



## **Intel 386 Microprocessor Design and Development Oral History Panel**

Participants:  
John Crawford  
Gene Hill  
Jill Leukhardt  
Jan Willem Prak  
Jim Slager

Moderated by:  
Jim Jarrett

Recorded: December 2, 2008 Mountain View, California

CHM Reference number: X5063.2009

© 2008 Computer History Museum

**Jim Jarrett:** Okay we're here today, December 2, 2008 at the Computer History Museum, to talk to the Intel 386 microprocessor development and design team and this is the first of two tapes; there will be another tape made that talks to the marketing team. So we want to begin with some self introductions by the members of the team. John Crawford let's begin with you.

**John Crawford:** Thank you. I'm John Crawford; I was the Chief Architect of the 386. I got an undergraduate degree from Brown University, 1975, Masters in Computer Science from the University of North Carolina in 1977 and I joined Intel right out of college. The first five years I worked as a software engineer at Intel developing software tools for the 8086. That prepared me nicely for the 386 job because I knew the instruction set inside and out of the compatible processors. I started work on the 386 in early 1982, and stayed with the project through completion. I went on to architect the 486. I co-managed the Pentium Processor Team and then later on I led the HP-Intel team that developed the Itanium architecture. I'm still at Intel working on various other chips and technologies.

**Jarrett:** What's your position at Intel now?

**Crawford:** I'm an Intel Fellow, which is the top of the technical career ladder; I was named a Fellow in 1992, long after the 386.

**Jan Prak:** I'm Jan Prak; I have a Masters Degree in Physics, and a Doctorate in Physics from North Carolina State University that I obtained in 1970. I started work in Silicon Valley in 1972 at a company called Intersil and I worked in consumer and medical chips in CMOS technology and I advanced to lead a group that started to work on microprocessors there. Then at the end of 1981 I joined Intel to work on a new processor project which was not yet called the 386 at the time. On the 386 project I originally was the project manager for the first half of the project and then I was in charge of the performance and layout as well as four of the eight units. After the 386 I became the manager of the co-processor focused group, which later also included the i860 product line. Much later, I was part of the Itanium Project that John mentioned, and currently I'm retired.

**Jill Leukhardt:** I'm Jill Leukhardt, I have an undergraduate degree in electrical engineering from Johns Hopkins University and a Masters degree in electrical engineering from Loyola. I went from my undergraduate studies to an engineering development role at Black & Decker in New Product Development and from there to Intel in the Field Sales Organization where I was a Field Applications Engineer and then a Regional Architecture Specialist, which was a senior position charged with teaching our customers about why Intel architectures were superior to Motorola architectures. I was recruited out of the field to come to Santa Clara in August of 1982 to work with John Crawford on the definition of the 386. After that job I became Chairman of the Microprocessor Strategic Business Segment and then I was Product Marketing Manager of High Integration Microprocessors. I then left Intel to move back to the east coast where I spent three years in a wholesale microprocessor and microcomputer distribution business, then spent another 10 years in a virtual private networking [SafeNet] business and retired to write a novel.

**Jim Slager:** I'm Jim Slager, I grew up in Illinois. In 1963 I went to the University of Illinois for an Electrical Engineering degree. In the summer of 1966 I was fortunate enough to get a summer job at IBM in Poughkeepsie and landed on a project which had the goal of using minimum amount of hardware and moving as much of the design into software as was possible. This was before minicomputers; nobody ever dreamed of a microprocessor but it got me thinking about the concept very early. In 1968 I went to Bunker Ramo Corporation and was scheduled to go on to a project with bipolar technology and I learned that there was an MOS computer in the works and I managed to get myself transferred onto that. It wasn't CMOS it was PMOS in those days. And so I got involved in MOS technology very early; my career was very fortunate in timing of having both the microprocessor concept and MOS technology from the beginning. Eventually after a few years I moved to Rockwell International and the descendant of that is I think called Conexant today. There I walked into an environment where we had a complete design system from beginning to end with checks on everything, all automated. We were designing microprocessors ---they were used for calculators and they were custom so the world didn't know about 'em in general --- but again I reinforced my MOS and microprocessor experience.

Eventually I moved north and spent a couple of years at AMD. Then in 1978 I joined Intel, the same day as Bob Childs. We were the first two engineers for the 286 microprocessor. At Intel I was somewhat surprised to learn that their design methodology was very primitive. It made me long to go back to Rockwell, but I didn't. Also I quickly learned that there was this ominous dark object hanging over Intel, which was threatening to crush it called the Zilog MMU. That resulted in the 286 taking a full year to define before the design started and resulted in something which I think we'll hear a lot about during the rest of this taping so I won't go any further. After the 286 I was pretty much the Logic Design Manager for the 386 or maybe RTL Manager. After the 386 I spent some time on the 486 but then I left for the land of RISC because I was convinced that the RISC companies would prosper and Intel would not. So with that I'll turn it over to Gene. [Jon Slager editorial note: For about the first half of the project I was the Logic Design Manger, but in about the middle of the project, Jan and I split the chip in approximately half with each of us managing the loci, circuit, and layout for our half. At around the same time, Dave Vannier took over the logic circuit and layout for the bus unit. Also, in general, at around this time the very concept of logic design was giving way to RTL, so logic design was becoming an obsolete term.]

**Gene Hill:** I'm Gene Hill, I got a BS in electrical engineering from Oregon State University in 1969 and my whole senior year I got into a graduate course on building integrated circuits and I got so fascinated, I was cutting most of my other classes to do that. So I went out to interview and I didn't want to build integrated circuits, but everywhere I went they kept cancelling my interviews and shipped me off to the Integrated Circuit Group. So after I graduated I joined Collins Rockwell doing an Integrated CAD Semiconductor Design System. So I cut my teeth on completely automated place-and-route, diagnostics, a very elegant system. I worked there for a couple of years then I moved up to Silicon Valley in 1972 for a company doing custom chips and each engineer would develop two chips a year and we did a lot of calculators, a lot of printed calculators and actually did a microcontroller for Burroughs. I proposed a group that would have one CPU chip and one I/O chip and we would take on all the custom designs that way. Well I kind of got booted out of the office and was told never to use the words programmable or anything else because it would ruin the custom design business. Then they gave Burroughs back the rights to their microcontroller in exchange for a fixed amount of ROM codes, so I figured well this is it, if I'm going to do programmable chips and CPUs I'd better get to Intel.

So I started at Intel in 1976 in the microcontroller group, and at that point in time, archaic isn't quite good enough to explain the design system. I took over the design on the 8048 microcontroller project and one

of the first things I did was collect all the napkins and have schematics drawn so we could document the chip that was already in manufacturing. I eventually was in the group that architected the 8051 microcontroller that Intel then subsequently gave away to every company that had a patent contention with them. They learned on the 386 not to give things away. When the group went to Arizona I stayed and took over the implementation of the 286 project and brought it out, put it into production and then halfway through the 386 project, took that over from Jan. I stayed as Director of Engineering through the 486, then I also went off to RISC land, did RISC chips at LSI Logic. I was amazed how much less resources it took, but it's a very tricky landscape for business. Sun pulled the projects inside,(???) I went to a start up, learned a lot of lessons but got no money, then went back to Sun to help them with their first RISC project that they did in house. Then I spent the last 10 years of my career at AMD doing X86 processors and I'm now retired.

**Jarrett:** Let's begin, let's go back to early 1982 and Jill can you give us kind of a sense of what the industry was like at that point, what the environment was like, where was the PC at this point?

**Leukhardt:** Yes I would say that there were three key elements in the competitive environment at that stage. In terms of the PC it was very early on in terms of that phenomenon. The PC had been introduced I believe in 1981; that was the 8088-based PC from IBM, and just starting on in terms of other companies hopping onto that bandwagon. There really weren't very many other entrants. We were not cognizant of the importance of software that was being developed for the PC and its user base, so what is obvious today in terms of the installed base of software and the vast importance of that compatibility was not at all obvious when we were entering into the definition phase for the 386. In terms of the competitive environment, the Motorola 68000 was a superior machine, and we knew it and felt it very strongly. We were competing with everything we had. We had put together a marketing campaign called "Crush" and we used everything we had with the 8086 and the 186 and the 286 and everything surrounding them: their peripheral components, our Field Application Engineers the higher clock speeds of those products, benchmarks, anything that we could throw and stick against wall that would make a good story as to why customers should design with our products instead of the competitor's product. We were scrambling. We perceived that we were going to be later than the 68020, the Motorola 32-bit product that we would be later to market with the 386 and so we felt that very strongly and that created a real sense of urgency in terms of what we were trying to do. Third, we were stuck with the 8086 approach to segmentation that was wildly reviled; everybody hated 64K-byte segments. They limited the size of data structures and that was perceived to be and was actually a limitation in certain applications. Programmers in particular and compiler writers and others just saw that as a huge limitation. Jim you want to say something?

**Slager:** Oh yes it's just humorous when you look back...that we thought it was so important to have a segment-based memory management scheme because of the Zilog MMU and it was defined into the 286 and we just thought it'd be the greatest thing in the world but customers just didn't agree, I don't know what was wrong with them <laughter.>.

**Jarrett:** So it's early 1982 you've got the 286 out; there's;nobody particularly wild about it, but it's moving along and now a new effort is beginning to develop: a follow on to the 286, and that's where you all come together. What was the environment within Intel at that point and the thinking about this new chip?

