Oral History of Gary Davidian, part 1 of 2

Interviewed by:
Hansen Hsu

Recorded February 5, 2019
Mountain View, CA

CHM Reference number: X8931.2019

Hsu: It is February 5th, 2019. I am Hansen Hsu here with Gary Davidian. Welcome.

Davidian: Hello.

Hsu: So let's start with where and when were you born.

Davidian: I was born in 1956 in New York City. And my parents were also born in New York City.

Hsu: And so is that where you grew up?

Davidian: I grew up-- yes. I was actually born in Manhattan but grew up in Queens.

Hsu: And what's your ethnic background?

Davidian: Armenian. Both my parents were Armenian. So I grew up in an Armenian household.

Hsu: What did your parents do?

Davidian: My father was a mechanical engineer and he worked for-- well, while I was growing up, mostly he worked for Sperry Gyroscope in Great Neck, New York. And my mother, while I was growing up, was just a-- a housewife, but, you know, she had-- she had worked before marriage and then later when I was in college she was-- she was working again.

Hsu: Do you have any siblings?

Davidian: Yes. There are four of us. I have an older sister. I'm second. I have a younger brother, and then the youngest is my youngest sister.

Hsu: Did you have any hobbies growing up?

Davidian: Well, there were-- there were actually some things that-- when I was growing up-- which were sort of my favorite toys. <laughter> One of-- very early, was-- one of my favorite toys was actually my-- was the vacuum cleaner. <laughs> My mother always said, you know, "He would take apart the vacuum cleaner and put it back together." It wasn't quite that. It was-- I would actually take it out of this hall closet. I'd drag it out into the room, put all the attachments in, unwind the cord, plug it in, turn it on, and have this big smile on my face. And then take it back apart, you know. Put everything back, you know, in-- put it back in the closet. Another toy that was one of my favorites was-- it was called Mr. Machine. It was made by Ideal Toys in the early 1960s. And it was a pretty interesting toy at the time. It was-- it was this little robot. It was a wind-up robot that walked and made some noises and had a bell and these rotating wheels. But it was also a puzzle. It came apart, and there were lots of gears and fasteners and, you know, it completely came apart like a jigsaw puzzle. And, you know, so you would take it apart and put it back together. I think at the time there were a lot of kids who got it, took it apart. They couldn't put it back together. Their parents couldn't <laughs> put it back together, and, you know, and that was sort of the end of it. But I used it all the-- that was one of my favorites. I would take it apart, put it back together. And they actually came out with the toy again in the '70s, and they changed it a bit. You couldn't take it apart anymore. And I think that was partly because people couldn't put it back together and also because child—product safety laws <laughs> had changed. The stuff we used to get-- get away with in the '60s, you know, just-- there were choking hazards and all sorts of other things in it. And then another favorite toy of mine was the erector set. It was just these-- you could build things out of these metal-- little metal beams and screws and fasteners and motors. So that was another favorite toy of mine. I was also-- bicycles were a favorite of mine. And I used to, you know, take the bicycle apart, put it back together. And I, in high school, I was working at a bicycle shop. So that was, you know, I actually got-- had paid work, working on bicycles, and used to enjoy-- I enjoyed seeing how things worked and that was, you know, sort of one thing that followed me and helped me in the future because it's-- if you understand how something is supposed to work, when it's not working you know where to look. And, you know, it's deb-- that's how debugging is with computer code. So that was-- that was some of my childhood. The toys and hobbies. When I got to high school things changed a lot. That was-- there was a class there called Computer Math. Our high school had been given some old computer equipment. It was-- I think it was handed down from one of the magnet schools. Brooklyn Tech. There was an Olivetti Programma 101. That's the first exhibit you have here. It was sort of one of the very first-- what could be considered a personal computer. It was really a programmable calculator. It had a paper tape printout like a cash register. It had a little keypad, numeric keypad with some alphanumeric characters. A very strange programming language. It had some magnetic cards for storing programs. It had, I think, 22-digit numeric accuracy. So there were small programs that we could do-- program on that. And that's where I learned just some of the basic elements of programming with loops and conditional-- conditional branches, and then just, you know, some arithmetic operations. We also had this other computer which I don't believe the museum has. It was-- we-- it was called the Monrobot XI 2000. And it was sort of a personal computer also. It was-- it was built into a desk. Sort of an L-shaped desk. And it had two electric typewriters on it that were used for input/output. It had a paper tape reader. It had, I think, 2000 words of drum memory. It had a little console with some push button switches on it. And, at the time, I learned to program it and programmed it in hex. It had a funny hex representation instead of-- you know, it had 0 through 9 but instead of A through F it had, I think, S through X. <laughs> And I believe it had 32-bit words, but it stored two 16-bit instructions per word. And that was, you know-- since it had a typewriter on it, you could actually do much more sophisticated input/output. We had game-- we made games like tic-tac-toe. You could also print out more alphanumeric results. It even had a small FORTRAN compiler <laughs> that you had to load through paper tape. But the thing I remember about that was I was learning to program this in machine code. In hex. So that was really my introduction to programming computers. It was at this very low level and I didn't really know much-- much else. So I was-- I didn't-- it wasn't until college that I learned about assemblers. <laughter> So it was just really programming in machine code and, you know, if you added an instruction you had to recompute all these branch addresses. And everything was typed in each time. We could store some things on paper tape, but the programs were fairly small. There wasn't that much storage. We would write-- we would write them on these coding forms, which were mostly like graph paper. But that was-- that was my real early introduction to programming, and it was at a very low level.

Hsu: So that-- what year did you-- was that?

Davidian: That was probably around 1971. I think it was probably about the year that-- I was in that-- I think it was-- I think it might have been the second half of my-- of tenth grade. It was tenth and eleventh and twelfth that I was involved in that. And the other thing about that class was we could stay after school. It wasn't just, you know, 45 minutes a day. So it was more-- there were several of us. We were all-- we all became friends and we would all hang out there after school pretty much every day. It was-- and we were all very interested. So it wasn't just a class. It was-- it was more just like this-- we had a lot of freedom and-- the other thing that the high school had which I got to really learn some-- some early computer history was the-- a lot of IBM unit record equipment. So we had, I think, two IBM 029 keypunches. There was an IBM 82 sorter. There was, I think, the 419-- the 519 card punch and the-- I think it was the 557 interpreter. And then the one I liked the most was the IBM 407 Accounting Machine. This was a monster. It was this huge thing. I think it used a lot of relays. But for programming it, it was actually this peg-- it was like a pegboard with wire plugs in it. And that's how you actually controlled it. The machine sort of had this-- each cycle, it could read a card and print a line. And then this control board would control how, you know, how the data-- the columns of data would move from the-- from the card reader to the printer and it had some-- some fairly primitive math. It had an accumulator. You could actually total things up and print final totals at the end. So that was-- yeah, that was another learning experience. And again, it was lear-- I was learning really low level, you know, how things were done at very primitive levels. So it was just really understanding how the hardware worked. And IBM actually had some very excellent documentation back then. The manuals for these machines were very detailed and really described exactly how everything worked. I enjoyed reading them. That was-- that was very good, you know, material. One other thing we passed over in my-- in my childhood, which was a really significant thing in my life, was my father passed away when I was eleven. So that was-- there was a lot of sadness after that. Getting involved with computers was something where I was-- it allowed me to focus on something that was really interesting and something that I-- I was enjoying. In high school I knew where, you know-- a lot of kids get out of high school and go to college, but they don't really know what they want to do. I knew. <laughs> I knew exactly what I wanted to do.

Hsu: So you went on to SUNY Buffalo?

Davidian: Yeah, it was-- getting out of high school, there was-- given that my father passed away, the-- it was sort of a given that I would be going to a school in New York State and probably a state school.

Hsu: How did your family make ends meet, by the way?

