think was a big surprise to everybody at Sun. It was like Google, they were extremely secretive. Bill is, by the way, the opposite. Always when I'm consulting for a firm, they're convinced that the academic's going to spill the beans. And we're under NDA and they always look at me. Everybody signed the same agreement. And so during the SPARC days when people would ask what's going on with Berkeley and RISC and Sun, "Talk to Bill." Bill would tell anybody everything that was going on, answer any question. As far as I know, commit. Hillis is absolutely the opposite. He was absolutely convinced if I give an interview and give any hint of any direction in the Thinking Machine he would call me up and bawl me out.

Joy: I always figured if I told people what I was doing, they would tell me what I was doing wrong.

Patterson: So it was no surprise at all when SPARC was announced, right? Absolutely nobody was surprised?

Garner: Oh, that was a total shock. Well, some people knew. And by the way, I think Bill saw what had happened to the old model where, for instance, at Xerox, we kept everything under lock and key, golden, you know, the instruction set was known only to the priests. We went to Intel and said, "Could you make a Mesa processor?" And they said, "What's your volume?" And they laughed. So Bill knew the downside of a closed system. I hadn't told, didn't tell this story earlier, but when Bill was focused on floating point and looking at my notes I also see that the systems guys were saying, "We don't have enough MIPS in the 68010, and we don't think there will be enough in the 68020 to be a decent server. We can't do NFS serving." We were actually missing disk rotations. This was a 3,600 rpm disk, and they said we were missing disk rotations, saying "We need more MIPS." So that was one aspect. Bill's aspect was the floating point. Let me just finish. So I said, "Bill, we're going to do RISC, it's not going to be a co-processor in the 68020." We had some ideas in the early days of Sun that-- we were a very successful 68000-based company. Why would we take this exponential ramp, 10 million, 50 million, 100 million, 200 million dollars, and destroy it all with a RISC processor? We had a great ramp. So there were ideas of RISC and 68000 together talking with the same cache, different boards in the same system. So Bill said, "Look, if you're going to do RISC as the main processor, you have to standardize it." And I said, "I don't care. You go standardize it. That's your job." And boy, did Bill do that. He talked to everybody about SPARC. I mean, constantly he was telling everybody about it.

Gwennap: How easy was it to convince some of these other companies to use SPARC?

Joy: Not easy. I think everybody we talked to about it then decided to do their own [RISC instruction set]. I mean, that's one of the problems when things are easy, I think.

Patterson: Startups would do it, right? Startups would embrace SPARC.

Joy: Yeah. But in general, Motorola started their own effort [RISC, the 88000] after we talked to them. MIPS just couldn’t come to an agreement with us. It was not easy.

Garner: I'll never forget Mike, Ross--

Joy: Roger.

Garner: Roger Ross. He had presented to us a 68881 floating point chip, which by the way was 80 bit. We had giant quandaries about 80 bit. David Hough and all those people thought 80 bit was cool. We thought, "Hey, no, performance is more important."

Patterson: David Hough was the floating-point expert.

Garner: Floating point expert, right. So we had thoughts of do we need to have a 68881 on the side of SPARC, just to keep 80 bit? Because we didn't want to go 80 bit, those thoughts all died. But anyways, Roger Ross came later and he couldn't believe that we had done a RISC chip and it was running. He said, "That's impossible. You can't have it ready by now." Because they needed full teams and they were doing full custom. We took him into the lab--

Patterson: You mean at Motorola.

Garner: At Motorola. He couldn't believe it was possible. I remember the day. We brought him into the lab, we showed him this board running and his mouth was just-- <laughs> just dropped. I mean, they couldn't believe that we had actually gotten it done.

Gwennap: And Roger eventually became a convert, right?

Agrawal: Yeah.

Joy: Yeah, that's true. He did Ross Technologies.

Agrawal: Ross Tech, yeah.

Gwennap: Right. And there was an effort to get Apple to adopt SPARC?

Joy: Yeah. I mean, Apple did a bake off, was it in the [CEO Michael]Spindler days, between SPARC and...

Agrawal: Power PC.

