Oral History of John Hennessy

Interviewed by:
John Mashey

Recorded: September 20, 2007
Mountain View, California

CHM Reference number: X4149.2008

John Mashey: This is John Mashey, a trustee of the [Computer History] Museum. I am greatly pleased to be able to do an oral history with Professor John Hennessy, President of Stanford. So, John, let's go all the way back. Where did you grow up? When did you think you were going to get into computing? How did all that happen?

John Hennessy: I was born in New York City but I grew up mostly on Long Island. My father was an engineer, worked in the aerospace industry. [He] probably got me interested in the early days in computing and I think probably what really got me going was the opportunity to do a little programming on a timeshare machine using good old paper tape to store our programs. And then in high school with a buddy of mine we undertook a high school science project which was to do some experiments, including design of a tic-tac-toe machine, which we built out of surplus relays because they were the cheapest thing we could find and having no money and doing this ourselves. We built it out of surplus relays, put it in a plywood box, and then covered it with black contact paper which is really the trick to make it look neat but it was wonderful. The amount I didn't know when I started that project was amazing but we figured out a lot of things about how to build a decision tree for tic-tac-toe, not very hard. But in those times stunning to people that a machine could beat them at tic-tac-toe.

Mashey: Okay, so that's how you got started, all right.

Hennessy: That got me started.

Mashey: Okay, so I guess when did you decide you were going to be in computing?

Hennessy: Those days you couldn't major in computer science.

Mashey: Yes, yes.

Hennessy: I started off in electrical engineering but then I already had this computing interest and it continued to build when I took my first programming course which I was in the advanced section because I had already had a little Fortran programming in high school.

Mashey: So this was at Villanova [University]?

Hennessy: That was at Villanova where I was an undergraduate and that continued to build my interest I think and I had the opportunity to work on a research project. There was a guy working on his Master's thesis building a self micro programming machine, a micro programming machine where you could download the microcode yourself out of the bit slice processors that were available then. So I got to work on that project during the academic year and a little bit during the summer, so that got me more intrigued with computer science. I decided to pursue my graduate work in computer science at that time.

Mashey: Okay, and so then that took you over to Stony Brook [State University of New York] ?.

Hennessy: That took me to Stony Brook. I finished my undergraduate degree in three and a half years. It turns out my then girlfriend but now wife of 30-some odd years we had been split off. We had met each other in senior year in high school so we were headed to different schools. We were eager to kind of get back and converge and she was a teacher so she was going to be teaching somewhere in the New York area. Finishing in the middle of the year I had to then find an institution that would accept me to a graduate program in the middle of the year somewhere in the New York vicinity so that took me to Stony Brook.

Mashey: Ah, okay. All right, so you're another one with a high school sweetheart turned wife.

Hennessy: Yes. Dave [Patterson] married his early. I waited until we were done with college but then we got married four years after in 1974. Four years after we started as undergraduates.

Mashey: This seems to be a common thread, what can I say here?

Hennessy: It's a good thing.

Mashey: At least it keeps life stable.

Hennessy: It keeps life stable in one part of it so you can pursue other things.

Mashey: So then what was your thesis work in Stony Brook?

Hennessy: My thesis work was on real time programming. I was extremely fortunate. Shortly after I started in my graduate work a guy who was a researcher at Brookhaven National Lab came along with an interesting problem. He was trying to build a device that would do bone density measurements to check for long term, low level radiation exposure among people who were working at Brookhaven. What he wanted to do was build a very finely controlled x-ray scanning device that would scan x-ray across an arm or so and get a bone density measurement. The problem is you had to control it pretty accurately because, of course, you don't want to increase the radiation exposure to people. And micro processors at that time were extremely primitive so the real question is could we build a language, a programming language that could do this kind of real time control in which you could express the real time constraints. You could then use compiler technology to ensure that they were satisfied. I was lucky, I started on that research and as I was finishing my Ph.D. thesis lots of people decided this was an interesting area in which to work and here I was working right in the middle of the area so the timing was particularly fortunate.

Mashey: So then you got recruited to come out to Stanford [University]. Talk about how did you end up here and who recruited you? How did that all happen? That's a long way.

Hennessy: It was fascinating. It was a long way. Actually, Stanford was the 14th institution, university I interviewed at. I decided I wanted to be an academic from the very beginning so I interviewed only at universities and I started interviewing all across the country and I think my first interview was at the University of Iowa. I had an offer early on from Wisconsin that was a really interesting offer because, of course, great computer science department. And then I came out to California and my wife liked the sunshine and we sort of liked the place. I came out and I interviewed and Mike Flynn and Ed McCluskey were recruiting me but I remember Don Knuth was on my interview schedule and, of course, for a young computer scientist meeting Don Knuth is a fascinating experience and I still remember what I asked him. I asked him how had he gotten so much done? And he said, “Well, I don’t watch a lot of TV.” I said, “Well, neither do I.” So he said, “And I love what I do. My work is my hobby.” and it was an inspirational interview. And at that time, the computer science department was fairly small at Stanford. It was in humanities and sciences and it couldn’t grow very much because humanities and sciences didn’t have very many resources. So electrical engineering was growing and adding computing people. Susan Wickey [???] had come the year before I had come and so they were trying to recruit and build out that department and I got the offer from them and accepted and been there ever since.

Mashey: Interesting I guess it’s a long way, yes.

Hennessy: It is a long way but it was one of these… it was a dream come true. You get an offer from one of the top computer science departments in the country. It’s a real opportunity.

Mashey: Okay, so then so you got out here and sort of talk about your research areas and what you were doing for the next few years. When was that?

Hennessy: That was in 1977 and I was doing a combination. My thesis was really a combination of language design and compiler technology so I started working in that general area. At that time, the S-1 project was going on at Lawrence Livermore to try to build a high performance computer and so I got involved in the compiler technology for that and I took over a piece of that and began working in the compiler optimization area which is how I got to meet Fred Chow and we did the work on register allocation and developing the U-code concept building on the old P-code intermediate form but really trying to experiment with this notion of a single optimization component working on a common intermediate format and what did the intermediate format need to have in it in order to make that work? So I got working on that for a while and did some work on that. Did some work on language extension, trying to think about how to make Pascal a more useful rather than just a tutorial language. Could we really make it a programming language that people could use because obviously all the people who were doing serious systems programming then were using C? And could we take advantage of some of the things in Pascal and make it useful?

I worked on that for a while and then Jim Clark was starting up what would become the geometry engine project and one of the challenges in that project was there was a lot of microcode to be written to do most of the geometric transformations, implement the floating point technology and he was interested in getting a tool that would help him write that microcode at some higher level and then compile down into what then were PLAs [Programmable Logic Arrays] as the implementation technology. So I took on that project and created a programming language called Slim and wrote a system for writing the microcode and that got me interested in doing a lot more in hardware design. It was a pipelined engine so we had to build a high level language that would make it easy to do the pipelining properties that the internal microcode engine could exploit. So I worked on that. It was a great project, got me much more interested in VLSI [Very Large Scale Integration] technology and exploiting that.

And along about the same time after Carver Mead had seen the presentation that I gave at the Cal Tech VLSI conference on the topic, he asked me to come over and possibly start consulting for Silicon Compilers which, of course, was in this business of building up tools to generate chips. So I spent some time with them. I actually worked on what became the MicroVAX I chip set with Dave Cutler at DEC [Digital Equipment Corporation] and a group of other people working on the project and Ed Chang who was then at Silicon Compilers. And that really helped me learn a lot about, not only about the VAX architecture but about microcoded engines and other aspects of that and got me more into computer architecture. And so, after we had finished the geometry engine project and Jim was starting up Silicon Graphics, which I then ended up consulting for, I continued to get interested in architecture and we started basically what would become the MIPS [Microprocessor without Interlocked Pipeline Stages] project by asking ourselves the following question. Well, we have some expertise in compiler technology. We have some expertise in VLSI. It seemed to us that the fact that processors were going to be single chip entities was a foregone conclusion.