Davidian: It was-- actually, well, some of it was Social Security. I actually-- I collected Social Security already. <laughter> It was my father's Social Security when I was young until I graduated college. And my mother did-- did go back to work after that. But also I was-- I was actually fairly lucky. In-- while I was in high school, the New York City Teachers Union, the United Federation of Teachers, they had a college scholarship. They-- one day I was called down to the principal's office and I didn't know why. And I was told, you know, I was given an application and was told that, "You should apply for this scholarship." And I think there were-- from each school there were two children that were selected. And I was selected. And that, at the time, that paid a thousand dollars a year towards college. College back then was not that expensive. <laughter> And also New York State had the Regents' Scholarship, which also, just based on scores, test scores, that covered tuition at state schools. So when I was looking at colleges there were a few coll-- I applied to all the state universities in the New York system. That was Stony Brook, Binghamton, Albany, Buffalo. Maybe Binghamton, did I say Binghamton? Also my older sister, she went to the state college in Potsdam. So I applied to Potsdam and in Potsdam was also Clarkson College, an engineering school. And I applied there also. So I-- I got accepted to Potsdam, Clarkson, and Buffalo was a little weird. I got a letter saying, you know, "We'd like to meet you." <laughter> One thing-- well, my SAT scores were lopsided. I had 790 math, 430 verbal, so it was a 1220 total, which-- but-- so they wanted to meet me. And there was an interview set up with the math department there because, in high school, math was what I was focused on, because there really-- that was sort of the closest thing to computers at the time. So I met with the math department there. I got to see the campus. In Buffalo, the campus was-- there was a transition going on. They were building a new campus in Amherst, and there was a temporary campus called the Ridge Lea Campus, which was kind of like an office park. It was just a bunch of small buildings. And that's where the computer science department was and the computing center was. And so I got to tour the computing center when I went up there to visit with my mother, and they had this CDC 6400. And you could walk up to it. Hand your card decks in. They came back. And also in the dormitories, that were on a different campus, they had remote job entry and printers to submit your jobs to it. That was-- so there was just this lot of access. And in Potsdam they had an IBM 360 [Model] 30. It was behind glass and like off-limits. And Buffalo was-- it was the university instead of a college. So that was where-- really where I wanted to go. I had also actually, while I was in high school, I did have some CDC exposure. One of my math teachers was taking some night classes at NYU and NYU had a CDC 6600. I was actually writing a program for the school to help them with scheduling classes. I was writing it in FORTRAN, but it needed a lot of memory. It needed a real computer. So he actually took me down to NYU and actually, you know, let me run my programs on the 6600 at NYU for the school. There was also a community college that-- Queensborough Community College. They had an IBM 1130 that we sometimes had access to. They also had remote entry to the City of New York-- CCNY's 360 [Model] 50, so I could run some FORTRAN jobs there, too. So I did have a lot of exposure in high school. But I was sort of learning things in just-- in a sort of random order. It wasn't like college where they sort of led up from one thing to a more advanced thing. It was just, you know, "Well, here's something. Go learn about it." That was-- that was kind of high school, and then college it was-- I knew what I wanted. And Buffalo was where I wanted to be, even though it was this sort of campus in transition. It was a small department. That's where I also got to meet some of the people who were fairly influential throughout my career. There was Dr. Gideon Frieder. He was-- one of his interests was microprogramming, and that's the field that I went into. Yeah, he was my advisor during-- during my undergraduate, and then when I was in-- going for my master's degree he was my advisor. There was also-- that's-- when I started taking the computer science courses there in order and learning things in the order you're supposed to learn it, that's when I took this class called CS241, which was where you started learning assembler. And it was CDC 6000 assembler. And that was also-- that was fairly influential, because the CDC 6000 architecture was very much like what RISC machines were like in the '90s. Late '80s, '90s. There was 3-address, load-store architecture. Large register files. And so, you know, when I saw some of the RISC machines like the MIPS, Motorola 88000, PowerPC, they were all in some ways very similar to what I started on. <laughter> The same thought patterns went into programming them. In college, we had-- one of the computers we had in the computer science department, the CDC 6400, which later was replaced with the CDC Cyber 173, that was the university computer. The computer science department had some of their own computers, and there was a Varian 620/i which had a CRT graphics display on it. That was used by some of the artificial intelligence people. And then there was a Burroughs 1726, and that was-- that was a microprogrammable computer which had-- Burroughs had this concept of S languages, which were-- they had an instruction set that was more tuned for the higher-level language that it was being used for. So there was like a FORTRAN S language. They had their own system programming language, which was somewhat like ALGOL. So they had an SPL S language. But they weren't-- they didn't sell it as a micro-programmable research tool, but that's what the university was using it as. So we were actually writing microcode which their-- Burroughs' customers never did. That was something that was done internal to Burroughs. So that's where I started writing some microcode. One of the things I programmed there was-- I did an emulator for a Data General Nova. And I was able to run the Data General operating system on it. So that was my first emulation experience to the level of where it was running an operating system. So that was a really fun project. Also while I was in Buffalo, the other thing I did was, for work, at the computing center they had student assistants which they referred to as "information clerks." So they had a little-- a little office in the computing center or at these remote sites where the students who were using the computers, if they had problems understanding an error message or things like that, this was the help desk. So I got a job working for the-- for the computing center as an information clerk when I was an undergraduate. Besides the pay, which actually-- I learned that-- I started as a federal work study-- that's how I-- it was sort of a financial aid job. And that's when I discovered that the federal government could actually pay below the federal minimum wage. <laughter> I think I was getting two dollars an hour and the minimum wage was two ten when I started. But the money wasn't the only reason. The other benefit of this job was you got a computing center computer account which had no limits on it. So you got computer time. Whereas the class, you know, in this batch environment, each class account had some dollar amount allocated for the whole class and it was-- I don't know how the-- I don't know if any real dollars were ever transferred between departments but these were limits. And so in the class accounts you were sharing with other students and there was money that was gonna run out. Whereas here you got a personal account and there might have been dollar limits but it always got refilled. So it was essentially-- you had unlimited computer access. So that was the real benefit of that job. <laughs> I definitely used to explore a lot of programming. So I definitely enjoyed having-- having the freedom there. And the job was fun, too. I'm trying to think of some of the other things about Buffalo. I also-- there were some other jobs that I had while I was there. There was-- other computers that I used. The computer science department eventually got a Data General Eclipse S/200. And that had-- had two-- there were two consoles, I guess you could call it. They were CRT consoles. And it was-- the other thing that we had for that was this Comtal-- I think it was the Comtal 8000 graphics display. It was very high-end at the time. Color. 24-bit color display. It was used-- this was around 1976, I think, and it was-- I believe JPL was using these for the Viking Mars images that they were getting. And it was-- this was acquired by a group there that was actually doing some of the very early research on what was-- what was becoming CT scans. The computer-aided tomography. So the display was for doing a lot of the display of x-ray-- 3-D x-ray images. And I got to write some of the device drivers for that display and some of the graphics routines that were used by the group that was doing the research. It was, you know, again, it was something where I could-- I could use the computer whenever it was available. Also, other people with that dis-- the display-- there was a trackball on it, too. Big one. The size of a cue ball. <laughs> And someone programmed the

Breakout game. So there were other things. And I think there was-- I think maybe Solitaire. Some sort of card game. So those were things that we, you know, that we played with. I did enjoy being in college there. There were a lot of-- lot of classmates. It was a small department. I think there were maybe 30 students a year that were admitted to it and we were all friends. We were all living in the computing center or the computer science department. Just, you know, it was like one family. I also went-- another company around that time that I worked for was Nanodata. Nanodata--

Hsu: So you were still in college at this point?

Davidian: This was still in college. This was-- it was during-- during my master-- the time I was getting my master's was when I worked for Nanodata. They were-- they built this microprogrammable research machine. There were a lot of-- probably more papers written about it than there were computers sold. They were a small company. It started with a research grant that the University of Buffalo got, and there were some people that had worked at the university who decided to form this company to build this computer and an actual product for-- to sell to others. So I don't know how many they actually sold in the end. The machine at the time was the QM1. Nanodata QM1. They did-- one of the-- they developed a Data General Nova emulator for it because, during development, they actually had a Data General Nova, so all of the software they wrote-- like the assemblers and, you know-- they wrote everything-- all the tools needed to program it were written for the Data General Nova so they emulated the Nova so you could actually run it on the QM1. They also did an IBM 360 emulator and, in later years, based on their experience with the IBM 360 emulator-- IBM 360 and 370 clones were becoming popular back then. They went into the IBM 370 clone business and they had a-- I think their-- the machine that-- the next product they were going to do was called the VMX. I don't know if they ever sold it. They eventually went out of business or got acquired by some larger companies. I was-- I had left by then. I was focused on the QM1. Actually it was the QMX, I think, was the follow on that was in development when I left. But I got to know some other people from then that were also influential. This one person, Bob Hanzlian, and he was—he had worked both at the University of Buffalo in the CDC systems-- in the systems group. They did the maintenance of the operating system and sort of the custom modifications for the university. And he was also at Nanodata for many years. He was an excellent microprogrammer and I learned a lot from him, just talking with him for years. And Gideon Frieder was also influential. He was the one who told me to go interview at Nanodata. He was involved with Nanodata, some of the patents, you know, for the original Nanodata. He was consulting on the QMX. So that was a connection there.

Hsu: So you graduated in-- you got your master's in what year?

Davidian: 1979.

Hsu: Seventy-nine, okay. And then right after that you went to Data General.

Davidian: Yeah, I interviewed-- there was actually-- there was a conference I went to in-- it was in Dayton, Ohio. It was around the time I was graduating. It was an-- I think ACM was sponsoring it. But one of the things they had there was a job fair. So your resume got-- there were like books of resumes, and books of job descriptions, and lots of tables, and letterboxes that you could just leave notes for people.

And so this big matchmaking-- and so when I was actually doing the job hunting, there were-- I think there were five companies that I was interviewing with. There was Wang in Lowell, Massachusetts; Texas Instruments in Plano, Texas; Data General in Research Triangle Park, North Carolina; and Hughes Aircraft in, I think, Culver City or El Segundo, California. And then there was this weird one. <laughs> Ford Motor Company. <laughter> This one I didn't, you know-- I didn't go looking for them. They-- a recruiter contacted me after the conference. Said, "There's people here that would really like to talk to you." And I said, "Do you know what I do?" <laughs> "Yeah, yeah, we know what you do." <laughter> So they flew me out to Dearborn, Michigan. It was very clear that they had no idea what I did. <laughter> But I was there for the day. They did not know the difference between microprogramming and programming microprocessors.

Hsu: Oh. <laughs>

Davidian: So I think they were more interested in some sort of embedded systems programming. But they also-- they used CDC equipment. So I got to talk with their systems programming group instead. I spent some time there. But eventually, yeah-- eventually the right thing happened. I did get a rejection letter from them. <laughter> From Ford. So that was the strangest.

Hsu: Were you-- I mean, were you particularly marketable given your specialty in microprogramming?

Davidian: Yes. At the time, microprogramming was still a big field. You know, there were microprogramming conferences that I would go to. In the mini-- this was sort of the minicomputer age. All the minicomputers were microprogrammed. So it was more limited to computer manufacturers. That's really the-- which is why a car manufacturer seemed odd. <laughs> But yeah, it was a big field, still, then. Hughes Aircraft, the reason why they were interested-- they actually had a Nanodata QM1. So that was their interest at the time. Someone who actually worked at Nanodata. I'm not su-- I don't recall exactly what TI-- TI was-- it was like this huge gang interview where like all the-- all the new grads started off in this big auditorium in the morning. And, you know, where you sort of picked up your interview schedule for the day, and you had a rental car and you drove all around the Dallas area to these different sites to interview with different groups there. And I do remember interviewing with-- you know, the people I interviewed with did know about microprogramming. And I think-- it might have been the TI-990 series at the time that they were working on. But I don't really recall exactly what the position would have been there. And Data General-- Data General, besides being-- meeting up with them at this conference in Dayton, they also did campus interviews. And Rich Belgard, who was one-- who had gone to University of Buffalo-- was one of Gideon Frieder's students. He was working at Data General, and he came and did the campus interview. And Gideon mentioned my name. And I had-- I had heard Rich Belgard's name. It's just-- I had actually been work-- I was working on some of the code that he had worked on when he was a student there. So, you know, I got to know some of the stuff he did and we knew a lot of mutual friends. And he did the campus interview and so I got invited down to an interview at Research Triangle Park. They were definitely doing microprogramming. So that was definitely something that I was interested in. It was-- they were attracting a lot of-- a lot of interesting people there, too. It was-- there were a lot of people from Burroughs, a lot of people from MIT, a lot of people-- well, some people that had worked at Data General in Westborough. The head of the project was Ron Gruner, who-- I believe he designed the Nova

1200 on his kitchen table, and so this was a very exciting project and I definitely wanted to be part of it and...

Hsu: And this was the, the Fountainhead Project?

Davidian: This was the Fountainhead Project, yes.

Hsu: That they-- you hired directly into.

