Oral History of John H. Crawford
2014 Computer History Museum Fellow

Interviewed by:
Douglas Fairbairn

Recorded: February 24, 2014
Mountain View, California

CHM Reference number: X7104.2014

Fairbairn: Okay--

Fairbairn: Okay. Okay, we're here at the Computer History Museum. It's February 24th 2014. I'm Doug Fairbairn. And I'm here interviewing John Crawford who is an Intel alum and a long time computer architect and software engineer and a nominee as a Fellow of the Computer History Museum. So, John, welcome, glad to have you here.

Crawford: Thank you, glad to be here.

Fairbairn: So, in this interview, we're going to just start at the beginning and talk a little bit about your background, where you grew up, what your family life was, what some of the early influencers and influences that sort of steered you in the direction of software and computer architecture and so forth. So, let's just begin. So, tell us where you grew up, what your-- tell a little bit about family life, your father, his career and so forth. So, let's start there.

Crawford: Okay. Well, I was-- I grew up in suburban Philadelphia. My parents bought a house, their first house, within two weeks of my birth. I lived there until I left home to go to college. So, I had a very stable childhood. My father actually passed up several opportunities to move, but declined in order to keep the stability of the family and the environment that we had there.

Fairbairn: And he was working for whom?

Crawford: My dad worked for Bell of Pennsylvania, which of course was part of the AT&T organization. And grew up through the organization, and eventually his top job there was operations manager for Northeast Philadelphia. So, he had responsibility for a quarter of that large city, keeping the phones on, the repairs, the installs, the billing, all the folks working in that environment. My mother also had a career that was very interesting as well. Later in my-- as I was a teenager in 1968, she ran and successfully was elected to the State legislature. And for the next eight years, she was a representative from our district, which was unusual at that time. Back in the late '60s, there were very few women in the legislature. And in fact, because she was a Republican, it was even more unusual. And that was a big influence on my life, as well. I think both of my parents had a huge impact on my life. And the careers and the experiences they had and what they were able to pass on to me was great. It was fabulous.

Fairbairn: So, do you have siblings?

Crawford: I have a sister, an older sister. She was, I think, fourteen months older than I was. And one of the things that helped me get going in school was the fact that she was ahead of me. In fact, I've been-- my parents related the stories to me that she would go off to school. She was in first grade. She was learning how to read and all the things that you learn in first grade, she would come home and teach me.

So, then I was able to start school with a good head start from that. And my sister and I have been fairly close throughout our lives.

Fairbairn: Did she pursue any kind of technical career? Does she have a career, or--?

Crawford: She was a lawyer. She went to law school and got a law degree in Pennsylvania. She practiced for several years in the district attorney's office in our home county. And after several years of that, decided to make a career transition, and became an FBI agent. So, for twenty some years, she was an FBI agent doing the things that FBI agents do.

Fairbairn: And you didn't know about, right? She couldn't tell you.

Crawford: Well, she would drop some hints. She was doing some wire taps. And towards the end of her career, she was involved in a lot of child porn investigations. And she said that was extremely difficult for her to have to deal with and process some of the things that she had to look at.

Fairbairn: So, let's get back to yourself. You were growing up in a stable family environment. Was your father technical? I mean was he-- had he grown up from the technical ranks of the Bell Telephone System? Or was he more business oriented?

Crawford: Well, I'm not really sure because by the time I could figure out what was going on, he was a certain level of management that would have been removed from the technical work. I do have pictures of him crawling up a telephone pole. And he was trained as an electrical engineer. He got a degree at Penn State for-- in electrical engineering, and then went off to work at the phone company. Although, in later years, he said the main thing he learned from his electrical engineering degree was that he didn't want to be an electrical engineer. So, go figure.

Fairbairn: So, as you grew up, when did you first develop an interest in technology? When did you first get acquainted with computers? And tell me a little bit about your evolution towards sort of a software and computer background.

Crawford: So, growing up, I had an interest in electronics when I was in high school. And I read-- I subscribed I think to Popular Electronics magazine and was very interested in that kind of thing, and actually did some of the projects that were published in that as a way of learning about things. I can remember building these projects with a printed circuit board and etching printed circuit boards in order to build these projects. The first one I did, I very carefully taped the resist onto the board and put it in the acid bath and so on. And it came out fairly good. The next one, though, I had-- I learned a tip. Well, you can just use a magic marker to-- and that's a perfectly good resist. And that really sped up my progress. I can't remember any of the projects that I did. And I'm sure they're long gone. But that was a great learning experience to teach myself some of the basics of electronics.

Fairbairn: So, what about the world of computers? Did you have any experience with programming or whatever before you went off to university?

Crawford: So, in terms of my introduction to computers, our high school was very fortunate. When I was a junior, I think I was a junior at the time, IBM had made available a 1401 computer to visit different high schools. And so, the AP students-- the AP math students were gathered together at one time and given an introduction to FORTRAN and this computer and how one would deal with the punch cards and run your program and so on and so forth.

Fairbairn: Just like the one we have in here in the Computer History Museum.

Crawford: Could be the same one. But so, I went to the first session. I read all the materials and read through the FORTRAN introduction. I said, "Well, this is pretty simple. This is-- what's this computer programming stuff? This looks really simple." And went to the lecture-- went to the session, they had invited, I suppose it was neighboring schools, as well that came. The computer was going to be serving several high schools. So, we had a crowd of about sixty students there. So, I looked at that. There were sixty students. I think there were two punch cards, one computer. So, I said, "This is great. But it's going to be an awful lot of waiting in line and maybe not very-- not as good as it sounds." So, I dropped out, made some room for the other folks, and went back to pursue my main passion in high school, which was my rock band. So, I wanted to spend the time with my garage band rather than waiting in line for the computer environment there.

Fairbairn: So, what instrument did you play?

Crawford: So, I played several instruments. Growing up, I learned the piano. I had piano lessons. And then in sixth grade-- fifth grade, I think, I picked up the trombone and carried that through high school, in fact, played the trombone in our marching band. And at age thirteen, my sister was learning to play the guitar. So, of course, I-- following in my previous footsteps, picked that up as well and taught myself how to play a few chords on the guitar and got started that way. So, in the early bands, eighth grade, ninth grade, I played bass, bass guitar. But then by the time ninth grade came around, I figured out I was a lot better guitar player than our lead guitar player. So, we had a little palace coup and I-- we formed a different band. And I was the lead guitar then from that point on.

Fairbairn: So, did you play at venues? Did you hire yourselves out?

Crawford: So, our band was very successful. And I would say our garage band was very successful because we made enough money from our appearances to pay for our equipment. So, we played-- we had kind of a steady job at the YMCA. Every weekend they would have a dance. And once a month we were one of the-- we would play it seemed like on that order of schedule. So, we had a pretty steady thing there. We played at the junior high dances. We played I think for one high school dance and a couple of private functions.

Fairbairn: Does it make you popular with the girls? Or did you have any groupies following your band around?

Crawford: Well, there were groupies with the band but they weren't following me. They were more attracted to some of the other guys. So, sadly I missed out on that part of the rock star phenomenon.

Fairbairn: So, did you actually end up doing any programming in high school before?

Crawford: No.

Fairbairn: Or you stick with the music?

Crawford: So, I did not do any programming in high school. The combination of this large group that was going to be sharing this very precious contended resource, I decided I wasn't that interested and wanted to spend the time on-- like I said on the-- on rock and roll was a lot more interesting to me at the time.

Fairbairn: Yep. So, what about your teachers in-- especially in high school. Were there any that were a particularly strong influence in terms of guiding your future career or helping you decide where you were going to go to school and how did that unfold?

Crawford: So, looking back to high school, that was forty some years ago already. There were many teachers that were influential. At my fortieth reunion, one of the experiences I had was a woman that had gone through the-- had taken several classes with reminded me that she was very-- she reminded me that how much I got beat up by one of the math teachers in ninth grade. He would pick on me, but I always had the right answer. So, I'm not sure. And I didn't remember being picked on. But she was-- she even apologized about the guy. But I can't remember who it was.

Fairbairn: But you don't know who it was.

Crawford: I can't remember the name and you know to be honest I don't remember a lot of the teachers. I do know a trig teacher I think I had in eleventh grade. A couple of times I corrected her, and-- which was kind of interesting. And it turned out I was right.

Fairbairn: How did that go over?

Crawford: Well, the first time not so good. But after that, I think she was willing to listen and work through, oh yeah you're right.

Fairbairn: So, you obviously showed some expertise in the math. You were interested in sciences. How did you choose a college and major once that became the thing to do?

Crawford: So, I knew I wanted to be-- leaving high school, I knew I wanted to be a math major. I had done very well. I really enjoyed it. It was a lot of fun to solve problems and to learn new things. So, that was pretty well set. So, I knew I wanted to start at least-- start in that direction. And in terms of what colleges to apply to, I knew I wanted to apply to Penn State. That's a state school. My father went there, and everybody in his family went there. So, that was a given that we would apply there. And then I made two other choices. I applied to MIT, and to Michigan State, and to Brown. Now, I don't remember why Michigan State and MIT, but I do remember Brown. They had invited me and several other students to an informational meeting. And I said, "Brown, I never heard of this place." And-- what is this college? Oh, it's an Ivy League school. I thought that was Harvard, and Princeton, and Yale, that was it.

Fairbairn: Right, that was it.

Crawford: But-- so, they invited me. I went there. I applied and I was accepted. So, I decided that I would go there. And fortunately, my father was able to finance the education and had a fabulous experience at that. And kind of a circuitous route to my degree. But that's probably for another question.

Fairbairn: So, you sort of found it-- you ventured off to Brown. Tell me about your freshman year. What was the-- were there any major revelations at that time? You were taking on a path in math at the time, right?

Crawford: Right. So, I guess there's one other aspect of going to Brown is, once I discovered it was an Ivy League school, it came to be known that my great grandfather was in the first class at Cornell. So, we had some history in the family. They didn't all go to Penn State, but I did have one relative that did go there. So, now that I've interrupted the question, you'll have to ask me again.

Fairbairn: So, you entered Brown, you knew immediately you were filed as a math major. And you started on that path. When did you get involved with computers and software in the university?

Crawford: So, I went off to college and knew I was going to be a math major. So, entered the program, took some tests, tested out of some classes, and signed up my first semester for a pretty good dose of math and a couple other subjects, and ended up befriending one of the math students there. I think I was taking a modern algebra course at the time. We would end up getting together and working homework problems with each other, trying to build off each other and encourage each other from that. And the second semester, he signed up for introductory computer programming course, or computer science 101 or something like that. And he convinced me to sign up too. And I said, "Oh well, I remember this thing in high school I was a little bit interested in. Hopefully, the lines won't be as long at this point." So, I signed up for that and really, really enjoyed it. It was-- that first course really got me hooked.

Fairbairn: What was the computational vehicle of the-- used in the course? Was it still punch cards? Or are you into interactive--?

Crawford: So, the computational environment there, actually before I took a course. In my freshman year, I did connect with Professor Barrett Hazeltine who was encouraging me to do an independent study in computation. And at that time, the independent study option was a console using APL as the programming language. So, the very first language I learned and did any programming in was APL, which is kind of an unusual language, a matrix oriented kind of a thing. My first project was to combine one of my other passions with computation and math, which was to compute poker odds. So, I used the-- programmed up an APL simulation of poker odds, different hands. Was it five card, seven card? Was it stud or draw and that kind of thing? So, it provided some insight to me and my-- some of my fundraising efforts.

Fairbairn: So, we've discovered a couple of your interesting sidelines. You're into music. You're into poker and cards. What else have we missed along the way in terms of your personal interests?