We couldn't have predicted and I think nobody really predicted how the microprocessor would take over the whole computer industry. That was not quite as well understood then as it is now so obvious to everybody but the beginnings were there. We certainly saw how microprocessor based machines could be competitive certainly with mini-computers that we were then paying hundreds of thousands of dollars for. So we started what would become the MIPS project with a class basically that was a brainstorming class where we just brought all the students in and said, "Well what do we know?" And I think the first real concrete insight we had which I gathered partly from my experience on working on the MicroVAX I and other projects was that using the compiler to try to eliminate the level of interpretation that would normally go on in a microcoded engine, say interpreting the VAX instruction set, was much more efficient than trying to do that interpretation dynamically. Why break a complex instruction down into lots of small instructions at run time when I can do that work once in compiled time? And that was really the inspiration for the MIPS project. We also, having worked on the register allocation work, knew a lot about register allocation and how to make it work. So having a machine that was register-to-register oriented, lots of significant number of general purpose registers that could be allocated by the compiler we understood how to make that an efficient target for a high level language and that was where the inspiration work on the MIPS processor came along.

Mashey: So how many students were in that class and have any of them gone on to do interesting things?

Hennessy: Oh, sure. There were about… initially there were probably about ten or twelve students in the class of which four or five of them became the core of the project. Thomas Gross, who did the work on eliminating, doing the software to eliminate pipeline interlocks, for example, really optimizing for a pipeline, scheduling for a pipeline machine went to Carnegie Mellon and now is at E. T. Haut [???] in Switzerland. Chris Rowan, who was at MIPS with us for a while and then was the founder and CEO of Tensilica. Norm Jupee [???], who is at HP labs and continues to do a lot of interesting work at DEC World on computer architecture. And Steve Shabilsky [???], who's an independent consultant but came with us at MIPS for a number of years. And then later, people who joined the team,Anon Tagger [???] joined the team later on and really helped implement a lot of the system stuff for it so it was a good group and Forest Basket of course was there. He was sort of my mentor in those early days and was the principal investigator for the overall VLSI project so he was involved with it. It was a good project. It's amazing when you think… I mean we had a team which by today's standards for what you need to implement a piece of silicon was tiny but we were naïve, willing to work hard and try some things that were a little outrageous.

Mashey: Well it actually seems when I watch this from outside that there are periods in computer history where, in fact, small university projects can actually build pretty interesting stuff and then there are other periods where it's just really hard, huh?

Hennessy: There was a real discontinuity. Basically the opportunity to build a single chip processor computer created a discontinuity. If you looked at what the Intel-Zilog war that was going on then, it was who can put in more instructions than the other guy? They were oriented in the instruction sets to try to empower assembly language programmers. The Motorola 68000 was out there but it was really inspired by the [DEC] PDP-11. It was a PDP-11 like architecture, a lot cleaner than the Intel architecture no doubt, but inspired by that. We thought, "Let's take a clean sheet of paper. It's not so obvious what's been done in the past is the right thing to do" and that's I think where universities really play a unique role being willing to do that and not be bound by what's been done in the past.

Mashey: So now right about that same time your buddy Dave Patterson was over at University of California,] Berkeley doing stuff so to what extent were you guys interacting at that point?

Hennessy: We were. We were. We certainly had seen the RISC [Reduced Instruction Set Computer] papers. We were certainly driven by this notion that if we were going to fit this processor on a single piece of silicon that we had to simplify the instruction set. I think we were obviously much bigger believers in compiler technology, but I think we all believed that simplicity would have value in terms of the overall architecture and performance. Although I think part of our difficulty and part of our difficulty that we all had in selling the RISC ideas was that we couldn't say why. We couldn't give a good… It wasn't until several years later when actually it was in the early days of MIPS, the company, and when the landmark paper by Doug Clark and his colleagues at DEC was published that showed that, in fact, the VAX was taking ten cycles on average to do an instruction. Then we came up with a classical explanation of the reduction in CPI [Cycles Per Instruction] exceeding the increase in instruction count. Had we had that earlier I'm convinced we could have sold the concept, although certainly others would have argued, "Well, we can implement a low CPI version of the VAX if we really wanted to." Although in retrospect they were never quite able to do it, right? And we did interact. We had a few group meetings where we had their students down to Stanford and we took our students up to Berkeley, so it was a good time for interacting. That was a very interesting time because the community that got built and, of course, John Toole was very instrumental in building that community. Duane Adams before that. The community was a highly symbiotic community so the rate of progress of the research was enhanced by the fact that Stanford, Berkeley, Carnegie Mellon, MIT and a host of other schools were working in similar areas. They were sharing tools. They were sharing ideas and it actually meant that the rate of progress was absolutely phenomenal and I think that's one of the things that leadership at DARPA [Defense Advanced Research Projects Agency] did really wonderful in creating that community and enabling it to advance at a very rapid rate because of that.

Mashey: So talk some more about that actually while you're on DARPA.

Hennessy: Yes.

Mashey: So what exactly was their role in this? I know there were some grants, right?

Hennessy: They funded the early VLSI work. They encouraged us first when Duane Adams was running the activity and then subsequently when John Toole was. They certainly understood the importance of building tools but they also encouraged us to think about this as a new media, as a new opportunity to experiment and rethink questions, of course first at Stanford with Jim Clark's rethinking of how to do high performance graphics in a much more cost effective way but then on the MPS project and subsequent projects. So it was that enabling of some out of the box thinking I think that was really crucial and encouraging us to think out of the box.

Mashey: Actually it is an interesting thing and maybe at some point we should address this. Once upon a time we had places like Bell Laboratories, where I was, right, where a huge amount of industrial R&D was going on and it seems that's shifted a bunch to a bunch of that work seems to be done in universities these days.

Hennessy: Yes. It's I think the great national assets that we had in Bell Labs, Xerox PARC [Palo Alto Research Center], even IBM Research as it was then which I think was less aimed at next generation products than one might argue it is today and I think even the IBM people would admit it is today. Those were national assets, of course, each dependent on its own peculiar monopoly, whether it be the phone system, copiers, or mainframes. And I think at that time they enabled so much of the work. That basically doesn't go on today… a little bit at Microsoft perhaps, a little bit here and there, but not at the scale, not with the open ended kind of thinking that went on. And you look at how many great people went to Bell Labs. I mean it was considered a great place for graduates to go and work and certainly as attractive: Bell Labs, Xerox Park, and IBM were as attractive as any academic position in terms of the freedom you got, the resources you got. That's gone and I think gone, oh gone because the industry has become so much more competitive and there's much more emphasis on quarter-to-quarter results and I think there's another interesting phenomena that we've discovered. Great breakthroughs in industry often become public goods. Unix became a public good.

Mashey: Yes.

Hennessy: The concept of the personal computer that Xerox PARC piloted became a public good. Lots of the basic technology that IBM invented in the computing industry became public goods and it's not so clear. If you're going to invent public goods, maybe it does belong in the university, after all.

Mashey: That certainly has been an interesting change to watch.

Hennessy: It has been an interesting change and I think it's been important for the universities to understand that they now carry this role going forward.

Mashey: But actually that leads to another question. You certainly have had industry connections from very early. How did that get started? Talk some about the mixture between industry and academe and what good interactions are because, again, you have some long history with that.

Hennessy: Yes. Well I started actually-- there's always been a belief at Stanford I think dating back to the days of Fred Terman, the legendary dean of engineering who was the person that put Hewlett and Packard together, that that interaction, whether it be through consulting or some other role, was a valuable thing, added to the not only enabled the university to have some greater impact but also added to the understanding and insight that faculty had as they became teachers and I think that's certainly true. If you think about the students in a typical classroom, most of them aren't going on to be academics. Even in a graduate program most of them aren't going on to be academics. Most of them are going on to work in the industry and the more experience I think that a faculty member has in that the better they are. They're certainly going to teach principles and fundamentals still because those are the things that have lasting value, but that experience and understanding how to apply those things can often be incredibly valuable for an instructor to have experience with.