Davidian: I hired-- yes, the project was-- I wasn't in at the beginning. The Fountainhead Project actually started in Massachusetts and it was named after the, I think, the Fountainhead apartment building. It was that they were sort of working offsite and this was, I think they had rented out some apartments and that was, that's how it got its name. It was the-- that's why it was called the Fountainhead Project. It was actually named after the building. It started in...

Hsu: So how was this different from the Eclipse? What was the relationship or...?

Davidian: There was none. That was the thing about this project. This was supposed to be Data General's architecture for the future and their instructions were to start with a clean sheet of paper, not worry about compatibility, and just think about the future and also to reinforce that, they were separated from the rest of the company. That's why they were in North Carolina. It was-- that was intentional, so it was, in some ways, it was treated like a separate company, like a startup company, in some ways, and so there were a lot of bright people and a lot of things that were done back then were early. There were things that you wound up seeing decades later, but...

Hsu: Like what sort of things?

Davidian: Very large address spaces, networking, everything network, unique identifiers so that things were, there was sort of this large, universal address space that all the computers lived in and it was heavily microprogrammed and it borrowed from what Burroughs did as far as having these S languages tailored to each high-level language. So there was a Fortran S language, and there was a COBOL S language. There was a SPL, a systems programming language, in some ways similar to what Burroughs had for a system programming language.

Hsu: So the-- by S language, so you mean that the, there is a different-- the, there's a microcoded implementation that optimizes for that particular high-level language?

Davidian: Yeah. In some more recent things in history that were like the S languages, like the UCSD Pascal system, the p-code system, and the Java virtual machines.

Hsu: Right, okay.

Davidian: So the, those are things that were somewhat similar to what S languages were like. So it was sort of an, it was an instruction set. There was a compiler that generated these instructions but the-- it was sort of an, there was sort of an interpreter, which was the microcode was interpreting these instructions. It was implementing that instruction set architecture, so, but there was one underlying hardware microarchitecture that was, that had different microcode interpreters that were interpreting these different S languages. So that's part of what FHP was like, so it was-- I got there in June 1979. I don't know exactly what year-- I don't remember what year the project started, but the goal was they wanted to demonstrate it at the National Computer Conference in 1980. It was, NCC 80 was the goal so that was, that would have been in the summer of 1980. In the meantime, there was, Data General was being supported by the Eclipse customer, the Nova and Eclipse customers and there was a lot of fear about those customers being scared away if they knew that something new and different was coming and why buy the-- I think the Osborne effect, but this was an early version of the Osborne effect. So yeah, this is something, so that was, yeah, there was some fear of that. So that's sort of how the-- that's why Data General felt they needed to sort of have a Plan B which was a 32-bit Eclipse, essentially. And that's what the Eagle Project was, and that was done in Westborough. Compatibility was the main goal, and it was to-- it was to compete against the VAX. So FHP was initially going-- I don't know if the VAX was actually out at the time that FHP started, but it was going to be Data General's 32-bit minicomputer and beyond.

Hsu: Right. Okay, so one is starting-- they're both 32-bit, but one is starting from a complete blank slate...

Davidian: Yes.

Hsu: ...no compatibility, and the other one is sort of like, okay, Nova compatibility is important...

Davidian: Yes.

Hsu: ...and let's compete directly against the VAX...

Davidian: Yes.

Hsu: ...more, I guess, more of a practical market consideration than the FHP, which is more of a pure, long-term technology play?

Davidian: Yes, yes, and so that's how Eagle started and *The Soul of a New Machine* documents this, the development of Eagle and the competition that was going on between Eagle and FHP. And there was-- the, I think the-- I don't remember when the book came out. I think it was before-- I think it started coming out actually before FHP[1] shipped, but it started coming out a chapter at a time in *Atlantic* magazine. And it was actually kind of funny because the first chapter that came out was a chapter titled "Flying Upside Down," so, and that got, it got pinned up on the bulletin board in North Carolina and everyone saw it and we thought the title of the book was "Flying Upside Down" but then more chapters came out and we started to see what it was like. So it was-- but we got to, we really got to see what, sort of see this

[1] The interviewee said "FHP" but he meant Eagle

competition that was going on that we didn't necessarily recognize as that but it eventually, it became clear that Eagle was going to ship and Eagle was going to keep the company alive and FHP's future was uncertain and, but we kept, we were told to keep going forward. And one thing that was, one benchmark that was important to Data General back then was the Whetstone benchmark. It was a Fortran benchmark and I think it actually, Data General started using it a lot when, I think, they actually benched fairly close to one of the CDC machine-- I think the Eclipse S/200 benchmarked fairly close to one of the CDC machines on this Whetstone benchmark. So that was an important benchmark to Data General so for FHP, working on the microcode, the Whetstone benchmark was something I was always focusing on. I was-- and I would look at everything in it. It did a lot of-- there was a lot of transcendental functions that are used, sine and cosine and things like that. On the Eclipse, Data General actually implemented instructions to do sine and cosine and they, the op code was actually followed by a table of some floating-point constants and those were actually used in the computation. And so I was looking at how the Eclipse did things and actually the person who worked on the Eclipse Fortran compiler, John Pilat, he was working on FHP so I would talk to John Pilat a lot. He wrote a lot of the math routines, so it was definitely focused on the performance of those instructions and I also started looking at just all of the computations that the benchmark did. And I discovered that it, of all the floating-point division that it did, a lot of them were actually dividing by two, but the, you didn't, the compiler couldn't tell that it was dividing by two. You couldn't tell at compile time that it was dividing by two but at runtime, you could so I said, "Okay. Well, divide by two is really easy." And so actually I special-cased the floating-division to have a fast path for dividing by any power of two and so that improved our Whetstone benchmark numbers and there was one time where, sort of after the dust settled with, between FHP and Eagle, I remember Steve Wallach was once down to talk to people on FHP to see if there was anything he could learn. And so I told them about this, the Whetstone benchmark and dividing by two, so I think that eventually made it into the MV/8000, the Eagle follow-ons. So FHP was also very good at COBOL. We weren't-- that was never really a big focus of Data General. They were more into the scientific market, but there was this benchmark called the U.S. Steel benchmark and it was a COBOL benchmark and we wound up performing very well on it. So that was one thing that was, in trying to justify the future of FHP, that was just this one thing where, just look at this. But in the end, FHP was just put on the shelf and nothing ever became of it but I do-- there is a remnant of it that I can...

Hsu: Yeah, let's, let's, let's do that.

Davidian: So this is one of the circuit boards from FHP. This was called the FM board. FHP had, they had what they referred to as the F box and the E box. The F box was sort of the Fetch box and the E was the Execute box so this-- and all the boards in the F box started with an F, in the E box started with E. So this was the FM board and the M stood for Microstore, so this is where the microcode memory was. This is the control store board for the, for FHP, and FHP used fairly large printed circuit boards. I think at the time, the Nova used 15 by 15-inch boards, Nova and Eclipse, and this was larger. I think-- I don't know if these are 15 by 17 or, but it was a departure from the past because they just needed to put more chips on the boards.

Hsu: So shall we move on to Rational?

Davidian: Yeah, the transition. So after-- well, once FHP's fate became pretty clear, Rich Belgard, who I had sort of followed-- he's the one who brought me in and he knew Gideon and he knew he went to the University at Buffalo-- he had moved on. He went-- he moved to the West Coast, to California, and he was working for Tandem and Tandem was sort of a spinoff from Hewlett-Packard. Well, a lot of ex-Hewlett-Packard people founded Tandem, and Tandem was doing fault-tolerant transaction processing computers. So, and they were microprogrammed and they knew what microprogramming was so Rich was-- yeah, I would keep in touch with Rich after he left. And he told me that there was this startup company that he was talking with called Rational, and he was probably going to go work for them. And so I was thinking of leaving Data General at the time because there was really, it became clear there was no future for FHP and, at the time, Research Triangle Park was in somewhat of its infancy. It was-- there was IBM. There were some drug companies but it wasn't-- the area there really wasn't a high-tech area and the main thing in North Carolina was tobacco farming. So I figured I was going to wind up in California one way or another and so I thought I'd interview out there and so Rich was going to go to-- he had decided to leave Tandem to go to Rational. And I said, "I'd like to interview at Rational," but I said, "I'd also like to interview at Tandem" because one of the things, one of the people at Tandem was Carl Alsing, and he was, in *Soul of a New Machine*, he was one of the leaders of the microprogramming team there and he had also written the, all the microcode for the Eclipse S/200. So he seemed like someone that I would actually like to work with so I wanted to just see what Tandem was doing and then also see what Rational was doing and Rational was an interesting company. It was started by two people that were actually from the United States Air Force Academy but they were going, they were also involved at Stanford after that and they were also working in-- they say they had a longer-term commitment to the Air Force and so they were also working in the Blue Cube, the-- I don't know. I forget what they called it at the time. It was this windowless building near Moffett Field, and it was the-- oh, it was called the Satellite Control Facility, the U.S. Air Force Satellite Control Facility. So while working where, they just-- they had a lot of observations about the state of software development and they, their feelings was that there was this software crisis and Rational-- that Rational's goal would be to mitigate the software crisis. And one of the things that was going on [at] the time was the Department of Defense was, wanted to develop a standard programming language for defense contractors, and there were a number of different bids, proposals for it. There was this one that was referred to as Green. It came in a green binder and that was the one that was eventually selected and that was the one that eventually became called-- that became Ada, the Ada programming language. And Rational's goal was to build a programming environment for Ada and there were a number of people at the Stanford AI Lab that were also interested in Ada and they got recruited into Rational and so there was a lot of early development that was actually done in Lisp on a DEC-20. So that was kind of the early development of this programming environment but Rational wanted to build their own computer, a large computer to run this programming environment and it was going to be microprogrammed and it was-- they needed a team to do that. So it was designed to run Ada and also to run this programming environment that was written in it, was going to be written in Ada. So I was, I think, the 20th employee that was hired in this startup. A lot of us, in the beginning, actually came from Data General, some from North Carolina, some from Massachusetts, and then there was the Stanford AI Lab people. So we set to work and Rational also had to-- I didn't realize it at the time, but they had a very interesting board of directors and investors and it was actually, when you look back at it, it was fairly impressive. You-- when I hear name, some of the names of people that I would run into at the company, so Arthur Rock was one of the first investors and he was-- he had, one of the initial investors in Intel and

Apple. Bill Perry, William Perry was, became Secretary of Defense. He was on the board of directors. We used to have the company picnics at his house. They had, Hambrecht & Quist was some of the venture capital. John Arrillaga built our buildings. I'm trying to think of some of the other names. Wilson...