Joy: Power PC. And ultimately, Motorola switched their architecture to Power PC in order to get Apple as a customer, and IBM. And so I think if Motorola hadn't switched we would have won the business. But once Motorola and IBM were behind it, the business context was too strong. We would have won the technical evaluation. And I think we were very far along. But that was a real surprise to me when that happened, and it really knocked us out.

Gwennap: To start wrapping up a little bit. So kind of looking back in general, how important was SPARC to Sun's overall success in the market?

Joy: Yeah, I think without-- the workstation business largely ended. I mean, HP bought Apollo and most of the other workstation companies disappeared because the software and hardware on PCs got better. So without SPARC, we wouldn't have been able to be a meaningful server company. And so we would have probably faded by the mid-'90s.

Patterson: Much like SGI did, I guess.

Joy: SGI did.

Agrawal: And mid-'90s I would say was a perfect storm where the dot com was coming up, we had technologies like E10K and UltraSPARC came around. And it was very, very--

Patterson: You had a scalable operating system. You had an operating system that ran just fine with 64 processors, which not many people had.

Agrawal: And I mean that's when Sun went to what, $15, $16 billion revenue.

Joy: Right.

Agrawal: That was E10 and then Niagara. And that was SPARC.

Joy: It was software like Oracle running on that machine that really gave it the-- as Dave mentioned before. And that's perhaps why Sun and Oracle got together in the end, because it really, it was the marriage of that powerful database with that server architecture. Now with the Niagara kind of architecture it's still an industry-leading architecture and a really powerful one and a very mature platform for that kind of database. People thought it was going to be client-server computing. But in fact, it was relational database that won, not client-server as a model. We had the machines that could run the relational database. We were trying to do client-server, but that was really kind of a disaster for the customers. So in the end of the day, SPARC and relational database on these large machines is what defined our business.

Patterson: I'd say on the other side of it, in terms of the RISC ideas, Sun made them successful. IBM, done that early on, they brought out the IBM, which many people forget, the RTPC. That was the commercial implementation of these great, you know, at [IBMs] Almaden [research lab]. And it failed in the marketplace. Now one of the reasons it failed is that like MIPS did, they did value-based pricing. Here is this single-chip computer they said, but look how many MIPS it is and how much IBM mainframe MIPS cost, and do they can charge this much. Nobody bought it. This was before SPARC came out. And I think there was at least one time when the SPARC project almost was killed inside of Sun. It's hard for me to believe we were on schedule. I don't know whose schedule you said we were on.

Joy: I wrote in pencil, not in ink.

Garner: All his notes were in pencil, that's right.

Patterson: It came close to stopping in the middle in part because look, IBM tried to make-- you can't make money at RISC, IBM tried it, even IBM failed.

Garner: We were grateful that IBM had the ROMP processor that they announced in '84, which was extremely slow.

Patterson: So it almost didn't make it, but then--

Joy: This machine was about the same price as a 68040 or whatever.

Patterson: Right.

Joy: So it was two or three times as fast but it was about the same price.

Patterson: Yeah. And then when it came out, this product, this board came out, and the salesmen at the launching event, it was shooting fish in the barrel is what they said. Here is this thing. It was much cheaper than what DEC built. It was two or three times the performance.

Garner: We had ported the software.

Patterson: What?

Garner: Application software, it all been ported.

Patterson: Everything worked. And this was part of-- what was the difference between MIPS and Sun at that time is Sun had very forgiving C compilers. Any kind of C it would take and produce code. MIPS was straightjacket, right? If it violated the C standard, then they couldn’t get the optimization and then it wouldn't run. So MIPS basically outperformed SPARC in the early years by 15 or 20 percent, but a lot more software sensitive. So Sun demonstrated the value of RISC. And then the ARM guys followed in that wake with that simple business model. And then the ARMs, there's an astounding number of ARMs made today.

Joy: The thing is, at the time, most of the customers who really needed the performance had their own code, their own applications-- it wasn't third-party software, just a piece of software that they'd written and they were already used to, from the VAX to the 68000, they were used to UNIX running on multiple machines. And so they had something that was taking too long, they could recompile one program and it ran faster, that was enough reason to buy a machine. That's not true today when you have most applications third party.