So I started actually doing some early consulting and then as I mentioned my involvement in Silicon Compilers was really my first and that was an interesting company. John Dorr was the primary funder of the company so it was a real Silicon Valley phenomenon with Carver Mead, John Dorr, Dave Johannsen, the guy who had done the OM2 chip, OM1, OM2 chip at Cal Tech. So a very interesting group of people and then after that, spending a couple years at Silicon Graphics. Then the way MIPS got started was we were working on this great technology. The people at SGI said, "Gees, it should be great if you could make a company out of this." And our old friend Gordon Bell was probably the number one instigator of that. We had basically finished our project, published our papers and, like any good hopeful academic assumed people in industry would just read these papers and would just go do it because it was so obvious that this was a good idea. That was naïve, to say the least, I think and we saw subsequently even the difficulty of getting in the early days Digital Equipment Corporation to accept the results of their own Titan experiment done unfortunately in Palo Alto, rather than in Massachusetts, which made it hard to sell. So by that time John Masoris [???], one of my co-founders at MIPS, had taken a sabbatical from IBM and come out and was working in our lab. We were talking about getting this technology out there. And then Gordon Bell came along and said, "You guys really need to start a company to get this out and it's really important and if you don't invest in it yourself and put yourself into it and get something started, it's just going to sit on the shelf." So Gordon talked to us about a variety of mechanisms, including funding it through a company he was getting going and actually suggested Skip Stritter as somebody we approach, my other co-founder there at MIPS.

And so we eventually put it all together and sold it to the venture capitalists with a business plan that I can only describe as naïve and embarrassing: basically a set of PowerPoint, not PowerPoint, hand drawn slides because there was no PowerPoint then, hand drawn slides that basically said how long it would take us to do the design, how much money we needed, nothing about analysis of the market, how we would sell this, gross margin, profit, return, none of that, simply this technology was so revolutionary that it was going to sweep over the industry eventually. And we didn't understand how to do the business part of it so they should help us with that and we would do the technology. And we found Mayfield [Fund] would fund us. Not many other firms would fund us but they would.

Mashey: Mayfield had funded Silicon Graphics.

Hennessy: They had funded Silicon Graphics and that's how we had a connection. We had a connection to them. But lots of other people turned us down either because of a conflict or for some other reason. I had, and I was quite open with the investors about it, I had decided that I was going to take a leave of absence from Stanford but that I was going to go back and people were nervous about that obviously, rightly so. We didn't have a CEO or a management team. It was really a technical founding team. They were nervous about all those things.

Mashey: But Mayfield funded you anyway?

Hennessy: They funded us anyway, right.

Mashey: So actually one quick sort of back step so I think you were running the [Stanford University] Computer Systems Lab is that right?

Hennessy: I was.

Mashey: And what was that?

Hennessy: We had, actually Mike Flynn and Ed McCluskey when they were at Stanford because we had separate electrical engineering and computer science departments and, in fact, they were in different schools, decided we really needed a laboratory to kind of bridge that. That would be a home for the people who were doing experimental computer engineering and computer science. Originally it was called the Digital Systems Lab and it was sort of a hardware focus, but over time it expanded to include software things and we renamed it the Computer Systems Lab and it still exists today. It's probably one of Stanford's most successful interdepartmental laboratories and the home of not only many great inventions but lots of great entrepreneurial activities as well. So that was the home. That was the lab that Forrest was in, Vint Cerf was in and then I came into and then after I received tenure I headed it up for a while. And we still hire a lot of faculty. I mean it's amazing how many joint appointments we've made over the years and it's really helped us create a culture that we can take students who come either through [the] electrical engineering Ph.D. program or the computer science Ph.D. program and when they come out the other side you'd have a hard time guessing whether they were originally electrical engineers or computer scientists.

Mashey: I think that interdisciplinary thing always seems to take work to make happen in universities and I think we'll want to come back to that later.

Hennessy: Yes, absolutely.

Massey: Okay, so MIPS, so you've described the technical founding team, so how did it go from then? How did you get hooked up with a fellow named Steve Black?

Hennessy: Oh, Steve, yes. So Steve was… we were looking around for somebody with some marketing expertise and Skip knew Steve from something else and so we started talking to Steve. He had a lot of crazy ideas. Steve is a guy who has more ideas than he can write down or get out.

Mashey: Yes.

Hennessy: But that was okay. We were all a little hyper in our own way so it fit in with the rest of the group. So we started talking to Steve about being at least in a consulting role in this and being involved with this. And Steve came onboard and headed that. Steve's really kind of more a marketing guy. At that time in the company we didn't separate out marketing and sales, so he kind of had to do both of it and, in fact, I did my first cold calls with Steve to basically his former customer list from Convergys.

Mashey: Yes.

Hennessy: And so we went to visit those people in all these places, Dayton, Ohio and places you don't want to go. It was an interesting experience for me because it really… Boy did I learn an appreciation of what it takes to be a good salesperson and an appreciation for the hard work that salespeople do in that job.

Mashey: Yes, well Steve has a long history of dragging technical people around to customers.

Hennessy: Yes, and that's what he did. And that's what he did and basically I went in and kind of gave the technical spiel and said, "You know this kind of technology is going to provide new opportunities for the companies that embrace it because it's going to be so much more cost effective and higher performance than other things out there." So it was a fun, yes, it was a fun time. Then we started to build the team. I mentioned Chris Rowan and Steve Shabilsky came to join the VLSI team and then we felt we really needed to hire probably one of the most important things I learned in that, hire great people from the beginning. Hire the very best people. So we said software is more important here than it would be in a traditional machine because it has to mesh well with the hardware and we're doing some new things in hardware. We need people who are creative. So we went and hired Larry Weber to head the compiler team and a guy named John Mashey to head the OS [Operating System] part. So we had a really terrific group of people and I think all characterized by the fact that they were willing to rethink things that had become conventional wisdom in the industry, whether it was in the OS side or the compiler side.

Mashey: Of course, in a few of our cases there was a predisposition to think this way. By amusing coincidence back at Bell Labs years before I had looked at a proposal from this guy named Dave Ditzel, right?

Hennessy: Oh, yes.

Mashey: Who was proposing a fairly RISC type chip to replace a fairly CISC [Complex Instruction Set Computer] type chip that we were doing internally.

Hennessy: Interesting.

Mashey: Bell Mac chips.

Hennessy: Oh, yes, the Bell Mac chips. You know I think it's interesting that everybody was involved in the early days of the RISC work, Patterson, myself, John Koch all had some experience with some CISC machine whether it was the VAX, the work we were doing on the MicroVAX in my consulting role or Dave's work and Ditzel's work on Symbol and John Koch's work on ACS. I mean they all had this experience with a machine that was quite a bit more complicated, tried to do quite a bit more in hardware, and concluded that that wasn't the way to do things.

Mashey: Yes. So anyway certainly there was a predisposition among some people that when something actually looked like it might work to be interested.

Hennessy: Yes.

Mashey: Okay, so let's see. As I recall in 1985 there was an incredibly ambitious schedule to design a whole lot of software and chips and get it all ready, so talk about that year.

Hennessy: Yes. One of our early calls was to Prime Computer and Prime had a problem that it still had good software but its hardware was no longer competitive and it needed a plan to kind of leapfrog rather than just kind of catch up because if it just caught up it would find itself still at a disadvantage against its other competitors. So probably only less than six months after the company started we got an MOU [Memorandum of Understanding] from Prime to deliver them basically a board for their new computer. There was a tremendous amount to do because, of course, a university project is one thing but a real product is a completely different thing. In doing the university project, we had cut all kinds of corners because that's what universities should do.

Mashey: Sure, sure.

Hennessy: They shouldn't try to… and we didn't really have an OS. We didn't have… we had a compiler that limped but wouldn't compile big programs so there was a tremendous amount of engineering work to do while we took some inspiration from the Stanford MIPS project. We basically redesigned the architecture. We had to implement the chip. We had to find a source of manufacturing source for that chip early on. Obviously build an operating system for it. We were trying some new things in terms of how we did memory management as well, so there was a lot of thinking that had to go in there and collaboration between the architecture team and the OS team. But we managed to pull it altogether and we had a delivery date at the end of 1985 from Prime to deliver them an engineering sample unit, not completely. And so I remember still Todd Baisch [???] agreed to fly on Christmas to deliver this thing to

Prime so we would make the end of the year. And he gets on the plane. He flies out there. He shows up the day after Christmas and Prime is closed for the entire holiday. But he left a note and we made that delivery which was absolutely amazing because if you look at how much engineering had to be done. I think it really shows you what a really dedicated smaller team can do and everybody jumped into the trench and did whatever was necessary to make it work. Near the end of the project I started writing test code because we were worried about, we were covering all the test cases that we needed to cover, so I think that kind of… That was a remarkable opportunity to get that done.

Mashey: Well clearly it was an insane schedule.

Hennessy: It was an insane schedule, yes.