**Hill:** Well actually that chip started as a non-chip; the 286 was so unsuccessful, Bill Gates called it “Brain Dead”. IBM said there could not be a follow-on; it was a dead-end architecture. So I had the job of finishing up the design of the 286, putting it into production and then disbanding the group because there was no 32-bit 386 that could be done. So that’s the start of the environment, I actually got Bob Childs, one of the architects of the 286, transferred to the 386 development team so he could write diagnostics. He worked underground for about six months and we would kick ideas around and there was a meeting where we made proposals for the 386. We always thought it was our brilliant work that got some traction when work began on the 386, but I actually think it was undercurrents of other things that were going on in the company at the time. So the 386 went from dead to a little bit of life breathed into it in 1982.

**Jarrett:** If you couldn’t do a follow-on to the 286, what was the next processor that Intel was thinking of bringing out?

**Crawford:** This was all in the shadow of the 432 microprocessor development at the time. The 432 was a very ambitious project that Intel was very firmly committed to and unfortunately it was also late and had slipped pretty significantly; so we had a number of gap fillers that were thrown into the breach. That was a little before my time, but my understanding is the 8086 microprocessor was the first of those gap fillers, followed by the 8088, the 186 and the 286. That’s another key piece of information here: the 432 project was really supposed to be our big thrust in the microprocessor market and these other efforts were really more or less gap fillers.

**Prak:** There was a variety of proposals to come up with a new architecture because there were a lot of people who realized the 432 was fatally flawed. So the project I joined originally was code-named the P4 and it was a whole new architecture that was very VAX-like. It was developed by Glen Meyers and the people in Oregon who had been responsible for the 432 wanted to try again. A number of them realized there were problems, so they wanted to do it over basically. So they had a proposal and as Gene indicated the 386, 32 bit follow-on was kind of the step child already in that discussion.

**Jarrett:** So what were the limitations of this 432 processor?

**Leukhardt:** Well predominantly performance; it was very slow.

**Jarrett:** So there was this nascent 386, there was the P4, and were there any other likely successors?

**Crawford:** I think that’s the roster. I also joined the P4 team late in 1981 or early in 1982, and shortly after I joined there was a big reorganization of that project to combine the P4 work with a follow-on to the 432 project. As it turned out, all the important folks on the architecture team were invited to go to Oregon and I wasn’t, but I was asked to stay down in Santa Clara and work on a compatible architecture, something that would be a 32-bit compatible upgrade to the 286.

**Jarrett:** So you’ve used the term “gap filler;” it was filling a gap between the 286 and what?

**Prak:** It was called a “single high end architecture” and Glen Meyers and the other people moved to Oregon to do it. The code name was P7 and they were combining the ideas from both of the camps, the 432 camp and the Glen Meyers camp.

**Jarrett:** So it sounds like there were sort of two camps within the company at that point or maybe three or four.

**Slager:** It had gone from three to two at that point.

**Jarrett:** So this 386 effort was launched and I guess the first thing we need to talk about is the definition of it. I guess Jill and John you were quite involved in that process; tell us about it.

**Crawford:** So the definition effort started-- I was there a little bit before Jill, just a few months before and there actually were two architects on the 386. There was the effort that Gene talked about earlier that Bob Childs had pulled together. It was a proposal for a 32-bit extension of the 286 and Glen Meyers had asked me to step in and do one, and so we had these two sketches of what an extension might be. So we had kind of the battle of the architectures going on a little bit. Part of the effort to resolve that situation involved a series of customer visits and that’s where Jill comes in -- you were there for those visits Jill?

**Leukhardt:** I came shortly after those visits but my understanding is that you and George Alexy took those two sketches to seven customers?

**Crawford:** Right okay, so let me continue that line of thought. George Alexy was the Marketing Manager and he took Bob Childs and me out to seven selected customers to bounce both sketches off of the potential customers and get feedback from them. We did that and we got some interesting feedback: one kind of strange suggestion that was made at one point after we’d done three or four of these was for Bob and I to swap and for him to pitch mine and me to pitch his. Well we never did that but it was, you know, for the most part a friendly competition that was going on. We of course both thought our sketch was the best, but we were trying to do the best thing for the company and for the project to make sure we had good feedback from customers. We also got feedback from the architecture specialists, somewhere along this process.

**Jarrett:** This is the Intel field ....?

**Crawford:** The Intel field yes that Jill mentioned earlier. These were the architectural specialists from our field that also were not shy in giving us good feedback on different proposals.

**Jarrett:** So what were the key differences between the two approaches: yours and Childs?

**Crawford:** My proposal was a simpler upgrade and I think kind of the essence of it was to focus on the key issues, which were to extend the address space to 32 bits and in particular provide a flat addressing of 32 bits. That was a key failure or lack in the 286 that I think was mentioned earlier. Second thing was

to make it a full 32 bit machine so have some way of giving it a full 32 bit instruction set. The other thing that we wanted to fix was to increase the number of registers. The proposal I put forward was a more minimal extension and admittedly it fixed two of the three issues. It provided the flat 32 bit addressing. It supplied a full 32 bit instruction set. It did not change the number of registers. The proposal and the instruction set looked-- the instruction setting coding was very similar to the 8086. So the instruction decode piece would have been a small change or a much smaller change. Child's proposal on the other hand tried to address all three and in doing so he ended up with a much more complicated instruction encoding strategy. In particular the 32 bit instruction set was quite different from the 16 bit instruction encoding. It did provide the opportunity though to increase the number of registers, which addressed the third point. Today I can't remember how well it did on the first two but he did have the distinguishing factor that it increased the number of registers.

**Jarrett:** The customers gave you input and chose your approach -- correct?

**Crawford:** Well I think we got mixed feedback from the customers. There were some customers that didn't care at all about 8086 compatibility. We wanted to see a broad range of customers, some of whom weren't even using our products, so clearly they wouldn't care much about the compatibility. Others were quite concerned about it, but I think overall the feedback was compatibility would be nice to have but not critical, and that is kind of curious looking backwards. On the other hand, our field application engineers gave us very strong feedback that we had to run the old software, and that would be critical for success of the project and also critical for continued success of the 286 and the other products.

**Jarrett:** Was this was PC software they were concerned about?

**Crawford:** Oh no, not at the time, I think it wasn't that long since August of 1981 with the big IBM PC introduction. It seemed that the PC was an interesting design but not one viewed as really the key thing to win and to do well on. Instead there was still a broad range of designs from the terminals to kind of little minicomputers to the personal computers which were just emerging.

**Jarrett:** How about the work stations? At this point was that considered to be a market to focus on?

**Crawford:** Better let Jill answer that one.

**Leukhardt:** Well we thought that was a key market segment for the 386. It was not a market segment where the 286 was doing well at all; it was a market segment where the 68000 was cleaning up principally because the 286 was not viewed as a machine that ran Unix well and the 68000 was viewed as a natural Unix machine. So when we were working on the 386 definition we wanted it to have as one of its attributes being able to run Unix well. So that's one of the things that influenced us in terms of wanting to have a way to run the 386 as a flat 32-bit machine, and that's one of the things that led to all the angst in the definition process about compatibility versus a flat 32bit machine and how they would coexist.

**Jarrett:** So that shapes up as a key challenge architecturally; how did you address that?

**Crawford:** So the key addressability challenge was how to be compatible with the segments that the, 8086 and even more so the 286 had in its architecture, yet provide a flat, basically unsegmented 32-bit address space. The problem with the segments was every point or two of memory object now had two pieces to it, you had to say what segment it was in and then where was it located within that segment and now the programmer had to keep track of not just one address space but multiple little address spaces in terms of these segments. So the challenge was to come up with a way to satisfy the need to address a 32-bit address space, which at the time was huge. 64 kilobytes was a generous PC at the time, you know. Four gigabytes: wow, four billion bytes of memory, who could ever imagine that? So the challenge was to come up with an extension of the address space that solved that.

In fact that was one of the most difficult architectural issues that I had to wrestle with: how to keep that compatibility with the segmentation yet provide this thing and I know we went through endless iterations and had a lot of advice from many people. In the end the thing that worked was inventing a fictional address space in between the programmer's virtual address space and the physical address that you go to memory with; in fact we had to invent a new name for it, so we called it the "linear address space" We kept the segments but we provided the ability to have a segment that was four gigabytes in size and that let the workstation guys and the Unix people address this four gigabyte flat address space basically by setting up one segment and having all of their programming objects within that one segment. They simply loaded the segment registers with the pointers of that segment and from then on dealt simply with offsets within that segment. On the other hand, code imported from the 286 could use the segments and operate them as they had had before, and so that was a way to have a compatible upgrade yet provide the 32-bit address space.

**Slager:** Of course there's controversy about all of this and, you know, it's incredible and one thing that I remember well is that the architect of the 8086, one of the two architects, Steve Morris, who started the whole X86 phenomenon...he was bitterly opposed to the segmentation model that was installed in the 286. When the 386 turn came he was very active-- he was still at Intel in software somewhere-- and he was very much opposed to the segmentation model because of the two-part pointers. In fact he called it software poison. This was the environment that John had to work in and he did come up with a brilliant solution. We ultimately ended up with the best of all worlds: we had the compatibility, but that whole segmentation model which was pretty much generally hated could just move over to the side and not interfere with the software. It still had to be there, we had to design it in and test it and that was pretty bad, but it could get out of the way and not prevent the success of the 386.

**Leukhardt:** Now in some ways we're getting ahead of ourselves because we got to the solution that included segmentation plus a flat address space, sort of the "have it your way" approach after we made the decision that we had to build a fully compatible machine that is an object code compatible 386, and that took a long time to decide. In retrospect that seems like a completely obvious decision but we wrestled with that decision for months and it was a tough decision-making process.

