Intel 8051 Microprocessor Oral History Panel

Participants:
George Adams;
Henry Blume, Jr.;
Shep Hume;
Ronald Towle;
John Wharton;
Bob Wickersheim;
Bill Wong

Moderators:
John Ekiss
David Laws

Recorded: September 16, 2008
Mountain View, California

CHM Reference number: X5007.2009

David Laws: Good morning. My name is David Laws. Today is Tuesday, September 16th, 2008 and we are at the Computer History Museum in Mountain View, California together with a panel of engineers who contributed to the development and marketing of the Intel 8051 micro controller family. Introduced in 1980, the 8051 product line grew out of Intel's experience with the 8748 product that was covered in an oral history recorded at the museum in July of this year. The 8051 went on to be one of the company's most enduring and influential devices. It was phased out of production by Intel only recently but continues to be manufactured by numerous alternate sources throughout the world. Our panelists estimate that the cumulative number of units shipped ranges in the billions. We will be recording this session in two groups. I will cover the period from conception to tape out of the first mask set in the Santa Clara, California facility. John Ekiss, who was general manager of the Intel microcontroller operation after the business unit transferred to Arizona, will moderate the discussion of subsequent developments. I'd like to start this first group by asking each of the participants to introduce themselves with a brief background on their early years, education, and work experience before joining the project at Intel. Henry? I'd appreciate it if you could get us rolling this morning.

Henry Blume, Jr.: Okay. My name's Henry Blume. I was known as Hank at the time. Degrees from Yale, MIT and a masters in electrical engineering from Stanford, with honors co-op, courses in the sixties. I worked for Fairchild for nine years mostly in design but I went through process development, applications, product engineering before I got to the design area. Worked for National briefly and then for Intersil for four and a half years before I joined Intel in 1974. I joined Intel when Faggin and Ungerman were leaving, though it hadn't been announced. Intel really didn't tell me what I was going to be doing but it turned out I was going to be working on low-end controllers.

Laws: Thank you. George.

Adams: Hi, I'm George Adams, and I joined Intel in 1975. Prior to that, I got my bachelor of electrical engineering degree at the University of Miami, subsequently my MBA at Boston University. And in my earlier years my first engineering job out of college was as a CPU design engineer on a mainframe computer. Unfortunately, it was the RCA Computer Systems mainframe, and they got out of the business two years later. At that point it was fortunate that I joined Motorola as a product manager for the ECL families and then became Motorola's first field applications engineer for the 6800 microprocessor. That was a period from 1971 to '75 at which time I joined Intel as product manager for the single-board computer product families that were being developed to be launched during that time. I then became product line manager and strategic business segment manager for what was called the "MCS48 single-chip products" in early 1977. And during that time we developed the marketplace for the 8048 and the 8748, while conceiving and bringing to market the 8051 family in 1980.

Laws: Thank you. John?

John Wharton: My name is John Wharton. I had attended Yale University as well for a couple of years studying physics and astronomy. Got more interested in computers, transferred to Northwestern University to get a bachelors degree in electrical engineering. Stayed on to get a masters degree there in computer science. This was in the mid, late seventies, just as microprocessors were starting to become existent. So we had fun trying to attend some of the very first classes taught on microprocessors that were still sticking into the curriculum slowly. When I graduated with a masters, January of '77, my first job was to work at Intel. I was in one of their new college grad programs. Spent five years at Intel as an applications engineer, strategic design engineer, and in various marketing support positions. And left after five years to become self-employed which I have been until now. During that time I've done a certain amount of technical writing, some chip design work, some system reverse engineering. I was a part-time lecturer at Stanford for 15 years-- 14 years.

Laws: Thank you. Bob?

Bob Wickersheim: I'm Bob Wickersheim. I graduated from the University of Illinois with highest degree honors in electrical engineering. Prior to Intel I had joined several smaller semiconductor companies, always doing logic design. At that time one engineer could design one chip, unlike today where teams have to do it. And during that time I probably was the first one to design a single-chip printing calculator at Bowmar just before they went out of business. I joined Intel in 1977. I was hired by Gene Hill who we'll talk about later. And I was the lead design engineer on the 8051.

Laws: Thank you, Bob. Bill?

Bill Wong: My name is Bill Wong. I graduated from UC Berkeley 1970, double-E computer science. I joined Intel 1975. Before I joined Intel I was the engineer at CDC Computer Corporation, designing the hybrid real-time emulation systems, interfacing with the mainframe Cyber 6400 and also with the PDP 11's and analog computers. In 1975 when I joined Intel, I was system engineer designing the four-way interleaf, four-point memory system to replace core stacks. At that time Intel was the frontrunner on all the memory technologies so we used the 16K memory, designed a semi-conductor memory system, to replace all the core stacks for Univac and also for PDP 11/70. After three years as engineer I was recruited to join the microprocessor development systems as the product manager for all the development tools. And I had the privilege to work with all the 8051 teams to plan the development tools for the microcontroller and subsequently plan all the tools for the 86, 186, 286, and 386 families. So I spent 18 years with Intel. I left Intel 1993. At that time I was involved with the development of Intel business worldwide including China, India, Brazil.