Mashey: So actually talk a little bit about management in the sequence that went on and how you decided to do that and how that happened.

Hennessy: Yes, well obviously I was and the rest of I think the founders were relatively naïve about building out a management team, both about general management in terms of CEO and in terms of sales and marketing and really understanding how to manage and build out a sales and marketing organization. So we took some guesses. I mean we really did. It was really experimenting. We got an initial CEO who I think was good for getting the company initially going, getting us funded, negotiating some of our arrangements, but probably who wasn't going to scale well with the company as it grew. So we had to kind of make the decision when was it time to make that change? I didn't understand even what to look for in a head of sales and marketing. I didn't even know what questions to ask the person when I was interviewing them. So I think for us it was hard to get the right team aligned and we had a technical team that was absolutely world class, as good pound for pound as any team that's ever been put together so for me that was a real lesson about the importance of building out the other team and supporting the engineering team. So we had to make a lot of changes in those early days as that team developed and stumbled along its way.

Mashey: So MIPS was able to deliver something at the end of that year?

Hennessy: Yes.

Mashey: And then, of course, there was still a lot more software work to do and things to get to be worked out.

Hennessy: There was a lot more to do to build a product out and I think one of things we discovered, and this is an interesting insight into how the business model had to shift over time. And one of the difficulties that we, I think discovered, was to really deliver and demonstrate this technology. You had to build an engineering team that could build, essentially, an entire computer system. Even if you weren't selling somebody boxes, you had to build all that technology to ensure that whatever you were selling them would really work. And so, I think one of the challenges we found was, we had to have an engineering team that was comparable in size to what a computer system's house would have, even if in the end we were selling compiler technology and boards with CPUs on them. And figuring out how to structure a business model that could support that was the real challenge we had.

Mashey: And certainly there were some interesting other challenges. I remember that I thought one of the big helps that we had was we started with some pretty good compiler technology. What's also the case, there were interesting interactions between good optimizing compilers and C and operating systems.

Hennessy: Yes, sure. Sure, I think figuring out now that was a time when compilers would get a lot more sophisticated, would move code a lot more than people might expect it to get moved. And one of the early notions behind C, which is that, when you wrote a line of code you could kind of guess what the instructions were going to look like. Well, that was no longer true because the compiler would have done all kinds of interesting things, including promoted variables into registers that, in fact, were not really just values, but actually memory locations that needed to be touched under certain circumstances. And obviously, when you start doing aggressive scheduling for pipeline, the sequence of instructions gets mangled significantly. And this was the early days of doing this because compilers had not been nearly so aggressive about reordering code in earlier versions. So we had to kind of figure that all out and learn by doing, I think.

Mashey: That was one where the volatile attribute of C came along just about the right time to have a standard way to talk about it correctly. But that wasn't before some of us had to do binary search on the Unix kernel, where it worked perfectly unoptimized, but when you turned the global optimizer on it broke because one instruction had been removed.

Hennessy: Yes sure, absolutely.

Mashey: Exciting times. So by the middle of 1986, MIPS actually shipped some early machines.

Hennessy: Yes.

Mashey: I remember we had plenty of surprises.

Hennessy: Yes, we did. We did and things, as you can always guess, when you ship early machines, not everything works out exactly as you thought and you're figuring that out and… but, and while we still had a lot of challenges, I think at least we were moving some hardware out and getting some things done.

Mashey: Now let's see, by this time, you were actually back at Stanford?

Hennessy: I was actually back at Stanford consulting a day a week and about half-time during summer probably, yes.

Mashey: So you were back running CSL [Stanford Computer Systems Laboratory] again?

Hennessy: Right, I was back running CSL and getting some new research started.

Mashey: So talk about where that research was going and where you were going from then.

Hennessy: Well, after the publication and the first RISC articles, I think the conventional wisdom in the industry was "Well, these RISC ideas are great but they won't really work when you have to do real floating point and incorporate that in the design. Or they won't really work when you have to do virtual memory or lots of other things of this form." So we wanted to work out how to approach some of those things and think creatively about how to address those issues. And then we began getting interested, sort of in the… a little later in the 1980’s in thinking about parallelism and multiprocessors, because we really became convinced that uni-processors were going to be dominated by microprocessor technology. It was far and away the most cost-effective way to build a machine. So it seemed quite natural to think that the most cost-effective way to build a big machine was to build a multiprocessor. And so, like the people at Berkeley and several other groups around the country, we began thinking about those things.

Mashey: So then that led to various projects at Stanford, the multiprocessor?

Hennessy: Yes. That led eventually in a similar model to what we had followed with the MIPS project, we started up another brainstorming class and we said, "Let's think about this." At that time, the conventional wisdom, based on a number of papers that have been published, was that you couldn't have scalability unless you were willing to give up on cache coherency. Which basically meant once you gave up on cache coherency, that the placement of objects in memory was going to need to become visible to the programmer. IBM had built the RP3 [Research Parallel Processing Prototype]. Obviously, long before that, Carnegie Mellon had built the CM Star. But both machines were fairly tricky to program because of this worrying a lot about where memory was and if you wanted to operate on some particular data object, you might actually contemplate migrating it to the local memory. But, of course, that wasn't transparent. That had to be done by writing code. So, there was a rapidly building consensus that if you wanted to build a scalable parallel machine, you would have to go to message passing. First of all, we didn't necessarily believe that was the case (A), (B) I also saw that this could create a very difficult dynamic in the industry, where smaller scale multiprocessors would use cache coherency and a shared memory model, hence. Larger scale machines would use a message passing non-shared memory model and we'd have a tremendous breakdown, in terms of the programming model. And I was worried, I think, in retrospect correctly so, that this would lead to fragmentation in the parallel processing market and thereby, slowdown the adoption and creation of software. And I think, in retrospect, that's what's really happened. I mean, people have avoided… I mean, we now find ourselves in this place where all these companies are shipping machines that have multiple cores in them, and we are so far behind on the software side still. And I think we obviously have to do some catch-up, but in some sense, the split between the very high end of the market that was willing to go after the Bell prize and do all kinds of things, where it probably didn't matter if it was too difficult if it was hard to program because you had PhDs doing the programming anyway and you really cared about one application running very fast. That split from that and the mainstream of the market which relies on lots of shrink wrapped software vendors just made it very difficult. So we just started with "Could we build a machine that would maintain a shared memory cache coherent model?" And began thinking about ways to do that and that led to the so-called

DASH machine to a model that had cache coherency, but a memory model that was distributed so the memory was distributed around the machine but objects could migrate to caches easily. And we, after thinking through lots of alternatives, settled on a directory-like structure to maintain the coherency.

Mashey: What did you actually build?

Hennessy: We actually built an implementation of that machine basically using the small scale multiprocessor boards from the SGI [Silicon Graphics, Inc.] machine, interfacing into their bus and then implementing the directory scheme in a set of programmable logic that would then go onto that board and so we could turn it into a, basically turn it into an extended, extensible machine and have it implement the cache coherency scheme that way.

Mashey: So maybe talk a little bit about where that research ended up going in relationship with SGI's later work and students involved and that sort of thing.

Hennessy: Yes. Well, there were two guys who were the early guys on that team- on the DASH team and they were Dan Lenoski and Jim Laudon. And obviously, by then Silicon Graphics was getting interested in larger scale parallel machines. They had built a very large, in the power series, a very large, aggressive, bus-based cache coherency machine. But we knew buses, if you built a bus-based cache coherent machine it wouldn't scale because the fundamental algorithm being used for coherency relied on broadcast. And if you use a broadcast algorithm, I don't care, you can build another bus, but you're not really going to get a scaleable scheme to it. And so in the power series, they had more or less basically, pushed the bus as far as it was going to go. So, we started talking to them about building something based on the DASH concepts and that's when Dan finished his PhD and went to work down at SGI on what would become the origin design which improved on the implementation that we used in DASH since we had done it with a lot of programmable logic and hacked into an existing architecture to do it. They redesigned it from scratch and that became the origin series after that.

