IEEE-SCV Life Member Affinity Panel
John Mashey, Anant Agrawal, and Dave House

Moderated by:
John Hollar

Recorded: January 7, 2013
Mountain View, California

CHM Reference number: X6745.2013

Hollar: It's such a pleasure to see all of you tonight. I'm John Hollar, the CEO of the Computer History Museum, and I see so many friends in the audience. It's just fantastic. It's like a museum night here or something. Thank you for the few museum pieces. No, I'm just joking. But it's really— it's wonderful to have you here this evening [at the IEEE Santa Clara Valley Life Member meeting]. And to let you know how excited we are to be putting this panel together for you. And Alan Weissberger really deserves a tremendous amount of credit for putting this together. Alan learned the term "herding cats" by putting this panel together, I think, over the last three or four months. Because the players changed, the subject changed, and but tonight, I think you're going to be thrilled with the discussion and the panel members. And I'm going to get to them as quickly as possible. But first you have been generous enough, [Chairman] Les Besser, to invite me to say a few things about the museum, so I do want to take this opportunity to do that. By the way, I do know the score of the game, so if anybody really wants to know the score of the game, I'll be happy to give that to you. But if you're waiting for it, then maybe I shouldn't.

Man: <inaudible>

Hollar: You could do that. The museum is undergoing a real renaissance, I think. There are three chapters in the history of the museum. The first chapter was the one that was written in Boston, a very distinguished chapter, written for many, many years, very successfully. The world-leading collection that the museum enjoys, which is the largest collection in the world— getting some feedback, Fred. Maybe I should—

M2: No, just keep on going.

Hollar: Oh, okay. It's the largest collection in the world. It's more than just large. It's irreplaceable. It could never be duplicated. In large part because the hard work that Gordon and Winmail did beginning in the 1970s, supported by founding board members— many of whom are here tonight— produced a collection that could never be duplicated anywhere in the world, and it is our competitive advantage, to put it into kind of private sector terms. That was Chapter 1. Chapter 2 was the transfer of the Boston Museum to California, putting down roots first at Moffitt Field, getting a base of people out here to love it and care about and support it as much as the group in Boston had over the years. We very proudly to still have the founders plaque displayed at the museum, and many of your names are on it. And so that sort of repotting of the museum, if you will, from Boston and California was Chapter 2. And Chapter 3 is the one that we're writing now, which is really the birth of a world-class museum that tells the history of computing on a global basis. Not simply on a Silicon Valley basis. And we're just having a tremendous time and a lot of fun doing that. I'm fortunate enough to have a wonderful board, a tremendous amount of support, not only here in Silicon Valley, but from a lot of places around the country.

We've grouped ourselves really into four areas that we're pursuing. First of all, the events. Many of you are at our events, and I see you there a lot. This Event Series that we put together, led by *Revolutionaries*, which has been on public television for more than a year now, and starts its second season next Tuesday night, really has transformed the museum into something that I think it's very important for it to be, which is a thought leader, about the importance of computing, its impact on society, and the implications for all this change on the future that we're all going to live together. And that's partly expressed through the Event Series. That's the area where we really do explore what's ahead in computing.

The second is in media. We have really been increasing the media footprint in the museum over the last three years. We now have 14 weeks of television on KQED every year. KQED-FM has just created a new series, called "The Computer History Museum Presents," which will be 14 to 20 programs. In this calendar year we had eight programs on C-SPAN this year, we're trying to increase that to 16. And we're now in discussions with NPR and Politico, the political blog about a series that will run on the implications of politics in technology on our future. So being a place where these ideas get spread through media is also very important to us.

The third is being a hub for Silicon Valley. I was just saying to Ed, Ed Aoki, at dinner, these as of December 31, we had done 381 events at the museum in 365 days in c2012Calendar 2013. So we really are a place now where people gather, where they hold conferences, where they meet, and where we're really kind of a resource for the community, which we're very proud of.

And the fourth is in attendance and just simply people engaging with the museum. All museums have to be gauged by how many people are coming in a year. We're now up to comfortably about 125,000 a year, which for a museum of our size, and a relative specialty, is really terrific. And it just shows you what appetite I think there is— not just here, but around the world for people to understand what's happening in computing and technology and to provide some context in their lives. I always say when you go through "Revolution," our new exhibit, that you're in here somewhere. Just walk through, and within about the first 15 minutes, you'll find yourself. About the second week that we opened, I walked in, there was a young woman standing in front of the slide-rule exhibition, and tears were just streaming down her face. And she was way too young to have ever used a slide-rule. And I said, "What have you found that's really connected with you here?" And she said, "You know, my father was an engineer, and he always carried his slide-rule in the pocket of his shirt. And when he came home at night, he would lay that slide-rule down on the kitchen table. And when I got home from school, if I'd been working late, or if I had some kind of activity, when I saw the slide-rule on the table, I knew my dad was home." So you never know how people are going to connect with computing, or their lives. But it happens in strange and mysterious ways. And I think that's one of the things that makes us so special.

As well as all of you. We have a great group of volunteers. A great board, a great donor base. And so we're just very fortunate as an institution to be working from this base.

The IEEE is a very, very important group for us. so I'll just close before I introduce the panel by saying we have so many ways that we touch the IEEE. And we're very, very fortunate to have a close relationship with you. So partly as a way of saying "Thanks" to you for all that you do for us, we're having an IEEE weekend, which will be Saturday, February 16th, and Sunday, February 17th. And all day on both of those days, all IEEE members and their families will be invited to come free to the museum all weekend long. You'll get a ten percent discount off of membership. You'll get a fifteen percent discount in the store and the café. And if you sign up as a new member at that point, then you'll get a two-year membership in any category for the price of one year. That's just our way of saying "Thank you" to the IEEE. We think we're going to have a big weekend that weekend, and we're really, really happy that so many of you could come by. That's February 16th and 17th from ten-to-five, all weekend long. So it's now my pleasure to bring the panel up. Dave, John and Anant, please come on up and take your seats here.

Hollar: All right, this is going to be a well-oiled machine. We got together, we rehearsed, we know exactly how it's going to go. It's going to be a lot of fun. The title of the panel as you know is, "How the Race for Microprocessor Leadership Transformed Silicon Valley." And I said that rather than try to introduce all these gentlemen with their extensive and distinguished resumes, I was simply going to ask them to do a brief self-introduction and talk to you about the things that are most relevant about them to the discussion we're going to have tonight about microprocessor history and their role in it. So I'm going to start with John Mashey. Take it away, John.