Patterson: It's binary.

Joy: The binary compatibility is much more important now than it was then.

Garner: Another thing I want to mention on that vein is we had the network use the computer, right? And this idea that on one network you could access data. So here we are coming out with SPARC. Well, okay, SPARC's the greatest, but you could have a VAX, you could have something else on your network and everybody could still talk to each other, right?

Joy: Right.

Garner: So the simultaneous announcement of SPARC and NFS and its ability to share data in a mixed network made people feel more comfortable. Remember, our phrase at Sun was that we were open and that if any time Sun OS could run on a different platform. This whole idea that you could move around platforms, it helped SPARC get adopted very rapidly.

Gwennap: In terms of kind of moving behind Sun but looking at SPARC's impact on the industry as a whole, it sounds like it really started to bring RISC to the forefront of the architecture wars. How much influence do you think SPARC had on that.

Patterson: Well, I think DEC was forced to adopt MIPS after-- with Sun's success against the DEC customers. Because at the time, people thought-- early on when Sun started, they thought the opponent was Apollo. But all their management talked about DEC. For some reason, you guys were sure Apollo was going to go away. But it was all DEC's the opponent. "We're going to take out DEC." And then once they came in the marketplace running a more popular operating system much cheaper, it really forced DEC's hand. The architects of the VAX, to their credit, wrote papers showing that RISC was better than VAX, and then if they didn't have the binary--

Joy: <inaudible> --when they did Alpha, right?

Patterson: Well, they first shipped MIPS.

Garner: First shipped MIPS. And UNIX was snake oil, remember.

Patterson: And then, to [John] Hennessy I think still wakes him up in the middle of the night. Because MIPS and Alpha are not gigantic differences, right? They could have easily done MIPS. And for business reasons, it's the same thing Bill said. They wanted to have their own instruction set. But they shipped MIPS because they had to have something for a while. And they abandoned MIPS because of NIH, they wanted to have their own, very similar instruction set. And so yeah, I think it showed that at least for that era, until Moore’s law kept going and the X86 basically adopted the RISC ideas with internal chip translation, to their credit, it changed the way people did micro-architecture and architecture.

Joy: But Alpha was really impressive with the clock rate.

Patterson: Yeah.

Joy: Really, incredible implementations, just lots of--

Garner: We were blown away by them, yeah.

Joy: Lots of little black magic at the circuit level.

Patterson: Well, they just had the best circuit design team out there.

Joy: Yeah, circuit design.

Garner: The best clock tree networks on the planet.

Joy: Fifty percent, at least half the power was in the clock tree.

Patterson: Well, and they were wasting their time--

Joy: Eighty percent of the power in the clock tree, right?

Patterson: Yeah. They were wasting their time doing VAXs, and then once they were released, then--

Joy: They just had a bigger ice cube than we did.

Patterson: No, once they were released from the VAX instruction set, you know, that group is still a great design group.

Garner: Do you think UNIX as snake oil had an effect on having them look at alternatives? I mean, they were so wedded into their OSes and VAX as one unit, and UNIX was this snake oil thing. Did that have an effect on DEC?

Patterson: Well, I mean, people were voting with their feet, right? The UNIX's were extremely popular. And of course, that's what made RISC possible, right? It's portable operating systems. The thing that Ken Thompson and Dennis Richie did, and Bill worked on, the fact that you could create a new instruction set and you wouldn't have to write an operating system in assembly language made this all possible and popular with many customers.

Gwennap: Looking back, if you could go back to the early days of the development and change anything, is there anything differently that you would do now in retrospect?

Patterson: I would have thought, because everyone's always asking me this question, "What about register windows?" I would have thought the answer would have been not do register windows. But it sounds like you think the compiler effort would have died?

Joy: Yes.

Garner: Yep.

Patterson: If we hadn't done register windows?

Garner: I'm convinced of that.

Patterson: I wish I had known that for the last 20 years. I remember I said, "It wasn't my decision. I asked this question, it was a very concrete question." It was good advice at the time. You can do register windows if you can't do graph coloring. But they subsequently did graph coloring. But I didn't know when that happened.