And then we went on to, in our research, to work on a machine called Flash. You know, second generation machines, even in academia, have a bit of the second generation syndrome. They tend to pile on the features. We started, actually… the start of our project, was a collaboration with Intel to build a machine that could do both a scalable shared memory protocol, and hardware-supported message passing. So the concept was to design, basically a communications processor that could do both well. Handle the cache coherency stuff and do very efficient message passing. And that became the cornerstone of the Flash design. And then Intel decided to get out of the high end computing business, probably about two years into our project, and we then had to figure out could we implement this thing in somebody's gate array technology, and we teamed up with LSI Logic and used their gate array technology. But we were going to do a semi-custom chip and all of a sudden you're in a gate array and we couldn't do all the… we had to redesign the whole thing and it cost of a lot of delay in the project. But we got it done anyway and it was certainly an interesting project and we learned a lot.

I think one of the things we learned a lot about software in that because, like our earlier projects, software was a big component of what we were working on. Mendel Rosenblum did the Sim-OS work at that time so that we could really understand the interaction between the operating system and the hardware architecture. Of course, then, that eventually led to VMware and the VMware technology. We also, at that time, began doing a lot of work on really trying to understand parallel programs and how they behave. And, I think, we were, and we still are in some areas, in a position of knowing relatively little about how parallel software behaves, compared to conventional sequential software. And when we did MIPS, we knew a lot about what instruction mixes would look like and these sorts of things, which enable you to optimize that. The difficulty in the parallel processing world is, it's not instruction mix you care about, it's things like how often are references to objects that are local versus remote, how much spatial locality is there? How much temporal locality is there for data objects that are shared versus data objects that are not shared. So you care about these much higher level properties and I think they've proven fairly difficult to characterize. They vary a lot between scientific programs and more systems oriented code. And that's made it much harder, I think, to say "Here is how to design the right kind of parallel machine for a wide variety of different workloads," and something I think we're still struggling with today.

Mashey: Back up a little bit back into the MIPS thread. As I recall, you kept consulting for MIPS right through the 1980's. As I recall, one of the interesting things that went on was the decision to make the MIPS R4000, a 64-bit processor, which I recall, you had some hand in.

Hennessy: Yes. Well, I think we made that decision. We saw that 64-bits was going to come and that we were going to run out of 32-bits of address space. And with the conversion and the drop in DRAM [Dynamic Random Access Memory] cost, that actually accelerated. And I think, talking to the entire team, obviously the OS people, probably the most acutely aware of how difficult the transition would be from 32 to 64-bits, and how long it would take once the hardware was actually available, and the same thing on the compiler side. As you began to understand this long chain between new hardware available and people actually using all the new software, the OS works. I mean, we saw this time and time again. We even saw it occur with DEC [Digital Equipment Corporation] as well. How long between 64-bit Alpha and actually 64-bit versions of the OS and compilers and everything else and shrink wrap software. That's a long process. And so I thought, we needed to get ahead of that curve. We needed to find a way to design a 64-bit version of the architecture that could run 32-bit code efficiently, with certainly, relatively little overhead, both in terms of storage and efficiency. So that's what we worked on is how to extend that architecture, I think. We came up with a very clever way to extend the architecture that worked well and decided to make the R4000 a 64-bit machine. And that turned out to have other benefits downstream. I mean, for example, out of that came the interest that Nintendo had in thinking about using 64-bit architecture because in the gaming and graphics world, moving data quickly is an important thing, and 64-bits move data more quickly than 32-bits.

Mashey: An interesting outcome. It also had a bizarre thing, you might be aware of this, but I helped sell the CISCO guys on using MIPS chips, and part of the sell was 64-bits, which was in 1992 or something like that.

Hennessy: Yes. I think our biggest difficulty in those days was the difficulties that existed with the ECL [Executive Control Language] machine caused a lot of delays in the R4000 because we had to pull people off. In retrospect, MIPS got into the ECL business in a funny way. We acquired a company that was interested in using this logic that BIT had that was basically, a new form of lower power bipolar technology as a way to achieve a higher performance processor. Lots of things happened in that process, depending on a single supplier and a single start up supplier, what turned out to be a real mistake. It turned out that even technology was, perhaps, not nearly as good as people thought it would be, and the relentless march of CMOS [Complementary Metal-Oxide Semicondutor] technology simply overwhelmed the advantage that the bipolar technology offered. From the company's perspective, it did enable the company to get a few key sales at a critical time in its growth period which helped launch it through the IPO [Initial Public Offering]. But in the bigger perspective, it probably was a distraction in the end and because we only had one generation of machines with that technology, certainly, not a good investment of capital. In retrospect, we should have just tried to march onto the R4000 and get it out earlier.

Mashey: I guess coming forward, then there was the acquisition of MIPS by Silicon Graphics.

Hennessy: Right.

Mashey: Talk a little bit about how that happened.

Hennessy: Yes. So MIPS, I think, struggled a bit in the system business. It was a tough business to be in. In some ways, you were competing with some of your customers because we were selling technology at different levels and so we were enabling companies like Silicon Graphics, for example, and then trying to sell machines that could be viewed as competitive with them, not so much in the graphics domain, but in the lower-performance work station domain, or even the server domain. It was a challenging business model to really make work. We had our semiconductor partners, I think, who were certainly helping and getting the architecture out there. But we were struggling a bit in that whole system side.

Mashey: We're actually mentioning semiconductor partners. Maybe you could talk a little bit about the different ways that MIPS interacted with semiconductor vendors, and how that changed over time.

Hennessy: Well, I think it's important to remember that when MIPS got started and actually through most of the 1980s, this notion that we have today of silicon foundries and fab-less [fabrication-less] semiconductor companies didn't exist. They weren't there. There were not a lot of opportunities to get. While there were a few companies who were kind of in the foundry business, they were not what (A) that was not their primary business, like you would say is today of a TSMC [Taiwan Semiconductor Manufacturing Company] or other company; and (B) they weren't leading-edge technology companies. So you had to sacrifice in terms of the technology that was available to you, and, you know, that put–-that would have put MIPS at a real disadvantage. The business model that today we call Fab-less Semiconductor, wasn't well understood, otherwise that probably should have been the business model we try to converge on from the beginning. But we didn't really understand how to make that business model, we didn't understand it, it didn't exist. So we needed to get deals with major semiconductor manufacturers that had access to world-class technology and could provide it. So, when we hired Bob Miller, he came in and started talking about those possibilities. And working with a number of semiconductor companies, both in the U.S. and in Japan. But we also saw the semiconductor partners, as ways to spread the architecture and expand the potential range of sales, far larger than MIPS could do as a small company, from the beginning.

Mashey: So as I recall, at one point there was almost Motorola as a semiconductor partner.

Hennessy: Yes, Motorola came very close to signing up. One of the things I've observed in the computer industry, from time to time people are willing to take a short-term, less painless road, even if they're aware that the probability it heads to a dead-end is very high. And this has happened in lots of discussions we've had with people. Whether it's discussions we've had with a variety of… and Motorola's a good example. And, they were trying to get into RISC late. They knew they were late. They didn't have the compiler technology they needed. And they knew that joining up with MIPS, probably in the long-term, would have been better. But they already had some short-term customer relationships and various other issues, and a team, which made it difficult to uncouple and make that decision. And of course this debate went on in lots of places. It went on in DEC, and they decided one way, and subsequently decided to go on their own again, so.

Mashey: Interesting times, yes.

Hennessy: Interesting times. And I think, one thing you learn about the industry is cutting into lots of small pieces doesn't make anybody's piece of pie bigger. It makes it smaller. <laughter>.

Mashey: So actually at some point along there was a relationship with Microsoft.

Hennessy: There was a relationship with Microsoft.

Mashey: I think you got involved in that.

Hennessy: Yes. Microsoft, I think came to us and said "We're really worried about the future of the PC, and that we're going to get the computing power we think we need for our software." And I think they felt Intel was not being very responsive, and was no longer pushing the performance curve as hard as it needed to be pushed. So, I think that they came to us and said "We think RISC is the future, and we want to get going in that direction, think about RISC as a host for [Windows] NT, their next-generation operating system, and a push in that direction. So, we got that all going. There were lots of grand announcements, lots of grand affiliations. I get very nervous about this. We were a small company still, and here we're in rooms with Digital Equipment Corporation and Microsoft, and all these guys. And, you know, elephants treading around, mice gets squashed in this. And, what was for them an absolutely negligible financial decision, for MIPS it was life or death. But it was a fascinating time because you could see this clinging to proprietary systems, and this unwillingness to really embrace something that was really open. Believing that somehow they were going to… by having something that was proprietary they were going to get a leg up, they were going to protect their gross margin somehow and have a higher gross margin out of this whole thing. And what they failed to see was that they were going to have products, which were uncompetitive, either because of costs or their ability to deliver them in a timely way.