Mashey: Great, thanks. Let me do a quick question to the audience here. How many people worked on, built microprocessors? How many people were computer hardware folks? All right. How about analog folks? Boy, it's a good mix. How about software folks? Oh, there are a few of you, too. Good. I'm really an old software guy. I was finishing up my PhD at Penn State when a Bell Labs Director walked in and said, "Have you thought of not going to academia?" I said, "You got an alternative?" "Yeah." "Okay." So I got recruited to go to Bell Labs and work on a thing called the Programmer's Workbench Version of UNIX. I'm talking way, way back. That was the thing that turned UNIX into something you'd let lots of people loose on. I managed the first UNIX system III-port to the Motorola 68000, which was exciting. But then the Bell System was breaking up, and either I was going to be a Bell Labs Department Head and be there forever, or I was going to go somewhere else. And I liked the weather better out here. So I came out— people remember the company Convergent Technologies?

Group: Yes.

Mashey: So I ended up being the Director of Software for one of the divisions of Convergent. And about that time, towards the end of '84 now, Steve Blank— anybody here know Steve Blank? A terrific marketing guy, did six or seven startups in a row before doing ePiphany, and then he just wouldn't have to work anymore. He had left Convergent. He met up with the three founders of MIPS, John Hennessy, John Moussouris and Skip Stritter, and he was helping them out. And he thought he would bring them over to Convergent to try this MIPS RISC stuff out on some systems guys. And so they asked me what I thought after I heard their pitch, and I said, "Well, hey, I like RISC. I actually evaluated Dave Ditzel's CRISP architecture back at Bell Labs, and said we should do it. I said, "I like that. But you know, John, you clearly had great compiler people, and great chip people on your team. But did you have any operating systems people?" And he said, "Well, no, but how did you know?" And I said, "'Cause no OS guy ever asked for your memory management unit, or your exception processing, you know?" They said, "You should come over and look at the real specs then." So there were a few months in late '84, where I was over there every Saturday, looking at specs and arguing with folks. Then they said, "You're here a lot, why don't you work here?" And I said, "No, no, no. I'm busy." But I couldn't resist, because for a software operating systems and compiler guy, the once in a lifetime chance to actually help design an architecture that you *liked*— I spent a lot of time on architectures that I could *live* with, but didn't like much— that was irresistible. So I moved there in early '85. I did a lot of work on the memory management and exception handling and other things. And to finish up, there were two things that I worried about. One was making sure it was healthy for UNIX and other substantial operating systems. But the other, oddly, even back then, was to worry about embedded control. I thought, there was action for us in the high-end of embedded things, like high-end laser printers and such. And I made a lot of effort to try to tweak the features so they, at least, would not get in the road of that. So I'm basically a software guy that strayed into doing chip architecture.

Hollar: Thanks, John. Anant Agrawal, please.

Agrawal: Hi. You know, when I doing my bachelors in India, my major was instrumentation. So I played around a lot with discreet analog components, and all kinds of measurement instruments. I never, ever thought that I will be doing what I did for the last few years, which is design and manage microprocessors. I'd taken one class in digital design, and that was my extent of digital knowledge. Then I ended up at Cornell. That's when I first saw a real computer using punch cards, you know, the IBM. That's when I ended up getting interested in computer architecture. And I had a fellowship for one year, and I decided that I will finish my masters in one year, because I couldn't afford to go to Cornell any longer. So after finishing my masters, I moved West, and started working for a company called Memorex. I don't know if you guys remember Memorex. <laughter> They used to design storage systems <static; inaudible> for IBM compatible mainframes. So guess what, my first assignment was to design an ECL processor using 10kh logic. And it was on two motherboards about this big. And it was really a control plane processor. I mean, data plane processors. The control plane was a PDP-11. And if you recall in those days, most of the companies including Storage Tek, were vertically integrated, they would buy these platters, and then assemble them in what they called HDA, head disk assembly. The goal was to certify the platters. The data rate coming off the platters was something PDP could not handle. So we designed this ECL processor. Well, from there, I moved on to a company designing IBM compatible mainframes in CMOS. That was the first attempt— and if you recall before '80...

<technical difficulties; microphone switching>

House: I think I'm the problem.

Agrawal: Yeah, so I joined this company designing a CMOS mainframe, and there I headed up the floating point Unit, just not me, but our team did about 17 gate arrays. And eventually that company went under. By the way, this company was not just designing mainframes, they wrote their own EDA tools, they designed their own gate array, had their own fab here in Sunnyvale. So after that company closed, I ended up with a very small startup called Sun Microsystems. <laughter> And how I got there is a funny story, which I already talked about last time. I was called by one of the guys who I worked with at IBM mainframe company, Bill Van Loo, saying, "You know, Sun is doing some interesting work, why don't you come and talk to them?" I said, "Look, they do workstation using 68000 processors. And I'm really into designing processors. He said again, "You gotta come and talk." So I went and talked to Professor David Patterson, who was consulting for Sun at that time; Bill Joy; I'm not sure if I talked with Andy Bechtolsheim; but Robert Garner; a bunch of people there. And I distinctly remember as I was driving back from the interview I just had, I said, "These guys are Looney Tunes!" <laughter> "They have never done a chip before, and now they're trying to do the most complex piece of silicon." And as I was driving some more, another thought came to me that said, "These guys don't know what they're getting into. At the same time, these guys are crazy enough, so if there is a group of people that can pull off a crazy thing like this, it's got to be these founders at Sun." And that's how I ended up at Sun. <laughter> At Sun, I initially worked on the processor architecture with other people. And then started managing, or started designing the first SPARC processor. There were actually only two of us who designed the first processor, the other person was Masood Namjoo. I designed all the pipelines in the processors. Oh, Don Jackson designed the Floating Point control unit. And then over time the responsibility expanded, and I stayed there for 18 years for those 18/17 years I was involved with the microprocessor, and probably for last ten years of my stay there, I managed the entire processor family. So that's kind of my story on how I got involved in microprocessors.

Hollar: Thanks, Anant. Dave House.