Hsu: Sonsini.

Davidian: ...Sonsini, yeah. Larry Sonsini was, I think, was on the board of directors. So it was just this pretty interesting group, and one of the first customers we were going to have was Rolm Corporation. They actually-- well, they made a couple of things. They were kind of a weird company because they sort of had this defense business where they built these sort of mil-spec Data General Novas and Eclipses, I think Eclipses, definitely Novas, and but then they also had this telephone PBX business. But it was the defense business that was interested in the R1000 and, but they were also, they were interested in getting Ada, an Ada compiler running on the Data General MV/8000 architecture. So that was one thing that Rational did was built an Ada compiler that ran on the MV/8000 and it was actually the very first validated, the first compiler to pass the DoD validation. So, but at Rational, one of the-- I think about sort of connecting the dots between things from my past. One of the dots that sort of became useful in the future was the initial computer, was it was designed as this massive-- it could have four CPUs in it and there was an I/0 processor in it that was actually a PDP-11/24. And going forward for a second product, we wanted to make just a single processor and a more simplified I/O processor. So for the I/O processor, we actually used a Motorola 68020, and this was in 1980-- well, we probably started this in 1985, maybe, and Apple, the Apple Mac II was the first 68020 from Apple and that came out in 1987. So this was a couple years before Apple used the, had a product with the 020 so for this product, I actually got to work on the I/O processor. So the software that actually, that had previously been running on a PDP-11, I was re-implementing on a Motorola 68020. So I-- that's where I started getting some of my Motorola, the 68K experience. Also during that time in 1985, I bought a 512K Macintosh and, which had a 68000 but a lot of tools on it, the debugger. So I was actually using the Macintosh a little bit for some of the development at Rational. Just it was a sort of a parallel learning experience so that was where I started on my 68000 skills. Actually, the reason I bought the Macintosh, it was not because of the graphics, graphical user interface or anything like that. It was because it had a 68000 processor in it and that's an actual-- another dot while I was at Rational was Rational had a, there was a cubicle they had that was the library and they had some magazines and things there. But there was-- I think it was the April 1984 edition of Byte magazine and it had an interview with Steve Jobs and the Macintosh development team and I read through all that and I said, "That's, that sounds like a lot, it sounds like it would have been a lot of fun. I really wish I could have been involved in that and could get to meet those people." So the, but that was-- I read it and just kept thinking-- I would always think about that, especially once I owned a Macintosh and so, and then I was getting 68000 experience. So that was, that went on for several years but it also became clear that as Rational matured that they were probably going to get out of the hardware business and be purely a software company which-- so the future of what I was doing seemed less sure. So the FHP boards were big. The R1000 boards were even bigger, and so this was a memory board. I don't-- this was either two megabytes or eight megabytes, I don't-- depends on which version it was and which density chips they used. So there were four memory boards in each CPU.

Hsu: Okay, and...

Davidian: This was called the FIU board. This was the field isolation unit. It was a bit-addressable memory, so this was where all the shifting and alignment and extraction and insertion occurred. So this was-- I was somewhat involved in the design of this. I spent a lot of time trying to get things to happen in one clock cycle that I was being told by the designers couldn't happen in one clock cycle. So while I was at Rational, Rich Belgard was another person I was working closely with. He eventually left Rational and he was working with some venture capitalists at the time and then got to work with some intellectual property lawyers doing consulting. And at one point, I was approached by some intellectual property lawyers who were interested in me being, me helping them out for a large intellectual property lawsuit and it would have meant that I would have to leave Rational. So I thought about it for, a lot, and I, it definitely seemed like something I wanted to do but Rational was also something I, yeah, I was enjoying. So I talked with the founders about taking a leave of absence and the two founders, one was more of the technical lead and the other was the business lead and the technical lead I talked with, and he was saying, he said it was-- I could do a leave of absence, come back when it's done and everything would be fine. As it got close to time to leave, I met with the business lead and he said that Rational doesn’t want to sort of set any, this precedence where people can just come and go and that I couldn't come back, which that hurt a lot and but looking back at it, it was actually a very good decision. So I wound up leaving Rational and to do consulting for a law firm and this law firm was involved in, they were representing NEC against, in a case against Intel and this was over the microcode for the Intel 8086 processor.

Hsu: Did you know at the time they approached you that that was the case?

Davidian: I knew that that was the case. I didn't know anything about the 8086 or the microcode for it or the NEC processors.

Hsu: Well, that was why they...

Davidian: Yeah, that was-- yes, they wanted me, they wanted to do a clean room where someone with no exposure would implement the microcode just from written specifications and there would be a very well-documented paper trail of every bit that went into it so...

Hsu: What was it about that that was so compelling to you that it motivated you to leave or at least be willing to leave?

Davidian: It was actually, there were sort of a lot of things, like going back to Data General. There was a team of microprogrammers working on FHP. It was not an individual effort. It was a big enough thing where multiple people had to do different parts and then at Rational it was a team, also, and with this it was going to be just me doing the whole thing and it was also the IBM PC was out. This was a very interesting processor to do it for. This was-- this seemed like a very big accomplishment that I really wanted to do. This was a challenge and I really, I wanted to do it and succeed at it but going into it, I do-- I was in the dark. I didn't know anything about it. Meanwhile, Rich Belgard was also working with these lawyers in developing the specifications that I would, that would be given to me-- so he and Gideon Frieder was NEC's expert witness in the case.

Hsu: Okay, so that's how-- yeah, okay.

Davidian: So, but that was also, it was presented a very difficult situation where now I was not allowed to talk to my friends. So, but going into it, since I needed to be clean even through the contract negotiations, I knew nothing about-- I knew none of the technical details about the microcode, microarchitecture, anything. So all estimates as to what it would take to do this were made by NEC and I'm sure they had their experience with how long it took their microprogrammer and then they had, they probably had some input from people who knew me. And, but it was also in this environment that they had no experience with, which they didn't know what kind of delays that was going to add. So they were also in a hurry because this was being done as an insurance policy for them. The case had actually gone to trial when they were talking to me but the judge hadn't ruled on it yet and so they were, in case they lost, they didn't want to have to withdraw the, their product. They wanted to have some alternate microcode to use, so they could continue to have a product. They were a second source to Intel and they had all the rights to the hardware but Intel was then saying that, no, the microcode was not part of the hardware. So they, that-- they could not use Intel's microcode. NEC also-- they had made some improvements to the design, so it wasn't an exact copy of Intel's hardware anymore. They had actually made some things that could make it perform better and they had, they were on microcode for it, but I don't-- they certainly had access to Intel's microcode when they were developing it and I don’t know how much they, that influenced their microprogrammer but Intel was claiming that there were all these similarities and it was definitely evidence of copying. So NEC's position was that these were-- this wasn't copying. This was due to constraints of the microarchitecture. This was the only way or the most logical way that anyone would do this and so in addition to this being an insurance policy, it also turned into sort of evidence that someone who had never seen either of the microcodes was producing microcode that looked similar to both of them and it was not copying. So they were in a hurry to get this completed because they were also, they were afraid that there might be an injunction or something like that, and so they felt that it would take six months to do it. But they, as incentives to get it done more quickly, they had-- there were bonuses based on each day that it was early with higher bonuses towards the-- for days in the first few months and then the bonuses got smaller as it got closer to the end of their six-month period. And this sort of got negotiated with, between me and one of their lawyers. They had a couple of younger lawyers that, I think, came from Fenwick & West and it was-- they were using a law firm here in Sunnyvale even though they're a Japanese company. So it was this law firm, Skjerven, Morrill, MacPherson, Franklin, and Friel, I think was the name, and the two lawyers that I dealt with mostly was Dan Siegel and Doug Derwin. And so Dan Siegel was the one who was negotiating the contract with me, and we actually met for the first time at this place called The Boardwalk that was on El Camino and Mountain View. It just closed a few years ago. It used to be-- and so it was negotiated over a pitcher of beer, and it was-- I still didn't know what I was going to be facing as far as how challenging the work would be. So I don't remember how long it took to get everything in place because they were, they had to build all the development tools and write all the specifications but eventually, I think it was the beginning of August of '86 that I started work on it and I was doing it from my home in Mountain View. And they estimated it would take six months and I started-- I'm always optimizing for something, so in this case, I was optimizing for dollars. And well, and there was another thing going on which was before all this. One of my good college friends was going to come visit me, him and his wife, and we had made these plans in advance, and that was about in mid-August they were going to come visit me. And there were things built into the contract where, if we reached some sort of an impasse where I was waiting for them for something, we would stop the clock, and so I was hoping to get to a point where we could stop the clocks <laughs> while my friends were visiting <laughs> besides the dollars. But as I got into it, they had some verification tests, and my first goal was to actually implement enough of the simple instructions so that I could run one of the verification tests just so there's enough implemented. And so I got that going, and I started running some tests and debugging, and it was moving along, so I would just go one test at a time <laughs> and implement instructions as needed. And there was also this back-and-forth where I would send them status and questions each night, and they would work on answers, and a lot of it came from Tokyo, so there were some time zone differences and things like that. So usually I'd work late. I'd send off the questions. I'd go to sleep. I'd wake up. There's answers. <laughs> I pick up where I left off, so it was kind of well pipelined <laughs> to work continuously. So that continued. Yeah, I was making good progress, and after 15 days I had passed all of their tests. And so according to the contract, I was five and a half months early <laughs> in bonuses on the six-month project. And so there's actually a book called *Inside Intel*, and there's a chapter about me. The title of the chapter is "Davidian's Bonus." <laughs> So besides it being a very interesting project, it also was financially rewarding.

Hsu: The chapter says that you got paid a bonus of 200,000 or something like that?

Davidian: The author asked me when he interviewed me. He asked me how much I got paid, and I said "I'm not going to tell you." And he said "Well, someone told me this number." I said "Well, you can print whatever you want, but I'm not going to tell you." <laughs>

Hsu: So the number he made up.

Davidian: So that number he made up. There probably is some court transcript someplace that says what it was, because that actually did get argued in court. I don't know, Intel's lawyers brought that up as saying "You're trying to show that somehow this was tainted in some way," I don't know, because of how much I got paid. <laughs> I was in the dark. <laughs> I didn't know. And as it turned out, the project really wasn't done at that point. It passed all their tests, but they were finding that their tests didn't have good enough coverage; their simulator didn't quite match the real hardware. So after my friends left, we restarted things and continued on, and in the continuation I think I was getting paid by the hour for that, so it was actually additional money. But it was definitely a challenging project, and I think there were times where I was really frustrating the lawyers, because once things were working, I was still trying to make things faster. So I started changing things that were already working, and so now there were multiple versions of routines, and one of them might've looked very good saying that this exactly matched <laughs> what they were accused of copying, but now there's a totally different version of it <laughs> that looks nothing like anyone else's. <laughs> So I think at times they kind of wanted me to just-- <laughs>

Hsu: Stop.