Laws: Thank you, Bill. Henry, you were involved with the previous product line, the 8748 and the other members which eventually led in to the demand for a new product family which became the 8051. Could you give us a little background on the 8048 and how that led to the need for the new product family?

Blume, Jr.: Well, in the time of the 8048, we went through a formal approval process that took some time. We had to shoot down the idea of embellishing the MCS40 line because we didn't see what we wanted to do but we were told to look at how we could embellish it more. And we got the approval for the 8748 to make use of the EPROM technology. In 1977, the group had more resources. By the middle of '77 we were starting to get recognized market success. There was a market and parts were being bought and we had pretty much cleaned up the design. We'd gotten a RAM design out. The 8748 ran fairly smoothly. And we also ran into memory addressability limitations. The 8748 had basically a 2K addressability for the program memory with a bank switch to 4K but that was it. So we knew we had to do something else. It's my perception that we kind of knew we were going to do something else, a successor product by osmosis. There may have been formal approvals but there was kind of agreement they were going to go ahead with the successor product. The issues were what was it going to be and I guess at that time we decided we had to help the systems group to do the in-circuit emulation and come out with a bond-out chip as the first thing we did. But the issues at the end of ‘77, start of ‘78, were “What was the architecture going to be and what was the instruction set?” But the approval, I think, went very, very smoothly.

Laws: And George, you were running the strategic business unit at this time that would involve deciding future projects. How did you get involved with the successor to the 8048 family?

Adams: Well, in early ‘77 I came over to be product line manager and also the strategic business segment manager for the single-chip family of products. What that meant was chairing a planning committee for looking at how we were doing operationally in our existing business. But also where were we taking the business from a new product and architectural standpoint. And in that timeframe, if you think back in that ’75 - ‘77 timeframe, that was the era where Intel had [gone] from the 8080, moving into the 8085 and then moving forward a little later into the 8086, this was all geared around a whole range of data processing capabilities, but all with a processor on one chip and everything else on a plethora of other chips on any system board or motherboard. Our focus was on bringing all the functions of the computer, the CPU, memory, ROM, RAM, IO, timers, counters, other functions, on to a single chip so that you could then embed that into any number of other applications. And within Intel, we were really wrestling with how far would you take the processing power of this kind of class of product. Obviously, it didn’t make sense for Intel to have competing architectures for the same applications. The 8080 and 80X later became the 86 or x86 family of products. They could cover a broad range of data processing applications, but were not as well suited for control applications. So if you build in things like a Boolean processor, like a limited range of instructions that can address registers and different IO configurations, all the things that gear a computer for controlling a man-machine interface or machinery of some type, printers, telecommunications equipment, home appliances, et cetera. So during that period, we evolved our positioning within the company, though terms like positioning weren’t used at the time. But there was the strategy for the whole data processing series of the 80X family. We saw a whole range of other applications that we could attack in the marketplace with a different architecture. And we were beginning to get some very good success with the 8748, 8048 families. We were growing that very rapidly from some $7 million in revenue in 1977 up to $70 million in 1980. This was both demonstrating that there was a much bigger market out there for a lot of other applications Intel could do. And that it needed an architecture that could evolve out of the success we were already demonstrating with the early 48 family of products. And so this led to some very interesting discussions about revolutionary versus evolutionary directions for the new product. Should we go with a 16-bit processor? Would that compete too much with the 8086 that was in the wings? Or should we continue with an 8-bit processor and really tailor other functionality in the chip? We could get into this later in the discussion but it all led to choosing the 8051 approach while evolving off of our 8748 base in order to define an elegant architecture for control that would complement the architecture Intel had for data processing.

Laws: John, I understand a free lunch at Intel led to one of the important decisions that was made here. And you were involved in the final decision-making process of the architecture. Could you tell us a little bit about that story.