Joy: I think the SuperSPARC was a big disaster for us.

Patterson: Microprocessor companies have disasters.

Joy: We somehow hired a group which didn't really-- they were implementing the instruction set but in a half-hearted way.

Patterson: Look at Intel.

Joy: We could have just taken, if we could just take almost the original pipeline that we had and just kept speeding it up, even without redesigning the pipeline, we would have been-- I think we made a lot of mistakes in that way. Gallium arsenide and some things were a diversion.

Garner: It was too big of a leap. It was too big of a step, that's how, you could say it differently.

Joy: We could have simply focused on really on the main line, just implementing the same thing, just doing what Alpha did. Just squeeze it out at circuit design, not at architecture. We were I think too interested in other-- enamored of different kinds of semiconductor technologies and with architecture, we would have focused more on implementation.

Garner: Well, we thought to be successful it had to be like an Intel 386. In other words, it could be in any PC shipping in the world, as a single chip. Any person in the neighborhood could design something. That was the mistake.

Patterson: But Fred Brooks spoke about this, right? The second system effect, right?

Garner: Yes.

Patterson: You had second system effect. Besides this project, besides SuperSPARC, there was Millennium, which fortunately for Sun never actually saw the light of day. But Intel, pretty successful company, it has disasters, right? They canceled a project that was-- they were bragging about 40-stage pipelines at Hot Chips. They justified a 50-stage pipeline, and oh well by the way, it was hotter than a nuclear reactor, right? They cancelled this. They had the 432, they had the Itanium, billions of dollars invested in things that didn't return a lot of money.

Garner: They're still shipping that though, Dave.

Patterson: It's hard to avoid these things. I think the funny thing to me as a consultant is, it was obvious that these were disasters. I freely spoke that Millennium was a disaster, but somehow that-- you think they're capitalists and trying to make money, but that wasn't sufficient. I think Sun is kind of, they like the people, all these people have been trying hard, they just weren't willing to go kill these projects.

Joy: I tried to fight the complexity that was creeping into the processors, like out-of-order execution, I never thought it was necessary. In the end, people went to multi-cores or multi-threaded and not to these pipelines that were-- because the farther you look ahead in what you're going to execute, the less likely you're going to be able to do it and the less likely it's going to matter, because it's probably going to get killed by a mis-predicted branch.

Garner: Remember the Russians we were working with? [Moscow Center of SPARC Technology, Boris Babaian, early 1990s]

Joy: And so just keeping things in order, relatively simple, and going to multi-core was I think the path that was the right path. We didn't quite get our-- we had one wheel in the ditch. We had one wheel on the road and one wheel in the ditch. We were kind of on that path.

Garner: We had two multi-cores.

Joy: We went away from it a little too much. And it was very expensive, because if we'd have stayed on the main path, the machines would have been much more competitive. And that's kind of what happened when Anant was talking about and called into Scott's office. Scott realized we'd gone into the ditch and we weren't executing. We should have had leapfrogging teams working on the main path.

Garner: But one thing we did, we did continue to keep some simplistic thinking. As you recall, we had the group we were working with in Russia, who had done the largely out of order machine.

Agrawal: VLIW.

Garner: The VLIW machine, which we couldn't-- again the branch behavior, prefetching too much, we just couldn't see the performance gains. That was one of the hardest moments of my life to tell the Russian team we were not interested in their NArch architecture. -- And remember we bought Intergraph['s Clipper design group, headed by Howard Sachs]?

Joy: But it would be interesting to look at the graph of the length of the pipeline versus time. And hopefully it didn't go exponential. I mean, hopefully we kept it relatively flat

Garner: We did.

Joy: We didn't ever do really deep pipelines, did we?

Garner: We did [try to keep the pipelines simple]. But we said no to [Babaian's group]. We said no to [the Clipper group]. So we tried to keep that focus on simple pipelines

Joy: I hope we didn't…

Garner: And we had the Niagra project, which was simple pipelines.

Joy: Yeah, 25 to 30 is terrifying. Twelve is terrifying.