House: Oh, BSEE, Michigan Tech. Went to Northeastern University for MSEE. Stayed in Massachusetts and was one of the architects of the in-route air traffic control system. Then stayed on to design the display processor for that system. A company called Computer Control Corporation had recently been acquired by Honeywell, was competing with Digital at the time, hired me to lead the development of the microprocessor, or a new mini-computer, I should say. I new mini-computer. During the development of what was known as the H-12 during development, first of all, I got introduced to Intel, because the lunatic guy that ran memories thought we should put a semiconductor memory on this machine. I told him, "There's no way I'm going to fund that, we're staying with core." He said, "I'll get some development money, let me do a parallel development that uses a 1KD RAM from Intel." And sure enough, we wound up getting the 1103 operating on the machine as well as the core memory. But during that time, I got to know the people at Intel. And after I won the Harold W. Sweat Engineer Scientist Award, which is kind of like being a Honeywell Fellow, got put on the architecture team for the merged— the architecture for the merged GE and Honeywell, after Honeywell acquired GEs computer business. And I'd been asked by

Computer Design Magazine to write an article. I chose to write an article about the impact of semiconductors on mini-computer architecture. And I'd observed each generation that I worked on in these digital machines. We'd go in from one flip-flop in a package to four-bit registers, to four-bit ALUs. And I was kind of like the fourth guy that discovered America. I kind of discovered Moore's Law way after Gordon Moore did, but I just didn't know he'd done it yet. And so I plotted the transit, <static; inaudible>. I am the problem. Speculated on the impacts on the computer design in this article. Well, one day I'm going by my boss's office, there's an article on the wall that talks about Intel's going to introduce the 4004. I went to the components guys and got the data sheet and read it, and I went home and I told my wife at the time, I said, "I learned two important things today." She said, "What's that?" I said, "One, my job's going to change industries." "What do you mean?" "I design a high-volume, low-cost computers. They're designed in system companies today. They're going to be designing semiconductor companies in the future." "What's the other one?" "My job's going to change coasts. This company's on the West Coast." Oh, now that was important! <laughter> Wound up doing a short stint at MicroData in Southern California where I was Director of Computer Development, developed the 32S machine, the 3200, software-based microcode on that one, 32S. And the Reality System. And then during all the time I was there, my buddies that had left Honeywell and went to Intel were trying to recruit me at Intel. And I started there in February of '74, before the 8080 was introduced, when the 1103 was the big product, and my job as Manager of Application Engineering was to make sure that the computers worked with our D-RAMS at the time. You know, the customer— Ed Gelbach, Head of Sales, would come into my office and say, "You got to go Burroughs." I said, "Okay, when do I go?" He says, "You go today." I said, "Okay, I'll go home and pack." He said, "No, no, buy the clothes when you get there. Here's your ticket." <laughter> "What's the problem?" "They called up and they said to stop shipments because they can't get their machine to work. Go out there and get it to work." And so that was my job was getting the shipments turned back on. Well, soon I was Head of Marketing for Microprocessors, and then I was General Manager. For 13 years, from '78 till '90, I was the General Manager of the Microprocessor Business from the— just after the introduction of the 8086, where I was the Marketing Manager to the Introduction of the Pentium II and went on and became the Director of Marketing during the time we created Intel Inside. And then one of my last two years '94 to '96 as Head of the Server Business. Before I left Intel, after 13 years on the executive staff there, to go into the networking business, became CEO of Bay Networks, and then President of Nortel Networks, and now I'm Chairman of Brocade. So I really came at the microprocessor at Intel from a totally different position than the rest of the company. We had, you know, Gordon Moore and Andy Grove and Bob Noyce were these wonderful semiconductor physicists, chemists, that understood this stuff at the atomic level. I didn't really understand that part. I was a computer designer. I understood the hardware and the interface and the software. And so my position on the executive staff was always kind of the representative from the customer side, "What do you really do with this?" And I was really blessed during my 13 years of running the microprocessor business there, of having the most phenomenal semiconductor technology in the world. That on which to build some of the architectures that, between you and I, were not necessarily the best. <laughter> And we had a very dynamic time, first with— competing with Zilog with the Z80. Killed us. The 8000— the 8085 lost to the Z80. Then we came out with the 8086 and the 68000 was better. And all you workstation guys were using the 68000. The Z8000, then the 68000, the 68020, and then there's SPARC and there's MIPS and the Power PC, and there was always a better processor. It was a better architecture, computer architecture. And our job was to change the rules of the game. That is, to make the story about the software, the peripheral chips, the development systems. Remember— how many people ever used an Intel development system, or circuit emulator? Right, see? Yeah, a few of you have used the old famous Blue Box out there. And basically, we've tried to change the rules of the game, and win despite the fact that we didn't always— well, maybe we never had the best architecture. <laughter>

Hollar: So there you go. Now you've met the panel. And three very different stories. Three very different starting points. MIPS, with John Mashey; Sun with Anant Agrawal; and Intel with Dave House. So we agreed that we're going to split the rest of the evening up essentially into three parts. One is Competitive Strategy. How did they go about competing with each other and thinking about owning their chunk of the world? Two was Outcomes. Three was Implications for the Future. Because after all, this is about how the race transforms Silicon Valley, and the transformation is still happening. So John, on Competitive Strategy, I want to start with you. When we met before, you essentially summed up MIPS by saying, "It was some good chip and compiler technology. Trying to figure out how to leverage yourselves into the market." And the quote I most remember, it was "Rock Soup." So you want to talk a little bit about making Rock Soup, and what the strategy was for MIPS?

Mashey: We were even crazier than these guys, though. Because you know, we really got going, December of 1984, and shipped the first chips in boards with software that was starting to work a year later. That's insane! Right? You've got new compilers, a brand new chip architecture, a new chip. You're this little company. You've got a couple VAXs and a couple Apollo Workstations, and you got to make all this work, and you've got to cobble tools. I mean, you know, it's crazy. And you've got to figure out who you're going to sell this to. So in our case, we weren't Sun,who already had a thriving systems business. So we had a lot of struggle. Were we a chip company? Were we a chip with software company? Were we going to sell boards? Were we going to sell systems? And we scrambled around a lot for that. So we knew, in our case, an awful lot of the action had to be in alliances of one kind or another. The first thing is if people remember back in the late '80s, there wasn't anything like TSMC that was a real foundry. So we had to go convince semiconductor companies that they ought to actually be willing to build this stuff. And we got Performance Semiconductor, LSI Logic, and Integrated Device Technologies. And they all had different reasons for doing what they wanted to do. But also we had to do three tape-outs for everything that we did. And then we had to find systems companies. What we were always hoping for were systems companies that had to compete with somebody that was building their own better chips, and so they would need us. And then things got more complicated later. You know, we went into a big thing with Microsoft and a big alliance. But we were always trying to figure out what big system customer can we get that will convince a big semiconductor company to sign on with us?

Hollar: The systems company that chose to make the processor. And so you had to consider your advantage to be design and software. Talk a little bit more about that. And why Sun believed it had to be in the processor business in the first place?

Agrawal: Sure, so when I joined Sun in '84. Sun was shipping its first workstation. At that time, Bill and Andy, and some others at Sun had this vision of network computing, using the internet, TCP/IP standards, and all that. However, they had a small problem. The processor performance that they were getting from Motorola and others, would not make that vision happen.

House: Especially floating point.

Agrawal: Floating point, and also even integer performance. So they went and talked to Motorola— I'm not sure if they talked to Intel, but—

House: Oh, yeah.