Wharton: There is no such a thing as a free lunch. I had been working at Intel for about a year, [as an] applications engineer. I was responsible for designing and demonstrating systems, first with the 8085 product but mostly with the 8048. I’d used the 4040 for my masters thesis at Northwestern. Had an exposure to a broad line of the Intel processors. Towards the last half of '77 my job was as a strategic design engineer, which was Intel's marketing effort to get their foot in the door in a number of companies and industries they thought would be key. And the way they were doing that was by finding about some major design that was in the works, [such as] a new microwave oven or an electronic engine control for Ford automobiles. And my group would quickly prototype a demonstration of that application using Intel processors instead of discrete circuitry or competing processors. The important thing there was for us to try to do designs within a month or so and demonstrate something and then be done with it. But that gave me a lot of exposure to a lot of different clients with a lot of different real-world applications. The incident you're referring to was in mid-December of '77. I came to work-- I was working for a fellow named Lionel Smith who was [the] best manager I've ever worked with. A very interesting person, good engineer, but also interesting people skills that I certainly lacked. I left my wallet at home, I think it was one Wednesday morning. Lionel and I would occasionally take each other to lunch and take turns picking up the check. So I went to Lionel and said, "I left my wallet at home; I'm broke. Can you take me to lunch today?" And he said, "Well, I can't today because I have to go to a, what they called a lunch bunch. "I have to go a lunch-time planning meeting." But he said, "They always have sandwiches there and there's always food left over so why don't you just come along and you can hide in the back and nibble on whatever is left?" So I sat in on this meeting which turned out to be George's product planning committee for the Low End. This was mid-December, towards the end of the year. It turned out there was an end of the year deadline to make some decisions about a number of different single-chip products. I may have not gotten the details quite right because I was more interested in the food and whether the potato salad would run out before it reached my end of the table and so forth. But they were talking about various offshoots of the 8048. Lower cost versions of the '48, lower-power versions, ways of enhancing the 8048 architecture, 16-bit machine that may be in the works, that sort of thing. For the 8048, what they had determined was that the logical growth seemed to be to expand memory on chip and also to expand some peripherals on chip. The original 48 had a 1K on-chip program memory that was followed about a year and a half later by an 8049 with 2 kilobytes of on-chip program memory and 128 bytes of RAM versus 64 bytes of RAM. The logical next step was to turn the crank one more time, double the memory another time, go to 4K of RAM, go to 256 bytes of RAM which would have totally filled the address space and would have been the end of the line for this product. But there was customer demand for some additional peripherals as well. There was need for additional timers, some sort of a serial port. A lot of the discussion was focusing on what sorts of peripherals to include and what sorts of trade-offs to make. Because the 8048 had initially been designed to solve the immediate problem just to get everything on to one chip. It was a remarkable product in that it was able to work at all. But there was sort of a mindset in that era that you'd figure out what the hardware facilities would be and then almost as an afterthought come up with an instruction set that gives you sufficient access to all of the things that the chip designers provided. So for expanding the 8049 to the next chip, which would logically have been called the 8050, the plan was to sort of retrofit the '48 instruction set and add bank-switching instructions in order to increase the address space. Add an IO switching instruction to get you to a second timer instead of a first timer. Various things that could be done in order to fill out the product but that would seem to be the end of the line, which I was sort of intrigued by. I was the outsider. I wasn't really trying to be part of that. The next afternoon was my weekly one-on-one meeting with Lionel and he said, "So what did you think of the lunch meeting yesterday?" And I hadn't really been thinking too much but I said that what they were discussing wasn't what I would have done if I was trying to design a successor. And he said, "Why not?" And I said, "Well, because in the design work I've done, and in talking with customers and so forth with the 8048, the problems that I run into aren't being addressed by this upgrade. If all you're trying to do is squeeze more features into a package that's already kind of tight, you have to do that by deleting something that's already there or by making the product a little more difficult to use." By this time there were about eight or nine different offshoots of the 8048, no two of which had exactly the same instruction set. Because as they added a serial port or as they added a parallel port with analog capability, new instructions had to be tacked on to access that. As they changed the bus from master to a slave, new instructions had to be tacked on to handle that interface. To do that they had to remove instructions they didn't think were being used. And all that seemed to be just snowballing and would get worse if we had further products beyond this 8050. Lionel told me to go off and come up with a proposal for a microprocessor that would feel like an 8048 so that we wouldn't lose our customers but which would attack the problems that I had been running into. And to come up with a proposal for something that would perhaps be different and more flexible and last further into the future. Friday, basically, I tried to summarize the issues that needed to be resolved over the weekend. Wrote up a proposal for a combination of instructions and addressing modes that would take care of that. Brought that in Monday morning to discuss with Lionel and unbeknownst to me he sort of went behind the scenes then and started riling up the program committee, suggesting that they should at least consider an alternative - that the path they were following as of the meeting the previous week might lead in a different direction. And where it led was eventually to what later became named the 8051.

Laws: So what was the process of comparing the original proposals with John's new proposal? Was there a lot of controversy about making this decision?