**Jarrett:** What were the other options?

**Leukhardt:** To make a machine where the source code could be recompiled to a new architecture. Because we did not understand the PC phenomenon and the value of the software that was going to be developed for the PC, we didn't understand the value of what that installed base of software was going to



be and all the object code that was going to exist. We were very fortunate that we had those regional architecture specialists from the field talking to us about how that object code compatibility was so important. They were out there selling 286's and the fact that code was being developed for 286's, and they had to be able to say to people "The code you develop for your 286 will run on the 386." That was an integral part of what they were doing everyday and that was really a key part of their message. We were getting that feedback from them and that got through to us.

**Slager:** I think the clincher was that there were stores now selling PC's and stocking diskettes with binaries on them and there's no way to change that. You can't make them stock two versions. By the time the PC had grown big enough, then it started to become obvious and the binary compatibility finally disappeared as an issue.

**Leukhardt:** But I can remember coming out of definition meetings that we had led and at one of them I remember going to my boss's office with tears coming down my face saying "I have no idea how we're ever going to get this resolved" because it was completely religious.

**Jarrett:** So John, you came up with this new approach that would accommodate both segmentation and paging. Was there any kind of a performance price that you paid as a result of this?

**Crawford:** That's an interesting question; in fact it was obviously a big concern because we were putting in two translation steps. We had the same rotation step which came from the 286, which involved adding a segment base address to whatever offset you had. Then we wanted to take that through a paging mechanism to translate that linear address, which is this new invention, into the physical address to go out to memory. That was very carefully designed -- in fact it was a key focus of the design -- to make sure we could slide that into the pipeline and not lose performance from that extra step. Of course it's a critical path on any computer, and that was a big concern all through the internals of the chip and even the BUS definition that was a careful concern. But the advantage was, I think it was mentioned before, we got the best of both worlds: we had segments for compatibility and paging for a flat address base; that was pretty much everybody.

**Jarrett:** Now internally, did everybody buy into this immediately or was there some debate on this approach?

**Crawford:** Well, I think that came at some point in 1982, but before that there were many proposals, obviously which didn't survive, that didn't solve the problem as completely. At some point there was a decision taken to go with my architecture as opposed to the Bob Childs architecture. As I remember, a key part of that was involved the software compatibility question and how efficiently we needed to run old software, and whether there would be a mode bit and basically have two machines in one or something more tightly integrated. I think for reasons of simplicity and for efficiency and time pressure we opted for the simpler model, which is the one that I had proposed that gave us the 32-bit extensions but without having a mode bit and without having a huge difference in the instruction sets. As I mentioned earlier, the price for that was we kept the eight-register model which was a drawback of the X86, but not a major one. I guess in addition I got half credit for the register extension because I was able to generalize the registers. On the 8086 they were quite specialized and the software people hated that too. One of the

things I was able to get into the architecture was to allow any register to be used to address memory as a base or an index register,,and was even able to squeeze in a scale factor for the index.

**Slager:** You know we went from huge controversy about everything and then it all came together and at that point nobody argued anymore. I can't remember the boundary there of the event when the decision was made and everything changed, or whether it just happened kind of gradually?

**Crawford:** Yes that's a great question, I don't remember. One big event I do remember was a meeting at the Le Baron Hotel in August of 1982 and the topic of the discussion was which architecture we were going to go with. I don't clearly remember exactly who the contenders were or what the points of contention were. About the only thing I remember from this meeting was my boss Glen Meyers it seemed like every ten minutes or every two minutes would say "It's not a new architecture" and enforcing some point as to we have to go this way because it can't be a new architecture and I would just get steamed at that and say "Well of course it's a new architecture, this whole 32 bit thing is brand new, there's no software for it, of course it's a new architecture." Anyway we both survived that meeting and hopefully something got resolved other than yes it is, no it isn't. So that was one event, I'm sure it contributed to some of the convergence, I can't remember how.

**Leukhardt:** I remember things coming together toward the end of 1982 but that there was still some controversy in terms of the common bus decision that we made at that time and we were all in "disagree-and-commit" mode that the bus for the 386 was going to be the same as the P7 bus, the P7 being the new architecture chip that was being done in Portland. The idea was that the two products would share the same bus so that they could share the same peripherals, and we agreed that the 386 would have that bus for that reason, even though there were risks associated with that decision. So I remember that still feeling unsettled but us going in that direction.

**Slager:** Yes that was a decision that was made and non controversial for a while, but then it became unmade, unstuck; maybe Jan you want to talk about that?

**Prak:** Yes the P7 bus was very far removed from the 286 bus and the 32-bit version of that bus. P7 bus was about as different as you could imagine; It was a packet bus, and it had all kinds of bells and whistles that I can't possibly remember now. It required the entire system to have new chips to go with the processor, so you couldn't talk to memory unless you had a very elaborate memory controller chip that I imagine they were working on also. Because there were multiple large chip developments, each of them -- if they slipped -- would cause the overall availability to slip. Even a prototype was affected because if you wanted a prototype of the 386 with the P7 bus you couldn't talk to memory unless you had this memory controller chip and it was already working.

In addition, we had, of course, a new process, which was a two-level metal CMOS process that was being developed in Livermore at Fab3. This was the first CMOS process that Intel had ever done and they struggled with that for quite a while. During the entire program they had difficulties with defects and also with the two-level metal. So we had all these external issues that we were worried about in addition to implementing the new 32 bit design.

To execute this common bus, Jim and I would travel to Oregon very frequently. I don't remember the Oregon people coming down here but they must have occasionally, and each time we went we would review where they were at and the schedule and so on and for instance, you know, when we were getting well into our RTL development and had something wiggling they had nothing comparable. So every few months the pressure built up of how the risk for the whole program was tied to this common bus decision, and Jill and I had numerous conversations, since I was the project manager before this bus decision was unmade.

**Leukhardt:** Yes Jan would come back from Portland and come and tell me this is getting riskier and riskier and I was SBS Chairman at the time and I would say "Well we made this decision and unless something dramatic has changed we're not going to go backwards" and I would frustrate him and he'd go away and two weeks later, he'd come back. This went on for a while and then finally he convinced me that too much time had gone by and that it was really becoming a risk for the 386.

**Jarrett:** So Gene, as the Program Manager for all of this, were you getting a lot of heat internally to stay with this bus unit from Oregon?

**Hill:** Let me be very clear about the start of the project: I wrapped up the 286 and put it in production. Jan, with his experience in CMOS and he's an excellent manager, Jim and Jan was the project manager for the 386, he was driving it. I was the Director of Engineering and since Intel developed each new processor chip with a new methodology. I had the job of trying to document one clean methodology for chip design for Santa Clara processors. So I was off the project for a small time period. We'd meet with them in one-on-ones to see how things were going, but I was a little bit hands off for a short time period.

There actually came a time period where since most of the staff from the 286 had left because there was no more design work we had Jim, we had Jan from the P4 work, and there was a big staffing recruiting effort that went on. We tried to build back up to have a team and one of the key problems Jan had was trying to find somebody that knew CMOS and was a good circuit design manager. I got reminded that there was this Intel class where they used the movie *Twelve O'Clock High* and one of the key features of the movie is that the hero demotes himself from being in charge of the whole base basically and becomes the captain of the lead plane on the flights over Germany. So he basically gets right back into the trenches to help fix the problems of losing planes and so forth. Jan and I got together and out of one short meeting we basically decided to demote ourselves. Jan jumped in to be the Circuit Design Performance Manager and I got out of documenting methodology and got to get back into project management. So that was the point at which I really got back into it with a 100% of my brain approach, which was great fun.

Another key factor that was going on was that Jim had been at Rockwell with a completely automated system and I'd been at Rockwell helping develop that system. I had gone backwards, every career move I'd made I'd gone backwards in methodology but forwards in terms of the economical viability of the projects we're on. So kind of what came out of the staff meetings was a move to do more automation and more methodical design work on the 386, that's a theme that we built in as we went forward.

**Jarrett:** How big was the team at this point?

**Prak:** It was pretty small, I think it was around 10, maybe 10 engineers plus or minus two or so and we were constantly worrying how to address the different areas where we should be working in parallel and whether we had enough people to work on each of the units.

After the bus decision was made and we went back to a 32-bit version of the 286 bus, we needed somebody to work on that. We were fortunate to get Dave Vannier, who had been working on peripherals, so he knew about buses and timing and so on. He took that over and of course he had a much shorter window because it was behind everything else we were doing. But he was able to pull that out and he was not the last unit to finish, so that was a good accomplishment.

**Jarrett:** How was this bus decision made?

**Leukhardt:** Well Jan convinced me and my recollection is that the two of us went to Ken Fine, who was the General Manager of Microprocessors at the time, and we convinced him, which was not easy, right?

**Prak:** Right.

**Leukhardt:** But once he was convinced, he was convinced, and he went and told his boss,,Dave House. Then they had to go and deal with the others who had to be notified and/or convinced, the most important person of whom was Les Vadasz. He was the godfather of the P7 and had been on sabbatical and was definitely not going to be pleased at this change in direction. To say he was not going to be pleased is a significant understatement. This was definitely-- this was a major change.

**Jarrett:** Now at this point were you still going out to talk to the customers and show them this new architecture? What kind of reactions were you getting from them?

**Crawford:** We did have a lot of customer visits, involved in the process and I think once we'd solidified on the model and were able to show them a flat 32-bit address space, a simple extension to 32 bits, and full compatibility, we got some pretty positive feedback from most of the customers we visited. One exception that Jill reminded me of earlier was that the IBM Austin team was not receptive to the design, but they of course were headed in a different direction. The IBM PC team seemed to like it; we had some good feedback from them and for the most part I think we got pretty good feedback.