Crawford: Yeah, so other personal interests-- well I got the music. You got the cards. You had-- and I did date quite a few girls in high school, none of them for very long. Sadly, as I reported, the groupie thing didn't work out for me but there were other avenues. So, of course there was some activity there.

Fairbairn: All right, so that's just a little sidelight there to make sure we hadn't missed something important. So, you took your first programming course. By the way, when you started down a math major, where did you think that led? Did that mean you were going to be a professor? Did you even know? Or were you just interested in math so you were going to do that?

Crawford: So, yeah I had no idea what I was going to do. In fact, that reminds me of a little anecdote that happened that first year. I was a math major. And I found myself flying home, probably for Thanksgiving or Christmas or something. And on the plane, I sat next to-- or I got on the plane with one of my roommates-- not roommates, but hall mates in our freshman dorm. And this was a guy-- he was the only black kid on our hall. And he came from a very wealthy family in the St. Louis area. And he was a lot more worldly than I was. And we got to talking a little bit. And he said you know, "What are you majoring in?" And I said, "I'm going to be a math major." He said, "Really? So, what are you going to do when you graduate?" Uh-- uh-- I obviously hadn't thought about it. It was just something that was interesting and fascinating. And in order to really top it off and zing it, he added this comment, "Well, you can't eat numbers you know?" So, obviously I was not--

Fairbairn: You hadn't thought a lot about that.

Crawford: I didn't have some grand plan in mind-- a plan in mind in terms of something to pursue. And I was just following my passion and interest at the time. And well, of course I was a freshman. I had a lot of time to specialize.

Fairbairn: Had your parents given you any counsel one way or another there? You know, go off to university. You're happy. Do what you want to do. Any guidance from--

Crawford: I think they were-- in the discussions we had, I think they were happy I was going to be a math major. They thought that was perfectly okay.

Fairbairn: All right, so you got through-- you're into your second semester. You get taking a computer science course. What-- how did things evolve from there? Did you actually change majors? Was there a computer science major? Or did you just continue to be a math major but take additional computer science courses?

Crawford: So, I started out as a math major. And when I started the computer science program, whatever introductory classes I was taking, I sort of decided well now I'm going to be in applied math person. The computer courses were being taught within the applied math department. And it turned out the applied math department was a lot more famous at Brown than the math department, which I didn't know until then. And I ended up taking several applied math courses in addition to the computer science focus. So, I had a mixture of classes from that. So, I went from math to applied math. And then my senior year, it turned out there was a-- I could get a degree in computer science. I was intending to get an applied math degree, but it turned out there was an option to get a computer science degree. And I said, "Well, that's probably more accurate. Sign me up and give me a computer science degree." At that time, there wasn't a computer science department. It was still a subset or contained within the applied math department.

Fairbairn: So, were there any memorable professors as you went through your undergraduate career that really helped guide, or shape, or interest you?

Crawford: Oh sure.

Fairbairn: You can tell us about?

Crawford: I do remember several of the professors. Certainly the most memorable was Andy van Dam who was the head of the computer science department there and actually taught several of the classes. He was a fabulous teacher. In fact, I think he taught the introductory class that I took that got me hooked. And one thing I thought was very strange about the computer curriculum at Brown at the time was that there was no textbook. And we would go to class. And the TAs would hand out mimeographed, I think, notes and--

Fairbairn: Smell them.

Crawford: Copies of papers that had been published in the last couple years. I thought well this is interesting. There's no textbook. What's going on here? Well, it turned out there was no textbook. And we were actually getting really leading edge kind of things that we were being introduced to that later on I came to realize was a very valuable experience. So, he was one. And of course he founded it and set the tone and was really a driving force for the computer environment there. So, huge debt of gratitude to Andy Van Dam. A second teacher was Philip Davis who was an older applied math professor at Brown. And I took a class from him that was-- I can't remember exactly what it was, but it was a course that inspired me. I think one of the things I learned there was about splines, which came back in later life, you know in several other things. I think he was also talking about fonts, and font design, and things like that. The third professor I want to cite is Infante, Professor Ettore Infante. And I took a course from him, actually a two semester course, on real analysis I think it was. Well, it turned out the subject of the course was taking things to limits. And it was just bizarre things that would take to limits. And that really stretched my mathematical brain at the time. It was a fabulous class. It was probably ten students at the most. We would meet in a small room. And several times he would ask me to stand up and work through something that I hadn't seen before. And the challenge of that and the encouragement that I got from him was very valuable. And it made me confident that-- and that was-- I can remember I was just barely hanging on to the concepts here. And we were pushing ahead so fast. You know I was kind of white-knuckling the thing through it. I ended up getting As in the courses but it was a real challenge and something that sticks with me.

Fairbairn: Skill and confidence that you could use later in your career, huh?

Crawford: Yeah, I think so. As part of that, when I did graduate, I did-- that year, they instituted a prize in applied mathematics. And the very first year they actually awarded it to two students. I was one of them. So, I think my experience in both sides of the applied math department, both the math, if you will-- the math part and the computer science part experience gave me a leg up on getting that award.

Fairbairn: Were there any major projects or things that you did during your undergraduate career that sort of stand out? You've mentioned a couple of them. I was just wondering if we've missed anything.

Crawford: Well, I wouldn't cite the poker odd study as a major thing, but it was a great start and gave me some interest. During-- one of the things I came to know-- came to learn is that van Dam used his courses as recruiting ground. So, he would watch for students that he could then recruit into undergraduate research projects. So, I had the opportunity then to be brought on to a research project involving the development of a programming language. He had a research grant to develop a language called LSD, if you can believe that, for the early '70s. In fact, he had to change the name to-- it was language for system development. I mean it was not a-- and I think at one point he changed the name to LSSD just to try to hide the obvious connotations. So, I was one of the-- several undergraduates supervised by several grad students trying to pull this compiler together. My assignment was to work on the symbol table. So, I had to develop the-- kind of the bookkeeping and the database, if you will, where symbols from the program would be installed. You understand what type it was, usages, and all the information that you would normally do. Well, turned out I was the first one to get done of the project. So, we were waiting then for others to complete to put the whole thing together. And it turned out then, at some point, I can't remember if it was my senior year. It might have been the summer after senior year. Or it could have been the summer between junior year and senior year. I was the project leader. I was promoted to project leader. After all, I was done with my part. And they promoted me and asked me to then pull the whole thing together. And that was a fabulous experience of-- first of all, a lot of software development, working on a team, and what it takes to try to pull things together, get things to work, kind of bootstrap something to get going.

Fairbairn: So, had you defined interfaces between the various modules that students were working on? And how did that work out?

Crawford: So, the interfaces were all defined before I showed up. So, that was done by probably the grad students had identified that. I may have refined the symbol table interface and worked out some of the details, but the broad partitioning was done long before.

Fairbairn: The notion of having clean interfaces that everybody could build toward was in place when you took on the--

Crawford: Well, I'm not sure about clean interfaces. But we did have interfaces. That was all defined before I joined the project. So--

Fairbairn: So, then you took on the project management role. And did this thing come to completion? Did you-- what was the eventual outcome of that?

Crawford: So, I think it did come together. We had fits and starts. And we did actually get something together. But then, I graduated as things happen, right?

Fairbairn: That's what happens.

Crawford: So--

Fairbairn: So, as you're nearing your senior year, you're nearing graduation, did you have questions about what the next step was? Were you thinking about going to work, going to graduate school, getting a PhD? What was your thought process at the time?

Crawford: So, in my senior year, as I was looking forward to graduation and what to do next, I did go out and interview. I did interview with a couple of companies. But at that time, this was the fall of 1974 and into the spring-- I don't know even. I got into the spring of '75. But the summer, fall of '74, the country was in a huge recession. I mean it was the worst once since the Great Depression I think was the way it was advertised. So, I didn't get a lot of nibbles. And I did get a job offer, but I wasn't interested. It wasn't exciting enough.

Fairbairn: Now, you had applied for computer programming jobs? Is that what you were looking for?

Crawford: So, when I graduated, I was applying for computer programming jobs. And even that was not a lot of--

Fairbairn: Not forthcoming.

Crawford: Availability in interesting areas. So, I decided then, at the same time, I put out some bait out to grad schools to see about pursuing a master's or a PhD. So, and I worked with van Dam. He gave me some advice. Some of the other grad students provided some advice to me in terms of what to do. So, I applied to-- I applied to Carnegie Mellon in a PhD program. I didn't get accepted. I applied to Stanford and Berkley for their master's program. I did get accepted. And I applied to North Carolina for their master's program. They accepted me with some financial aid, which the California schools did not offer. So, van Dam actually had a close contact at North Carolina, Jim Foley. And so, it was a good connection there. And I decided to go to-- and I knew that Fred Brooks was there. And computer architecture was one of my great interests at the time. So, I decided to go to North Carolina and accept the-- work for a master's degree there.

Fairbairn: So, you mentioned-- so, by this time, you had developed a real interest in computer architecture. That was something that had occurred to you not just around programming?

Crawford: So, we just had the AC kick on. Is that--? Oh, it's a projector. If that's a--

Fairbairn: Yeah, we can take a break. So, you'd made-- So, you were entering North Carolina. And had you developed an interest in computer architecture going into the program? And what was your interest and goal in terms of where you wanted to focus in the master's program?

Crawford: So, I had developed an interest in architecture, in computer architecture, in my undergraduate years. Part of the training, part of the teaching there, was taking things down to the machine level. In machine language, assembly language-- in fact, we had an opportunity to write a number of programs at the assembly language level. And I found it to be very interesting and really enjoyed that. In fact, a couple projects, some of them off the record, that I undertook at that time-- maybe start with the on the record ones. One of the exercises that they gave us in one of the classes was to write a micro program for-- they had some graphics engines that they used in the lab that were micro programmed-- micro programmable engines. And they posed a problem, which was-- I can't even remember what the instruction was. But it said write a micro program for this instruction. It was a fairly complicated thing. And they gave us an abridged list of instructions that the computer would run. And I remember sitting down and really enjoying, really chewing on that exercise. And I came up with a series of instructions for that micro program, finished it all up, handed it in. And the next day, the TA came to me and said, "Well, John. That was a really good job on this, but it won't work." I said, "What do you mean it won't work?" Well, yeah, based on what we gave you, it would work. But it turns out the machine, you can't do that on the machine. There was some restriction of this instruction and that instruction. But it turned out I had generated the program that was the shortest in execution. So, there's one little example. So, I had a lot of fun with that. A second one was actually earlier-- I can't remember if it was freshman year. I think it was sophomore year. I worked with a friend of mine, actually a guy from my high school. There were three of us from my class that went to Brown. And Ken Hoadley was another computer-- involved in computation. And we took on the challenge of trying to break into the computer oper-- break into the virtual machine that we were running on at the school. And, in order to do this, well the first thing we had to do was figure out what the code was. And so, we wrote a disassembler for 360-- we were running on a 360 Model 67. And it was a virtual machine environment. So, we wrote a disassembler, and went and disassembled the low core. So, it was the various machine interrupts that would manage system services at that level, and got that working. And then we're tracing through each of the supervisor calls. And finally found one that turned out to be a graphics one that had a very curious property that it would-- you would give it some offset and a value. And it would store that value in an offset within some buffer. Well, we figured out that's close. But the buffer's up here. And we really want-- we needed something else. So, we ended up with a two-step process. We would call it once and have it store something in the code for this supervisor call, changing the instruction, and then the offset in the instruction down to something else. And then-- or actually I think it was loading-- it would load a pointer. And then it would offset from that pointer. We just had it load zero. And then we could feed it whatever low memory offset we wanted. So, then stored into that. Boom, we got supervisor privilege. And I remember one night, we had a-- we had another student who was not at all involved with computers. Every student got an account. So, we said we don't want to use our accounts for this mayhem. Let's-- so we got that guy's account, ran a little thing. And the way we tested it to see if we actually got supervisor privilege is that we did a-- we sent a broadcast message that could only come from a privileged thing. And sure enough, we went through the process, boom. And the--

Fairbairn: Everybody got the message.