Agrawal: Certainly with Motorola. Looked at the roadmaps, and were not convinced that these companies can actually give Sun what it needs to create that— to make that vision a reality. So that's when they decided, "What the heck— we'll design our own processor." And that's how things really got started. Now the big difference at Sun, was, you know, you're right, you were Looney Tunes. But—

Mashey: Yeah, yeah. We were more Looney than you were.

Agrawal: Yeah, exactly. The big advantage Sun has was for us processor designers, we had a customer, right? Every processor designed, before the design started, we knew what machine it's going to go into, and what the requirements were. Right? In fact, at Sun, the chip design process was not necessarily driven by the chip architects alone. It was a joint effort between system guys; chip designers; chip micro-architects; process people— and I'll touch on that maybe even if we get time, on how it was different— operating system people; compilers— all of us got together, before we made a final decision of what the product would be like. And that gave Sun a distinct advantage in the marketplace. Not only in addressing what our customers were looking for, but also bringing out processors and machines a lot faster. Because when the processors came back, and those of you who have debugged these complex chips, it's a very difficult process. However, since the operating system and compiler people were part of the original design team, they were quite familiar with what was going on, so the bring-ups were a lot faster at Sun.

Hollar: Dave, you've already alluded to one of the strategic issues that Intel faced, which was it never necessarily had the best architecture. So in the face of that, what was the strategy, the winning strategy that Intel had to come up with?

House: Well, we kind of got into the microprocessor business in an altogether different way than either MIPS or Sun did. If customers came to us and said, "Would you do a custom chip?" And we didn't have enough engineers to design those custom chips. So we thought, "Well, if we could make it programmable, maybe we could sell it to a lot of different people." And I think one of the most important initial decisions at Intel, first meeting I ever attended at Intel was where the debate was we hire people called Field Application Engineers? That was a new concept at that point in time. We decided that we needed engineers in the field that worked for us that were design engineers that had designed products like we were selling. And the Field Application Engineer for us played a very critical role relative to setting our direction and where we went. So unlike National and Motorola and even Zilog to— although they did it to a greater extent— we were very early on creating a complete solution of IO chips and clock chips and memory interfaces, and development systems and in-circuit emulators, and PL-1, PLM language, and the Isis Operating System, and providing a more complete solution. But we always were at the position where we didn't control the whole environment. Unlike MIPS, that created the whole environment out of whole cloth; or Sun, who had a basis based on another microprocessor, we never controlled our environment, so Microsoft was always— we had this real yin-yang relationship with Microsoft, during all the time that I was there. We're real love/hate relationship, where the two of us were totally wedded together, like Siamese twins who hated each other. <laughter> Yet we're dependent on each other. And so we couldn't develop the operating system, we couldn't develop the compilers to the extent that we needed optimizing compilers, like MIPS really succeeded at. If we didn't control the rest of that environment we were always trying to get UNIX running, people to run UNIX on our processor, simply so that there would be an alternative to Microsoft. And so that the dynamics from our side were quite different. We probably— you know, we kind of stumbled into our architecture, starting out with data point, that wanted a shift register, and we said, "Well, why don't we do the processor for you," to an 8-bit machine, then we extended it to a 16-bit machine, we extended it to a 32-bit machine, that bite addressability, and Big Endian versus Little Endian issues. And a couple different memory management schemes. And the stuff that— we needed a John Mashey very badly <laughter>, somebody who really understood operating systems that could come in and tell us what we were missing. Because we did it more by trial-and-error and stumbled along. We had the best silicon technology, we didn't have the problem that either of you had of having to go to foundries. And we had the advantage...

Mashey: Well, if we had had foundries that would have helped! But the...

<overlapping conversation>

House: That's right, there weren't even foundries. We were even more complicated. And we had quite a profitable, actually EPROM business in the early days, wanted the development of silicon technology, and build our fabs for us, and the microprocessor guys could ride on that wave until we were producing these 90 percent margins and could afford to optimize presses, the presses technology for the microprocessor itself. But I remember extremely envious of both of you, because of the— you controlled the whole vertical stack, and you had expertise in all levels. And I kind of felt like we were kind of at the foundation level, the fortress, but we were struggling the further up we went. And I'm sure that's why our architecture wasn't always the best architecture. But in the end it didn't matter. I mean, architecture really doesn't matter. It's software. It's all about the software. And once we had the IBM design, the PC design win, you know, after that it was just mopping up the rest of the side markets like workstations. <laughter> Servers.

Hollar: So your long-time boss, Andy Grove wrote this book with the famous title, *Only the Paranoid Survive*. How paranoid was Intel, at any point, about what these other two guys were doing?

House: Well, Andy was my direct supervisor for 13 years. And the one thing—

Mashey: Whoo!

House: One— I think I should get a Medal of Honor for that. <laughter> Shouldn't Congress give me a medal?

Hollar: I think so!

House: Andy was not just a little bit paranoid. Andy could find twelve out of every three problems. <laughter> And Andy had us focused on everything that possibly could go wrong. And the majority of the time, that wasn't the biggest problem, but he always had us focus on the problem. And so, how are you going to solve— you know, how are you going to get recognized, I mean, start out with the IBM PC PS2 announcement when I came back so proud that IBM asked me to speak, and didn't ask Bill Gates to speak. And sort of this relationship. And Andy say, "Yeah, but nobody knows the reason the machine's faster, because it's got a 286 in it instead of an 8086." And so you know, "It's got the same power supply, it's got the same keyboard, it's got the same monitor. Everything, same memory, everything is the same. It's the processor that makes this thing better. Nobody knows that." And you know, a few years later, "Intel Inside" was the result of Andy's paranoia about that particular fact. But Andy did keep us focused on the problems. And so there was a heightened awareness of what MIPS was doing and what Sun was doing. Before that what Motorola was doing, what Zilog was doing during that whole time. Andy had, you know, he was a problem-focused guy.

Hollar: I want to get the same from both of you. So Anant, describe the competitive pressure for issues at Sun. And you said at one point, and this is really the conclusion of where the competitive pressure led you, that Sun put all of its wood behind the SPARC microprocessor, and there's a great story about how vividly that was exhibited to everyone. But can you talk about the competitive situation that led to that?