Davidian: <laughs> Stop. The lawsuit then took a couple of interesting turns. So when I started working on it, the trial had ended; the judge had not ruled. Since I completed it so quickly, it turned from being an insurance policy to being new evidence. They got the judge to reopen the case and accept my work as evidence in the case. Then <laughs> somehow they discovered that the judge owned I think 80 dollars' worth of Intel stock. <laughs> And it was on a disclosure form that he actually signed, and it was through some investment club, and I don't think he realized that he actually owned it, but he did, <laughs> and they got the judge replaced. <laughs> So there was now a new judge, and they essentially went through the trial again, and I testified in the new trial. I just testified about my work. Intel's lawyers at the time were Brown and Bain. They were also Apple's lawyers, <laughs> and at the time of my deposition, I was actually working for Apple. <laughs> There were a lot of delays. I think my deposition actually-- the first day of my deposition I don't think I actually said anything. There were some arguments from the lawyers, and then I think there was a postponement for a year <laughs> until the second day <laughs> of my deposition. So, yeah, when the deposition resumed, it was now these lawyers that were also Apple's lawyers that were giving me a deposition, and I was an Apple employee at the time. <laughs> So it got interesting. <laughs>

Hsu: Because you're on the opposite side in the case.

Davidian: Yeah, yeah, yes. Yeah, yeah, yeah. <laughs>

Hsu: But you work for another company they represent.

Davidian: Yeah, and there were questions in deposition: "Well, what do you do at Apple?" and I say "Well, I can't tell you, <laughs> but you could ask people." <laughs> So, yeah, I did enjoy one part of the deposition, because the lawyers are always thinking-- when they ask a question, they know the answer before they ask it, so they're expecting a certain answer. Well, in the whole clean room process there was a loose-leaf book that had the specification, and they would periodically send me updates to it, and they would come with repla- the updates would be replacement pages, and I'd take the old page out, put the new page in. So during the deposition, they handed me one of these packets and said "Did you receive this on such and such a date?" Seemed like a simple question <laughs> that was a simple yes answer, but I said "No." <laughs> And so they're trying to figure out "Okay, why did he say no?" <laughs> They said "Well, are you sure?" "Yes, I'm sure I didn't receive it on such and such a date." "Well, how can you be sure that--" <laughs> and I said "Well, look at the dates on some of these pages. They're not that date." <laughs> And they're saying "Hm." So they thought about it a while. They're trying to figure out why <laughs>. They said "Well, can you explain that?" <laughs> I said "Well, okay, so this is what-- <laughs> I would receive the new packet. I would put the new pages into the binder, I would take the old pages out and put them in the packet, so this is now a packet of old pages that I removed <laughs> from the specification instead of a packet of new pages <laughs>, so this was not what I received that day; it's what I created," <laughs> and it was created of old pages and <laughs> not of new pages. So that was just one of the things where they had their idea of what the answer should be, and it was just an unexpected answer but a correct answer <laughs>. So it eventually went to trial, and I was at Apple when the trial occurred. I testified at the federal courthouse in San Jose, and I think I was there for at least one full day. And eventually the ruling came out that-- the lawyers always use the shotgun approach. They were saying "Well, microcode can't be copyrighted. Even if it could be copyrighted, it wasn't copied, <laughs> and if could be copyrighted, well, Intel didn't mark the copyright. The copyright wasn't filed properly." There were all these different ways to fight it. And in the end NEC mostly won.

The one thing Intel won on was that the judge ruled that microcode could be copyrighted, but he ruled that Intel didn't properly mark the copyright notice on it.

Hsu: On the chips.

Davidian: On the chips, yeah.

Hsu: On the third-party chips, on the NEC--

Davidian: No, well, even on their own chips <laughs>.

Hsu: Oh, okay.

Davidian: Yeah, and they didn't have the second sources mark it. And the judge also ruled that and then even if it was copyrighted correctly, it wasn't copied. I think some of the words were "the clean room microcode was compelling evidence that there was no copying." And it was interesting during the trial; the judge had to be educated. There were two expert witnesses, Gideon Frieder on NEC's side; Dave Patterson was Intel's expert witnesses. <laughs>

Hsu: Oh, interesting.

Davidian: And the judge actually had to learn about microprogramming and just x86 instruction set architecture, just a lot of things for a judge to absorb, but he was actually very good at really seeing it and recognizing things. In the clean room microcode, I took advantage of some of the hardware changes that NEC had made. I took more advantage of it than NEC did <laughs>, and so in many cases, my microcode actually performed better than NEC's microcode. So, yeah, he recognized that I did things like that.

Hsu: But unfortunately, because NEC won the case, they didn't actually need to use your microcode?

Davidian: That's correct. <laughs>

Hsu: Even though yours was better in some ways.

Davidian: And they paid a lot of money for it. <laughs>

Hsu: Yes.

Davidian: But they won, so they didn't have to use it. But they did produce some chips; they actually did fab some of the chips beforehand when the project was completed. And I do have—I could show—take a little stop here.

Hsu: Actually, one more question, though. I think the chapter in the book, *Inside Intel*, mentioned that you were actually given a choice of two different contracts, and because you didn't know enough at the time, you couldn't make a decision, and so you worked with the lawyer--

Davidian: I think I was given a choice with the bonuses.

Hsu: Oh, okay.

Davidian: I think whether it would be a flat bonus per day or whether there would be this higher bonus for the earlier days. Yeah, I don't remember exactly, but I think it might've been where they were proposing one way or the other, and I didn't really have any way to guess. And I don't remember what the book said. I haven't read <laughs> the book in a while, but it probably was fresher in my memory when I was talking to the author about that, so probably whatever the book said was--

Hsu: Right, okay. Yeah, because it says that you didn't have an opinion as to which one, but you ended up going with the one that was loaded upfront.

Davidian: Yeah.

Hsu: And ended up getting you the massive payday.

Davidian: Yeah. One of the lawyers said "Imagine pulling the handle on a slot machine <laughs> and having it ring its bell <laughs> for a week." <laughs> So, yeah. So these were two souvenirs that NEC produced after the trial. This one has one of the chips that they fabbed that has my microcode in it, and this other one has one of the pages from the judge's ruling where it mentions "the Davidian microcode was compelling evidence." And there's a little mark on this one. It's where this fell off a shelf in my office at Apple during the 1989 earthquake, <laughs> so that's where the little smudge came from. But this was a real highlight in my career. This was a big, high-profile project.

Hsu: So you mentioned that the second deposition was a whole year later, so your part of the project, the actual development work, was finished before the trial was completed, correct?

Davidian: Yeah. There was this new delay because of the judge owning the stock, <laughs> yeah, so they had to get a new judge, and then I believe they redid the entire trial in front of the new judge. So I was involved in the new trial, and there was a lot of stuff that went on before my deposition. My deposition-- I'm sorry, not my deposition-- my testimony was actually quite boring. It really consisted of me reading my deposition <laughs> into the record. I didn't catch onto this at the very beginning when the lawyer was asking these questions. I was thinking "Okay, this question sounds familiar," and I just answered it from my memory as I would've answered it if anyone asked. I didn't realize that it was-- I knew they had given me my deposition transcript in front of me, but I didn't realize that this was verbatim from <laughs> the transcript at the time, and then it was kind of pointed out to me, and so then, okay, I'm just supposed to read this. <laughs> To an observer, it seemed like the lawyer was really grilling me, but in fact she was asking me questions that I had been asked a year before <laughs>.

Hsu: Oh, okay. Right, when the previous judge was still on the--

Davidian: No, the trial actually happened in ’88.

Hsu: Oh, okay.

Davidian: And I think it was in ’87 that the deposition happened.

Hsu: Okay. And so this is long after--

Davidian: It was August ’86 that I was writing the code.

Hsu: Okay, right. Yeah. Okay, so there’s actually a significant gap.

Davidian: Yeah. <laughs> Yeah. Yeah, there’s definitely a significant gap, and also during that gap, sort of the importance of the 8086 and 8088 <laughs> processors was diminishing. <laughs> 286, maybe probably even the 386 might’ve been out by then <laughs>.

Hsu: How disappointed were you that your microcode wasn’t actually used in the shipping product?

Davidian: At that point, I don’t think I was that disappointed. The fact that I got to do it was-- and I also got to know that it actually was compatible. It actually passed all the-- it was capable of being shipped. It actually could’ve been a product. So the challenge was satisfied.

Hsu: And what do you think of the results of the case?

Davidian: I don’t really have much of an opinion. During the trial, there was an exhibit that the NEC lawyers put together, which had three columns. [First] It was the Intel microcode; [second was] they called it the Kaneko microcode-- Kaneko was the name of the NEC microprogrammer-- and then [third] there was the Davidian microcode, and it was just some of these routines or micro sequences that were being contested, where Intel was saying this was copying. So when the case was all over, one of the NEC lawyers gave me a copy of that exhibit, so that was the first time I actually got to see anything, and it was after it was over. <laughs> And I didn’t really have any opinion about whether it was copied or not. I don’t even know whether the selection of which sequences they picked to argue about was done well <laughs>. And it was many years later that it was over, and I was onto other things.

Hsu: But did you have an opinion of the ultimate legal consequences for the--

Davidian: Yeah, well, yes. That was interesting, because in the ruling, the ruling was that microcode could be copyrighted, but Intel did not properly mark it; they did not properly copyright it, and even if they did, it wasn’t copied. The Mercury News article on this, the headline was “Major Victory for Intel.” <laughs>

Hsu: Oh, really?

Davidian: Yeah, and the last line in the article was "Intel has not decided if they're going to appeal." <laughs> So the only victory for Intel was that microcode could be copyrighted, which I think for Intel, that did help them, because then they were later using the same strategy with their second sources for the 286, 386, 486, and I think they figured out how to mark things. <laughs> And I did think saying that microcode could be copyrighted, that it was essentially just like software, was the proper decision. So I do think that was good.

Hsu: So, yeah, shall we move on to you joining Apple then?