Crawford: Everybody got a message. And then-- but the response was pretty quick. I think the next day, it didn't work again. So-- they caught on to us. So, that was another one. And later, in my senior year, another interesting assembly level computer architecture kind of project was to do an interpreter or an emulator. Basically, it would trap every instruction. And most of them, it would then execute directly. And then a few would have to interpret. What we were looking for was a way of monitoring time. So, we would have that ability to instrument the program and track time at different points. And in order to do that, we had to interpret the instruction-- some of the instructions that were there. So, I had experience in writing a 360 emulator, if you will, or interpreter. And as part of that, we exposed bugs in the virtual machine monitor's dealing with the clock. The 360 had a clock that was in low memory. And you could just read it. And it I think went up with cycle time. Well, it turned out the counting for that wasn't exactly right. But we worked with the system folks, and they got it tuned up. And then we were able to apply that to our compiler project and try to tune some of the performance of the project.

Fairbairn: So, you were getting familiar with sort of the guts level, the machine level, assembly level instructions of the computers and understanding how they handled things?

Crawford: Yes, so that was-- dealing down and getting down into the machine instructions and working very-- in different aspects of kind of cycle by cycle, really detailed, low level activity, I was-- I found that to be fascinating. I enjoyed and enjoyed the give and take of the rules and some of the difficulties that come with that.

Fairbairn: And sometimes things that weren't as advertised.

Crawford: Exactly. Right. Like the timer that wasn't quite accurate. Because we would-- the first couple of runs we would say, "Well, this didn't take any time." And then all of a sudden this thing took forever. So it was obviously the clock was not being updated properly in the virtual machine.

Fairbairn: Okay. So you're off to North Carolina. Tell me about your experience there. What were some of the major events, influencers, professors, developments that took place there?

Crawford: So let me start with my experience in North Carolina with a personal story and not so much a technical story. One of the first things that happened when we arrived they had a nice tour of the department for the incoming students. So we had one of the older grad students took us around, showed us the offices, "Here's this. Here's that. Here's if you want to do this computing."

Fairbairn: I'm sorry. Can we stop? Sorry. "When we first arrived in North Carolina." That would just really help out in the edit start off with that. Thanks.

Crawford: So when I first arrived in North Carolina, one of the first things that they did, of course, was introduce you to the department and we had a nice tour of the department one afternoon. They took us around and showed us the computing facilities. There was a separate lab in a different building. We went to visit that and at the conclusion we had all the new grad students were there in the lobby and were kind of looking at each other. And so I look around and say, “Well, who wants to go get a beer?” And I think I only had two takers, Mark Moore, who became my roommate at grad school and Kathy Finney , who was another grad student. So we go off and have a beer. But the reason I want to relate the story is that really started my friendship with Mark and we had a good relationship in grad school and remained friends even to this day. I was the best man at his wedding. He was the best man at my wedding. And we have kept in touch. In fact, he and his wife lived with me in Santa Clara for sometime. They rented a room in a house that I had just bought as a single guy and so on so.

Fairbairn: So you got through the introduction. You started down the course. Did you have an idea as to what you wanted to focus on or how did you determine what you want to do a thesis on?

Crawford: So the decision process for what to focus on-- pause. In deciding what course to take and what to focus on, I think the first year I was taking classes in some of the basic computer science courses that they had and exploring options for a thesis. I was looking at either a project, some kind of computing project-- write a program, get it to work, get some results, and publish it or a paper project. And I decided on a paper project given my little experience with folks that were choosing the other route tended to have the delays involved in their graduation. So I decided on a paper study, which was combining compiler technology, which was something I had experience with at Brown and a software engineering aspect to it which is something that I got more exposure to in North Carolina. In fact, one of the-- at the time I was there, David Parnas, was a teacher there, was a professor. And I got him as one of my advisors on my thesis to work on the software engineering aspects. And Mehdi Jazayeri was a compiler professor there. He was really helpful -- he was my main advisor. So my project was module specifications for a program optimizer. And so I had to work through some ideas for how to analyze programs in some kind of a new way. And then how to describe that in terms of what language to use and what description to use to describe the modules that would be required to make this optimizer work.

Fairbairn: So you worked through your program. You graduate on time? You graduated in the time that you had envisioned? And any other major developments during your Master’s program?

Crawford: So my strategy worked. I was able to graduate on time in the Spring of ’77. And actually published-- was able to publish things. I published with Mehdi Jazayeri -- Professor Jazayeri, a paper at a local-- a regional ACM conference among other things and continued the work. We published in several other places some of the work from my thesis. In addition to that, one of my other roles, which was Fred Brooks put me-- had me assigned as a research assistant in a lab, a graphics lab that they had. And the lab was doing x-ray crystallography, which was fascinating. They would-- something would take molecules and crystallize them-- large, organic kind of molecules and crystallize them and then bombard them with x-rays to get an idea of what the structure was. And then they-- you would come back and you knew what the molecule looked like. You knew what the chain of atoms was, but the idea was now to try to conform it to the 3D information that you got from the x-ray activity. So they had this idea of you would manipulate the bonds and twist the angles a little bit in order to try to conform the thing. So it was a very interesting kind of combination of, I guess, biology and computation. So I was involved in some roles on that. The one role that I-- maybe it was part of that or a separate role is I was assigned to be the disk czar. And I thought, “Well, this is a waste. Can’t somebody else keep track of who has how much space?” So my job was to track of all the projects and all the grads especially who was using what space, what their allocation was, and then basically dun people that were over the quota, and try to identify issues before it became-- before it became a thing. So that was an interesting experience. So part of the process of making that livable was to develop some additional tools that made my job easy so I was able to do that and then pass that on to my successor.

Fairbairn: So you mentioned Fred Brooks. Did you have any major interactions with Fred, classes, or whatever, in your time there?

Crawford: So my interaction with Fred Brooks was one of the reasons I went there. I wanted to have an opportunity to meet this man who was a famous computer architect and had done so well at IBM, and then came off to start this department at North Carolina. So one of the best things I did-- best experiences I had at North Carolina was to take a computer architecture class from him. One of the things about that he walked us through some of the basics. But then each of us had to pick a computer and do a term paper, if you will, on that computer. So various folks would pick at the time, computers that were available. You know, maybe the IBM 360 wasn’t available. No. That wouldn’t have been fair. A lot of the students went for the big iron so some of the CDC machines, CDC 6600, the Star. In fact, I think Mark Moore, my roommate took the CDC Star as his project. I went in a different direction and I said, “This Intel 8080, this is an interesting thing.” So I did my project on the Intel 8080 and did my term report and so on and Brooks was not impressed with my paper. He wrote some comments on it. I think I got a B on the report and a B in the class. And the comments may be even worse than the B on the report, his comments were, “Well, this is not very imaginative. Where can this technology go?” But off the bat you could tell my interest was in-- I already had sparked an interest in this microprocessor-- the microprocessor side.

Fairbairn: So you came, I guess, Spring of ’77 it came time graduate. Did you have thoughts about going on for a PhD at that time? Or decided to test the waters in the job market? How did that evolve?

Crawford: So when the time to graduate with a Master’s degree came along, I was done. I had no intention of-- I don’t think it even crossed my mind to go onto a PhD. I think I made that decision. Or actually, it was made for me. —When I graduated from Brown, I kind of threw it out on the water, you know, threw the bread on the water and the fish didn’t bite. Carnegie Mellon didn’t take me in as a PhD student. So and that was fine with me. I was-- would have gone either way and I had pursued this Master’s degree. And said, “Oh. Okay. Now, it’s time to go get a job.” Fortunately, in the two years since I had graduated in the spring of ’75, in spring of ’77, recession was over. Business was booming. And I ended up flying around the country. I think I had 12 job interviews.

Fairbairn: Pretty different environment.

Crawford: At, you know, a lot broader selection of companies. And, yeah, it was a radically different environment. So one of the things I remember-- it's kind of a funny incident. I was, I think, flying up to Boston to do some interviews up in the Route 128 area and was in the Newark Airport. And I hear this announcement over the loudspeaker, "John Crawford, please come to the white courtesy telephone." Or, you know, something to that effect. So I went and answered the call. It turned out it was a call from Intel. I had sent my resume out to them and it was a call to come out and interview to piggyback on an interview with HP that I had set up previously. So they had negotiated, "Okay. We'll keep him for an extra day and you have him this day. We'll have him this day and we'll do like that." They had to contact me at the airport. I'm not quite sure why but that was-- so that was that. Before they invited me for the interview though the process was-- I'm going to answer a different question.

Fairbairn: So you toured around. You visited a variety of companies. Then you came out to HP and Intel at some point.

Crawford: Okay. So but before we get there. The process of getting an interview at Intel went something like this. I had, of course, participated in the on-campus activities of the companies that would come and try initial screening of graduates to decide who to invite for an interview or not and then extended that by sending my resume out to several other folks. And I think it was Brooks that encouraged me to send my resume to Intel. I had done this paper, but not a very good one. Not, you know, not a very good one. But not, you know, I could have done a lot better on this 8080 and, you know, why not see what's going on over there. So I sent my resume in. And it turned out that year there was the-- I believe it was ISCA, the computer architecture conference, was in Raleigh and some Intel folks were coming out to Raleigh. And they said, "Well, can you come out to Raleigh and we'll do lunch and we can talk a little bit about this." So I was able to meet some of the Intel people that I would be potentially hired by at this conference at a conversation at lunch. And apparently I did well enough for them to invite me back out for interview. The scheduling which was a little difficult, but the phone call at the Newark Airport took care of that. So I came out to interview with HP and with Intel. And at the end of the interview-- the night after the second interview, I remember going up-- taking my rent-a-car driving across the Golden Gate Bride, and going up on the highlands that were sort of to the west up there. And the beautiful view back-- it was a clear night, had a beautiful view back across the Bay and into the city. And I stopped the car, turned off the engine, and got out. Tried to ignore, I guess, the lovers lane activity going on in some of the other cars. And just sat there and thought about it. Because I think that was my last interview. It was on-- perhaps almost my last and I had been through quite a gauntlet of things and kind of a came to the decision that I did want to come out to California. I figured, "Well, you know, this is a whole different environment. You know, if it doesn't work out, I'll be here for a couple of years and then go back East. Get a job back there, whatever." But here's an opportunity that I really wanted to go for.

Fairbairn: So what was it about Intel that made you select them?

Crawford: So, to be honest, at that time one of the things that attracted me to Intel was they had a vacation policy that when you're first hired you get three weeks vacation. Everyone else had two weeks vacation and you had to-- you had to have been there for a while in order to get more. So I thought, "Well, this is a good thing." That wasn't a major-- that wasn't a major consideration. Obviously, Intel was by far the smallest company that I had interviewed with. And, you know, my experience with the 8080 and understanding that and wanting to explore-- jump in and be a part of that. I thought that to be very intriguing, very exciting aspect. Perhaps, a more interesting policy, personnel policy was that they offered stock options to new hires coming in and I think they were the only company that offered me stock options coming in. And I thought, "Oh. Well, this is-- this sounds great. You know stock options, three weeks vacation, a great chance to be involved in this microprocessor thing, live in California, you know, at least a couple of years. This would be fabulous." So I decided to come on out to Intel.