Agrawal: So before I get into that, one more thing about Sun. Sun as I said I was there for 18 years. And one thing that was always there was Sun was always looking ahead, "What is the next trend?" What is it that customers would be looking for next?" and they this aligned with investments based on their predictions and their knowledge or their looking into the future. So if you look at the history. We started off doing RISC, just because we needed more performance. Then Sun moved to workstation market, and we became the leader in workstation, and designed microprocessors that were suited for workstation. In fact SPARC Station 1 was probably the most cost effective workstation of its time. <coughs> Excuse me. Then in late '80s, they saw this wave of internet and telecommunication infrastructure coming. So that's when we started working on 64-bit processor architectures with support for multi-processing built in. Visual instruction set for voice-over-IP processing. So that was another transition Sun made, and became really a leader in the server space. The machines that were running the backbone of telcos and internet. Then in the late '90s, and here is another example, we saw a big wave come in in 2000s. How to handle and process large amount of data in parallel. And this is what we call today as the big data analysis, or cloud computing, right? So we saw that coming in late '90s. And as a result of that, we switched all of our products— microprocessors we were designing at that time— to go to multi-core. Okay. Well ahead of the industry. And from there we moved to multi-thread. Again, took an initial step into something that was totally different. So that has been the philosophy of Sun, which is trying to figure out how best to serve not just today's customer, but how do you plan for what's coming through? I don't know if we were really a lot paranoid about others. Yes, we always had MIPS on our radar; yes, we always had X86 on our radar. In fact, in 1995, when we introduced Ultra-SPARC-1, actually our performance was significantly better than Intel at that time. And I know this, because one day I met with the GM of Pentium [ph?] who used to—

House: Yep, yep, worked for me.

Agrawal: He worked for you, and he’d say, "When you guys came out with Ultra-SPARC-1, we were sweating bullets." So I think Sun really focused on a few things, you know, building great systems, providing a great system solution. We did struggle at times with manufacturing, semiconductor manufacturing. Though we formed a lot of good partnerships and eventually it all worked out. So yeah.

Hollar: Okay, John?

Mashey: Well, we weren't paranoid. When you're a mouse, and there are elephants rolling around, you know, you're not paranoid, you're just sort of rational. <laughter> Again, MIPS was clearly the smallest player in all this. And of course, what we always— what we were always worried about was that like Intel would come out with a RISC architecture that really worked. And we worried about Motorola.

House: We would've if we could've.

Mashey: What?

House: We would've if we could've.

Mashey: Yeah, I know. Well, but you tried hard to market the Intel 860 that way, right? There were some minor software issues.

House: Yeah.

Mashey: I, for weird reasons, ended up sort of running the competitive intelligence and counter-intelligence for MIPS. Don't ask me why, but I was. And so we watched everything, because we had to convince serious companies to bet on this tiny little startup. Again, the view was there were elephants around. Sometimes even wounded elephants can kill you. <laughter> When they're rolling around out there. So I don't think we were paranoid. We just had to be ultra-competitive wherever we could find whatever wins we could get. You push up against the wall everywhere. And if you find a hole, that's where you go through, right? And it was just always complicated. Our designs early had hardware systems folks, and OS folks, and compiler folks, and chip folks. But we didn't control the end product either so much as these guys did. So we were always dependent on alliances and shifts. And we went in and we did this thing called the Ace Consortium with Microsoft, who was just as eager to get you out of there.

House: That's right!

Mashey: As you were them!

House: We were trying to get UNIX, they were trying to get <inaudible>.

Mashey: Yeah, yeah, so it was funny, you see, because some of the time we love you because you pushed UNIX, because I'm a UNIX guy, I own the California UNIX license plate, so you were good. And sometimes we were with Microsoft, and I loved Windows NT. <laughter> But anyway, I think we were always running scared. It's not anything irrational.

House: But the Compaq, Microsoft, MIPS, Ace initiative was scary for me.

Mashey: Yeah, I know.

House: Very scare for me. You know, your second biggest customer at the time is the person, the company that controlled our software base, and this little startup that had a good story.

Mashey: Yeah. Yeah, I saw the charts, and I think you must have put them together that were Intel things whacking MIPS. I saw those.

House: Yep, yep.

Hollar: You know, if we peek ahead a little bit...

House: Because I had your face on MIPS.

Mashey: Yeah, okay.

Hollar: Sticking with MIPS for a moment, we now live in a world of very low instruction set, very low power processors, how much of that was on the horizon when you guys were working on this architecture?

Mashey: Literally the architecture was mostly designed, in the first three/four months of 1985. And then we're filling in holes. It had to be small and lean, because— in fact, I'll just pass this around for fun, right? I'll pass two around. So this is a die plot of the MIPS R2000, which was a whole 110,000 transistors. These days you couldn't even see it. We had two micron CMOS, that's what we could get. Okay, now in '91, this is was the first 64-bit micros, the MIPS R4000, that was about ten times as much, and 1.2 micron. But this was all we could get, right? And so we had to be lean. I had given software talks called "Small is Beautiful." So I like RISC, minimal. I actually had operating system guys from other companies go through a list of questions, "How do you do this? How do you do this? I don't understand, how do you do this?" And I'd always tell them how you did it. And they'd say, "Do you realize this chip only just barely works?!" And I went, "Yes! It's RISC" <laughter> So here's the thing, our initial chips had to be good for UNIX, because if you can't possibly get huge volume, you've got to have some systems revenue business to get enough margin to be able to keep going.. You can't— suddenly get infinite volume, because we weren't Intel. But again, I always had in mind that, with Moore's Law, chips would keep shrinking and by keeping it minimal, and having lean exception handling, and other little things, that there would be embedded applications. Remember I was from Bell Labs, communications, networking. I had a feeling sooner or later we'd get into that. About 1991/'92, IDT, I think it was, said, "Hey can you help—" I often ran around with the chip partners— "Can you help us with this little networking company?" And I went over and I convinced them that 64-bit MIPS chips would be a good thing for Cisco. Right? <laughs> And I left the meeting saying, "I don't know which of these networking companies is going to win, but these guys seem pretty smart, I'll buy some of their stock," which turned out to be okay. But to this day, there's way more MIPS chips in communications and networking products. The MIPS guys tell me that there was like 700 million chip cores that shipped last year? And a whole lot of that's in consumer products with TV sets and routers.. So it's sort of ironic, as we started by making sure we could build UNIX workstations and servers, But you know, I thought we'd be in embedded.

Hollar: Was this same trend, Anant, was that partly on Sun's mind when they decided to put all of its wood behind SPARC, strategically?

Agrawal: So, actually, I didn't answer your last question.

Hollar: It's all right. You knew I'd get back to that.