Davidian: Yeah, so <sighs> once I was done with the NEC project it was time to think about what to do next, and I did think hard about the whole microprogramming industry, the field, and I felt that it was a dying field, and there were probably going to be only a few companies that would be doing microprogramming, and I thought it would be Intel, Motorola, IBM and DEC.

Hsu: Is that because of the shift to microprocessors?

Davidian: Yes.

Hsu: Was RISC part of that calculation also?

Davidian: Probably in a little bit of a way, but I think in some ways it was Moore's Law. It was the number of transistors was getting greater, and the ability to hardwire was coming. Microprogramming really was an implementation technique because it simplified the logic, and the transistor growth was eliminating that need.

Hsu: I see. Yeah, because we've been looking at a lot of-- the Alto was a microcoded architecture, and a lot of the people we're talking about, all the benefits of that in the '70s-- so this is an interesting sort of shift that you're mentioning is that-- so microcode was really beneficial in the '70s, and it was losing that; the benefit was sort of no longer-- in the '80s?

Davidian: Yeah, it was that, and it was-- microprogramming really started in the late '50s. It actually goes back that far, and probably the biggest use of it or the case where it really shined was the IBM 360 architecture. It's where they had one instruction set architecture and different hardware implementations, and they were all microprogrammed to implement the same instruction set architecture. So that was really where--

Hsu: That sort of proved the case.

Davidian: That's really where microcode showed its advantage and where a whole family could be produced of varying levels of performance and cost yet retain the software compatibility. But that's why IBM was one of the companies on the list that I <laughs> thought would continue, and I think IBM still is probably still right. <laughs> There's still microcode <laughs> on the-- I did feel that the microprocessors were turning into what were minicomputers before, so whole companies were disappearing <laughs>. The microprocessors were killing off companies that would've been writing microcode. And the DEC VAX was going to live on for a while, and x86, 68000 and the 360, 370 architecture; that's where I saw the future. But when I moved to California, one of the companies that was right across the street from Rational was Sun, and at the time, it went from all these minicomputer companies to they were referring to them as JAWS, "just another workstation," and it was all these 68000-based UNIX workstations that were popping up everywhere. That's what all the companies were doing at the time, so that was sort of replacing the minicomputer industry. Yes, so after NEC, I realized that microprogramming was a dead industry or a dying industry and that it was time for a career change, and I was thinking about "Well, what do I want to do next?" And I had financial freedom at that time to think about it. I definitely wanted <laughs> to continue working, but <laughs> I wanted to do something that was fun, something I'd really enjoy. And I think back to reading that Byte Magazine interview with the Mac team, and I was thinking I'd really like to work at Apple. Well, 1985, Apple saw a bad time. That's when Steve Jobs got kicked out and they had running financial problems with the Macintosh, and they were not hiring, and I also didn't really know anyone at Apple. Up until this point, sort of networking is how my career developed. And so I just kept thinking about it, and then it was in February of '87, and there was a Mercury News ad from Apple. They had four job openings. There were four positions they were advertising for in the Macintosh system software group. And I looked at them. I said "Well, I could do all those," <laughs> so I sent in my resumé, and I heard back from one of the recruiters, their HR person, and did a phone interview, and he said "Okay, we'll get back to you." So they got back to me. They brought me back for interviews several times. This went on for months. <laughs> And I talked to different groups, different groups, and I was wondering "What's going on?" So finally they made me a job offer, and I had to think about it a bit, because it was actually a cut in pay from what I was making at Rational. But it was a career change, and the guy that was hiring me, he knew what microprogramming was, but he also was saying "This is sort of Apple's pay scales." But he said "We can do things like we can give you a signing bonus, and we also do performance reviews every six months, and we can sort of guarantee that your next performance review will get you back up to equal." But he said "We're taking a risk, you're taking a risk, so we'll see how things work out," and that seemed fair. But I also warned them that I'd probably have some issues with signing their standard employment agreement and nondisclosure agreements and things like that due to the Intel NEC <laughs> case so that I'd have lawyers that needed to look it over.

Hsu: Oh, right. And that also you were still involved in the case and still had--

Davidian: Yes, and that I would need to be involved in that. So that didn't turn out to be much of an issue. So when I finally got hired, I said "What took so long?" <laughs> and they said "Well, we knew we wanted to hire you. We just didn't know what project we wanted to put you on." <laughs>

Hsu: There's too many choices.

Davidian: Yeah. And it turned out to be none of the things that were in the ad, <laughs> but it was still in the Macintosh System Software group. So the project they actually wanted me to work on <laughs> was kind of odd based on what I was expecting. It was actually to write 6502 code. <laughs>

Hsu: Really?

Davidian: Yeah.

Hsu: So that was the floppy disk driver and ADB manager for an I/O processor?

Davidian: So there was this project going on. One thing I learned about Apple was that, on the hardware side at the time, there were a lot of competing projects, because back then there weren't that many Macintoshes that came out. There were long gaps. It was the 128 and the 512, which were essentially the same machine, then the Mac Plus, so it was from '84 to '86 one machine, <laughs> and then in '87 they came out with the Mac II and the Mac SE. So the group that was working on this project actually came from Lisa, <laughs> and they were really trying to apply some of the things that Lisa did to a Macintosh, and they were sort of looking at the high end. And the project was called Modern Victorian, and one of its features was that it had these two 6502-based I/O processors. And they built a prototype that was this big card cage where every card in it was one of the ASICs that they were designing, but it was a TTL implementation of the ASIC. And so this was something new and different, but in some ways it drew upon some of the work I did at Rational on the I/O processor that was 68020 code there <laughs>, but here was going to be 6502 code. And it was low level. I was getting hardware specs that were like data sheets, so it was kind of the stuff I was used to, because as a microprogrammer, you were the interface between the software people and the hardware people, so you spoke both languages <laughs>. So this was very much something I was used to, so I enjoyed working with the hardware engineers that were working on this, and they enjoyed working with me, because it was like "This guy really understands this stuff." And I was actually asking them if they could make some changes to the hardware, because there's some issues with trying to write a driver with the way they had organized some of the control bits and things like that. So they were very receptive, so I got to know this team very well. But, yeah, as time went on, there were sort of more hardware engineers at Apple than there were products coming out. So they were all working on competing products or product proposals, and it was kind of like Data General <laughs> where one's going to survive and one's going to die. <laughs> So Modern Victorian eventually didn't make it, <clears throat> but some of the ASICs did. They stuck around, and the people did, and it eventually became-- well, there were actually two other projects that came out of it. It sort of split into one called Four Square and one called F19. Well, in the meantime, since I was in the System Software group, I was working with all of the hardware projects, so I was working with the team that was doing the Mac IIci, and we were also-- actually the Mac IIx, cx and SE/30, they were all essentially the same design. Those were coming out. Those were really just modifications of-- well, the SE/30 was cramming the Mac II into an SE box.

Hsu: So they were an evolution of the [Mac] II.

Davidian: They were an evolution, but the thing that was going on with those was that the floppy drive was getting upgraded to what was called the SuperDrive then, <laughs> which--

Hsu: Right, the high density.

Davidian: Which was the high density floppy, so I was involved in some of that, because also on Modern Victorian one of the I/O processors was going to use that floppy drive also, so it was working with the floppy people. <laughs> So I was working with the [II]ci group. So I knew what all these different hardware projects were, but I knew that they weren't going to all succeed. So Four Square was using some of the chips out of the ci and some of the chips out of Modern Victorian, and F19 was doing a new memory controller and was going for higher clock rate. So I think Four Square was going to be 25 megahertz, maybe 33, and F19 was shooting for 40 megahertz, and then they were both 68030-based. And so Four Square died, and F19 lived and became the Mac IIfx.

Hsu: And what would Modern Victorian have looked like had it shipped? What would the specs have been?

Davidian: It would've looked a lot like the- it would've looked a lot like the IIfx. A lot of what was in it wound up in the IIfx. I don't think it was 40 megahertz at the time, and the memory controller was different I think. I don't remember what other things in Modern Victorian, but I think a lot of the ASICs did live on to show up in the IIfx. Okay, so this is one of the I/O processors from Modern Victorian, which eventually the same I/O processor wound up in the Mac IIfx. This was the one I was working on that ran the floppy controller, ran the floppy drive and also did ADB. On the back here is the ADB connector. That was actually an afterthought. That wasn't one of the original intentions for this I/O processor. So the logic on this board eventually became just a single ASIC on the Mac IIfx board.

Hsu: Yeah, I notice that's a pretty big board for a floppy controller.

Davidian: Yeah, well, yeah, and there's a 6502 processor on it. You know, there's also two serial ports on it, because there were two of these. One of them did serial, and the other did floppy and ADB.

Hsu: So there were a lot of architectural similarities that continued?

Davidian: In the-- into the IIfx?

Hsu: Yeah.

Davidian: Well, a lot of things at Apple, at that time, were kind of-- they were kind of one-shots. Because, like, the I/O processors didn't-- they didn't really live on. I think they actually got-- they got used in some of the LaserWriters. But, you know, a lot of the-- you know, there wasn't a whole lot of thoughtful architecture going on in the Macintosh line. It was-- partly because there were all these competing groups, instead of one-- you know, one architecture group that was designing an architecture, and, you know, groups implementing it. It was really this kind of Wild West approach.

Hsu: So each group would just do whatever it thought was cool?

Davidian: Yeah. And-- yeah, that's kind of how it really was. In some-- I think there were some things that did sort of wind up being, you know, more system-wide, where some of the-- some things like some of the sound chips, or video, things like those might-- especially when it-- you know, when it came to NuBus cards, you know, then the NuBus card could-- you know, the computer would just have NuBus slots, and the cards-- you know, the stuff that got reused was actually on the cards. So-- but I-- you know, there were just-- during-- you know, I started in '87, and the fx came out in, I think, March of 1990. So it-- you know, there's a long span there, where a lot was happening. The development cycle was really long at that time, partly because of all these competing projects. You know, and going through different lives, and reincarnations. So the-- I was also working on-- I worked on the IIci and the Mac Portable. And there were-- one of the things I was doing at the time there was actually, to make my life a little easier, for development-- for ROM development, there was this in-chip-- or on-chip-- ICE. In Circuit Emulator, is what they called it. And, you know, we had a lab where there was a Mac II board, and so that-- that's sort of how we would test out new ROMs, was the In Circuit Emulator could also house the new ROM image, and boot from that. And you know, it was a slow process, and very limiting. There was only one of them. So I wanted-- you know, I wanted to come up with a way to actually test the ROM image on a-- just production piece of hardware. And this was before flash storage. So I wanted to actually execute the ROM image out of RAM.