**Leukhardt:** I think we knew we were on the right track when we got good feedback from AT&T because they had been in the 68000 camp. To hear them embrace what we were presenting and warm up to the option of a 32-bit flat address space and the performance characteristics and the paging option and so forth made us feel like we were on the right track. But it wasn't all a smooth path in terms of customer presentations because we went out on customer presentations before we had a final definition. I can remember airplane flights when we were going through our foil sets and saying "Well maybe this should be different and maybe that should be different" ,and if it had been in the days of PowerPoint I can imagine that we'd actually have been changing slides, but of course that wasn't possible at the time.

**Jarrett:** Gene, you were interested in bringing in some more automated placement and routing and wiring; can you talk about that a bit?

**Hill:** Okay there were a lot of constraints all a long on the project to try to fit in; trying to get resources on board fast enough was a constraint. We were also within the constraint as we worked on the chip plan, we got in the constraint of how big a chip you could actually flash onto a wafer. We were also in a schedule constraint because we were behind and we had to get the thing out and Jan had done an excellent job of setting the schedule but trying to hold to that was hard and at one point we finally made the decision that we should go with automatic place and route. Neither one of those things existed at Intel and the concern was could we get it done in time and would it blow up the areas of the chip so that they wouldn't fit and then it would all fall apart and we'd have to do it by hand. So what we did, we got an automatic placement program from a grad student at Berkeley, it was called Timberwolf and we checked it out and it seemed to do an adequate job so we had his software. He moved to MIT to work on another project and we actually had a terminal set up in his campus room where he'd fix bugs in the auto placement program as they came up. But luckily the whole thing came together and worked. There are several points in time where we'd get stuck and have to be waiting for him to fix his program. So that would take the individual cells and put them within a rectangle in an optimal situation for speed. We then had to hook things up and that was traditionally done at Intel by mask designers, which was a very laborious process. But the CAD group had a CAD designer that had done a two-level metal router before, so we got him transferred onto the team and in two months he designed a router that would actually function and inter connect cells. Now it wasn't pretty: the end result had to be edited by mask designers to sometimes fix the interfaces. Then, as the forbidden gap problem showed its head, we had to continually re-tweak the metal and the spacing of width, so there was a lot of mask design that went on but the automatic place and route worked out pretty well. There was a lot of readjusting to make things fit but it helped us hold the schedule.

**Prak:** That was at the end: auto place and route was really close to the end because we did a lot of the data path and the large blocks like the microcode ROM and the TLB's and the PLA's that were used in instruction decode, all those had been designed much earlier and the entire data path was all laid out by hand and we also had resource issues in the mass design because we used a CALMA System and we didn't have enough terminals. So getting more mass designers wouldn't help in a way because there was no place for them to sit down and work. I think we farmed some pieces out to people across the street in another building but we were definitely very limited in those areas as well as in building up the team. So for a long time probably, the first three quarters of the project, we definitely felt like the stepchild and to us it seemed that the P7 group was getting enormously more corporate support and assistance.

**Slager:** I was just going point out that if management had known that we were using a tool by some grad student as the key part of the methodology, they would never have let us use it.

**Prak:** Yes, yes.

**Jarrett:** Now this "forbidden gap" term, what was that?

**Prak:** Well it had to do with the two-level metal; this was before the polishing of wafers that is done today and allows using the five, six, seven layers of metal that are common today. That hadn't been invented

yet, or if it was invented, Intel wasn't using it and they had a lot of problems with the second level metal being deposited on top of the very hilly landscape after the first level metal was completed. So the first level metal you could have it close together or far apart but there was area in between that wasn't allowed -- that was the forbidden gap.

**Slager:** Some people thought the forbidden gap was 8:05 to 9AM, or if you showed up, you just sign any name on the sign-in list.

**Jarrett:** This is the time with the late list at Intel.

**Prak:** The forbidden gap was not really critical; it was more of an annoyance. We also had to deal with two kinds of changes in process. One was design rule changes where the further we got along the more expensive it was and I think we had several-- we weren't too bad in that area, I think they gave us a design rule change that close to the end but it was rather easy to implement or later than we expected it. The other was performance. The technology people would deliver the files that determined the circuit performance and we decided how to simulate the circuit. We had a very conservative methodology with extra margin so when two or three times during the project they gave us new files and they were always slower than the last one. So we were definitely worried about critical path evaluation and Jim was really instrumental in developing a method for dealing with the critical paths.

**Slager:** Purely manual method because we didn't have any automatic method like everybody uses today which just goes across the chip and finds all the critical paths automatically. I had to figure out a way to do it manually but which could be enforced across the project. It was primitive but it worked. Another new tool that we used on the 386 was the MAINSAIL programming language. Across Intel there were so many different computers in use that if you did things in one language it might not run on some others. The CAD group had standardized on MAINSAIL as a machine independent sail language and they convinced us that we should use it. But it worked out well and we wrote our RTL language describing the chip in MAINSAIL. Of course it was never designed for that, but it could work and we had like seven or eight units and seven or eight RTL designers, each one had its unit to work on and programmed it in MAINSAIL. We developed a micro-architecture spec which the earlier X86's had not used. Each designer wrote down what was in his unit, how it worked and everybody else could read it. He also described his boundary of all the signals that came in and out and in fact each was represented I guess by a procedure in MAINSAIL. We actually had a script that would go through the MAINSAIL and pull out those boundaries and then they were published with the micro-architecture document, and that's the only way you could keep it accurate because things are changing all the time and your documentation would be out of date very soon if we didn't have some automated way of doing that. But we found that the engineers were automating things by writing their own scripts where in earlier days you might have to go to ask a CAD person to come and do something for you --and that's difficult to do. Much easier if the engineers can do it themselves and I think that all came about because we instituted Unix for the 386 design. Again if management knew what we were doing they wouldn't have let us do it.

It all came about under the table by Ed Hudson one of the circuit design engineers in the early part of the project. He walked across the street from Santa Clara 4 to Amdahl and they had a Unix that ran on 370 computers. So he went over there and got a tape and brought it back, sent it over to Phoenix where the mainframes were and told 'em to load it. They did, not knowing what was on that tape because they never

would have done it if they had known and so we started bringing up 386 on Unix. Most of us had to learn Unix. Pat Gelsinger was a Unix expert; I don't know how he did it but Pat's an amazing guy, as most people know today. He was a very young guy at that point, like 20, and he had an Associate Degree and he had hired into Intel in the QA Department and he became the QA Department's Unix System Manager somehow. He taught the rest of us and that allowed us to bolt together all the different parts of the methodology that weren't designed together. It would fill the gaps -- I think I heard that before here -- "gap filler." But it allowed us to get automated and become much more productive. So we were able to take a design team of people almost none of which had ever worked together before, almost none of which had any experience from the 286 or any 86 earlier. In fact I believe I'm the only one, and later Dave Vannier joined, we were the only two that had had previous experience on the architecture and we had a few people that maybe had two years experience in Phoenix doing something or other and they came over. We also had a lot of new college grads because Intel encourages the hiring of new college grads, and we found that we could get authorized to hire a bunch of those where we couldn't get experienced people because we always had headcount limitations. But the new college grad was like free talent to us. So I'm trying to think, maybe it was the graduating class of 1983 that we got like four or five new college grads; they came right onto the project into key roles and they had on-the-job training.

**Prak:** I had only worked on projects with very small teams like two or three people before, so that part was very new, but I was used to documenting and checking everything over and over because at the company I was at before, we had no tools whatsoever. So everything had to be checked. For all the things that we didn't have tools for, we developed little spreadsheet-type things; they weren't actually spreadsheets but pieces of paper where you would check off various steps along the way of work on cells and blocks and schematics. People would check the other guy's schematics and we had a lot of that type of manual work and that was quite successful. The overall quality of the design was a lot better than the 286 if I may say so.

**Slager:** Yes that's true, I'll agree.

**Prak:** The new people fit into the system and in a way for some of them it was probably good they didn't have experience because they didn't say "Well I'm not used to doing it that way, I have my way." They didn't have their way because they had never done any of this, so there were a lot more people were ready to go along with the systems that we instituted.

**Hill:** Jim and Jan were very key in the project because they were experienced; they knew how to get it designed and the pieces that were required for it and they would set up methodologies. I'd work together with them but what was really key was to be able to bring in such young talent which was more fluent with programming than the old crotchety designers that had always done it a different way. So they had very good raw talent, but their big problem was to bring them in and build them into the team and into the methodology. It worked out excellently and we had to bring in a lot of tools that hadn't been there before. We had to be careful that we didn't overdo it in bringing in tools which could bring down the whole project. So it was a very good balance of experience trying to make the right decisions and young energetic talent that was adaptable to any methodology. One of the things we did to try to cut down on documentation is we redid all the cubicles so we could have the logic designer of a unit sit with the circuit designer of the unit because there was no automated system for tracking timings and so on and so forth. It was much easier to just turn around and ask the question; then Jim came up with the methodology of documenting the timing. But a lot of the first order questions about how something was supposed to work, what the

general times were, were just handled by looking over your shoulder and asking the other person on the unit. As a project, the methodologies came together really very well. There was a lot of strain getting them to work but once they were together, they were iron clad and they worked really well.

**Slager:** One aspect of having a young team is that it was rather easy to get them feeling urgency. In some other projects, you got experienced people, senior people, and they don't want to be pushed around so much. They can get it done next week instead of tomorrow. We really didn't have much of that, you know. Jan talked about checking schematics and things and I remember Steve Douglas had to have his schematics ready by Friday morning and he stayed up all night Thursday getting them ready.