Fairbairn: So did you take a break between college and work? Or you just pack up your goods and drive to California?

Crawford: So I didn't take a break. I had another friend who took a break and was a ski bum between graduate school-- graduating and he went. But I didn't do that. I had a couple of months. My start date was figured out to be August 1st. I don't know why, but it was August 1st. I was graduating. I was pretty much done in May so I had the period between May and August to move out of North Carolina and then figure out how to get out to California. The process there was I did take a break and had the opportunity to travel with the drummer from my high school band, Tony Marzulli. He was a year behind. He was a year younger so he was still in-- I think he just graduated. But he was able to take some time off and we drove in my Chevy Vega stuffed to the gills with important gear like stereo, and speakers, and some camping gear. So we drove across country. I think it was a three-week trip. We stopped at several national parks. Did a fair amount of camping. Spent some time down in Southern California with his brother and then came up here. He delivered me and then flew back.

Fairbairn: So who did you go to work for and what was your first job? What did Intel want you to do as a new employee?

Crawford: So my first job at Intel was kind of a jack-of-all-trades for some 8086 software tools. I had my finger in a number of small projects. And for example, one was working with a contractor in Southern California on a curious tool. It would take 8080 code and convert it-- source code-- or assembly code and convert it-- kind of transliterate it into 8086 code. The 86 was not binary compatible with the previous chips. So they had this idea this would provide a bridge for people that had a substantial code base in assembly that they couldn't just recompile.

Fairbairn: And the 8086 had just been announced at that point or was it recent?

Crawford: I think it was announced in '76. I don't know when it-- that was the period where we would announce the chip before we shipped it. And I don't know-- often there was a one-year lag between volume shipments and-- announcement and the volume shipments. So it was certainly on its early ramp. So that was one. And a second thing was we needed an execution vehicle to test the software tools. So they had a development kit for our customers that had a very small amount of memory. It was a board.

Had some memory. I'm not sure what the IO capability was. And, of course, the processor and my experience at hardware design then was to take that-- the schematics for that, Xerox the memory part, and hand it to a wire wrap technician and said, you know, kind of red penciled in some extra connections, "Here please let's get this thing more memory." So developed a board that had enough memory to really test out the software tools. So that was another project that I had. A third one was writing a simulator for the 8086 and then test programs to check out the simulator. So we had a-- I wrote a software simulator for that. In fact, that might have been my first project coming in was to write a software simulator for the instruction set of the 8086, which then was used to provide some initial kind of low limited testing then of the tools. But then later we were able to bootstrap that onto the hardware device to get a more thorough testing of the tools.

Fairbairn: So a few months in what was your impression of Intel at the time? Was it what you expected? Were there any major surprises, any issues, any second thoughts about, "Did I make the right selection?"

Crawford: So my initial exposure to Intel, I moved into a leased building, a kind of small environment, which was great because it was-- the software team was pretty much contained in that building. I shared an office with Bruce Ravenel who was-- he wasn't exactly my boss. But he was my project leader, I guess. So he would provide me kind of day-to-day direction as I was developing this simulator. My first boss was Kapil Nanda who was very calm. It was an interesting contrast between Bruce and Kapil. Kapil was very calm. You know, very hard to rattle. Bruce was a very energetic kind of guy. Interesting contrast there. In terms of what I expected? Was it what I expected? I don't know. It was-- I had a blast.

Fairbairn: You were enjoying it.

Crawford: It was a lot going on. A lot to learn and some great people. Just really smart people to work with. It was a great experience.

Fairbairn: So you initially started as software engineer. Were there-- so I want to sort of move on towards the Intel 386 experience.

Crawford: Let's not go there. Let's not go there yet.

Fairbairn: Yeah. Fill in some of the gaps that led you in that direction and major accomplishments along the way.

Crawford: So there's one little story I'd like to relate that was-- I think could be-- I thought was very interesting in hindsight. One of the projects I was working on, I was project leader for the 8086 assembler-- the second release of the 8086 assembler. The main feature that was added at that release was support for the 8087 floating point co-processor. So we had to work out, "Okay. How are we going to identify the data types and the instructions?" And particularly the data types. The 8087 had a stack-- a register stack.

It was not a directly addressable register set so we had to have some way of dealing with that. And at the time, I got into-- I think the software team came up with a strategy for that, but it was not-- it was something that John Palmer, who was a numeric expert who was one of the architects of the 8087 had a big problem with. And I remember working-- trying to work it through with him and it didn't work out. We certainly had a fair amount of conflict going on. One day I come into the office and in my inbox and that's back when we had paper inboxes, I pull out-- pulled out a stack of two or three memos that had flown in the last day or so. And there's a memo from John Palmer and it was a very short little memo and it was addressed to me and then copied Gordon Moore, Bill Davidow, my whole management chain. They didn't copy Andy Grove because I think he was on sabbatical, but he would have copied him too. But it was a huge blast memo of how this decision was terrible and it was unacceptable and he used some very-- and it was interesting the audience that he addressed in this list probably would not have related to the arcane issues that he was citing in there. But anyway, my blood pressure went phewwww, "Ha." What is this all about? And then I-- after I got over that initial shock I said, "Oh. Wait a minute. He probably didn't-- he didn't send it to these people. Why on earth would he do that?" And then I decided, "Oh. Well, I will take care of him. I will send it to those folks with a cover letter about memo etiquette." I sat down and handwrote a memo for that. Fortunately, I then went and talked to my supervisor. Showed him what's going on. "This is what I'm going to send them." "No. No. Calm down, John. Settle down. I'll take care of this."

<Laughing>

Fairbairn: So he was decrying your decision about something about the interface between the 86 and the 87?

Crawford: Right. I can't even remember what it was at this point. But there was a difficulty there. And, you know, frankly, it was a little curious. Looking back on it, I don't know why I just didn't take what he said because he was the expert. But apparently there must have been something difficult, or something awkward, or something, you know, very difficult in that aspect of things. Part of the resolution of that actually was to get Professor Kahan involved and I think he-- his counsel to Palmer was, you know, "Let the software guys figure that out. You know, this will work. We can make this work too." And we proceeded then along whatever lines that it ended up being. But as a new engineer, I was 18 months out of school, right? I was-- I had just really started. I was still probably the youngest kid in the department. Get this thing that purportedly went to the highest echelons up next to God himself up in the Intel hierarchy was pretty shocking.

Fairbairn: So that all blew over? And you moved on?

Crawford: Yes. That all blew over. We released the assembler, 8087 was released and had some great success. A very nice product for Intel and really Intel's participation in that IEEE floating-point activity with John Palmer as our lead was a very important part of Intel's microprocessor history and development. Being able to participate in that aspect of the market and having some really leading products. I think if I remember right the 8087 was the first product out that could be compatible with the new standard as it was evolving even. So, yeah, I think that blew over. John and I were on good terms. I never resolved it with him. I never did talk to him, “Why on earth did you do this?” But, you know, we managed to work through and carry on.

Fairbairn: Moved on.

Crawford: Yeah.

Fairbairn: So are there any other major roles to this time when you got to the transition to the 386 team?

Crawford: So I think the thing that provided a transition from my software engineering activity at Intel into the microprocessor team was doing a code-generator for the Pascal compiler for the 8086. That was kind of the trailing edge of 8086 tools that Intel was developing. And our main programming language was a thing called PL/M, Programming Language Micro. But there was some demand for Pascal as a language at that time. So the company decided to build a compiler for that. I was assigned to do the code-generator for that compiler. And as part of that I worked with Susan Graham and one of her students up at Berkeley to do-- to use a technique they had recently developed. She had developed with a guy named Glanville. So it was the Graham-Glanville code generation strategy. The thing that made it unique or made it interesting to follow the story later is that it was based on using a table-driven parser basically to parse the intermediate language that the semantic phase of the compiler produced. To use a table-driven parser to chew through that, provide the structure then to hang the code generation activities off of. So it provided quite a good degree of flexibility in how to generate code that made it first of all easy to develop and tune the product itself, the Pascal compiler, but then gave me an entrée into the microprocessor team later with an idea of porting that to a different architecture.

Fairbairn: So you described earlier in our previous conversation the transition what helped move you from the software development group to--

Crawford: Let me take a-- have a Marco Rubio moment here.

Fairbairn: Yeah go ahead.

Crawford: Okay. Fire away.

<crew talk>

Fairbairn: So describe the transition from the software development team doing the code generator for the Pascal compiler to involvement with the 386 development operation.

Crawford: So that's a-- the story of how I came to be the architect of the 386 is an interesting story. It actually began two years before I made a transition while I was still working in the software development team. And a colleague, Rick Schell, had been on-- been involved in with the software development and went over to the architecture team as a software guy involved in activities to produce a brand new 32-bit microprocessor. And so he worked for a couple of years there and decided to transition onto something else. And right at that time I was finishing up the work on the Pascal compiler. It was done but went through another rev or two and it was time for me to look for something new. Rick comes to me and says, "I'm moving on. But they still have a need for a software expert within this environment. Why don't you throw your hat in the ring? Come over and interview and see if that will work out." Well, so I did and I did get the job. And I was pulled into the team really to do what I described earlier, which is to take this Pascal compiler with the table-driven code generator and port it to this new 32-bit computer that was being developed. The lead architect there was Glenn Meyers, was leading the project. He was an IBM fellow that Intel managed to hire away and was leading that project. So I come over, move my office, and pretty much within a month of me arriving just barely getting started on things there was big shake up in the company. A big reorganization of the whole project and this-- that particular project was combined with a follow-on to the 432 program up in Oregon and all of the important architects were asked to move--pack up and move and go up to Oregon. Well, I was not asked to move. Instead, Glenn comes in my office and tells me that, "You know, you've heard about this reorg and we're moving up there. But what we want you to do is stay here and work on this new project that we're getting started, which is going to be a 32-bit product that's compatible with the 286 and the previous 86 and so on. So I was a little bit disappointed but excited to get going on this other thing. So I started out then as the first full-time employee on the 386.

Fairbairn: So what was, just very briefly, what was the microprocessor landscape within Intel at that time? the 432 was a major project in Oregon was seen to be the future of the company in terms of microprocessors, is that correct?

Crawford: So at the time, I can't remember exactly where the 432 was in its lifetime. I don't know if it was already out in the market or was about to go. I think it was probably about to go out to market because the fact that they were then reorganizing and trying to figure out where to go next. And, of course, you would have the architecture team go first and then the chip designers would come along later after the initial direction was set. So there was a 432 team up in Oregon and they had been working for several years on that very ambitious project. In Santa Clara was the 286 team. The 286 team was just finishing. I'm pretty sure they had first silicon at that time or they were very close to it. So there was a full team of chip developers working on the 286 at the time. The 8051 team I think had already moved to Arizona but they started in Santa Clara and then moved off to Arizona as a different product-line in the Intel universe of activity. So the idea for the 386 team was to really transition people from the 286, hire some new folks, and build it there in Santa Clara.

Fairbairn: And that was to be the future direction for the X86 compatible processors?

Crawford: So a polite way to say what the role of the 386 was to be the future of the compatible 8086, 286 processors. Maybe a more-- a jaundiced view is well it was a stop gap in a long line of successful stop gaps at Intel for I think, in a sense, the 8086 was a stop gap and the 286 was something that was there to continue the market while the 432, you know, a very ambitious program was being developed. So the idea with the 386 was to build on the compatible front not to be as ambitious as this new project, which was going to be a 432 follow-on. But to serve, you know-- but to serve some part of the market then that would value the compatibility and be there kind of to hold the fort, if you will, while this other project came on.