Hsu: Oh.

Davidian: And I also wanted to be able to run on lots-- you know, run on all the different machines that were produced. I didn't want to, you know, have to build a ROM for-- you know, for a Mac II or a Mac IIx, you know, all these differences. So-- well, the ROM source code was all-- it was sort of based on hardware products. And so it had all this conditional assembly, based on, you know, was this for the-- are we building for a Mac II? Are we building, you know, for a Mac Plus? So I wanted to change things. So, you know, we were adding all these new machines that had a lot of similarities. So, you know, building for a Mac-- you know, if it was a NuBus machine, it was, you know-- a lot of the places where it says it's building for a Mac II, you had to go change that to say a Mac II or this new machine. So any new-- so I wanted to change the source code so that, instead of being based on the products, it was based on the features of the products. So, you know, so does it have-- you know, and you would just set bits saying, you know, it has NuBus, it has, you know, Color QuickDraw, it has, you know, it has 32-bit addressing. It has 68020 or 30. So I was doing a lot of massaging of the ROM sources to do this. And then I was also doing changes to the code at startup, to detect what kind of hardware it's running on, so that things that weren't done at compile time, there were-- you could make decisions at run time as to what hardware is actually there, and we could build what I was calling a universal ROM, that could run on multiple different hardware products. So, that also allowed, you know, easier testing. Because a lot of the ROM was the Toolbox, which was not dependent on the hardware at all, it was-- so that didn't really have to change from product to product at all, that could-- the bits could stay exactly the same. And so the-- you know, they really, you know, since you're not changing anything, you know, there's less likelihood that something's going to break. It also simplified the patching. In the early days, there was-- you know, the system was based on a-- was floppy based. It had to fit on a floppy. And it was also-- it had to patch the ROM-- you know, fix bugs in the ROMs. And so it had to do all this patching. And for all these machines that has slightly different ROMs in them, you know, there had to be different patches. And with these-- with the universal ROM, there was, like, one set of patches that applied to a number of different products. So it simplified some of the software maintenance. So--

Hsu: Did this work lead, later on, to the New World architecture? Was there a relationship?

Davidian: I forget what the New World architecture was defined as. What that-- because there was--

Hsu: I think there was-- I think, partly, it was--

Davidian: Was this post-- after Steve Jobs or was this before Steve Jobs?

Hsu: I think this was just before, but I think it was--

Davidian: Was this like the--

Hsu: I think it was party-- part of the CHRP spec, and also it-- they moved the ROM into RAM, so that a lot of the clones could do that. And then, I think, it was finally implemented in the Power Mac G3 Beige Series, I think.

Davidian: Yeah, that was-- that was different. It was-- it was different than what I was working on. That was-- I wasn't at Apple at that time. I do remember hearing Apple kind of struggling with that, and I remember talking to Jon Rubinstein, saying, you know, I did this years ago. You know? But my goals were actually just to make the software easier to maintain, and also, to allow it easier to test builds, as you were developing, so we didn't have this one, shared lab machine. We could do it at our desks. Because when I'm developing code, one thing I always want is quick turnaround. Make a change, quickly test it, and say, okay, that's done. Next. You know, I don't have to think about that anymore. Let's think about something new. And that's what that-- you know, that was another benefit of that. It was also so I could work at home. There was-- you know, a lot of things were personal motivations. You know? But I-- it was a fun project, and it wasn't actually-- it didn't really start as a project. It started as something I just wanted to do. And, in some ways, it was also to make sure that I understood the hardware, you know? Whether, you know-- was all the written documentation correct. And that, you know, that it actually did behave the way the documentation says. And I got to test it on all these different machines. So.

Hsu: You mentioned you worked on the IIci and the Portable. Could you say a little bit about what you did on those two?

Davidian: Yeah, well, the IIci was actually the first machine to ship with this new, universal ROM. And one of the things-- it was also the first 030 machine that was done with the 030 in the design to begin with. The IIx was really a Mac II with an 030 slapped on. And so I was doing-- there were a lot of things I was doing to pay attention to performance. The memory controller-- there was this burst mode where it would suck in four words at a time, where the-- there was a longer delay for the first one, but then the three following it were quicker. And so, knowing that-- you know, how these blocks were aligned, and the speed of things, and that you were going to be fetching these other blocks-- these other three words in the cache block, I would align code and try to pack code better, so that they would, you know-- things would work better, with the burst fetching and the caching. And-- I'm trying to think of all the other things. I-- you know, I worked on the floppy driver that was in there. I don't remember if we shipped this or not.

But when I was working on the Portable, I worked on the ROM-- the Portable had a RAM disk. And I know, internally, I had the RAM disk running on the ci. I don't remember if we shipped it or not. But I remember we used to-- we had a couple machines where we actually had the maximum amount of RAM in it, or the max we could afford at that time, and had a big RAM disk, and would actually do the build off of the RAM disk.

Hsu: That's what every programmer wants to do.

Davidian: Yeah, it was-- so it was a really well-performing machine. And-- oh, one of the other things that I put in during the development of that ROM was-- and it was all-- it was actually just for internal use. It was-- but everyone knows this now, is the reset PRAM sequence. And everyone has it wrong too. Everyone says it's command option PR, for PRAM. That's wrong. It's command option RP, for Reset PRAM. Of course, it's the same, because you're pushing the same keys all at once. But I named it. That's what it was. And it was implemented to fix a specific problem in the lab. When we were developing this, when I had the universal ROM, or the new ROMs, and we were actually testing it on a Mac II logic board, there was-- we'd made some changes where the way some of the information for each NuBus slot-- each card had a little bit of PRAM allocated for it. We changed the format of some of that. And it-- we ran into a problem where, if you-- if we booted the new ROM on this Mac II logic board, when you went back to the old-- the original ROM on it, it wouldn't boot, because the contents of PRAM looked corrupt to the-- you know, it looked wrong. It was confusing the card, or the slot-- the slot manager, or something. So on the Mac II logic board, the battery was soldered to the board. So you couldn't just pull the battery out to clear the PRAM. So, to solve this problem, I implemented this key sequence. And starting with the Mac IIx, the battery was socketed. So it wasn't needed, you know, there was an alternate way to fix this problem. So it was to fix a problem on something that never shipped. That's why it was there. But, you know, it did come in handy for other things, you know, because it saved-- you didn't have to pull the battery out, and--

Hsu: Right, for the user, it's much easier to press several keys, rather than to open up the case.

Davidian: Yeah, but we never even thought that there would be a case for the users to want to reset PRAM.

Hsu: That's true. It's become one of those things that a user does.

Davidian: Yeah. And the-- and there's all this folklore that you have to do it three times or something. So that's where it came from. It was to fix this bug, because of a soldered-in battery.

Hsu: So what are the actual, legitimate cases that a user might need to do this?

Davidian: I've actually run into a couple where I've-- where reseting PRAM actually fixed it. Because, these days, they're storing all sorts of stuff in PRAM. They're actually changing the way the boot sequence works. And--

Hsu: You mean in modern Macintoshes?

Davidian: The modern-- yeah, in the Intel Macs. Yeah. You know, decades later. So there have been some, where-- because they do some stuff where, when it does a system-install, where, you know, it reboots as part of it-- as part of the install process, and it knows that it's part of that, and sometimes it kind of doesn't erase that information, if things got interrupted in a funny way or crashed, or-- so sometimes, you know, things are left behind that, you know, I've experienced, and reseting PRAM does--

Hsu: But on those machines of the day, like the IIci, you know, those machines, what would cause-- what would be legitimate reason for the user to reset PRAM?

Davidian: There-- I would think, the only things, then, if you wanted to get it back to the out of the box settings-- because I think it's-- I think there were things that were saved, like the keyboard repeat rate, and the mouse scroll-- you know, the mouse acceleration rate, the double-click rate, things like that were, I think, were stored in PRAM, that, you know, people might have wanted, you know, if they wanted to get back to the factory settings on those. But there really was-- we never intended to document it. I don't know how it got documented. You know? It might have been Developer Technical Support in a TechNote, might have mentioned it. But it was just for us.

Hsu: And then it got out, and-- I mean, was-- were user--

Davidian: Well, we left it in, because we did think it might-- you know, it was useful for this case we hadn't thought of, it might be useful again. No need to remove it. It also, that-- the same code that was detecting that key sequence, we did something similar for the hidden ROM pictures, because they also used a command option sequence to activate those.

Hsu: The hidden ROM-- so there were secret pictures in the ROM, that you could--

Davidian: The IIci, there's a picture of the ROM team. And I think the date has to be September 25th, 1989, which was the date that the ci was introduced. And I think it was command option CI. And during boot, if you hold it down, it will bring up this ROM picture. And, you know, the date was actually picked so that we could do it easily at the introduction, if we-- and that was one case where we actually had to know the name of the product before the ROMs froze, which was actually a kind of a rare thing. And there's a similar key sequence on the fx, where we had to know the fx product name ahead of time. So it's the-- and it's the-- you know, for the fx it's the same picture, but it's the different date and different key sequence.

Hsu: Oh, okay. Just fun little Easter eggs. These products all came out when Jean-Louis Gassée was still there.

Davidian: Yeah.

Hsu: Can you maybe talk about what he was like as an executive, and, you know, what his influence was?

Davidian: I liked him. You know, I-- he had an office that was near-- well, when I first started there, we were all in De Anza 3. We were on the fourth floor of De Anza 3, and he had an office there. I don't think he was there all the time, but you'd run into him. You'd talk to him. And he seemed-- you know, he understood what we were doing. It was-- but he left us alone. It-- I didn't really feel like there was a whole lot of supervision going on, you know? We got--

Hsu: But he was generally supportive?