**Crawford:** An all nighter.

**Slager:** Yes, yes.

**Jarrett:** Earlier you had talked about feeling like a stepchild and other teams getting more resources and I remember somebody was saying that one of the required readings for the new team members was Soul of a New Machine.

**Crawford:** Right yes.

**Jarrett:** So did you promote this sense of being this very small ragtag team that was doing all this?

**Slager:** I don't think we had to be promoted.

**Crawford:** It was there.

**Hill:** You didn't have to build it into the culture.

**Crawford:** But there was a lot of parallels between the situation described in that book which took place at Data General with a well-funded new machine being done in North Carolina versus the extension machine being done back at where Data General was first started in Massachusetts. So it was a great book, made a great read, and we could certainly empathize with what those guys went through.

**Jarrett:** You were talking about going and getting the Unix tape and running it; were you getting enough computer resources within the company at that point?

**Slager:** No. Like at the beginning of the 386 project, Intel had one mainframe computer and it was I believe five MIPS at that time and then later it went to like 10 MIPS. Every PC has-- my iPhone has --100 times more capability, and that was used by the whole company. We also had some VAX's for the project, but they were very low performance. It was awful in those days trying to get computer resources.



We were still doing memories in those days, and the memory people had a DRC (Design Rule Check) run which would take like a week to complete and they would turn the mainframe down once a week for maintenance. So they had to time the launch of their DRC very carefully so that it wouldn't get blown away by routine maintenance. But the machines were always overloaded; it was always a problem. I don't know, they were a very expensive machine like 20 million dollars for a 5 MIPS computer.

**Prak:** Yes, and we had the remote connection, and it wasn't satellite and it was rather primitive. They were in Folsom, and whenever there was a thunderstorm in Folsom my simulation would go dead and I'd have to start over again and this was true for everybody. So a fair percentage of work was always lost because of these communication links to the computer, or you'd have to be waiting because the computer was overloaded.

**Slager:** Well if you think that's bad, on the 286 we were using the DEC 10 computers in Oregon and they went down because Mount St. Helens was spewing ash on the computer center. We lost like two days due to that.

**Jarrett:** So as the design phase continued there in 1983 and 1984, the PC industry was blossoming; did this impact you at all?

**Slager:** Oh yes it changed everything because we went from stepchild to king. I can't remember exactly the chronological sequence of it all, but probably over a 12-month period we went from stepchild to king. It was amazing because the money started pouring in from the PC world and just changed everything.

**Leukhardt:** Well, and we got a really dramatic demonstration of the absolute importance of bit-level compatibility with the 186, which was object-code compatible in terms of its instruction set architecture, but had an I/O architecture in terms of its on-board I/O which was not completely compatible with the PC. Therefore PC's that were built with the 186 all failed and so it completely reinforced the definition of the 386.

**Jarrett:** So as the design process was continuing Gene, you were undoubtedly making some trade-off decisions among performance and features and manufacturability; how was that process handled?

**Hill:** Actually that happened fairly early on with the chip plan. In other words, getting early designs of the cells, looking at how the 286 went together and allocating space on the 386. There was actually a small cache that eventually got exited because it didn't have enough performance for the size of the cache that we could put on board the chip, and the problem was if you made the chip bigger, it literally wouldn't fit inside the lithography machine's field of view, to flash on the chip. So we had these hard boundaries that we continued to have to fit into. We had to fit into this circle and so each unit in the chip had to fit in its jigsaw place and the whole thing had to add up to not be bigger than the reticle field. Actually there's a spot on the die picture there we don't have bonding pads because that particular block on the chip pushed too close to the outside of the field of view.

**Slager:** Yes that might have been the last big controversy, the on-chip cache. We had it in from the beginning because the 68020 had a cache, and ours was going to be twice as big. Marketing would have loved that. Then we got in trouble with the chip size and we just didn't know how we were ever gonna solve that. We went through a period of indecision but it really just had to go and then oh, marketing hated that. We'd have no cache and 68020 had a cache. Now the real truth is that both caches on both chips were virtually useless because they were so small and added almost no performance in either case; it was purely an image problem. But it had to go in the end and it did and from that point on I think it was pretty smooth sailing. Well it wasn't smooth, but all we had to do is get the chip out. We didn't have to argue about it, we just had to get it out.

**Crawford:** Yes tossing the cache off was the biggest thing kind of in the middle of the project, and I want to acknowledge Les Kohn's contributions for that activity. He was instrumental in pointing out the performance issues with the cache that we had and how marginal it was. He was able to work with Dave Vannier to make sure the external pin timings, the external bus timing was set up such that our customers could build a very capable off- chip cache. It turned out that that was a much better solution than having this little tiny cache on chip that really wasn't very effective.

**Slager:** Yes Les Kohn was a very helpful person. I don't think he was ever assigned to the 386, but he was just kind of floating around and he always had good ideas.

**Hill:** So once the bus unit was done and once the cache unit decision was done, what was left was small tweaks of the chip plan, you know, something would grow a little, something would shrink a little. We called 'em the range wars and the buck stopped with me on who gave up space, a little bit of space, who gained a little bit of space. But they were all pretty straightforward, nobody wanted to give up space but it was clear who had to have the space. So for a while there we would move boundaries on the chip plan and then things pretty much settled out until the final stages of trying to put in the place and route blocks. But once the architecture decisions were done, it was pretty much go -- we had to get the thing out.

**Jarrett:** How about the interface with manufacturing and technology development, was that a fairly smooth process?

**Prak:** At times it was a lot easier to work with them than with the P7 people, but we had some ups and downs for sure because as I said earlier they were definitely struggling in that period. There was a history at Intel of developing a process and technology development and then transferring it to the production and the people in production would redo everything. They were trying to get out of that way of working things and at the same time doing this new process with CMOS and two-level metal. It was definitely quite a challenge; they went through at least two teams of people working on it. There were people working on it that had left and then they got another group of people. I remember that but we didn't meet with them as often, it was usually every three or four months we would get together and when they had anything new that would change the rules or the performance then we had to have a big decision. But as we got closer, when we got to this period where we felt no longer the stepchild but more the top project in the company, we also were getting more support from the other groups like the technology people.

**Slager:** Yes and product engineering --on the 286 getting product engineering support was kind of hard because they were busy -- 386 they were coming over to us to talk about things ahead of time because

we were no longer the stepchild, we were the king at that point. So we got real good support from product engineering as I remember it and they even took our RTL model and were running it themselves to generate test programs for the tester which was a new experience at Intel for sure.

**Crawford:** Yes that was a great deal, they loaned I think two engineers to me, I was responsible for the validation test programs and we got them to start early, imagine that.

**Slager:** Yes it was great, really great.

**Crawford:** They got a chance to learn about the chip and really contribute a lot to the validation effort by generating a number of test programs that we were then able to use not only on the tester but pre-silicon and chase bugs out of the RTL model; that was very helpful.

**Slager:** Yes they had the ability to reformat the patterns and change channels and everything all by themselves without coming back to us, so that was a good outcome.

**Prak:** One of the difficulties of dividing the chip into units, which was probably more pronounced on the 386 than on previous chips, was that the majority of the engineers knew their unit but very few people understood how the chip as a whole really worked. So when we got to the debug on the first silicon we only had maybe a few people who really could get their overall view of things and figure out where to go and what to do. I remember Pat Gelsinger and Greg Blanck were in particular very helpful because they understood more of what was going on between the various pieces of the chip, and I had worked really hard to understand all this myself including microcode. I learned how the microcode worked which was interesting and...

**Crawford:** I didn't give you any microcode to write though did I?

**Prak:** No, no, but so that definitely helped when we had some serious problems on day one. Our first test program consisted of three instructions: NoOp, NoOp, Halt and it didn't work.

**Jarrett:** So at that point....

**Prak:** Was somewhat discouraging.

**Jarrett:** Was that normal at that point for silicon not to work?

**Crawford:** Yes.

**Slager:** Well it seems like everybody that does a chip will tell you that it worked the first time, but maybe it had some focused ion beam or something and then you noticed that each stepping was said to have

worked the first time, A step, B step, C step and then you wonder well why are you having all these steppings if it worked the first time? So in reality, I think very, very few chips ever worked the first time, but maybe we had an even more unpleasant outcome right at the beginning because we couldn't do no-op

**Hill:** We had this one section, one PLA on the chip that was dead. It was supposed to be dynamic and it didn't work at all and Jan very quickly diagnosed what was wrong with the circuit and someone came up with an excellent way to change a few polygons and make it static so it could work. I think Pat Gelsinger got involved; At first we took the actual glass mask that was used to make the wafers and we took it to a mask vendor and we tried cutting metal with the laser to implement these changes. The problem was the laser cut the metal alright but it hit the bottom of the glass, bounced up, cut another piece of metal, bounced back down, cut another piece of metal. So we had these mouse bites all over the reticule; it was trashed. But we found another vendor that had a scheme called ion milling where they'd actually go in and focus an ion beam on the mask. It was mostly for removing defects in the plate to make it useful but they could actually remove whole polygons, and by cutting a little prisms into the glass they could make opaque layers, and so we implemented this change in this PLA right on the plate. We didn't go back to the design database and we were able to turn the wafers around pretty quickly and those wafers would run no ops, so that was a major step forward.

**Prak:** This was a metal change so it could be the fab had wafers available before metal and so we got parts turned around and there were still quite a few bugs but the chip was much more functional at that point.