Fairbairn: So at the time you stepped into this role as an architect for the 386 you didn't know that that was going to be the flagship product at the time. You thought perhaps that this program in Oregon was to be the long-term direction for Intel, is that correct?

Crawford: So when I stepped on the beginnings of the 386 were very humble, to put it mildly. So the expectations were low. I think the idea was they wanted to get something-- a simple extension of the 286, get it out to market, get it underway and the expectations were low, which is why they had-- I mean, why did they put me on as the architect? Well, I had no experience at computer engineering. I had a lot of software experience. I knew the 8086 instruction set backwards, forwards, upside down, every little bit which was-- a very important piece of the puzzle. And I had a good software background, including this machine level understanding and kind of a passion for computer architecture. So at that time, Intel took a risk. They took a risk on me stepping in to lead the architecture of this new product, but the risk was mitigated because they had <laughs> great expertise on this other project and, in fact, my boss remained Glenn Myers for some period of time. He was both the lead architect of this new chip as well as supervising me on this little side project <laughs> compatible thing.

Fairbairn: Can I ask a short question, which is why would they send teams to different locations? It sounds like they would start a team up and they would send them to Oregon, somebody was in Arizona and then you were here. What was the logic of doing it that way?

Crawford: So even at that time back in the- at that time it was the early eighties Intel was very consciously moving things out of California. I think they'd taken the decision never to build another factory in California but wanting to expand in directions, have the growth occur outside of Silicon Valley. So they were constantly moving projects out that might've gotten started in the Silicon Valley area but to pursue the growth elsewhere. And the politics, the environment, price of electricity and water, the price of housing, I mean the cost of living in the area plus the competition. So if you hired somebody at Intel two weeks later they might go off somewhere else. <laughs> You know, there's a lot of movement within-- for a variety of reasons they wanted to expand elsewhere.

Fairbairn: So you were appointed the architect of the 386. Was there a clear goal for the 386 at that point? What did it have to accomplish? What speed/performance improvements, major capability improvements over the 286? That's what it was to be the follow in for, correct?

Crawford: So the initial goals of the 386--it's a little hard to remember now--but certainly some of the key goals for the 386 team were first and foremost make it a real 32-bit computer, so they wanted the 32-bit address space, 32-bit registers, 32-bit operations. That was kind of point number one. Point number two was to include paging as a memory management technique. The 286 had adopted a segmentation approach of-- 64K segment was the biggest chunk of memory you could manage on the machine, and it had a property of having kind of a two-part pointer. To identify a piece of memory you had to have a segment, a pointer for the segment or a handle for the segment, if you will, and then an offset within the segment. So the second aspect was to come up with a memory management scheme that provided a full flat 32-bit address space. And the third but maybe the first was be compatible and be able to run 286 and 8086 software, binary software.

Fairbairn: Binary compatible.

Crawford: Binary compatible, yes.

Fairbairn: Okay. At the time that you took on this task did you feel that any of these major specifications were a stretch, a reach, difficult to be done, or you thought, "Okay. This can be done. Let's roll up our sleeves and figure out how to do it."

Crawford: So the goals were pretty broad, I mean pretty open, and I thought, "Oh, sure. I'm sure we can do this." Part of the history of the 386 team at that time was we had two books that were required reading. So the first one was "Soul of a New Machine," which was the story of Steve Wallach and his team at digital-- not digital, <laughs> Data General and their challenge to stretch a machine out to 32 bits at the same time conflicting with a team in North Carolina that had the grand and glorious future, right? So it was very eerily <laughs> similar. So that was the number one required reading. The other required reading was "Mythical Man-Month" by Fred Brooks as a way of grounding us in doing a good job on the project. So we knew it was possible. Obviously the Data General experience proved it that there was at least one way to do it. So we sat down and set about proceeding with that. An additional goal, which caused quite a bit of conflict between myself and my boss, Glenn Myers, was that it's not going to be a new architecture. So that caused me no end of heartburn and a lot of arguments. In fact, at one point in I think it was August of '82 we had this big pow-wow meeting. Maybe it was '83. We had a big pow-wow meeting about the direction that we were going to take to accomplish these goals to stretching to 32 bits and then getting the memory management to support the flat address space. And Glenn kept saying, you know, "It's not a new architecture," and I kept saying, "Well, of course it's a new architecture. There's not a line of code written for this 32-bit thing. It- it's- you know, of course it's gonna be a new architecture." So we had this kind of semantic conflict. His idea was it would be very, very, very, very, very similar to the 286, and I was pushing for some minor changes to alleviate some of the difficulties with the 286 instruction set.

Fairbairn: Were your changes still within the context of binary compatibility or would they be in conflict with that?

Crawford: So the changes I was proposing gave a slightly different flavor to the address modes in the memory reference instructions. So on the 286 if I remember right there were two registers you could use as a base register and a different two that you could use as an index register. And I found a way of scrambling the bits when you're in 32-bit mode to generalize that, so any of the registers could be used as a base and any as an index. And, oh, by the way threw in a scale factor so you could scale the index by two, four, or I think eight at the same time and mash it together. And in fact that was one of the- we actually had quite a go-round between the engineering team, marketing team and Glenn in terms of trying to push that very subtle change--I mean it was a very small change--through, but he was so adamant that it was to be as minimal change as possible. He had dug in his heals and said no, so that's when maybe my political genes I learned from my mother- maybe I inherited those from my mother with her legislative career-- kicked in and I discovered, well, if I'm going to override my boss I'll need to have a good reason for it. So I shopped it around to my buddies in the software teams and they thought it was fabulous. "Wow! We could have any- you mean any register as a base register?" <laughs> "Really?" And then the marketing team loved it because they had heard no end of complaints about various difficult parts of the instruction set. And then the engineering team said, "Well, this is no big deal. You know, it's a couple-- it-it'll make the instruction decoder a little bit different and, you know, it's another shifter in the- in the uh.. address generator, but that's nothing in the grand scheme of things," and they could see the value of it in addition to listening to the marketing and the software guys be very strongly in favor of it. So I was able then to overrule my boss and get some of those changes included as part of the instruction set development.

Fairbairn: So were you the architect? Do you have a team of people working with you on the architecture? What was the size of the team working on 386 at this time and did that mushroom over time? Tell me a little bit about the development program itself.

Crawford: So the way the 386 architecture development proceeded was I was the lead architect, say, from the architecture side. So Glenn reported in to the general manager, and I was the architect from that. It turns out at one point there were three architects that thought they were architecting the 386. <laughs> So architect number two was Bob Childs, who was the architect of the 286, and he had a completely different model that was quite a big change from what the 286 was. I think he still had a strategy for it being compatible, but he had scrambled the bits so that now there were more registers, 16, and it was probably the main feature of it. And a guy who ended up working for me, Rakesh Agarwal, had another proposal, and I can't remember what it was. But he was not a long-lasting <laughs> architect, but he did have some ideas in the flow. He also worked for Glenn Myers, so he was kind of a colleague of mine on that side of things. Bob reported into the chip development team. He was on the 286 and reported in through the chip development. So at one point in time we had two competing directions for the 386 architecture. Not only did we compete internally but we had this fabulous idea of taking a road show out to visit some of the key customers and get some feedback from them in terms of, “Well, here’s door number one, door number two. What do you like?” So we had this curious thing of going out to various customers, getting some feedback. And at one point Bob decides, “Oh, you know, let’s swap. <laughs> You talk about mine and I’ll talk about yours with the next customer.” So we did that. <laughs> And so at the end of the trip we got back, we regrouped, and I can't remember what the feedback was. I mean it was so long ago. But as things turned out, of course, I ended up being- the thing I was proposing ended up being chosen, and we proceeded ahead with that.

Fairbairn: So how long did the architecture take? Did the outcome match your expectations and the goals? How did things-- just very quickly sort of summarize--

Crawford: Let me answer the rest of your answer, too - the previous question on the team. So after we worked through, “Is it going to be door number one or door number two,” and we had settled on kind of a general direction the architecture team was tiny. It was me. And we had hired- I think at one point I had four or five people working for me, two of them writing microcode and two of them writing test programs. And we worked with the logic guys-- or we must’ve developed an instruction set simulator as part of the team, and we were able then to try out test programs and check out the instruction set to kind of work out any kinks that might’ve been there, but really that was it. I ended up writing a third of the microcode on the 386 and had two other people. Rakesh Agarwal joined the team, and he wrote the floating point code. And Greg Blanck joined the team and he was involved in some of the- he picked up the protected mode and some of the other aspects, some of the more complex actions of the instruction set, emulation and the microcode.

Fairbairn: And so this program went on for approximately five years from beginning to actual delivery of the product, is that correct?

Crawford: So the project started in March of 1982. That’s when Glen Myers came into my office and gave me the opportunity to join the team. And we had a product announcement October of 1985, but that was back in the day when you announced the product before you shipped it. We didn’t end up shipping until sometime in 1986. So over that scope of time it was about three years to first silicon. First silicon came in a little over three years after the start of the project.

Fairbairn: And when did it become clear that this really was going to be the flagship processor for Intel versus this program that the whole team had gone to Oregon to work on, the follow on to the 432?

Crawford: So I think there’s two aspects as to what-- so one of the things that was just fabulous about working on the 386 project. We started with this very humble beginning, right? “John, can you- can you come on and, you know, be- be the architect lead for this compatible 32-bit product...” that was to be no new architecture and really a minimal change to be compatible and provide that path. So very humble and then started up very slow. But you start low, well, there’s no place to go but up. <laughs> And it was fascinating because every time a new team would come on-- so we'd had some architecture and we worked with a lead circuit and a lead logic guy early on, which then blossomed on. As the 286 finished up we would bring some of the logic guys and have a little more logic. They developed some RTL through RTL simulations and so on. Every team that came on was really excited about it. They thought, "This is really good. You know, this is- this is a- this is turning out to be a great project." We're fixing some of the problems with the 286, things that people had complained about and that we had difficulty selling to customers, and the team was gelling together and things were moving forward. Then the circuit guys came on. They were jazzed. The layout team started to come on as the 286 got out the door, and later on we had some of the manufacturing folks come in and marketing folks. Each team that joined just kind of raised the crescendo of excitement about the product. Now coupled with that was the start from scratch big bang 32-bit product was delayed. You know, no surprise there. <laughs> They ran into difficulties and they were getting delayed. In fact, one of the tenants for the 386, another requirement that I didn't mention earlier, was that we would share a bus with this new product, and the bus was being defined by the new product, and part of our role, I guess, was to act as a bridge. So we could have the hardware environment that would take either chip. Maybe people would start with our chip and have the software compatibility and then have the pathway, then, from a hardware perspective to take advantage of this new thing. So we started out. That was fine. They were in Oregon. We were in Santa Clara, but we more or less worked for the same company, so the communication paths were there and we were proceeding along those lines. We had a nice internal interface that we had to meet. Of course it changed. There was some flux to that but not so much as to really be a huge problem. But it got to the point where we were completing our work and the bus was delayed, delayed, delayed. So it got to the point where we had to have a decision. Are we going to seriously delay our chip to market and use this bus or are we going to break off, have a different strategy for hardware interface and then maybe the next product would have this bus compatibility. So the decision was taken to break compatibility, and we had a last-minute scramble headed up by Dave Vannier who was one of the lead logic guys on the 286. He came on and quickly defined a bus that was an evolution of the 286 bus, and we got the logic and the circuit and layout done just in time, pulled it all together and got the product out in- like I said, the first silicon came out in 1985.