Garner: He's just a software guy, but he still likes to design pipelines.

Joy: Any double-digit number at the pipeline stage is terrifying. I'm not counting retirement of floating point instructions, I'm talking about the rest of it. Not divide, that’s … who cares.

Patterson: Go ahead

Gwennap: I'm waiting for Bill to finish.

Patterson: That could be a long wait.

Gwennap: That could be a long time.

Patterson: Long wait.

Joy: I'm just thinking about the divide instruction. I don’t like-- well, whatever.

Gwennap: Tag, you're it.

Anant: We did an synchronous version for SPARC.

Joy: Yeah, an synchronous divide.

Gwennap: I think today SPARC has become much more than just Sun and now Oracle. With SPARC International, the open standard, SPARC licensing. What can we say about where SPARC is today, what is it being used for, some things maybe that people aren't generally thinking of off the top of their head?

Garner: Fujitsu for one is continuing SPARC in a big way in their supercomputer lines.

Agrawal: So probably Dave, you have the latest information. Because we are all--

Patterson: One interesting vehicle, it's in most of the European satellites. And the fact that it's the only standard instruction set, the only instruction set with a standard and a verification suite makes it very attractive for kind of small groups. So there was a person hired to do an FPGA implementation so it could be radiation hardened for satellites. He went with SPARC because of that advantage. So it's got this kind of funny following there. Sun and Fujitsu continue to use it. There's talk about it in this open source software movement, open source hardware movement where you would have Verilog and gate arrays and FPGA implementation. So it's relatively popular there. I don't think it's in any embedded device that I know of. It never got that following that MIPS did. And when we were talking on the phone about it, I think the fact that we had a two-cycle delay for loads and three-cycle delay for stores in that first gate array implementation affected people's perception of SPARC. Whereas, the MIPS guys didn't do that.

Garner: And we had a very large register file.

Patterson: That would have been bad for embedded-- what?

Garner: We had a very large register file too, I think, for embedded people.

Patterson: Well, but we didn't have a split instruction and data cache.

Garner: True.

Patterson: So a lot of byte loads and stores and stuff like that were hit.

Garner: But I was just also saying I think the embedded people were turned off by the size of our register file.

Patterson: Okay.

Gwennap: I think yeah, even today, there are licensable SPARC cores now. So I think we're seeing, because of the opening of the instruction set, that it's turning up in a lot of places.

Joy: It took a long time. The discount rate makes it beyond 25 years or whatever. That anybody's still using it after 25 years is astonishing.

Patterson: Yeah. Successful instruction sets last a long time.

Garner: Bill, that's your vision of making it open.

Patterson: Yeah.

Joy: Well, but I mean, if you were to sit down and do something today, wouldn't you do--

Patterson: You'd do something new.

Joy: You'd do something different.

Patterson: We've done it, Berkeley's got one we call RISC 5. So it was RISC 1 and 2, SOAR is RISC 3, SPUR was RISC 4, so we just did RISC 5. So no register windows, no delayed branch. It's got the option of 16-bit instructions. And we did it so that we could use it for education and as a base point for technologies going forward. And I heard there was a project inside of Sun Labs like that, of a clean instruction set as kind of a useful research vehicle. Yeah, you wouldn't use-- all these RISC instruction sets have major flaws in them that you wouldn't do if you were starting it fresh today.

Gwennap: Robert, any last stories that we missed or last words on SPARC's impact?

Garner: One other comment, I guess. This is all SPARC focused, which is good. But one of the main things that fueled Sun's success was Sun OS and Solaris. And again, I was saying that one of the things that SPARC enabled us to do was to build arbitrary systems. And that combination of the best-in-class systems and the best OS really helped propel Sun's success. And I want to say as a last comment that I'm really proud to have worked with such a great team of brilliant people to actually work on an instruction set design that not only was successful for a company but has lasted this long. I mean, most, these days you can compile instruction sets on the fly. It's an anachronism almost, and I'm really happy to have worked in kind of that closing era of computer design.

Gwennap: So Anant, any last stories?