**Slager:** There's probably something we should insert: you ask about support from other groups, so when we taped out, we had great support from fab. They had three groups, each hand carrying the wafers through fab, racing against each other to get the wafers back, which was fabulous. Then we knew we were king; you get the fab people have three groups hand carrying it, 24 hours a day.

**Prak:** It took 11 days and when they first started on that process I think it took them four months to get a wafer out and by the time the wafer came out about a third of the process had already changed that they used to make the wafer so their whole feedback loop was kind of trashed by this really slow through put time. That also made the process development really difficult. When they got to our wafers it was very, very fast.

**Jarrett:** Now you had mentioned earlier that this was Intel's first CMOS product.

**Prak:** Correct.

**Jarrett:** And what led Intel to go from NMOS to CMOS?

**Prak:** Well Intel was really late; there was an overall trend towards CMOS and I think people realized that CMOS reduced power consumption because if a particular node wasn't moving it wasn't consuming any power and then the other benefit was it provided cleaner, logic levels. The zeros were really zero and

the one's were one's and NMOS didn't do that. They had the zeros but they didn't have the one, the one wasn't very good.

**Slager:** It was only about .8.

**Prak:** So other parts of the industry, I can't give you a blow by blow, but there was a general trend towards CMOS both in the memory world and in the microprocessor world. There was a perception that higher performance products in the future would be CMOS. Other companies had a CMOS process so they would move one of their microprocessors to their existing CMOS process. Intel didn't have a CMOS process, and they didn't want to copy what other people were doing, so they wanted to invent their own CMOS process. So Intel's CMOS process was different from what other people were doing at the time. But after our project I don't think they ever did another NMOS processor, so it was definitely the beginning of the new world in technology.

**Hill:** The big concern with getting CMOS to work was the two-layer metal forbidden gap problem which destroyed the yields and made the reliability really flakey. With so much terrain the metal had to go over, there were cracks in the metal; there was little whiskers of metal that would touch. So it was a big problem getting the 386 yields up and getting it to pass reliability testing so we could sample it. We did have constraints on how reliable parts had to be and we had to implement special voltage stress tests to try to blow up the bad ones so that they wouldn't get into customers' hands. That's a whole story in itself how the two fabs got their yields together and were able to provide the samples and the early production rung that were needed.

**Prak:** The other concern with CMOS was latch-up. When CMOS started, people were saying "Oh it's scary and you don't want this because it can latch up." Latch-up means that there's a change in the substrate where you get a short essentially between VCC and ground. If the latch up occurs, the areas that are at particular risk are the I/Os, where you could have high voltages injected into the chip and things like clocks and stuff like that. So we had to provide guard rings in those critical areas and we had to use special designs and I/Os, but the technology overall was pretty latch-up immune, and I don't think I've ever seen a latch-up happening on a 386 chip. But it was definitely something that everybody was very worried about; it came up in every executive meeting.

**Jarrett:** So you talked about two fabs running the product; there was Fab 3 in Livermore and another one.

**Hill:** In Israel, yes.

**Jarrett:** Had the methodology come into play at that point of "copy exactly"?

**Hill:** I think that's true.

**Slager:** No I think that came a long time later.

**Crawford:** It came a lot later. I think you mentioned that they redid the process. It seemed like the technology group would develop it and then they....

**Prak:** They did a lot of work. After we taped out and we were debugging the chip and they were trying to do some sort of ramp, we would meet with them; Jean Claude Cornet was then the microprocessor group general manager, and we went to Livermore regularly to talk to them about the yields and the design changes. It was a very close interaction and there were people there who were very focused on the production side of the technology. They told us how happy they were with the design, that it worked over all these different process corners.

**Crawford:** There is another character that we should give credit to: there was a fellow named Vinh Dham who actually transferred over from the Technology Group to help resolve this problem of trying to get the 386 to work in two fabs. He really helped pull the teams together between the two different fabs. He did a tremendous job -- had no idea what a microprocessor was -- but helped solve the problem of the forbidden gaps in bringing the volume up in multiple fabs.

**Jarrett:** The product was introduced in October of 1985, and at that point how healthy was the product? How were the yields?

**Crawford:** Well I think we had just-- It was only a few weeks since we had gotten parts that worked well enough to run some software. But we had some real creative system guys that got something up and running to do a demo. I can't remember exactly what it was, but we did have something running. It was nowhere near in good enough shape to ship, but we made a lot of progress since the No OP, No OP, Halt. We got somewhat beyond that.

**Hill:** One of the advantages of the 32-bit 286-style bus was that there was a 16-bit mode, so you could actually make a little socket that took the 32-bit pin out and mapped it back to a 286 pin out. So with that little socket we could plug the 386 into a 286 PC and have an instant design without having to go through a whole board design of a 386 to be able to demo things.

**Leukhardt:** And that's in fact what they did for the introduction of the product and the demonstration of the product.

**Hill:** But there were a lot of little bugs, luckily not killer ones, that some very creative apps people would trap and go emulate what was supposed to be happening and go back to the application. It was kind of a miracle that the demo software worked as well as it did and as fast it did.

**Crawford:** It was also careful selection of demo software.

**Hill:** Yes.

**Prak:** Yes, the floating point had a lot of bugs because we hadn't simulated the 386 with the floating point coprocessor; just tried to conceptually work on how the interface was supposed to work. And that was a real failure. So when we were working on the 387 later, they created a combined model where the two RTL's were working together. That was the way to do that correctly, but we hadn't done that for the original 386 tape out.

**Crawford:** If I remember right, most of the bugs were in my responsibility, which was the microcode, and a lot of those had to do with paging restart. So you run an instruction and it targets an address that gets a paging exception. The machine is supposed to back up, and take a clean interrupt, allowing you to take care of that exception, and then cleanly restart. Well everything worked up to the cleanly restart part. In all too many cases we hadn't had the right microcode sequence or the right operation at just the right time to take care of that situation. One of the things we did late in the process as we discovered this was build a microcode rule checker to run against the static verification of the code and point out areas where we had left stuff out.

**Slager:** At some point you implemented a flush twice, didn't you? Why don't you tell about that?

**Crawford:** <Laughs> I vaguely remember that.

**Slager:** I think something in the hardware in some operation needed to get cleared better. And microcode can launch a flush, which would flush the pipeline out. But it didn't quite work. So John did two flushes, and then it worked. And his slogan was, "Flush twice, it's a long way to the kitchen."  
<Laughter> Back from the dorm at whatever your school was, right?

**Jarrett:** What kind of reaction were you getting from the field and from the customers as this product became real and as you put it in their hands?

**Slager:** Well, I made a trip to the field, I don't know when but it was towards the end of the 386 and I think before there was silicon. And the field loved me. That was after we had left the stepchild phase and had become king.

**Crawford:** I had a nice little anecdote from that time period. Some time in 1985 I had come to work and there was this big box in my office. It turned out it was a box of Veuve Clicquot champagne sent by the Paris sales office. They were so happy that we had gotten this product out that they sent this case of champagne along with a nice little note: "Thanks to the whole 386 team for pulling this together." And they also gave careful instructions not to share the champagne with any of the marketing people, but only with the engineering team.

**Jarrett:** You were still working on debug for how long?

**Prak:** I stayed with the debug and Jim started work on the 486. I had a number of good people working on the debug. And we had to run all these test programs, and they didn't always test the right thing,

which we found out later in the ALU problem. And then we had the tester, the product engineers had their test program, and they would give us shmoo plots, which is a graph of functionality versus frequency and voltage. And those were valuable to pinpoint the areas where we had circuit issues. And we had some critical path issues. And of course, we had to rework those PLA's that didn't work on the first silicon. So we had plenty of work in those areas, and then we also found a number of logic problems, and then we found a lot of these microcode issues. So the next stepping we did was not totally clean, but I think it was getting a lot closer.

This ALU problem I was referring to was data dependent-- it was a circuit problem in the ALU in the Carry Look Ahead circuit. The Carry Look Ahead is critical because, especially when you go to 32 bits, when you're doing an add, the simple way of doing it you have to go through 32 stages of logic. And that is never fast enough, so the Carry Look Ahead reduces that to about six or eight stages. The Carry Look Ahead circuit needed to be tested for different data patterns, and somehow we had some data patterns that we had left out. And those were the really critical ones. So after we started shipping revenue units, we found this problem. It depended somewhat on the process corner. I think some parts were still functional, but a lot of them had this problem so that they would not work. The part would be 16 megahertz but for this particular problem would only run 12 MHz, something like that. So we had to scramble to deal with that issue. Actually, it was the next people who were scrambling, because at that time I had also moved on.

**Jarrett:** So how important was the 386 to Intel?

**Slager:** Look at everything that followed after.

**Hill:** Well, first of all, being able to say we had a 386 boosted 286 sales. There was sort of a reluctance to sign on to the 286, being a dead end product, knowing there was a big inflection point coming. So I think there was an instant boost in 286 sales even without having 386 parts. So that was a gap-filler in itself. And then the 386 was a great success.

**Slager:** The 386 allowed PC software to be 32-bit, and it allowed the PC to take off. If we had just done faster 286's you wouldn't be using a PC today. You might be using a Macintosh, but you wouldn't be using an IBM PC, because it would have been stunted.

**Jarrett:** Now 1985 and 1986 were pretty tough years for Intel and the PC industry. And Intel dropped out of memories during that period and really was focused in on the microprocessor industry. So did you have this sense that you were carrying the company on your shoulders?