Fairbairn: So it sounds like the progress to product was fairly straightforward. Were there any points of major doubt of, "Oh, my God. We've gone down this path and we didn't think about this," or "The performance is nowhere near what we expected," you know. Any major surprises or just more typical engineering challenges along the way to solve and get over?

Crawford: So with the hindsight of history I don't remember any major hiccups in the program with the exception of this compatible bus and having then to break off and do a new bus, things proceeded well. We were fortunate in that we were the first chip on Intel's CMOS process, high-performance CMOS process. So that gave us an advantage in terms of power, which also then translated to speed. Actually, the 386 micro-architecture was very similar to the 286, so even given that we were able to run at a much higher clock rate than the 286 had because of the new CMOS process, with the speed and the power advantages that it brought. And frankly if you remember the goals that I set earlier there was no goal for performance. I mean they wanted it to be faster, and I think the expectation was that we would at least ride the technology, which we did. But in terms of major microarchitectural advances there was not a real push for that, although we did reorganize the memory access path in order to accommodate a two-cycle off-chip cache, so we did have provision for a faster memory interface than was available on the 286, and that was an additional kicker on the performance.

Fairbairn: So let me come back to one previous question, and that is at what point did you realize that this was in fact the future of Intel? The 432 follow on was being delayed and at least my interpretation is that the status of this product must've changed over time.

Crawford: So over time the project just got better and better. I mean it was <laughs> just a fabulous experience to ride along that wave. One incident that comes to mind is a visit to IBM Research in Yorktown. Glenn Myers and I went there to describe the two products, and when we were there one of the key people we were meeting with was George Radin from the 801 design. He was there. He listened to both of us and at the end came up and gave us some feedback. You know, he congratulated Glenn, and I suppose they had worked together before. But then he turned to me and he said, "You know, your project is very important and, you know, it's going to be uh.. you know, very important for- for the company," and had some nice words to say about it. So there's one incident that comes to mind. Another thing that happened, in about August of 1985, after we had first silicon, I come to work and in my office is a big box, and turns out it's from the Paris sales team. And I open it up. It's a case of Veuve Clicquot Champagne.

<laughter>

Crawford: So I open it up and it's a nice letter in English <laughs> from the French sales office that said, "Congratulations on first silicon," or something like that. "We're really anxious to- to get the product. Uh.. you know, please continue your efforts at full speed. Uh.. You know, customers are- are anxious and- and waiting for the product." So that was fabulous. That was the one and only time I ever got alcohol delivered to me at Intel and any kind of praise from the field team. They were normally a very critical bunch, but this was fabulous.

Fairbairn: So you mentioned just in passing the IBM person representing or having participated in the 801, which is perhaps the earliest implementation of a RISC-based machine. During this period that you were working on the 386, the '82 to '85 or '86 timeframe, I believe, was the time when RISC architectures were becoming very hot items in the computer architecture world. How did that affect things at Intel? Did it affect you at all? Did you take note, or you were just on this path and you had to develop a binary compatible 386 and that's what you were going to do?

Crawford: So the RISC storm was gathering at that time. There were a couple of interesting chips that had come out, but the storm was just gathering, and it wasn't really until the 486 when the RISC-CISC debate really warmed up. In fact, we were so far along we couldn't have done anything, anyway. We'd basically have to scrap it and start over if we were to entertain any changes. But we introduced the product. It was nicely successful, and got us a foothold into 32-bit computing. And I had mentioned the goals early on of being 32-bit compatible or fully 32 bits both in the instruction set and in the memory addressability and then being binary compatible. Actually, one of the key decision points for the instruction set was the memory management. How are we going to do this thing? We got these segments. We need to carry those to be compatible, but we want a flat 32-bit address space. And that actually involved quite a bit of mental energy in the early stages, and trying to get that resolved right was an interesting challenge. And in the end we ended up with something unique, which is inventing a third address space, kind of an intermediate address space between the virtual space that the machine instructions would use and the physical space that went out to memory. We had this idea of a linear address space. So the segmentation would transpose the virtual address, two parts, into a linear address of four gigabytes, and then we would apply paging to that - we would then apply that into a paging algorithm and come up with a physical address. So that was really a second key aspect that made the product successful. First of all we had the full 32-bit flat address space. It was available to you. It was easy to set up the segments to ignore them, <laughs> which I think was the approach adopted by most software. The segments were still there available for any kind of operating system that wanted to use the smaller compartments or to compartmentalize things differently.

Fairbairn: so the 386 ended up being an incredibly successful processor for Intel. It sounds like you delivered pretty close to on time. There wasn't a performance spec. Was there a delivery spec, and how close did you deliver to on time with respect to initial expectations?

Crawford: Well, I'm sure we delivered on schedule.

<laughter>

Crawford: Of course, the joke there was-- in fact, the marketing manager- one of the marketing guys was Claude Leglise, and he very cynically pointed out that everyone delivers on schedule. You know, it's the schedule you set the week before you actually ship. Now in terms of where we were I'm sure we were a little bit late. I think we were later than what we had wanted. The bus confusion cost us some time, but I don't think we were more than a couple quarters behind the original schedule. What the original schedule is I have no idea. That's lost in the mist of time.

Fairbairn: So the next program was the 486, and are there any sort of major transition points? What did you expect to be your next challenge coming off the 386? Was it obvious that 486 was the next thing? Tell me about the transition to taking over on the 486 program.

Crawford: So after we got the 386 out the door- in fact, during the process of getting it out the door I was assigned to get us going on the next thing, which of course what do you do after the 386? The 486. And in order to get that going I had one employee, Pat Gelsinger worked for me as the microarchitect for the 486. So Pat worked out some options for the integer part of the instruction set. He worked for me for about six months in the October ’85 to maybe early ’86 timeframe. And we had a couple things we knew we wanted to do. We wanted to include a cache on chip. We had a transistor budget that would support that. And we went back and forth as to whether we’d have floating point integrated on the chip or have it as a separate coprocessor. At the time Pat was there he focused on the integer portion of the instruction set. The key item there was how to integrate a cache and how to improve the microarchitecture to get a much faster, much better clock per instruction (CPI) kind of an engine. So the challenge there was really to make best use of the cache. And Pat and I went back and forth. I remember the one discussion we had was whether the memory instruction time should be two clocks or one clock, and it turned out Pat was very adamant in saying, “If we just add this extra pipeline here, do the address computation in a parallel pipeline we can get the cache access down to a one clock per instruction throughput rate, so why don’t we do that?” And I said, “Well,” I said, “but it’s extra stuff over here.” And he said, “Oh, that doesn’t matter. It’s- it’s almost nothing.” So Pat kind of blazed the path for that and set the microarchitecture direction, which we then followed into the 486. So as people rolled off the 386 Pat then went off to do some other things, and other folks then rolled on, and we went about the 486. And rolling through some decisions there, a couple of key decisions on the 486, again, first of all was how do we get the performance and how do we compete with RISC? And it turned out because of Pat’s work we were able to have a path to a one-clock per instruction throughput rate. Does that sound familiar? That was really the big selling point of the RISC chips was exactly that. Now we certainly had some non-RISC things. We had these things called task gates that were miles of microcode you would wind through to transition state-- it was from the 286. It was kind of a hardware task switch kind of thing, which never did work out quite right, but all that was just microcode. And operating systems rarely use those, anyway, if at all. So a lot of the CISC stuff got to be- you could ignore it. It was there, we had to do it, but it didn’t really impact the performance much and it was tied up in microcode, which was just bits in a ROM, so...

Fairbairn: It was there for compatibility, but it was not required for any of the new stuff that would be created for it.

Crawford: Right. The majority of the-- if it was used it was used so infrequently that it was not a performance issue. In the load, store, branch that was all the meat and potatoes of computation that was all one-clock per instruction. Well, branches not quite but almost.

Fairbairn: So were there battles in Intel about now you’re onto the 486. What had become of this follow on to the 432? What were the tensions or steps in terms of setting the direction at this point?

Crawford: So when we got started on the 486 the 432 follow on team was still working on their product. We had shipped the 386 and they had not yet come out with their product. So in fact our internal rival shifted from that team, if you will, to the 860 team. The 860, Les Kohn [ph?] was the architect of that chip, and he got started before we got the 386 out and was proceeding with a RISC focused instruction set. And that had a lot of traction within the company. I think we saw the RISC activity in the competitive environment and said, “Well, maybe we’d better have- you know, have something in that regime as well.” And curiously we were both in the same building. It was both in Santa Clara in the Santa Clara Four Building on Northwest Parkway. And they were on one side of the building. We were kind of on the other, and it turned out we had a rivalry going. But it was a friendly rivalry; in fact, Les Kohn was very helpful in helping us set direction for the cache that we were going to put on the 486. I think he provided a cache simulator and a couple of other things we were able then to change to try to predict how things might operate and what size we should try to focus on and some of those kind of questions. But as things proceeded then as the 486 and the 860 proceeded the whole RISC-CISC thing really started to bubble up and boil over. And one thing that was disturbing to me about that was a meeting with the marketing team, the 860 marketing team at one point in time. We were comparing benchmarks that we had of the 486 running certain things comparing that with benchmarks of the 860. And now there's no question, I think everyone agreed that the 860 would do better. It was better performing on the set of benchmarks, but the 860 marketing team was upset that the 486 was close enough <laughs> they didn't believe it. They said, "That can't be right. You- you must've cheated," and so on and so forth. No, we didn't, but it was--

Fairbairn: They thought you cheated. They thought the 486 team--

Crawford: Yeah, they thought we cheated. Yeah, I mean that we had cheated and we were just making it up or whatever because it couldn't be. I mean it was a RISC chip. Theirs was a RISC chip. Ours was a stinky CISC thing, so it couldn't possibly be that good. So, well, it was.

<laughter>

Fairbairn: And was that largely because of this sort of one instruction per cycle capability that you were able to sort of build in enough RISC-type stuff into the "CISC" environment, or was it other aspects of it?

Crawford: So I think that the key to success of the 486 was not that we were going to beat RISC but we were going to be able to keep up and we were able to adopt a lot of the same engineering techniques, maybe not the same but we were able to get performance that was consistent with that. Well, we had to spend more transistors. It was more complicated. But we had a big enough market to justify the investment, so our goal was not to beat all the RISC guys. Our goal was to be within, I don't know, 20 percent or something of performance, at which point we thought that other considerations would carry the day, and we still could be quite successful. Yeah, the key thing was the one clock per instruction throughput on the 486. And a couple of interesting things about that. I had mentioned Pat Gelsinger's push to accommodate that within the execution pipeline. Even before that I remember a conversation with engineers from Sun and Bill Joy was in the audience, and we were probably trying to sell him on the 386, and back and forth questions. And a question that he asked that really struck me as profound was, "Well, how do you know when to start..." because we had this complicated instruction set. "How many clocks does it take you from the time you start decoding one instruction to know where the next one starts?" And that's a key parameter that we had to deal with in order to get one clock per instruction throughput. I mean in order to get one clock per instruction throughput everything has to happen at a one clock pace. So you start decoding one instruction. Next clock you better be decoding the next one or you missed it.

<laughter>

Crawford: So it turned out we couldn't do that in general because the instruction set was complicated. We had prefix bytes and variable-- you know, the instructions themselves would be variable lengths and so on and so forth. But it did turn out that we were able to look at just a handful of bits within the first couple of bytes to have 90, 95, probably a dynamic point of view 99 percent ability 99 percent of the time to say, "Okay, the next instruction starts here. We're sure of it, so start cracking it there." So that was a key thing. The other thing we had to do, we were a microcoded engine, so we had to figure out how to get-- and the microcode would take a clock to access the ROM to get the first word of microcode out. So we had to come up with a technique that moved some of the basic-- again, for 90+ percent of the instructions we moved the first word of microcode into the decode ROM, so as we decoded it would come out with an address to look up in the microcode in case there was more than one clock worth of stuff, and then the first instruction microcode, which for most of the instructions there was only one. It would come out of the decode ROM and, boom, off we'd go. So there were a couple of key angles of getting that one clock instruction throughput from this very complicated- well, not very complicated but certainly non-RISC kind of instruction set that made it happen.