Agrawal: Thanks for reminding me. So it was around '87 timeframe, Sun had started shipping SPARC machines. They were being very well received in the marketplace. At the same time Sun was also shipping 68000, and lo and behold, we were also, we had started shipping X86 machines. So we had these three architectures. And you know, Scott, had one of the meetings, strategy meetings, and Scott said, "You know, this is not sustainable. I want to go to one architecture." "What is it?" "SPARC." And most of the people in that room looked at him saying, "Do you really want to take that risk? We just started this effort, and who knows what's going to happen in the future?" Here are these established, some of them are companies that are in the business of supplying microprocessors. He says, "Nope, I made a decision. It is going to be SPARC." So, that's where that phrase came from, putting all the wood behind the one arrowhead. In fact, at Sun, at least in early days, we were known for pulling off some great April 1st pranks. And in this instance, when Scott walked into his office on San Antonio Road on the Fifth Floor, there was probably a 25/30 foot arrow going through his window, okay? It's about this thick. And this was always done by a team of software engineers, and they managed to pull that off. All the wood behind one arrowhead.

Hollar: Yeah.

Agrawal: What is your next question?

Hollar: Pretty vivid. Just, the same sort of architecture, driving your strategy. So John, by the way, our first processor that we did was 80,000 transistors.

Mashey: Yeah, right. They were gate arrays, right?

Agrawal: Yes, yes, yeah. You know, even though the first processor was very simple and small, and there's a reason for that, because there were a lot of naysayers in the company who wanted to kill the SPARC effort, and my goal was to get it done as soon as possible, to really do two things. One is to demonstrate that RISC is commercially viable, right? And second was, of course, go to market before anybody else did using RISC technologies. So but then as we moved to different markets, like servers and all that, we really didn't think about embedded. Many— well, maybe we thought about it, but we never succeeded in that space. We did put a processor core in the open domain, and really didn't catch up. So our focus always was to go after markets we were serving. And most of the time that meant enterprise server market, which means large, honking chips that burn a lot of power. So embedded really wasn't a part of SPARC. No, embedded was part of a line of processors, but then that really didn't go anywhere, which was the Java processors. But on SPARC, absolutely not.

Hollar: I want to move now just a little bit to how this all gets resolved. Dave, when did it start to occur to you and to Intel that this— the microprocessor race was starting to get resolved. Even if that's the right word. When the choices were starting to get made, and they weren't going to be reversed.

House: You know, we were always paranoid. There was always something to be afraid of. We had a lot of reason to be paranoid in the days of the 8085 versus the Z80. And in the days of the 8086 versus 68000. In August of 1981 when IBM announced the IBM PC, we knew that it was something big, but we didn't know how big it was actually going to become. Because of its open architecture and for people to develop compatible units, it became huge, and of course, that drove the application business. And once all the office applications were up and running on that architecture, it was sort of over at the desktop level, the PC and the notebook level for that generation. At least for the PC generation. Now we're in the mobile generation, that's another story. But it was really clear then. But then the RISC architectures came on the scene, and all the press, the *Wall Street Journal*, *Business Week*, and all the trade press, RISC versus CISC and RISC is going to win. And you could not tell anybody on Wall Street any finance organization that that wasn't true. And you couldn't tell the executives at any of our customers that that wasn't true. I think the most telling argument at the time was that Moore's Law doubles the number of transistors every 24 months. Well, it doubles microprocessor performance every 18 months, because these things are smaller and closer together, and you've got more of them, and you can do more architecturally. So you really double every 18 months. And the question always was what kind of performance advantage did the RISC architectures have? And my engineers, at least, would come back and argue about whether it was 50 percent or 100 percent— 50 percent or a 100 percent. I talked to the technologist and I'd say, "Well, if we double every 18 months, then 50 percent is, you know, it's a logarithmic scale, you'll understand, but you know, maybe that's four to six months; and 100 percent, maybe that's twelve months." So if we could just get our silicon technology out faster, and have better integration there, we can keep up in the mass market. And RISC architecture is where they did succeed early was with the workstation, but soon Intel architecture took over the volume in the workstation marketplace. Because just those economics. And then the server marketplace is probably 90 percent of the units are Intel architecture today, so it was basically able to move up. Where we really were paranoid that the RISC architecture— that the common belief at the time was that RISC architecture was going to move down. And the Ace Consortium, I think, was kind of the pinnacle there, of Microsoft and Compaq standing up and saying that the RISC architecture was going to take over the desktop. And that was probably the darkest days from Intel's standpoint, the most paranoid days, you know, "What are you going to do about that?"

Hollar: Was it Intel's vertical integration then that ultimately was decisive?

House: I think that it was our controlled silicon technology. The volumes that we had, our ability to work with and control the semiconductor equipment manufacturers. The silicon technologists that we had. Absolutely, we hired the best people in the world. Because we were making 85 to 90 percent margin on our microprocessor business, we could afford to hire the right people, to spend the money in R&D, to develop and build the fabs, to build these, you know, billion dollars and two billion dollars and four billion dollar fabs, and then I called it the virtuous cycle. You know, one thing fed the other, and it just became a snowball effect in a machine. And you know, today, as we enter the kind of the fourth generation in computing after mainframes, then mini-computers, then PCs and now onto mobile devices, we've never seen a single architecture make it from one generation to the other. It was the 360 with mainframes. It was DEC at the mini-computer, it was Intel at the PC level. And now ARM architecture is the architecture of the mobile devices.

Hollar: I would like you and John to talk, before we move to questions from the audience about the implications of the way your companies, and companies like yours influenced the way Silicon Valley is today. The way Silicon Valley works. It's kind of an open-ended question, but this whole— the integration of software and hardware, the reliance on partnerships, the speed, the need to make decisions quickly. Talk a little bit about how that helped to transform Silicon Valley.

House: Well, Intel started earlier, so maybe I can start, with stock options. Stock options played a huge role, 100 percent of all Intel employees got stock options. The operators on the line had stock options. There was equity kind of a basis, which became part of the culture. Extreme focus, high value for technology, high value for engineering and science, were a key part even in those very— carried over from Fairchild into Intel. And I think really at Intel, that culture that became the Silicon Valley culture, really magnified and amplified and grew and spread out into other companies. We were— started before venture capital was what it was when your company started. So I think that companies like yours took that culture and took it to another step.

Agrawal: Yeah, I would agree with the stock options. All of our companies created a lot of wealth in the Valley, and it was being noticed all over the world. You know?

House: We were changing the world.

Agrawal: Yes. We started a phenomenon that eventually spread to the rest of the world. Similar things, maniacal focus, you know? Focus on the right thing. Again, quick decision making. At Sun, it didn't take us too long to make decisions. The decisions were actually made very quickly. Like the decision to use SPARC as the only architecture. The other effect it had was really attracting a lot of talent into the Valley. Intel, Sun, certainly MIPS, were the places to be. Just like today what Apple and Google are. So a lot of people moved to the Valley to come in more for these companies. And we had the jobs for them. When I joined Sun, it probably had 250 people or so. And when I left Sun in 2001, they were at 44,000 or over 44,000 people. So it really created a financial cycle growing in this Valley that wasn't kind of seen before.