**Hill:** Oh yes. There was the sense that we were carrying the company. Plus we had this product that was as big as you could flash. And we were on a process that people were concerned in the beginning could not yield at all. So we actually got the resources of the RAM memory group and technology group in Oregon. There was a new process shrink technology developed, and we taught the circuit design team up there how to do a 386 layout and circuit design and critical path understanding. And that group, Joe Schutz and various people, just did an excellent job. What they did was compact the 386 to the process shrink. They didn't have to go through all the micro architecture and bus debates: are we going



to have a new bus? Are we going to have a cache? So they had a fixed definition, and they could really focus in on trying to get a compact layout, get the circuit speed. And they did an excellent job. The first shrink of the 386 really allowed the production volume to take off. That was a major economic thing that happened to help the 386 be as pervasive as it became.

**Slager:** But one consequence of going from stepchild to king is that you do realize that it's all on your shoulders. That helped us get the urgency down to the engineers, but we felt it ourselves also. Any more screw ups or problems or definitional wars, anything that got in the way, we just couldn't let it happen.

**Jarrett:** There was also the 386-SX, which was a 16 bit bus version.

**Hill:** Yes, after the 386 became so successful. And I think this was the point we had the shrink, we were asked, I think by Dave House, to come up with a lower performance, much less expensive 386 to try to protect the ASP of the 32-bit bus machine. And so there was a whole effort to define the new chip design. But as I looked at the cost structure of the 386, we had reached a point where the die cost was roughly equivalent to the package cost. So I proposed that we take a 386 shrunk die and put it in a plastic, I think one dollar plastic package, strap to the 16-bit bus, and have a low cost version. We took that to the SBS and looked at all the pros and cons and that was adopted. I think that charter was then given to the old 186 team to productize and bring out and make that happen, which worked really well.

**Slager:** If I remember right, we had the 16/32 bit capability in the bus from the beginning. Is that right?

**Crawford:** Right.

**Slager:** Whereas an 8086 had a 16 bit bus. One of the most astonishing things that happened is that somebody thought well we'd better have an 8088 chip to do an 8-bit bus to save on memory cost. And that's what ultimately, I believe, tipped the IBM decision to Intel, because they wanted an 8-bit bus in their PC. Everybody agree with that?

<Several respondents agree.>

**Hill:** It was that cost-reduced 8088 that brought about the IBM PC.

**Slager:** It was the 8088 that won the PC business for Intel, and the consequences of that are just astronomical. So we thought it could be a pretty good idea to put that in the 386SX but not require a new chip. It was designed in, and you could do it just by a pin, tell it to be 16 or 32.

**Crawford:** In fact, that was a key part of Vannier's bus definition. We needed to be able to use the 8086 peripherals, because back when we were the stepchild, we certainly weren't going to get a whole fleet of peripheral chips. This strategy worked great. You could put the 16-bit peripherals on the low ends of the bus, then have it pull the 16-bit pin when it responded and gave you access to all that stuff.

**Prak:** So just the memory was 32-bit, all the other stuff was 16-bit. Is that the way it worked?

**Leukhardt:** That's right.

**Slager:** It could change dynamically between I/O and memory.

**Leukhardt:** And that was key to the early 386 designs, because they used all the old peripherals.

**Slager:** We were lucky that Dave Vannier was just sort of available at the point we made the bus change. And he could come in, and he knew all that stuff very well, and be able to do the spec and manage. The bus unit started well after everything else, but it was not a limitation.

**Jarrett:** You had talked about the impact that this had on the PC industry. How important was the 386 to the PC industry, and did it change it in any way?

**Hill:** Well, one of the things that really struck me, one of those "ah-ha" points: the 386 started with Compaq and then it began spreading and spreading. At first there were operating systems written with the 386 and then eventually several years' later application codes. They were 16-bit code ported over to run on the 386. But at one point *PC Magazine* awarded the design team "Man of the Year." And they had a big to-do in Los Vegas. And Jan and I went, and I got up and accepted the award. But what really sunk in my brain was to look around the room, because all the awards that year were for the 386 boxes, software, boards, chips, peripheral chips. It was all 386. So here was a whole ballroom full of people, the elite in their various industries that were getting awards just because of the 386. So it really struck me how many jobs and how much businesses had been created from the 386. It became not only the king of Intel, but the king of many industries, not just the PC industry. And that kind of froze in my brain the impact it had.

**Slager:** And to show how things had changed from the 286: after the 286 Gene, and I won an award in some magazine as unsung heroes, because the 286 was unsung. But the 386 was very well sung. And I think it really launched this massive PC infrastructure with thousands of companies around the world, three computers to everybody's house. And it never would have happened if we hadn't got to 32 bits. And it was the 386 that did it.

**Prak:** And people wanted to buy a machine that was future-ready, because if you went to the computer store in 1987 and looked at a Compaq machine, it might be a 386 machine, but there was no software. But the customers didn't want anything. They wanted to be forward-looking. So that was a big pull as well, I think.

**Hill:** That was the concept level--

**Prak:** To look at their customer in the PC world as well as in the imbedded applications.

**Hill:** One of the concepts that took a really long time to get through our brain, especially my brain, was that if you came out with a new architecture, the operating system could immediately be written to that architecture. And that would ship with the box. But the applications people were writing for the installed base. So it's very nice that you have a 386, 32-bit architecture out there, but the people that had existing software were sticking with the old model until there were enough boxes out there to actually invest in the port. There were early adopters that had new applications, but a lot of software people were milking the 16-bit software that would run on the new machine. The compatibility mode ran most of the software for years. I think some people said it took up to ten years to get to the point where people would adapt and write all of the software to the 32-bit mode. So it was the promise of the 32-bit mode that carried a lot of the new designs, and the speed, which was immediately available in new applications like workstations. But it wasn't clear how much that installed base, not the software, but actual machines, would run the new code. The 386 machines were actually adopted very early on for the promise that wasn't fulfilled for a lot of years.

**Slager:** But the 386 ended the architecture wars.

**Hill:** That's right.

**Slager:** And until recently when the architecture went to 64-bits it was a very stable architecture for 20 years. The whole industry could just focus on making everything faster, cheaper, better. You didn't have to reset things by arguing over architecture and then changing the software. So 20 years of stability in which the PC industry just mushroomed into something fantastic.

**Crawford:** At the time, the idea of providing a 32 bit computational engine and a complete 32-bit address space just seemed ridiculous. You know, four billion bytes of memory, even the U.S. government couldn't afford to buy that much memory. But at the time, of course, Moore's Law proceeded and doubling every 18 months to two years. Well pretty soon we started seeing megabytes and tens of hundreds of megabytes on computers. And the four gigabyte thing started to hit. But it took a long time, 20 years out, 20 years of life.

**Slager:** If you look at the microprocessor business from 1971 with the 4004 up until the 386 it's pretty much the history of people underestimating the need for addressability. You had little chips with not very many bits and never had enough addressability. So every few years, oh yes, we ought to increase the addressability. And the 386 just ended that for 20 years.

**Crawford:** One of the things I learned at the University of North Carolina from Fred Brooks who was the architect of IBM's 360 is several times he reminded his architecture students that the reason most computers fail is because of lack of address bits. I was made very aware of that, and we certainly didn't have that issue with the 386.

**Slager:** In all the earlier battles for some reason the addressability was way down the list. And we spent huge amounts of energy arguing about some little instruction or something that some customer wanted. And we just didn't get the attention on the addressability. And now it's easy to see how it should have been done at the 286 definition. If it hadn't been for that Zilog MMU we might have stumbled into the right

approach. But there were really only two priorities. One was 32 bits, which we just totally missed, and the other was a path toward virtual memory. And the right path would have been not to do it with the 286 but to leave it for paging in the 386, which is really what other architectures did. They didn't do a segmentation model at all, they just did paging and it did all the protection that they needed and all the memory management. I don't know how things would have changed if we had stumbled onto the right path, but we sure had it backwards in those days. To look back at it, it might have given Intel an advantage to have the complex segmentation, which was largely unused from the 386 on, because it made it difficult for other people to second source it. It was a problem for us to design, but that same problem was there for everyone else. Eventually AMD mastered it, but I don't know if you ever master it, because it takes a lot of testing and a lot of design capability. And Intel had it by those days and AMD maybe was still struggling with it.

**Prak:** It's still true today. There was a start-up not long ago called Montalvo Systems, and they were going to do an X86 machine, multiple cores, etcetera. And they failed. They had over 100 people, and they couldn't do it. They didn't have the test base. I don't know exactly what they failed on. It's a tough problem.

**Slager:** It helped Intel establish the single source capability, which has mostly been true, although AMD has been a competitor in more recent years. Of course, the whole user base would like to have multiple sources. I think that once Intel got behind it 100 percent, Intel could supply all the parts that the industry needed at a price that allowed a good profit margin so that Intel could just be devoted to it. I bumped into one of the marketing guys years after I had left Intel and talked to him about what's it like at Intel now. "Well, there's only one product, and Andy Grove's the product manager." So total devotion to it. Focus, never had it before, but after the 386 and the success of the PC industry it became very apparent. Another thing that I observe looking back is that, I think Jan alluded to Joe Schutz joining the 386 near the end and then doing the compaction. I think Gene mentioned it too. That's very important, because to come out with a new generation chip, you really don't have time to do everything that you'd like to do to optimize the layout. What happened was that a new chip would come out on the current technology and it would be an expensive die, but the users then would redesign new machines, new generation of PC's and that would take a while. And by the time they ramped into full production, Joe Schutz would have his compaction ready for it for mass production at a small die size and low cost. And it fit so well.