Mashey: So, coming back to Stanford, during this time that you were still director of CSL, right?

Hennessy: Mm-hm [Yes].

Mashey: And, as I recall, you got a lot of funding from outside industry.

Hennessy: We had some funding from outside industry. Never a lot, but we had some… we had had some funding from HP [Hewlett Packard] along the way primarily to work on complier technology, a little from IBM early on. So we had a number of companies. Primarily DARPA was the big funding agency for us, yes.

Mashey: So then somewhere in there, I guess what 1993, 1994, you became department head of computer science?

Hennessy: Yeah, I became department chair in 1993, 1994, yes.

Mashey: So how is that different from being CSL?

Hennessy: Well, you have a bigger group of people to worry about. I became chair as we were building, and about to move into the Gates computer science building. It was an interesting time for computer science at Stanford. Computer science at Stanford had never been in one building since maybe its foundational days. But when I arrived in 1977, it was already scattered, and CSL was in a different building from computer science, and the AI [Artificial Intelligence]Lab was up on the hill at this remote camp, which was really a different kind of place. And Numerical Analysis Group was over in this little thing called Serra House. I mean, it was scattered all around. And then when we moved to the quad, to Margaret Jacks Hall, it looked like it was big enough, but the day that the department moved in, there wasn't enough space for everybody. So a group of us stayed back in the old ERL and stayed there. So, actually getting the department into one building, where we'd actually be able to have all the CSL people and all the rest of the department in one building was… we viewed it as a real opportune time to get more cross-communication, get more collaboration going- and more interaction going across the department. And I think, in fact, that's probably… that has happened. And we've seen technologies migrate. We've seen a lot more interaction, for example, between graphics and vision these days, than we would have ever seen in the earlier days, and I think a lot more interaction between the theory people and the systems people, for example. So that's been a positive thing.

Mashey: So I must admit, watching this from outside, that there were those of us at… well, at that point it was SGI, all right, but old MIPS types you said "Well, we'll see. Okay. So, Hennessy is now a department chair. We'll see if he can either stand the administrative stuff, right, <laughs> for a couple years, then go back to being a professor. But if not, he'll end up being president of Stanford." <laughs>.

Hennessey: There's a long gap between being a department chair, which is a relatively short sentence to administrative work, usually, and being even dean, let alone the president. But...

Mashey: But anyway, this was our observation, was professors either can tolerate administration for the trade-off, or they can't. We've seen many professors who couldn't, and that was the interesting question.

Hennessy: You know, I think certainly there are some skills that are required. You need to have--obviously you need to be able to interact with people, and you have to have good people skills. You've got to have a lot of patience in any of these administrative jobs. It takes a long time to do things in academia and to shape them, and build a consensus around something. And then what probably the most important thing is, you've got to be able to celebrate and enjoy the successes of others, that you've helped enable. Because when you're a researcher, you're in the middle of it, right? It's getting a paper published. Even when you're in a company, I mean, it's delivering that product, seeing that first MIPS box go out the door. It's a thrilling moment. When you're in an administrative role, it's completely different. It's seeing that student win a major scholarship. Seeing a faculty member win a Nobel Prize. Seeing a great discovery occur. And you have to find…you have to be able to say, at least inside, you know, somehow I helped to make that happen, and I feel great about that; we were able to do that.

Mashey: But I think you still had PhD students?

Hennessy: I did, certainly through my time as department chair. And department chair job is some half-time, something like that, job, really in terms of the administrative role. So I still a very large group actually through the time I was department chair. And I started downscale it a little bit when I became dean. But at the peak, I had 15 PhD students at one point. It was a little insane, and I began to scale it back a little bit.

Mashey: Keeping in this same vein, of keeping your hand in, one of the things that's always amazed all of us watching this, is the books. So let's go back to the books.

Hennessy: Yes. So the first book...

Mashey: How on earth did you manage to keep doing that?

Hennessy: Yes, it's an interesting question.

Mashey: How did that work?

Hennessy: I picked a really good partner to write the books with. I think that's probably the first and foremost answer. You know Dave [Patterson] and I started writing the books. I was still doing a lot of consulting for MIPS. It was before the IPO, so late 1980s, probably. Dave and I were both very disappointed with the textbooks that were out there. Lots of people were teaching even their first-level graduate course in computer architecture which might have Masters as well as PhD students in there, they're teaching using collections of papers. Papers have no common terminology. It was very hard to stitch it together. And, the only books out there were what we used to call supermarket books. You know, here's one from column A, one from column B. And they'd basically contained synopsis's of various papers, that's really what they were, rather than any kind of, here's how to think about computer architecture. So, Dave and I felt that there'd been a lot of interesting things happening in the RISC side. And as much as the particular architectural insights, probably the methodology for thinking about computer architecture was actually in some ways, more important in the long term then. And thinking about it as a quantitative science that you can think about and measure performance. Of course the CPU performance equation had then, at least in the research world, really stirred up some interest, and people really understood what was going on. And that seemed like the bedrock of, kind of core CPU performance. Similar things can be done in a cache world, harder in the IO world, but even there you clearly have performance measures. So we decided to try to write a book crafted around a quantitative approach; an engineering approach to computer architecture. And that was the beginning of that book. We both took some time off from our respective roles at the university, and hid out at DEC WRL [Western Research Lab]. And they made a little home for us, and gave us feedback periodically and gave us a group we could talk over lunch to.

Mashey: Well, so for people not familiar, just describe DEC WRL. Where is it?.

Hennessy: DEC WRL was the DEC Western Research Lab. It was in downtown Palo Alto.

Mashey: Convenient for you, anyway.

Hennessy: Convenient for me. You know, it had a number of people we knew, headed by Forest Baskett , who was formerly my colleague at Stanford, several, PhD students from Stanford and Berkeley, and other institutions were there. It was a group of maybe 20 researchers, something on that order. They had done a Titan, which was this early experimental DEC RISC machine. So it was a group, we had a lot of overlap with, in terms of interests and things. And they provided us with a home, and we did our work there. And it was a great place to do the work, and brainstorm with people. I think what Dave and I… we both wanted to do… our view was, don't do a book unless it's really going to be excellent. So we developed a working relationship that would allow each of us to push the other person to do better. And, it's that little extra push you sometimes need. You write a version of the chapter, you write a section describing something. You know in your heart of hearts it really needs an example, but you're going to try to get away without it. And then you get the words… the comments back from your reviewer, and they say "You really need an example here." <laughs>. And that pushes you over the edge to do it. We got a great set of reviewers, and we'd listen to them. We got really great people to read the chapters, and we really tried to listen to their comments and take them to heart, which I think that's the most important thing for a reviewer, right? It's to have some impact.

Mashey: I think you did some process that almost felt more like releasing software.

Hennessy: Yes, we did like a beta release. The very first version of the book was really a collection of no more than notes that we experimented with on classes at Stanford and Berkeley. And I still, if I see somebody from that class I say "You know, you were in this first class that had just the notes." I said "You are the guinea pig class" I said "You've done a great service to the rest of the people studying computer architecture." So there was that version, the notes version. Then we kind of did an alpha version. And literally, we were shipping chapters to each other, like the day before Dave had to teach out of it at Berkeley, or I had to teach out of it at Stanford. And then we did a beta, and we released a beta version of the book in softbound copy for people to comment on, and that was to get even more instructors beyond the reviewers engaged in the process and get feedback. And that worked very well. People really made a lot of helpful improvements in the book. And then we went and did the final version of it, and it survived.

Mashey: How did you get hooked up with the publisher?