Hollar: John, your thoughts?

Mashey: When I was back at Bell Labs, Bell Labs itself was 25,000 people. You know, R&D. And Bell System, because there was one Bell System. It was over a million people. And oddly enough, we could still be pretty entrepreneurial, if you knew how to play the game. You can get terrific leverage, but it was not the speed of motion that you could get here. Among the effects on this was people. Remember way back, Frederick Terman was unhappy that his students kept going back East for tech jobs right?

House: Yeah.

Mashey: And so he go these two of his students named Hewlett and Packard to start a company here, right? And—

M2: But he sent his son back East.

Mashey: Pardon?

M2: But he sent his son... <laughter; inaudible>

Mashey: But after all, you know, once upon a time, the heart of the computer business was New York and Boston, right? That's where it was, right?

House: Yep. Really Boston. Well, New York, IBM.

Mashey: New York -you can't ignore IBM, you know, they're kind of there, right?

House: ____________, the mini-computer.

Mashey: Well, I mean, here's the thing, I think in a lot of respects, if there hadn't been the speed going on out here with the weird combination of competition and cooperation, I'm not sure that motion would have happened anywhere near as fast. There's a good book, Anna Lee Saxenian, who's up at Berkeley, she's from back East. She wrote *Regional Advantage— Culture and Competition Between 128 and Silicon Valley*. Really good book, right? She observed that back East tended to be much more vertically integrated, right?

House: And hierarchical.

Mashey: Yeah, and hierarchical. But you know, like we always said in the Bell System, "Well, if we need semiconductors, we need to buy a beach, you know? And get that sand." And out here nobody was big enough to do that. You guys even weren't big enough. You might have been big enough, but you weren't.

House: We weren't big enough to do everything.

Mashey: No, you weren't big enough to do everything. We were little, but we would gnaw on these guys as much as we could.

Agrawal: Right.

Mashey: On the other hand, when Anant's guys had an NFS connect-a-thon, it would be our engineers, and DEC engineers, and HP engineers over in their room where they would keep the marketing people out. Because they'd be horrified to see engineers from horrible competitors helping each other out, right? People familiar with the spec benchmarks?

Agrawal: Mm hm.

Mashey: Well, when we started that, it was Sun and us and Apollo and HP. And it was funny, because we were always taking a lot of time on performance benchmarks, and just duplicating a lot of work. And then having to argue— people remember the Dhrystone Benchmark?

Agrawal: Mm hm.

House: Yeah.

Mashey: Ugh! That's like a vampire. Put a stake through the heart. Yeah.

House: Dhrystone and Whetstone.

Mashey: How about Stan Baker? Anybody remember Stan Baker? *EE Times*? Well, Stan had this thing about Dhrystone MIPS ratings. And I dumped on him and so did somebody from Sun. And he said, "Okay, you industry guys say Dhrystone is bad, do something better." So we had a meeting down in Stan's bar in Campbell. That was the origin of SPEC. Anyway, it turned out to be a lot of weird things like that. As ferociously as this place competes, people get together and they get standards to happen, and they trade stuff. And I think that style of working, in part, because of the microprocessor wars, got amplified, because of all the partnerships and arrangements and even having to cooperate with fierce competitors.

Hollar: Yeah, yeah. All right, let's take some of your questions now, maybe for five or seven minutes. Alan, you want to ask the first one, and then Dick? And Chris.

Chris: Yeah.

Alan: Yeah, what's very, very interesting is that Sun and MIPS went about the microprocessor completely different. Sun <static; inaudible>

Hollar: Wait, wait.

Alan: Sun envisioned microprocessor giving them a performance edge in their workstations and later their servers. And MIPS was always trying to sell the microprocessor to systems companies. The question I have is, in the early days, did you ever think about going into the other's business? Did MIPS ever think about going into the systems building business?

Mashey: We were. We did.

Alan: I wasn't aware of that.

Mashey: Oh, yeah, sure.

House: Oh, yeah, because we had a workstation.

Mashey: No, we did—

Alan: And did Sun early on— I know you did later consider selling the SPARC on the merchant market? Later on.

Agrawal: Yeah, it wasn't until early '90s, or mid-'90s that's when Sun really thought about— that's when Sun broke into what we call this Five Planets. There was SunSoft and Sun Systems, Sun Microelectronics and 2 others. I had the charter to support the sales team in growing external business for SPARC chips.. Wasn't very successful.

House: We had debates at Intel. <static> We had debates at Intel in the '70s about whether we should go into the computer business. And I had a small under-the-cover project to make a personal computer before IBM ever made like a personal computer. We had a proposal, and I took it to the executive staff, and Gordon Moore, and I said, "We should make this small personal computer. Gordon had invested early on in a company called— the watch company—

Alan: Micromic.

House: Micromic. And he'd entered the end-user product business. And he had felt so burdened by that, he said, "No, we're never going to make a product that we sell to the consumer. No way that we're going to make a computer." And so there went my proposal down the drain. There were people later on in our systems division, who wanted to go into the computer business, and this was after IBM had come up with the PCS. He says, "No, no, no. We don't want to make those. We just want to take the profit from it. Just bring the profit in, guys. Turn them into distributors of our products, bring the profits to us. We don't need the business. We just need the profits."

Agrawal: Now...

Agrawal: There is that. So we did have some customers that were buying our chips and making computers out of them. But the difficulty was trying to design the system around the chips that we had designed for our own machines it was a difficult task. It was taking them too long. So then what we decided to work around that was and go into embedded boards business. There, we would design the entire board, which was not a Sun Workstation, which was not a server. And that became a huge business for Sun. Now, I started that business, by the time it was let it go into a different business unit, we were doing about $400 million starting from zero. So no, we didn't sell chips, but we did sell other things that were not Sun products, or branded as Sun products.

Mashey: So in our case, we shipped the first systems, 5-MIPS machines, in the Fall of 1986. They were more like development systems to make sure people could get going. But we also sold some of the boards from those to various vendors. Then we really got in more of an OEM systems model because there were a bunch of mini-computer vendors and mainframe vendors who needed something. And so we worked with Control Data, for instance and other folks. So, a reasonable sized chunk of the revenue was actually systems, because we didn't have the volume. And meanwhile, we would try to get the chips into other things. Of course, Silicon Graphics was a pretty early customer, and they wanted to do their own systems pretty quick.

<overlapping conversation>

Mashey: But they were pretty close, because the first SGI IRIX port was actually managed by me. They lent me a bunch of guys, and I took some of mine, and we did a port. And so, you know, there were different— there were all different combinations. We had the weird problem, I'd say, fairly— people remember Daisy systems?