**Crawford:** That was a key strategy that Intel had and still has. Just recently we reemphasized that, calling it our tick tock model. So the idea is that just like clock work, tick tock, tick tock, we come out with a new process every two years. And the first year of the process is a tick and that's where you take the previous design and Joe Schutz or his successor will compact it and really focus on optimizing the layout, the speed, everything, and not have worry about a lot of things that were worked out in the first design. Then the tock comes on the second year of the process. And that's where you change the design for the next generation. That's the tock, and that's done just in time for the next tick team to pick it up and hit the next process window. Fill the Fab with that small cost effective die and on you go.

**Slager:** So you're always developing a new processor on existing technology and a new technology on a compacted processor where all the bugs are wrung out and everything -- a really nice strategy.

**Hill:** It's a really nice strategy. It wasn't always that way. And I think there was a point in time where Intel management had to place both bets. They had to bet on Joe Schutz and the compaction and they also had to break ground on fabs. I think they actually got over-capacity on fabs when the compaction worked well. They actually throttled back on how quickly they were going to bring up the third fab because the compaction relieved the need for that third fab to be able to supply the world. Now I think there's a lot of confidence and capacity can be planned well.

**Slager:** Another issue, which we probably ought to mention, is RISC versus CISC. In the late 286 and early 386 periods, it seemed like the whole world was going towards RISC. And we as designers, at least I thought it would be great. Get rid of all the stuff that nobody really wants or is useless and concentrate on just the important stuff. In fact, in the 286 while it was still in the definitional phase somebody hired Dave Patterson from Berkeley. Some people called Dave Patterson one of the fathers of RISC, although he told me that it was really Seymour Cray and John Cocke of IBM. Anyway, he came in and looked at different architectures and made recommendations. I remember I think I was with Bob Childs and we were telling him all the great things about the 286 and he said, "Well, do people really want that? Don't they just really want large address spaces?" Oh man, <laughter> because he had it 100 percent right, but we didn't listen. And the other thing he did was tell us about RISC and how you could do really fast processors if you just kept the definition as simple as possible and just piled on performance. And that was spreading. And new companies were being founded to do RISC. And we were feeling kind of inferior because we had a CISC machine.

After I left, I think it was maybe the P6, somewhere around there, Intel came up with the perfect solution. To do the chip, the P6 chip, you think of first inside as a RISC chip but take the X86 instructions and translate them in a rather small unit -- ten percent or less of the chip. And internally it's a RISC; it can run as fast as any RISC. I don't know who came up with that idea, maybe it was Gelsinger or maybe it was in--

**Crawford:** Bob Caldwell. Maybe to follow on the story, Gene, Jim, and Jan all got seduced by this allure of RISC as they described in their intros. And I decided to stick on and pound through this CISC thing, the 386 and follow that on. And with the 486 we actually ended up maybe three steps toward getting to where the P6, or the Pentium Pro and those processors came along. With the 486 we were able to do the same trick the RISC guys did, which is to retire an instruction every clock. So we were able to re-pipeline the machine such that we do the same thing. The RISC guys were out promoting, "Hey, we can do an instruction every clock." We said, "Oh, we can do that too even with our complex instruction set." It takes us-- it's a little more complicated, it takes us more logic to decode it. We had to have a microcode ROM to handle mostly the 286 segment protection model and a lot of the complexities were swept into a microcode ROM. But the RISC part of the instruction set never touched the microcode ROM. It was set up to issue and run through the pipeline without interference from the microcode ROM.

**Slager:** That was very important. I think Pat figured that part out in the 486, and that caught Intel back up in performance pretty close to the RISC people. And then later the whole interior of the chip was a big RISC machine.

**Leukhardt:** But in the 386 generation the RISC versus CISC debate begs the whole question of compatibility again. And that was the debate that we had at the time during the definition. And for me,

the key lesson learned in the 386 in terms of the definition is never under estimate the value of your installed base. It was hard to know that value at the time we were doing the definition, because as I said earlier, we didn't really understand the value of that object code of the PC software. We only really started to understand it as you all were doing the design and the PC phenomenon was developing the incredible life of its own as it was going on and on. But thank goodness we had the inputs from the field and our own common sense that led us to come to the conclusion that the code that our customers were developing had such value and the investment that they were making in the 186 and the 286 had to be preserved. And we had to have it as a part of our pitch, that that investment would be preserved in the 386. And preserving that investment was what we were selling. And therefore, that installed base was basically what we preserved in the definition of the 386. Now in some ways that makes you want to scream and say, "Well, if you follow that logic, do you ever do anything new?" Or "Who ever does anything new?" And I would just say, well if you're successful in investing in your installed base, then you get to place bets selectively on other things. And that's the way you can do it. But if you don't stay loyal to your installed base you don't get to do anything. That's my conclusion of the 386 lesson.

**Jarrett:** Any other lessons learned?

**Hill:** I had an interesting observation, because I did go to the RISC camp. Because the design was obviously superior and it was much easier to do. No longer did you have to have 100 people on the design. It was actually more fun. You could enjoy yourself and the chips would run faster. But as I got into the RISC world it became very apparent to me that RISC as an economic entity survived because of the high ASP's on the X86 product lines. The 386 and 486 volumes were so high that you could actually bin out the highest performance CISC chips and they would be performance competitive with anything that RISC could do. But the RISC guys wanted so much lower ASP that it made no business sense for Intel to sell those products. They'd have to be selling the very highest speed products for the very lowest prices. So the CISC performance was actually there to do the workstation business, but it made no sense to sell it. It would have destroyed the business model. So maintaining those ASP's created an umbrella for all the Sun's and the LSIs of the world to develop their RISC chips and come in on a cost-based model and do the workstation thing. It was kind of an interesting observation that the CISC business model was allowing the whole RISC thing to exist. Another observation of the 386 was these guys [Slager and Prak]. This is an incredibly talented team of first level managers that knew how to get it done with a minimal amount of tools and got it done with very young, inexperienced, but very capable engineers. There had been an awful lot of projects in the industry in the past that had had various levels of experience, various levels of design tools. But these guys pulled it off with a very minimal amount of tools. And it wasn't until the end of the project that we actually got a lot of resources. So it was bare bones all the way. There's a lot of skill sitting here.

**Slager:** It is kind of incredible how successful the family has been. All the controversy and pitfalls, but it's almost as if something was guiding it, some invisible hand.

**Hill:** Okay, you may not want to include this in the video tape. But I myself am a devout Christian and I had always prayed for the design projects. I had prayed for the 8051 and it came out very successful. I prayed for the 286, and it came out an ugly duckling.

**Slager:** We needed your prayers.

**Hill:** So I was actually praying, and I was saying, what's up? If there is a God and he influences things, why is this seemingly uncorrelated? And a verse in the Bible came out; it says you have not because you ask not. Well check that one off, I always ask. And it says, and because you ask with the wrong motives. So I said what is the right motive to pray for a design project, and what eventually came to me was that God would be glorified and that jobs would be created. And so throughout the 386 project I prayed for it that way. And I had in mind at the time preserving the jobs on the 386. I was astounded when it came back to me at the awards ceremony where the entire room was filled with the industry, businesses that were created, jobs that were created by the 386 project. So yes, Jim, I think there was guidance going on in the 386 project. So I'll leave it to you guys whether to include that or not.

**Jarrett:** Anything else?

**Crawford:** I actually became a Christian as a result of the 386 project. Shortly after, in 1987, and through the process of working on the project and particularly Pat Gelsinger working on me, we had a lot of interesting discussions. So one of the things I took away from the project was a newfound faith.

**Hill:** That's great.

**Prak:** Well the guy who got the UNIX tape, he was a devoted atheist.

<Laughter>

**Prak:** I just want to be clear here that religion was not the determining factor.

**Hill:** That's true, although Pat Gelsinger had programmed his computer so everyday when it booted up it put up a new Bible Scripture on it. And I think we drove him off the team. Another thing that happened in Las Vegas was that I gave God the glory for the 386 project. And after that I was surprised how many different projects and companies' people got up and said that God had had an influence on their projects that they were getting awards for.

**Jarrett:** Interesting. Anything else that we should cover? I think that's a wrap then. Thanks very much.

<Begin next segment. See video for images>

**Crawford:** What we have here is a poster of the 386 layout. And a special feature of this one, of course, is we had signatures on the mat from a lot of the folks involved in the development of the 386. A fun thing for us in looking at this is actually to see the regions of the chip and remembering what blocks were there in terms of the data path here on the left hand side, the microcode ROM over here. Maybe I'll let Jan describe some of the other features of the chip.

**Prak:** Okay, these are the two large PLA's that caused our initial problem. This is the entry point that provides the address to the microcode where the instruction is going to start executing. And this is a test PLA that is used in the exotic segmentation instructions. Then the instruction flows down to here where it's parceled out to all the different execution engine components, so the data unit where the registers are is over here as well as the adders that create the addresses. And then they flow up through the segmentation unit into the paging unit. And this is the TLB of the paging unit. Page information is stored over there. And the bus unit, as you can see, is wrapped all around a chip except for here where we ran out of space when the data path got too tall.

**Crawford:** That was the time one of the casualties here was bus parity. We had to remove some pins and decided that we had to do without parity on the bus.

**Prak:** In this picture you can that the signals are routed from the bus unit, which is here, all the way to the pads. Of course, in this technology the RC (Resistance x Capacitance) delay was not big enough to cause a big problem, so we didn't have repeaters and things of that type. In this picture one of the metals is blue and the other one is red. So that's where you see the red and blue. And then when we were finished, the master designers added the teams' initials which you can see over here and here.

**Crawford:** All over the place.

**Prak:** All over the place.

**Crawford:** I guess that's it, exhibit A.

END OF INTERVIEW