Hennessy: We started looking around for a publisher. We wanted someone who would (A) give us a lot of freedom, in terms of how we were both going to write the book and support in terms of getting it out there, somebody who had connections into the computer architecture community. We felt that if we wrote a good book, the book would sell itself, if you could get it in front of the right people. And that's what we really wanted, rather than perhaps a more conventional publisher's approach to it. So we tried that, and that seemed to work well as a way to do it. So that's how we got lined up with Morgan Kaufmann. And they were willing to do some things that other publishers just balked at. Like, we wanted to pipeline the chapters through the book process, because for computer books, time-to-market is really crucial. From the time you write the last word of the book, you want to minimize the time until the book gets in people's hands, especially at that time, in the 1980s and early 1990s, the field was still changing so quickly. Six months could mean you'd left out key ideas, or that the things you'd put in the book were long. I mean, they were out of time sync. So, Morgan Kaufmann was willing to do that. Lots of other publishers were not. And I think, lots of other publishers, the problem that often exist is, most authors never delivered their books, and certainly don't deliver them on time. So they're reluctant to set up a process that would really count on authors being efficient. But we said we're going to do it.

Mashey: So we'll come back to the books a little bit later, but that, as I recall, that book immediately sort of took over as the standard textbook.

Hennessy: It did. It still is. I think at the graduate level it still is.

Mashey: Yes, but to me, I've seen textbooks live for a very long time. That one was astonishing in the repetitively with which it suddenly became the text you had to use.

Mashey: Okay so let's sort of get back to the general career at Stanford, so at some place in there you became a Dean of Engineering?

Hennessy: Yes, I became Dean of Engineering in 1996.

Mashey: So talk about how that happened and, clearly there's a big difference between being a Department Chair of a department you've been in for a while and suddenly taking on a whole lot of other activities.

Hennessy: Yes, there is a big difference. So Jim Gibbons had been Dean of Engineering for a number of years and I think served probably at least a decade, maybe a little longer and so it was a natural time to change the Dean. They started doing a search and eventually a group of representatives from the search committee came to me and said would you consider being a candidate and I said "Well, maybe." <Laughter> I was a little uncertain I think, not as uncertain as I was at taking the provost job perhaps but certainly a little uncertain and I thought about it long and hard and I think the big piece of motivation for me was that computer science was doing tremendously well as a department and in some sense the challenges it had to face were challenges that were bigger than the department. They were everything from what did faculty compensation look like in the school to what were we going to do about housing to inter disciplinary activities between computer science and other departments in engineering that needed to be promoted. So in the end they asked me if I'd take the job and I took it. Condi [Condoleezza] Rice actually took me out to dinner for the final interview and asked me if I would take the job.

Mashey: So you did. Talk about the new challenges and, how that went and what things you tried to do in that time, I mean again this comes back, perhaps some of this interdisciplinary area's very interesting.

Hennessy: Absolutely, I mean it was a time when engineering was doing well at Stanford and it was really its strongest departments were getting stronger. But also a time of change, a number of disciplines were going through key changes and possibly the most important thing to work out in those days was what would be the role of the biomedical sciences and how to develop bioengineering and think that through. We had, at that time, a number of key collaborations with the medical school but they were in separate departments. We had a good medical imaging group. We had a good biomechanical engineering group. We had a few people in electrical engineering doing work in neuroscience areas but it wasn't a collective activity, it was fragmented along all the parts of it and our activities in the biotech sector, which really is more chemical engineering, we're really quite nascent. So we decided we needed to build that up. We made some key hires in chemical engineering, in electrical engineering and began to explore really then, I as Dean of Engineering and the Dean of the Medical school, whether or not we really should contemplate a joint department; how that department would interact with the Medical School because it would probably need to interact both with some of the basic science departments in the Medical School. But also with the Clinical Departments because engineering after all, engineering is a translational discipline. Engineering is about taking basic science insights and applying them to real world problems and in some sense it had to be a bridging discipline between the clinical side that's often primarily focused on seeing patients and seeing how new therapies work in the hospital setting and the basic science departments. So we had to figure out a strategy for doing that and begin to put in place what would eventually become under the next Dean, the Bioengineering Department. So one of the first, if not still the only Bioengineering Department that actually sits in two different schools. So it doesn't sit in medicine or engineering, it actually sits between them so it's joined and I think we wanted to have that structure from the beginning because it really was key to make it work well. We also started a number of other activities, civil engineering really accelerated their transformation into a department that really was focused on environmental issues as the predominant focus of the department and its research in those days.

Mashey: So actually talk a little bit more on the biomedical thing and of course you've got some funding from Jim Clark for building and I heard some interesting stories about how you actually encouraged the cooperation between engineering and medical faculty.

Hennessy: Yes, certainly building that bridge between medicine and engineering was absolutely crucial and while there were many commonalities in terms of how people thought about what they were trying to do. There were a lot of differences in how they did their work, how they financed their work. I mean engineering has a lot of students that are teachers and the education component is a big part of it. Medical schools: it's not a lot of students running around, you have a small number of medical school students, you have some fellows and post docs but it's primarily research and clinically driven. So you had to think about how to make these two work together and overcome the bureaucratic pieces. At the same time I have felt that in the same way that physics was such a transformational science for the 20th

century and chemistry for the 19th century that biology was going to be the transformational science for the 21st century. If you look at the rate of scientific understanding and the growth of scientific understanding, let's say since the discovery of DNA 50 years ago, that rate has just blossomed in the biological sciences and when that happens, that's a time when there's tremendous opportunity for progress to build on that tremendous new set of insights. So I saw that happening and I saw many more interdisciplinary things, not just bioengineering but biophysics, enhanced interactions in biochemistry growing so we then tried to capture this and there were a group of faculty who were really doing this bottom up at the same time and I think it's an interesting lesson about how the academy works because in universities it can't be purely top down and it can't be purely bottom up because eventually somebody at the top has to say "Yes, this is really the thing to do." and we had a great confluence of those with a group of faculty from literally across the university who had this concept that there would be a blossoming of these interdisciplinary collaborations with the biosciences at the center of that. And Steve Chu, our Nobel Prize winner was very involved in that. Jim Spudich, Bill Mobley from the Medical School so there was a number of people Channing Robertson from engineering. Out of that came this concept which today still has the name that was our working Bio-X where X is the free variable, can stand for engineering, physics, chemistry, lots of other things. We always thought we'd put a more formal name on it, but the Bio-X name seemed to fit well and caught on quickly. So then we went to my friend and former office mate Jim Clark who by then had not only been successful at SGI [Silicon Graphics, Inc.] but of course at Netscape and talked to him about providing a naming gift for the building and we decided to put the building right on the corner opposite computer science, next to the medical school, diagonally across from electrical engineering so it was really at a conjunction of those points because we visualized the building as being the nucleus in a larger atomic structure. Not everybody who was involved in these collaborations could possibly fit in one building. There are today 200, 250 faculty that have some affiliation with Bio-X, but we could put enough of them in there and we could put enough shared space that we could create a nucleus that bound the entire activity together. So that was the vision from the beginning, I think we got a great architect, Norman Foster.

Mashey: Yes, Sir Norman.

Hennessy: Sir Norman, who did a marvelous job on the building and really made it a meeting place with a structure that really captures that and captures I think the vision we had in the building. So that's turned out to be a real success for us and then Bio-X became the incubator for our new bioengineering department and really helped it grow and blossom.

Mashey: So now you were having a good time as the Dean of Engineering and then?

Hennessy: As my wife says, it's the best job that I've ever had. It's a great job, being a Dean, lots of people say but it's a terrific job, you know, at slightly over 200 faculty but the range of intellectual disciplines, I could understand what everybody was doing. I certainly couldn't be an expert in what somebody in mechanical engineering, but I could understand enough about what they were doing that I could appreciate it. I could know all the faculty on a first name basis which was a wonderful, wonderful opportunity. So then Condi Rice decided that she was going to step down as Provost and take a leave of absence and then, of course, subsequently return to work on the Bush campaign and the Bush Administration. President Casper then conducted his own search with a faculty advisory committee for Provost and ended up asking me if I would take the job as Provost. At that time he was approaching the end of his 7th year. I think we all figured he'll be President for 10 years. I said. "Okay I'm safe." I won't have to make the bigger decision next. But still that was a very hard decision because you're going from a range of disciplines you know well, a set of faculty you can know on a first name basis to an entire university where you can't possibly know all the faculty on a first name basis. You can't possibly understand what they all do and what's important in these wide variety of different disciplines. So that was a tough decision but after hearing a couple of people speak and thinking about it I decided to… One person said to me "You know, you've never worried before in your career about taking new opportunities and you don't know where those doors will lead to and what new hallways you'll walk down." So I decided that I would accept the offer and I became Provost and then probably a month after I was Provost, President Casper said "Next year is going to be my last year." so all of a sudden the whole process was accelerated and I thought I would have a little longer to sit in the Provost job and contemplate what I wanted to do next but he had decided after 20 years in university administration between Chicago and Stanford that 20 years was enough and he was probably right.