Fairbairn: So, the-- were the design goals of the 486 clear? What was the-- what did you think-- what do you remember as being the major challenges? And did you think that it was a-- at the beginning, did you think this is all doable, engineering challenges? Or were there major issues that caused you some significant heartburn and weren't quite sure whether you could pull it off or not?

Crawford: So, the goals of the 486 were to be compatible with the 386. We really didn't envision any major instruction set changes, at that point. I mean we'd just been through this huge sixteen to thirty-two bit transition. So, a key goal, at least in my mind and I think in much of the engineering team, was to keep some stability. The second goal was to get a substantial increase in performance. And I think we had a goal of 2X, so basically doubling the performance over the 386. And it turned out for-- that we studied the performance quite well for the integer instruction portion of the instruction set, and we ended up getting, at the same clock rate, we got two and half times the performance, which was fabulous. And that was never to be equaled again in any of the subsequent generations. In fact, it's trailed off from there. You can imagine the law of diminishing returns. We were unable to duplicate that. Why was that? Well, the 386 was done on an old micro architecture. It was borrowed from the 286. We hadn't really pushed to improve the clock per instruction rate there. It was just more focused on other things. So, the 486 had an opportunity then to include an on chip cache and really speed up the memory access through the inclusion of that mechanism. And the one clock per instruction throughput rate was a huge improvement. Now, we didn't have-- the branch instructions were not one clock. Well, they were one clock if you didn't take it. But it was several clocks if you did end up taking the branch. So, that was probably the one area where we had more than one clock per instruction. And through that combination of things, we got a tremendous improvement in performance from the 486, never to be equaled again.

Fairbairn: And so, the 486, you were the architect of that for a couple of years. Was it a much larger team? Did you have to-- or could you handle that again mainly as sort of one, two, three man show to do the 486 architecture?

Crawford: So, the 486 team grew over the 386 team. In fact, that's another-- one of the many Moore's Law kind of corollaries if you will, or adjuncts is that a lot of things grow at some kind of compound annual growth rate. At one point, we looked at several successive teams at Intel and said it's about one point six. So, we get about sixty percent more manpower on the next generation from the current generation. And the 486 was right in line with that. So, we had more folks on the architecture team itself. So, instead of three or four, we had six or seven of us. And the logic, circuit, and layout teams were larger by a corresponding amount.

Fairbairn: The 486 development proceeded apace. And major surprises or huge challenges or big breakthroughs or other things you want to highlight as part of that development program?

Crawford: So, one of the key decisions we had on the 486 was how to deal with floating point. The previous thing about how to get performance I think was resolved very early on. And thanks to Pat. And I came up with the microcode strategy, and the decode strategy was also thanks to Pat. We were able to get this one clock per instruction throughput. But what to do about floating point was a big discussion point. Intel had a very profitable business in coprocessors. We didn't sell a lot of them, but there was a fabulous margin on those products. They sold for a lot more money than the main processor. Now, the attach rate was low, but it had that. So, we were looking at two alternatives. One was a coprocessor that would be a lot more complicated. So, we would have probably a similar interface between the 486 and a 487, if you will. And we would get performance by speeding up the internal operations within the coprocessor. Well, that would have resulted in a much larger chip, a much larger coprocessor chip, a big development effort and so on. The alternative was take the core of the 387 and put the hardware into the 486 and then include the-- of course we could eliminate all the overhead. We get performance how? Well, we would eliminate overhead of passing instructions and data between processor and coprocessor. So, by including that all on chip and completely blowing up the interface, or removing most of that overhead, we were able to get performance even though multiply still took the same amount of time. All the instructions were basically using the same hardware that we had on the 387. So, we went back and forth with that for quite some time and eventually took a decision as it turned out of course to include the coprocessor on the chip.

Fairbairn: So, there were the normal engineering challenges, but the 487 rolled out on a predictable schedule. And meanwhile these other programs, the 860 and the whatever-- the follow onto the 432 kept chugging along but never got in the way of the programs you were responsible for. Is that-- tell me about how that environment evolved.

Crawford: So, the 486 development proceeded apace. I'm sure we slipped our schedule. Nobody-- in fact, you probably need to set a schedule fairly aggressively so when you slip it's not too bad. It's still salvageable. But we weren't-- again, it wasn't maybe a couple quarters of from a gleam in the eye stage when we shipped this to when we actually got it out the door. It proceeded apace. The 860 had its own activity. I mean it was very active in the graphics market and had some success there, and in some of the scientific computing environments. They had a lot of interest in that. The 960 team was involved with this-- and I wasn't directly connected, but at the time, they had this big joint project with Siemens and again a very 432 inspired, very complicated, very interesting computer system they were building with them. And another spinoff, the 960 as an embedded chip came out and actually shipped for quite a few years. But it was all separate. It was not really competition per se. I guess I-- years later I had heard rumors of well, there were various points when somebody or other tried to kill the 486 program. But working on the project, I was blissfully ignorant of that if it occurred. It could have been just a fog of war rumors or something. But yeah we proceeded apace. One of the things that was interesting about the launch of the 486-- now in contrast to the 386, which started small and just grew, and was just a fabulous launch and hugging and kissing all over the place. When the 486 was introduced, we actually had a number of people who were disappointed. They said well what's new? And so-- which really frustrated me. Well, it's two and a half times faster and you had integrated floating point. You know, and the bus is so much-- more throughput on the bus, and so on and so forth, different clocking scheme to make it easier to build systems. There were a lot of things-- innovations in the 486, but there was no-- very few changes to the programming model, which again I thought was extremely important to keep that stability. And so, we actually had mostly from hardware folks, hardware oriented kind of folks were a little disappointed that there wasn't some huge new thing in the 486.

Fairbairn: Then you moved on. Tell me about the follow on. What happened after that? You moved on to a Pentium program that decided to call it Pentium instead of 586. Was this again just sort of roll over to the next generation? Tell me about that.

Crawford: So, before we move onto that, I'd like to relate some personal items that happened during this timeframe. As you can imagine, there was a lot of excitement and lot of my energy went into work with this whole 386, 486, and then that Pentium processor progression. So, starting in '82, up through '86, '87 was a particularly exciting time in my life. As I was working on this, Intel took a risk on me, gave me the opportunity to work on this 386 chip. And that was my work life. We described that in quite a bit of detail. Now, in my personal life, in 1982 I met a woman who was later to become my wife. So, July 3rd of 1982 we met at an Intel party. And we had a little bit of a rough start to our relationship, but then in the middle of 1984, in April 1984 we got married. So, this was right in the middle of the whole 386 activity. And then--

Fairbairn: She was still working at Intel at that time?

Crawford: Well, she didn't work at Intel. Her roommate worked at Intel and invited her along to this party. So, she had been involved kind of on the periphery. But she was-- we managed to bump into each other continued to see each other, obviously. And so, we got married. And my first sabbatical interestingly was June, July of 1985. So, we took off for three months. The architecture work was done, but there was an awful lot of work then to pull that all together and get the first silicon out. In fact, they probably were happy to send the architect away so that I couldn't make any changes on things. So, we went off on a sabbatical. And then we come back. We come-- during sabbatical we come to find Norma's pregnant, so first child was on the way. And then in February of ’86, my son was born. And we’re still going through all this. And in the meantime, I had been going through the process of getting back-- reconnected back with the church and had been participating in a Bible study with Pat Gelsinger, who actually took me on as a challenge to try to see where my leanings were and to see what-- so, I ended up then in 1987 becoming a Christian, getting baptized. So, in this span of-- while the 386 and then the early 486 was going on I had met my wife, had had my first child, had become a Christian. And all of this was going on.

Fairbairn: That really was a tumultuous exciting time wasn’t it?

Crawford: Yeah. Yeah, it was quite a-- yeah.

Fairbairn: Okay. So, just tell us about the transition, the major differences and advances from the 286 to the 386.

Crawford: So, the 286 was a sixteen bit processor. And a key feature of it was it had a protection model, a memory management protection model that was included. It was a segmented model. And being a sixteen bit chip, the segments were sixteen bit addressability. So, we were limited to 64K per segment. The physical address was larger. It was up to twenty-four bit address space. But it was awkward to get that because you had these smaller chunks you had to deal with. So, a key-- the key imperative for the 386 was to fix all that. And to have a real thirty-two bit chip, one that had thirty-two bit addressability, one that had flat addressability within a thirty-two bit space, and one that could support a full thirty-two bit physical space, a thirty-two bit, four gigabyte address space. And in addition to that, it had to be fully thirty-two bit computationally. So, we needed thirty-two bit registers for that. And it needed to be backwards compatible. It needed to run binary software from the older generations, the older sixteen bit generations. Moving ahead from-- so, with the 386, we had a thirty-two bit chip. It had full thirty-two bit addressability. It had paging as a key memory management technique. So, moving ahead to the 486, we didn’t want to have that kind of disruption. We wanted to maintain a good software, a consistent stable software model, but instead focus on performance improvements. So, the key innovations there were producing one cycle per instruction throughput rate in the microarchitecture and also including floating point on the chip as-- in order to complete the programming model. So, we got away from the coprocessor for floating point by integrating it on chip and providing it all in one package.

Fairbairn: Okay. So, tell me the major transition points from the 486 to the Pentium processor.

Crawford: So, with the 486 processor, we had a full thirty two bit machine. And we had transitioned into a RISC like one cycle per instruction throughput. So, the challenge for the Pentium processor was to again improve the performance. And the avenues we chose were again under the covers. The software model was not changed very much. But we took on the challenge of having two instructions executing in parallel. That was one key technique for performance improvement. The second one was a branch predictor in order to speed the control flow transitions. And a third angle of it was a complete redo of the floating point hardware to include a much more aggressive floating point engine in the Pentium processor.

Fairbairn: Okay so, the-- another major change for you I think in the Pentium processor was that you became design manager of the whole program as opposed to only the architect or architectural manager, is that correct?

Crawford: Right. So, my role changed from being the architecture manager, if you will, on the 486, into a project management role on the Pentium processor. And I actually co-managed the project. There were two managers. I was responsible for half the chip, Avtar Saini was responsible for the other half of the chip. And we changed-- we started out with me responsible literally for geographically half of the chip. So, I had a logic-circuit-layout for maybe the left half of the chip, if you will, and Avtar had it for the other side. Now, given our backgrounds, I was given the parts that were more software intensive, so the floating point, the protected mode and so on were in my half of the chip. And Avtar had the more circuit focus, the cache and so on. About halfway through the program, we shifted that and reorganized the project in a more traditional way. So, I had the front end. And he had the back end. So, I had the architecture, logic, test program validation aspects of things. And Avtar had the circuit, layout, physical design aspect of things.

Fairbairn: So, can you tell us briefly any major challenges? Or what was the most interesting and perhaps the learning experiences in the Pentium processor relative to the earlier designs you had worked on?