Agrawal: Same thing. You know, I think I feel very honored to have worked with such great people. And some of the best in the industry, and that's what really made SPARC happen.

Gwennap: Bill?

Joy: Yeah, no, I think as we head towards the limits of lithography and you think about a 14 or 10 nanometer chip, I still think there's got to be a better way to organize what's going on in there to extract more value. And I hope, maybe it won't be a SPARC, but I hope somebody does something new. It's always disturbing to me, if you look at the generations of these processors, the number of transistors that are switching for every bit of computed state, has just gone--

Garner: Shrinking.

Joy: --to the ridiculous. And if you think about the number of zero-one and one-zero transitions you can have for a given amount of power on that chip, and the actual amount of work we're doing is so small that there's got to be like a factor of 1,000 headroom for some new idea to come in. And I hope somebody has a bold idea like John Cocke's and Dave's idea of RISC, something else as a reaction to what we're doing now so they can really do what the physical medium is capable of and not just what the historical stuff mapped onto it. Because no one anticipated, I don't even know how many transistors you could put on a die like that, but it's just some ungodly number. Something better should be possible. Somebody should go try it.

Gwennap: Leave it to Bill to take our history panel into the future. So Dave, you're batting clean up here.

Patterson: Yeah, I think Sun deserves credit as a corporation. It probably wasn't the best engineering organization in the world that set a corporate target and everybody hits it, right? Everyone marching the same order. But was a bold company. Kind of chaos would happen, but the management would see something and embrace it. Like Java, right? Java was this failed-- that wasn't what it was called. Oak, right? It was this failed project. It was actually a bold vision to reinvent a programming language for embedded devices because that was going to be the future. That didn't work out. But somehow, at Sun they saw that and matched it to the dot com and embraced it and it became a bigger part. Sun, here's this-- how many years old was Sun when they did SPARC? Three years old?

Garner: Two.

Patterson: Two-year old SPARC, startup was going to do its own instruction set. And the management was willing to support that. So one of its legacies was this bold company that would pick things-- moved into servers, right? Bought Crays. The thing that SGI wanted to kill, Sun bought it and turned it into an important force in the marketplace. So Sun always was a bold corporation. And going from Xerox PARC, which is used in business programs as a bad example, right? This is why you shouldn’t have a research group, because you can't turn it into products. And Sun was just the polar opposite of that, right? Could recognize ideas, embrace them from many companies, turn them into products and make things happen and lead the industry that way. So I think Sun, the leadership and Sun as a corporation deserves credit. You guys will know how that extends into Oracle Sun. I’m hoping that Sun will take that way. But I'd like to thank them for whatever, let's see, 28 years of consulting, and the boldness to be able to embrace new ideas and make exciting products.

Gwennap: Great. Well, I'd like to thank Robert, Anant, Bill, Dave, for participating here. We've heard some great stories today and learned a lot about the history of SPARC and its influence, not just on Sun, but on the computer industry as a whole. So thank you very much.

Garner: Thank you.

Patterson: Thank you.

Joy: Thank you.

Agrawal: Here with me is the first SPARC microprocessor ever produced. This is indeed the chip number one. And this is what really started the whole revolution for Sun. So this is a die photo of the same chip, the first SPARC microprocessor. As you can see, this is a gate array with one custom block which is the register file. And this gate array was produced by Fujitsu Limited in Japan. What you see here is a chip here which is the UltraSPARC-I. this was the first 64-bit microprocessor Sun did. And this is the chip that took Sun to 14, $15 billion in revenue. Well, Sun did do crazy things. At one point, we designed a microprocessor in ECL technology. And even though Sun did not productize that microprocessor, it did finish it. It was used by Cray Computers, which eventually Sun acquired and created a multibillion dollar business out of that. So this was one of the ad campaigns we put together when we introduced UltraSPARC-III, our second generation of 64-bit microprocessors. What you see on your right is a package die for an UltraSPARC microprocessors. And what you see on the left is a package die of a MicroSPARC microprocessor. At COMDEX in 1996, they celebrated 25 years of achievement for microprocessors, and Sun was one of the companies they recognized at that event.

END OF PANEL SESSION