Davidian: He-- oh yeah, yeah. And I do remember one thing, when working on the Mac Portable, I had to-- there was a feature they-- that the hardware is supposed to have. It was called-- they were called SLIM cards. They were kind of like, in later portables, they were PCMCIA cards. But they were just-- they were memory cards, and it was to expand memory. But also-- they could also be like a removable electronic disk. And they were actually RAM, and they had a little coin cell in them, to keep them powered on. And there was just a slot-- I think it might have replaced the floppy-- you know, I think it was an optional thing, and it would go in place of the floppy, on the side of the Portable. And it had some problems that-- you know, one problem was that it was connected, directly, to the 68000 bus. So sometimes, when inserting and removing it, it would cause glitches, and the system would crash. Another problem was, it had a battery that was, you know, keeping the contents alive. And there was no way to read the battery level. So you didn't know if the data would live, you know, long-- you know-- you didn't know when you were going to lose data. And there was also no eject mechanism or locking mechanism, so you could yank it while it was, you know, writing-- so it was unlike the floppies, where you-- you know, there was locking. So I remember, you know, I had to, you know, explain the issues, you know, sort of-- to present the software's opposition to this hardware feature. And, you know, I had-- just had to explain to him, because someone had to make a decision, you know? I-- and, you know, and he definitely understood, you know, what I was, you know, all the reasoning I was giving. And so that was going-- you know, eventually that was going to get-- the hardware people were going to redesign things, and it was going-- the second generation of the Mac Portable would support it. But it never happened, because, you know, this-- even with the second generation, nobody really wanted a computer that heavy.

Hsu: Right. But you feel like your arguments helped Gassée make the decision to kill that feature?

Davidian: I think-- I don't know if I even needed to make the arguments that strongly, because I think he got it pretty quickly that you know, we went through this with floppies, and you know-- the pulling, you know, just being able to not lock it in. So.

Hsu: Did he-- I mean, you mentioned that there were a lot of competing projects. How much influence did he have in deciding which ones should live or die? Was he the guy who ultimately made those decisions?

Davidian: You know, I don't-- I remember him more as the software side. I don't really remember-- well, the Portable was definitely his thing. But it seemed like there was separate, you know, separate hardware management that was making a lot of those decisions about what lived. I forget who it was. So.

Hsu: On the software side, what was-- did you see any overall, strategic vision, or long-term plan, or mile[stone]-- you know, I don't know. What was the long-term, sort of, trajectory, for the system software?

Davidian: It got-- at one point, there was this whole focus on, you know, the new features, and there was the-- you've probably seen the-- Pink and Blue.

Hsu: Oh, right. Yeah.

Davidian: So that was, you know, all these new-- you know, all these new features for the future, and are they sort of short-term or long-term? And the shorter term were, you know, on blue cards, and were sort of intended for System 7, and the longer terms were on pink cards, which were intended for never. Well, not at the time. But that's--

Hsu: Right. Became vaporware.

Davidian: Well, so-- with the Pink and Blue thing, you know, when Pink spun off, my experience at Data General, you know, influenced, you know, my decisions about what-- how things were going to go. And it's like, you know, like I said, you know, okay, well, I've seen that movie, I know how it ends. And-- but you know, that was Pink-- you know, Pink was, you know, sort of this clean sheet of paper, no application compatibility, and a schedule that was two years in the future. Continuously. So.

Hsu: So you stayed on Blue.

Davidian: Yeah, I stayed on blue. But there-- you know, they had what they called the Blue Book, which was all the things that were going to go into Blue. Most of the things that I had in that Blue Book actually shipped in System 6. System 6.05 or 6.04-- 6.03. One of them-- one of them was the Time Manager. And the Time Manager was originally implemented in the Mac Plus, and it was sort of implemented out of necessity, because the SCSI driver needed it, something like it. And it wasn't really thought out very well, and it was implemented even worse. And so there were lots of problems with it, and it was also difficult for application developers to use. So I wanted to solve a lot of those problems. And I also, at the same time, wanted to give a way to do high-resolution timing, to have a microseconds timer-- microsecond-level timer. So-- and I also wanted to have the ability to have sort of fixed frequency interrupts. So even though, you know, the routine got delayed a bit, the next time it got called, it would still, you know, be occurring at the frequency you wanted. So it would-- you wouldn't be drifting, as you-- you know, as time progressed. So, you know, I implemented this new Time Manager, and-- but, in the meantime, in System 6.03, that shipped with the Mac II-- I'm sorry, the Mac SE/30. They were having a problem during testing of that, where running it as a file server would hang, every now and then. And only on certain machines. And it was-- and it seemed to be hanging because a time-- a Time Manager task wasn't getting called when it should be. So they-- you know, they wanted me to debug it, because I knew about the Time Manager. And I said, well, I really don't want to work on the old Time Manager. I'd rather, you know-- hey, I've got this new one, can we just, you know, try it? So I replaced the Time Manager, and the problem continued to happen. So, well, that's interesting. But at least, you know, now it's my code. I know what's going on here. So I debugged it even more and more, and I tracked it down to a hardware bug in the-- there was the 6522 VIA chip, that was used in-- starting in the original Macintosh, and it had the timers that the Time Manager used, and generated interrupts. And it was just-- it was not generating an interrupt when it should. And then, after investigating it more, we found that there were two different vendors that were being used for this chip. There were two different sources. And it was only happening-- this problem was only happening with the ones that were manufactured by Rockwell. So this became known as the Rockwell VIA bug. So I isolated it even more, and I-- you know, and I finally came up with a way to work around it. I figured out exactly what was going on. It was-- had to do with if the counter was counting through zero at the same time that you were writing a new value into it, it would not generate an interrupt correctly. So we actually contacted Rockwell, and we had some meetings with them. And they had bought the design off of-- I think MOS Technology was the company that originally-- the same company that did the 6502. This was a 6502 peripheral chip. So they really didn't know anything about the design. And it took them months to even agree that, yes, they could see the problem happening, and where it was a fairly obvious problem. And then came to-- do we want them to fix it? And, you know, I'd already come up with a software fix for it, and I really didn't trust-- you know, trust any change that they might make to it, given, you know, that it seemed like they didn't really understand their own chip themselves. So we said, no thanks, you know? We'll-- but from that day on, whenever there was any discussion at Apple, any meeting that had anything to do with the VIA chips, I got called into that meeting. So that was-- you know, that was the-- one thing that I, you know, just remember that it was-- you know, once you solve a problem, it kind of follows you forever, even though it, you know, might have nothing to do with the problem that you solved. So-- yeah, that was-- so, then, in System 6.03, my-- the new Time Manager actually shipped. And it was actually good that the Time Manager shipped in System 6, because, later on, there was this other group that Bruce Leak and some other people were in. There were-- well, actually, there was a-- at the developer conference one year, there was a slide that went up, describing this thing that none of us that worked there had ever seen before, and it was called QuickTime. And Bruce and some other people said, well, maybe we should go implement this thing. And QuickTime was heavily dependent on timing and the Time Manager. And QuickTime actually shipped on System 6.

Hsu: So that wouldn't have happened without the new Time Manager in 6.03?

Davidian: It probably wouldn't have. You know, it-- you know, I got to know-- you know, I got to know the QuickTime team pretty well. I used to hang out with them. And yeah, they-- you know, abused the Time Manager in all different ways that I had never thought of. But yeah, they were very appreciative of the Time Manager. And-- although, they-- I think, eventually, they stopped using it, because there got to be this weird-- you know, a lot of weird problems, because there were all different clocks within the computer, and they all weren't quite synch-- they weren't synchronized with each other. So the sound chip was running on its own clock, and-- you know, which was-- which wasn't quite in sync with-- you know, there was variation in all these crystals. And they-- so it varied differently than the Time Manager's crystal. So they would run into problems where the sound and the video were getting out of sync. So they kind of switched to using-- I believe they switched to using the sound-- the speed of the sound driver as their synchronization.

Hsu: Yeah. So maybe talk about the things that you worked on for System 7.

Davidian: I don't think-- I don't know if there was anything left that I was working on for System 7, other than maybe, you know, helping out on some of the-- you know, some of the stuff that went on there. Maybe-- I think the Quadra 700 might have been one of the first to ship with System 7. So I was sort of helping out a little bit on the Quadra 700, but not quite, you know, the same. I was kind of focused on other things then. So maybe some of the stuff to support the Quadra 700. But by the time System 7 was coming out, I was focusing on other things. I was kind of the one engineer that System Software was allowing to work on RISC-based Macs.

Hsu: Yeah. So, in the run up to the-- to you starting on the RISC stuff, between working on the IIfx, maybe just go through the various things that you had done.

Davidian: Yeah. I'm trying to think of all the-- you know, I think some of that was just decompressing. It was-- there was-- so, there was kind of a sequence of events that happened. So it was in September of '89 that the ci and the Portable got shipped. So everyone that was working on that was done, except those that were working on the fx, which was me. So I still had the fx, that came out in March of '90. And in-between that, in October of '89 was the Loma Prieta earthquake, which was a very disrupting thing. The-- we were on the fourth floor of De Anza 3. I was actually standing near the stairwell when it hit. And, you know, the wallboard in the stairwell was crumbling, and we just went down the stairs. I left my keys and my briefcase in my office, and the-- some of the water lines-- well, I don't know if it was the water line. The sprinklers went off. And sprinklers can only be turned off by the fire department, and they were kind of preoccupied that day. So the building was filling with water. It was crumbling-- you know, kind of crumbling. And so it was sort of decided that De Anza 3 would not be reoccupied. There were plans to remodel it anyway, and they just accelerated that. So we were moving into Mariani 1, which was not quite ready for us at the time. So that was something that was going on. And, you know, also just-- I was involved in this other project that, you know, still had to ship. But it was actually-- you know, I do have one memory of being at home-- you know, I think there was probably about a week that we couldn't go back to work, and I was actually home, without electricity, during part of that. But dialing in with-- I had a Mac Portable, and I could actually dial into AppleLink at the time, and communicate with some people. But around that-- shortly before the earthquake, I remember having a meeting with Phil Goldman and Erich Ringewald. They were-- I think they were working, sort of, on the-- what was called the Jaguar Project at the time, and Pink. And it was 88000-based. And they-- you know, Phil Goldman, when he was working on MultiFinder, he was very focused on compatibility. You know, he wanted to-- you know, if they were breaking an app, he wanted to know why. And you know, I think the-- you know, so I think when he looked at Pink, he was also seeing the-- you know, the same things-- the same movie I saw. And so he was talking to me about, you know, what about emulating the 68000? And they gave me copies of the Motorola 88100 and 200 data books and said, you know, why don't you think about it? So that was really the beginning. And so I was reading through those data books when I was stuck at home, during the earthquake. And that's when I really started thinking about emulating the 68000 on the-- you know, sort of mapping the 68000 onto the 88000. But it wound up going no-- you know, it wound up going nowhere. I think both of them left the company after that. You know, I don't know how soon after that. I know Phil went to General Magic, and Erich was at Be. So-- but that's kind of what-- you know, sort of the first inklings of the 68K emulator.

Hsu: Right.

END OF THE INTERVIEW