Crawford: So, our challenge in starting the design with the Pentium processor was how to get a performance improvement. And already we were spoiled. We'd gotten two and a half X on the 386 to 486. So, expectations were high. And fortunately while we were finishing up the 486, there was a small architecture team in Israel headed by Uri Weiser that pulled together some simulations to explore parallel execution. So, Uri and his team really paved the way I think-- again I was skeptical just like in the 486. I argued with Pat Gelsinger about-- let's be more conservative. Well, I was skeptical of the parallel execution. Well, how often are you going to get two things together? Maybe it's ten percent of the time. It's no big deal. It's not going to help very much. But I think what Uri showed was with the right selection again it's-- the dynamic mix is a lot different than what you see in the instruction book. So, the instructions flowing through the machine, ninety percent of them are simple. And by focusing on that and being able to pair up the most popular combinations, we were able to get a pretty good boost out of the parallel execution.

Fairbairn: And was it a major challenge for you to take on the whole design management activity versus the role that you played earlier?

Crawford: So, my challenge in moving into a project management role, I was comfortable managing the architects as well as the logic designers because I had worked closely with them in the past even though I didn't have oversight responsibility for them. But taking on the circuit and physical design was a challenge. I had had some interaction obviously, but very limited. And I had no skills in that area. So, part of the organization, that early organization where I did have circuit-physical guys-- carefully took the strongest circuit manager or circuit supervisor and had him report to me. So, I came to rely on him for the expertise and for the knowledge. And then my challenge was kind of auditing what was going on and learning how to ask the right questions and check the right data to make sure things were on track.

Fairbairn: So, so our time's limited. Let's move ahead. Over the next decade, you became an Intel fellow, architectural manager. One of the major programs you were involved in was the cooperation with HP in a joint Intel/HP team to define and build the Itanium processor. Can you tell us about the major steps that that was to accomplish and why Intel chose to team up with HP in that regard?

Crawford: So, after finishing the Pentium processor, the direction that the market wanted to move was pretty clear. They wanted to see a path to a sixty-four bit instruction set model that four gigabytes seemed like a lot of memory back in the '80s, but by the time we got into the mid '90s, there were already server systems that were bumping into that and were definitely in need of and wanting more memory. So, sixty-four bits was clearly a requirement. So, my first task was to head an architecture team to explore that in terms of what would Intel's sixty-four bit product look like. So, we explored a number of things, a number of models, both instruction set changes and microarchitecture changes to get there. That occupied '92, '93, up to early '94 timeframe. And early-- sometime late '93, early '94, Hewlett-Packard came to Intel and said HP Labs has been working on this new instruction set. We would like to partner in terms of building it. We've got the systems expertise. You've got the semiconductor expertise. We think this would be a good thing to get together. So, we formed a-- we investigated it, obviously decided to partner and proceed with kind of a joint development of the instruction set. Intel would produce the chips. HP would produce systems and software. And Intel then was free to sell the parts on the open market, as well. A key decision point on that was not only was HP interested in partnering with Intel for our semiconductor expertise, Intel was very interested in finding a partner in the server market, particularly in the high end server market that we could partner with and get more of an entrée into that market. So, there was good business reasons for this project to come together.

Fairbairn: So, was this processor to be in any way compatible with the previous generations you'd' worked on? Or was this a clean slate using the instruction set model that HP had delivered or developed?

Crawford: So, in terms of being compatible with the existing base of x86 software, the approach HP was interested in as a system vendor was we can do this with binary translation with interpretation and so on. We'll just make sure that the new chip is fast enough that whatever overhead is there is acceptable and we can get good enough performance on old stuff. And we'll carry over new stuff and transition, then, into a new environment. Within Intel, when we were defining the first Itanium chip, we had an idea of more of a hardware translator at the front end to smooth the transition so we could actually execute binary code-- binary x86 code through a process of-- related to kind of hardware translation, if you will.

Fairbairn: And were there other competing programs within Intel to continue the compa-- the strict compatibility plan? Or was this to be the future path for--?

Crawford: So, within Intel, I was responsible for defining a sixty-four bit instruction set extension that was compatible to previous things. So, we were working on this at the time HP came to us. So, we had developed kind of the sixty-four bit core of the instruction. We were in the process of working out how we were going to provide backwards compatibility, and how compatible were we going to be, what-- first class instructions, second class instructions, and how to resolve all that. HP came in with this opportunity, and we basically abandoned that. Now, in the meantime, the Pentium Pro team up in Oregon was in the process of developing their out of order x86 microprocessor-- or microarchitecture. And they were working on that while we were proceeding with this discussion with HP. And they actually participated in the earlier sixty-four bit stuff. And they were proceeding, of course, to deliver a thirty-two bit product that had the out of order engine and the consequent performance improvements they got from that.

Fairbairn: So, I'd like to now sort of switch to a higher level and ask you sort of throughout this, illustrious career at Intel, what were the-- what was the thing that you were most proud of or the events that you were sort of most notable in your career? And let's just start with that. What do you feel your major contributions were to the company and to the industry?

Crawford: So, looking back, I think my biggest contribution was actually the 386 activity, that defining a good thirty-two bit extension of that instruction set and establishing really the x86 kind of architecture that then was the key product in the industry for so many years. The instruction set definition of course carried on. The microarchitecture, the product itself, was no great shake. I think I had indicated before, I mean it was a tremendous accomplishment. But it was really kind of built on a shoestring, if you will, or less than a great budget. But it provided a key product at the right time. And then be able to continue that with the next couple of generations, and then hand it off to the Pentium Pro team, I take great satisfaction with that activity. So, it's either-- on the one hand, I think it's the instruction set definition and be able to fix some of the major problems with the previous products. The other thing was this marathon. I view it as a marathon. We started the 386. And before you were done with the 386, you've got to get going on the 486. Before you're done with the 486, you've got to get going on the 586, which became the Pentium processor. It's like running a ten year marathon, was another accomplishment, and being able to provide continuity, in fact, more continuity than we had experienced in the past being able to carry over more of the chip designers to the next project and keep the momentum going. I think that would be another thing I would cite.

Fairbairn: So, you obviously made a decision to sort of continue on this path. Do you have-- looking back, do you have any regrets about decisions you made along the way, both maybe choices within Intel, or choices on the program? And also, you must-- you became a very visible person in the industry as a spokesperson for the highest volume selling processors out there. You must be constantly pinged by outside companies to come join them. What-- and you made a choice to stay with Intel throughout that period. So, both in terms of your internal career as well as other choices that may have presented themselves, do you have any-- would you have done anything differently? Any regrets or not?

Crawford: Well, I would say one regret that I have looking back was a half billion dollar regret, was the floating point divide flaw on the-- within the Pentium processor. So, that was on my watch. I was the project manager. And looking back, we had one of the goals of the Pentium processor was to dramatically improve floating point performance. We completely blew up the hardware and had a significant new hardware engine, all new microcode, all new algorithms, and so on and so forth. As part of that, we recognized that the transcendental instructions, which were included as part of the instruction set on the chip, were going to be a challenge. We had brand new algorithms. All kinds of things could go wrong. Numerically, they could be-- we could miss stuff and so on and so forth. So, we had a huge validation put in place for those instructions. We had the multiplier and the more basic operations, multiply and add, were thoroughly tested kind of by nature of running anything floating point, we were able to get good coverage of that. But the divide was something different. Here was an operation that complicated. It was slow. And consequently, many floating point algorithms did whatever they could to avoid doing divides. So, we had limited coverage. And that was a blind spot for us. We did not have-- if we had focused just a fraction of the effort we'd put on the transcendental instructions onto the divide, things would have turned out differently.

Fairbairn: So, when did that-- I remember that was a major issue within the industry. Just briefly, how did that unfold to you? And when did you realize we really do have a problem here? At first was it there must not-- couldn't be? Or tell me a little bit about that experience of discovery.

Crawford: So, looking back through the fog of history trying to remember what was going on there, I had already moved off of the-- moved out of the project group and was into this idea of looking ahead for a sixty-four bit instruction set. So, of course, I was still in touch. And I think the first discovery of it was within Intel. We had discovered that there was a mistake and would occasionally give you a few bits of-- would give you inaccurate answers of more bits than we had specified. So, as I recall, there was a fix put in a place. And we were spinning the chip to produce that. Before we got that completed and the first silicon-- or move the production over to that, the whole Professor Nicely discovery and the publication of that hit the fan, if you will. And there was huge uproar of the computer industry. Well, here's this chip. And what caused the uproar, or what—it didn't cause the uproar, the problem caused-- the flaw caused the uproar. But what added gasoline to that fire was Intel's insistence, "Ah, it's no big deal. It's a few bits and most of-- very few people are ever going to run into a problem with it." And they stuck with that line for a couple of weeks while the press crescendo built and built and built and built. I think at one point IBM then said we're not going to ship any more of these parts until Intel fixes it. That and some other events happened. Eventually, Intel said, "Okay, mea culpa. We've got a problem here. And we will replace any chip. If anyone wants us to replace our chip, send the old one in. we'll send you a replacement, no questions asked." And at that point, the furor died. The next day, there was some press. And the day after was almost nothing. So--

Fairbairn: And you already had the fix in process, so it was--?

Crawford: We had the fix in process. So, the challenge was-- when I got involved it was not one of finding what the problem was because that was already known. The problem was how to educate our management on what was going on, try to better understand what the implications were, who might be affected, how to present the information, and then how to get-- how to work internally because we knew it was a problem. And being engineers, we said well this isn't good. We have to replace these parts. Well, then the business guys would say, "Well, that's a lot of money and does it really matter?" All these chips ship with some problems. I mean there's not such a thing as a perfect chip. So, there's always things that-- errata and workarounds and so on and so forth that had to be accommodated. But this was a case where there wasn't a satisfactory workaround. So, working through that and being-- fortunately for me, I was kind of one step removed. I was involved in the day to day operations of it, but the floating point architect and micro program developer, Harsh Sharangpani, really took the brunt of that. He was the face and wrote a bunch of stuff and talked a lot. But we worked through that. And Intel, I think, liked to say we learned a lot about being a-- at that point, we knew we were a consumer brand. And we knew we had to learn some new lessons about the responsibility that comes with being a household name and wanting to hold our brand up as high quality, which one of the things you have to do is, in a case like this, say, "We'll stand behind the product and no questions asked, we'll take care of it." And we developed a process, then, of publishing errata. Before that, it was secret, and we would give it under non-disclosure to our customers. But it was not anything that we would make public. So, moving on from that, we started publishing the things, fortunately small things or issues that come up with the chips.

Fairbairn: So, that was internally a regret. What about externally? You must have been pursued by other companies dangling significant monetary and challenging jobs. How did you handle that?

Crawford: So, I never really was tempted to leave Intel. I was very much for that period of time, it was exactly where I wanted to be. And they were at the center of the microprocessor market place. Their products were the leading products in the industry. I had the opportunity to be a part of that and to set direction and develop products that millions of people would use. So, there really wasn't a temptation for me to move away. And I was happy with the compensation that came and being named a fellow was a great-- was a nice thing. And I enjoyed that role and just really never thought about moving on to something else.

Fairbairn: Okay, let's sort of wrap things up. Is there any-- what other message would you like to give, major point you'd like to make in terms of that? And do you have advice for somebody such as yourself just coming out of the university in terms of things to pursue? And what do you find exciting in the world today? If you were just starting out, what would you want to pursue?

Crawford: So, my advice to someone come out now and-- my advice now to someone coming out and joining the workforce in the computer industry would be to look for a role that's at the boundary of computation and biology because I think there's an awful lot of very interesting things coming in that space, bioengineering, man-machine interface, and a lot of activity there, as well as maybe big data with cancer genome stuff. There's just a tremendous amount of innovation and very important work going on at this combination of biology and computing.

Fairbairn: All right, John, thanks very much. We very much appreciate the time you spent with us today and fascinating and tremendously successful career, so congratulations.

Crawford: Okay.

END OF INTERVIEW