Mashey: He'd herded enough cats.

Hennessy: Yes, he had herded enough cats and we had had a very hard time with the hospital. Our hospital had been a lot of financial problems. We'd done the merger with UCSF [University of California, San Francisco]; we undid the merger that was a very tough and grinding process. So I think he decided he would move onto other things.

Mashey: So there came the opportunity to be President. How hard did you have to think about that or was already being Provost sort of set it up?

Hennessy: I think being Provost was probably in some sense the harder step because it's a step away from being an administrator. Clearly the job of President brings extra demands on everything from travel schedule to constituency and so I had to think about it some. On the other hand by that time I had at least some notions about where I thought Stanford could go and what we could do and so I probably decided I would do the job fairly quickly once they decided they wanted me.

Mashey: So sort of getting towards the end here. I know you've done, sort of in this general period a number of interesting things with industry, I'm thinking of some of the energy and environment things. Maybe talk a little bit about that, how did that get to happen?

Hennessy: Sure, well I think as you mentioned earlier John, I think universities are increasingly becoming the primary source of innovation, breakthrough thinking and if you look at the challenges that we face around the world, they're if anything more complicated than they've been in the past. We've got the challenge of energy and global warming and resources in general, so we've got monstrous environmental challenges. We've got the challenge of everything from threats to peace to seeming inability to create the opportunity for countries to see economic growth and improve their situation and in the medical area, the challenges we face are really tough. I mean the age of infectious diseases is not over, the challenges we now have to do and to work on things like autoimmune diseases or cancer. These require deep fundamental understandings that are grounded in science, first and foremost, and so I felt it was a real time for the university to think about what it's role would be in addressing these problems. They were going to have to be addressed in a different way. Not by one professor working alone with one small team of graduate students but in a much more collaborative interdisciplinary fashion.

So I think we started early on with the environmental area and the international areas, areas where we already had a team that was quite broad. Say in the environmental area all the way from conservation biology to earth sciences, to civil environmental engineering to over even into the medical environmental area and begin to pull that together. We had this opportunity to become the lead institution for GCEP, the Global Climate and Energy Project. That is industry funded, funded by a consortium of energy manufacturers, energy generators as well as some automobile manufacturers and really work on that problem. And it’s a different problem because I think if you can find new solutions to the problem of reducing carbon loading, these are things that you’re going to have to get big companies to play a role in, to get them to scale which is what we need to do, you’re going to have to collaborate with big companies. So this has been an area where I think, unfortunately, there hasn’t been enough government funding too. We should have been funding decades ago back in the, you remember the oil shock crisis of the late 1970s. We started up all these alternative energy research programs and they went away as the price of gas fell again. They should have never gone away; we should have been working on it all along. So this is an area where you need to have multi-disciplinary interaction, where you’ve got to have a collaboration with industry if you’re going to get that technology transferred. So we’ve started to do some experiments to try to figure out how to organize those kinds of research projects and approach them differently.

Mashey: So let’s finish up here with a few quick questions. What would you call your most important life lessons from this whole long process?

Hennessy: Don’t be fearful. Be bold. Be willing to take risks. Recruit and find the very best people you can to work with and appreciate that there’s technology alone does not guarantee success.

Mashey: What would you say your proudest moment has been in the career or anything else? You’ve got a lot of things…

Hennessy: I’ve got a lot so I’ll probably have to say two things. I think when we shipped the first MIPS box, that was a really monumental moment because there’s so much effort had gone into that and creating the company and building the company and it was the first time I was really in the middle of getting a product out the door as opposed to a research paper.

Mashey: It’s a little different.

Hennessy: It is a little different. You know I think the other, when I visited China a few years ago and we’re walking down the hall in a research lab in Chunghwa and my wife points out, she says “Look there’s a copy of your book, why don’t you sign it for that young lady?” So I stopped to sign it and in 30 seconds 50 or 60 students pile out of their little cubicles holding copies of the book and you realize that through this book you’ve touched people worlds away. In my university role last year was an incredible year for us, we won two Nobel Prizes in the space of a week. That was an incredible time, obviously, really a testimony to the investments that were made, long before I was an administrator at the university but a wonderful time and a wonderful reminder of what those investments can really do and turn into.

Mashey: In some sense, looking back it almost seems more like a straight line getting you here than a lot of people have followed in their careers but sort of what turning points were there for you in making decisions or was this kind of something you were thinking you would do way back?

Hennessy: Oh no, absolutely not. Neither was starting a company nor was being a university administrator in a leadership position. Those were both never on my horizon. I love being a faculty member. I love teaching. I love working with graduate students in a research setting. I really thought that was some of the best moments of my life. When we decided to do the company it was almost like we felt we had to do it because we had to get the technology out there. But that was a tremendous learning opportunity for me. I learned skills that would have been much harder to learn in the academic setting. In an industrial setting you have the pressure of time, you can't wait, you can't just not make a decision because the answer's aren't purely black and white, you've got to make that decision because time is your enemy and I think you have to do that in leading a university too because there are lots of complex decisions like that.

Mashey: And certainly in a company like MIPS, there were interesting personalities to go with it.

Hennessy: Yes and I never thought I'd be an administrator. I think when I took the Dean's job I really discovered that I could enjoy that and enjoy the accomplishments of others that I helped facilitate.

Mashey: So in terms of what you might else have done I guess the obvious thing is you would have kept on mainly as a researcher and teacher instead of taking the fork in the road?

Hennessy: Yes, start a company but, you know, those things they blend in a way that's really interesting. You don't ever quite understand how the interaction between different things you've done and experiences you've had will blend. I mean because we did MIPS I really was forced to think much harder about how RISC worked and what it's advantages were which of course led to the CPU equation, which lead to the thought that we would write a book that was really a quantitative explanation of computer architecture which of course, brought me around to other contributions.

Mashey: So I guess if you look back at the impact of your contributions, you've kind of talked about that some, but is there anything else that's worth mentioning there particularly?

Hennessy: No, I have a lot of students out there that I've been privileged to teach and work with in the graduate setting and, of course, for all the great research accomplishments or things I've contributed in that side, what they will do will make all my accomplishments look small because they're a tremendous magnifier and I think that's one of the great things about universities: the role of students as magnifiers, is really a tremendous one.

Mashey: So how about the other direction how about role models. Were there any particular people that mentored you at some point or other?

Hennessy: Sure I think, you know, Mike Flynn was my first early guardian at Stanford and helped me get settled in and get my research going. Forest Baskett really became my mentor in the whole area of how do you build and experimental research team and pursue an experimental computer science, computer engineering kinds of project and helped really create the umbrella that got us going early on. So those were sort of my academic mentors and role models. Today you need a different kind of role model as university president. I think actually my entrepreneurial experience was really terrific for… because understanding how to change… Universities are not institutions which change very quickly, in fact, we're built to change slowly because we're built to avoid fads and things take a long time. A university is in many ways like an oil tanker: it takes a long time to change direction but once you get it going it can coast for a very long time on its own speed and my entrepreneurial experience was really key for me to understand how to make those kinds of changes in it.

Mashey: How about advice for the current and future generations of students, faculty and anybody else?

Hennessy: Well, the one piece of advice I often give is people should not be timid. Helen Keller once said "The cautious are caught as often as the bold" and I think it is true. One rarely accomplishes much if you're not willing to take some risk and every year I encourage the students to take some risk and that's all kinds of risks. That's going through a door that you might not have contemplated going through and possibly not trying to program where that path is going to lead to. I think working with great people: I've been privileged to work not only with great colleagues at MIPS and the university but to have tremendous students and I think they've contributed so much to what we were able to do over the years and really help innovate and create new things. I think it's one of the magic of universities. Students don't know what approach shouldn't be tried because the conventional wisdom is that's been tried and I think that's a very healthy thing. You know, the computing field is an amazing one because it reinvents itself and has this wonderful ability not only to stay fresh and new but to also have tremendous impact on the world and it's a real privilege to be in a field like this where you have that tremendous opportunity and in some ways it's the best of both worlds and I've been lucky to be in it.

END OF INTERVIEW