Group: Oh, yeah.

Mashey: Okay?

House: Spinoff from Intel.

Mashey: Yeah, spinoff from Intel.

M3: Workstations.

Mashey: Yeah, they were a workstation company, and they used Intel. Everybody else was using Motorola 68000s and they were Big Endian. And Daisy said, "We're interested, but only if they're Little Endian." Tom Riordan said, "I can do that pretty easy, it's not that many gates." So we got Bi-Endian chips. Well, that meant we had every flavor of OS and compiler and everything you could imagine. And I'm not sure it was good. Turned out they didn't buy it after all.

Hollar: I'm going to go to Dick and Chris, and then I think we're going to have to wrap up, because it's getting overtime. Sorry, guys. Go ahead, Dick.

Dick: Okay, RISC started meaning reduced instruction set. Today's RISC is not a reduced instruction set. So when did RISC change? And how would you define RISC in today's environment.

Hollar: So just to sum that up so everyone can here, Dick was asking about RISC starting out as reduced instruction set. Today, not so much. How's it defined today? So go ahead, John, first.

Mashey: Well, I actually wrote a piece on, if people remember, Use Netcomp.arch. I kept telling people, "It's not the number of instructions. It has nothing to do with it. I did this big matrix of characteristics of different architectures. I said, "Look, you know, CISC is really a crazy acronym, because it's a whole bunch of different things that are completely different." I said, "It's probably a RISC if it has a fixed size of instruction, 32 bits, f it's a load-store architecture, if here's no indirect addressing and other funky gimmicks like the VAX. Motorola 6820, for instance. But not Intel.. And there were a bunch of other characteristics. And not everything was pure. I think the modern chips are still RISC even with a lot of extra instructions, because it's the form of the instructions and avoidance of some features on older machines. You would have a load, or an add instruction that would pick up an indirect address. And if the right bit were set, it would go to the next place, okay?" And those things were impossible to pipeline. And they're just drive you nuts. And the VAX was extreme. I'd sit with VAX designers and they'd say, "You RISC wimps! You have no idea what it's like to do a VAX!" And then they would say, "There's this 56-byte instruction that can make an X-number of different memory references if everything is wrong. We have to cater to that. And if it faults, we have to back everything up. You guys don't know anything!" I think the modern chips are still RISC

House: Well, Intel basically said, "We're going to do everything they did with the instructions that we execute the most. We're going to take our— there are few instructions in our about 80 or 90 percent of the execution, and we're going to make them single-cycle, and we're going to make them— implement them without microcode, etcetera, etcetera." And that's what the 4AM.

<overlapping conversation>

Mashey: You see, as ugly as some of that stuff was, you avoided indirect addressing and I actually went through this at one point. I said, "You avoided some of the things that are just real killers."

House: Must have been ____________. <laughter>

Mashey: No, no, it wasn't that. But for instance, Motorola went from the 6810 to the 6820. They added a bunch of real funky addressing modes, that are really ugly to make go fast.

Agrawal: Now, in answer to your question, I agree with John that RISC is RISC, and I've probably heard, if you talk about the complexity of the implementations. If you talk about extensions that RISC guys have put in, like the visual instruction set that Sun did, or instructions put in for encryption, or for accelerating some other functions. Yeah, it is creeping up. But if you keep the number of instructions aside, and if you look at the sheer complexity, the number of cores, the different caches, you know, hierarchy of caches, different kinds of IO, cache coherency, I mean, I couldn't tell a difference between a RISC and CISC chip as with the die size of RISC is smaller. Heck no!

House: ____________ kind of blended.

<overlapping conversation>

Hollar: Chris:

Chris: When this all going over like this, I have heard 68000, and other— and I haven't heard from any kind of members there was any of the architecture going to 32000 CDs of the National Semiconductor. What happened to them? What do you think about that?

Hollar: Did everyone hear that? What happened to the 32000 series from National Semi?

House: Sequent [ph?] went out of business.

Mashey: So now I'll wear my UNIX guy hat, all right? All the UNIX guys thought that National Semi 32000 was really pretty cool. We thought it was pretty elegant. It felt like a cut-down VAX.. And people were from VAX UNIX at that point, right? If you look at the MIPS design, I was ruthless in what the memory management unit was allowed to do. I didn't want the hardware guys to put in stuff they normally did. They said, "OH, you'll need this." I said: "No, I won't. We always work around that. Don't put that in" The 68000, the 6810s, National Semi chips, but especially those, had weird problems with exceptions and memory management. And it's the kind of stuff that ages an OS guy. I mean, I'm only 30, you know? <laughter> Because here's what happens: you make something work, and then systems crash, and it's because of some weird special case that only happens every once in a while. You spend all night, trying to figure this out. And there was just enough of that in the national chips that people tried— really good OS people with system people try to make them work. The bugs were in these ugly nasty weird things.

Hollar: Anant, Dave, you guys have some thoughts on that?

House: I don't think that National had as complete of a solution. The thing that MIPS and Sun were able to do to Intel was to have more complete solution. You had UNIX architects, and you had compiler and generators and people at top end, and, you know, Intel as big as it was, struggled so much on that end. And I think companies like Motorola, I talked to the Motorola team leaders, and I talked to the National Team leaders on the 32032 that had— or the Computer History Museum, we've done over two dozen video histories of the teams that designed the microprocessors, because the analog chip might be designed by one person. And the memory might be designed by one person, but microprocessors are done by teams. So we've interviewed the teams, and I've interviewed the 32032 team, and the 68020 team and the 68000 team. What they said is, "We just didn't have the support from the corporation to develop the rest of the story." We had an excellent processor, but we just couldn't put the rest of it together.

M3: Let me ask you that. I was on the 32016M instead of 32032.

House: Yeah?

M3: There were a couple of problems. And one is the core team itself was headed by somebody from Israeli. And the chip, of course, when it came out had a lot of bugs in it. And exactly what Dave said, we could not get enough people in the industry, in the core system, to actually work on the compilers and so on and so forth. So I was at TI, we were the second source. And we found out TI and National were competing for a four percent market share at that time. It was just not working out.

House: ____________, you were involved at that time.

M4: Well, I have a little bit different perspective on some of these things, because I actually worked on the SPARC, I worked for Intel for ten years, and I also worked at MIPS. <laughter> So I've seen all these things go round and whatnot. But the one thing that Dave mentioned earlier that I think we all still miss, Intel had field application engineers who were trained, who knew what they were doing. By the time you got to second generation, we had specialists in operating systems and software and compilers, and they made a huge different slice. Nobody else in the industry has ever met that criteria.

Hollar: Please join me in thanking John, Anant and Dave House.

<applause>

END OF PANEL