Adams: Oh, there were clearly a lot of different points of view. Intel has a culture of constructive confrontation, and people were doing that quite actively to each other. I found throughout the rest of my career that that was an extremely valuable element of the culture. But [with] all these different architectures and about six or seven different alternatives on the table and now with John's on the table as well. We really looked back to a process we started earlier in the year of a task force that was looking at, what are the applications where people are going to need control-type processors rather than data processing processors. It was an intense look at the telecommunications industry, the automotive industry, the consumer appliance industry, the computer peripheral industry. Out of this came a range of applications you want to get into, what are the features you would need or the functions you would need on a single chip or a couple of chips, to really do that more cost effectively. And so what we did was to say, "Look here are the ones that we see as the biggest opportunities. Here are the growth rates. How do each of these alternatives really map into that?" And it didn't take long to see that the evolutionary element of what John was proposing as well as some revolutionary capabilities and adding in the number of registers, how you address those, how those are needed for things like whether it's anti-skid braking or engine control, or counters on line printers at the time, that type of thing, mapped extremely well. Also, it would get us out of this quagmire and dilemma of forever permuting the instruction set for all the different flavors. So that led to proposing that [architecture] and driving that through the next couple of sessions with management reviews to get buy-in to go ahead with that design as the next generation for microcontrollers.

Laws: Henry, were you involved in this process at all based on your extensive experience you'd had in designing [the previous products]?

Blume, Jr.: Yeah, no, I was involved. I was one level removed from the single chip. I'd hired Gene Hill and he took over single-chip design when I was promoted in early '77. I promoted Gene and Gene was the single-chip design engineering manager. And I think we got John's proposal, I would guess as early as Tuesday, and it was a thing of beauty. It was really an aesthetic thing to look at compared with some of the stuff we had left over. As I remember, the 8048 even had BCD arithmetic because it came from the 4040 calculator line. We didn't have enough nerve to get rid of it. But I think [what] John did created some space. But Gene had an engineer named Mike Malik who was a very bright engineer and he had a proposal that [included] most of the things John was doing. It was more 8048 like. And I spent one week having a seven o'clock in the morning meeting with Gene, with Mike, with another engineer, to look at the two engineering proposals and look at John's proposal. And I think by Friday I came to the conclusion that while John's was probably more expensive [according] to our ability to calculate in advance, it wasn't worth fighting about. It just wasn't worth having an argument about which architecture we use. We bought into John's architecture. He and Gene Hill may have fine tuned it over the next few weeks, not changing anything that John had but realigning stuff a little bit. And we went with John's architecture and, you know, it's lasted for 30 years. I mean, it …

Adams: Speaks for itself.

Laws: Were there any major technological decisions to be taken here as you had in bringing EPROM onto the earlier family? Or was it just an extension of technologies you had?

Blume, Jr.: I think that while there was a new EPROM denser technology coming, there was a denser logic process used on 8086 or the successor. And with a bond-out chip being first, we did not have the issues of an EPROM first. We had a bond-out chip, right. We had circuitry and logic right. Then we laid it out for the ROM version. We laid it out for the EPROM version. I think Bob Wickersheim could address just how simultaneous those projects were.

Wickersheim: Yeah, what we did, we determined the layout for the '51. We just left large holes in the layout where the ROM array or the EPROM array would go. It was a great advantage from a design standpoint in that we could focus totally on the logic functionality of the chip and not have to worry about the interfacing, both with the EPROM design groups. And interfacing the circuitry on the chip to EPROM. It was just less work to do it that way. So it was two advantages. One, the bond-out chip was useful right away and secondly, we could cut down the total design time by not having to worry about that circuitry.

Laws: Were you involved in the decision process on the architecture at all, Bob? And having inputs on how best to do it? How to implement the design?

Wickersheim: Not a lot of the major architecture. Gene Hill hired me in May 1977 and while he was focused on the 51 he assigned me the task of managing the yield improvements on the 8048, 8049 family. That was good for me because it gave me a chance to learn Intel's design methodology. Prior to coming to Intel [at] the small companies that I'd worked for I would have had to develop my own transistor models using fabrication data, before I could even do circuit simulation. I was extremely delighted when I got to Intel to find there was a whole department tasked with the job of creating the transistor models, setting up the simulation environments for us. So we could focus, actually, on the circuit simulation. Early on, I think around mid-1978, Gene assigned me the task of leading the team to do the 8051 design. And I recall in the early stages of our design, John Wharton coming to us with minor functionality tweaks because the architecture, the logic was still being finalized. And as long as it didn't impact the design work we'd already done, we could squeeze it in. One of the things, I think, that impacted the design, I would highly recommend not moving a product in the middle of its development cycle. When Intel announced that they were moving the microcontroller group to Chandler, and had set a move date of around July 1979, and so I sold my house in July. The other members on the design team that were electing to go made arrangements to move. And by July 1979, I was the only one left on the design side. All of the mask designers were still there. And I spent from July to the end of 1979 finishing up the work to do tape out. There was a decision made to actually complete the tape out, that is releasing the data to manufacturing, before the project totally moved.

Laws: John, could we finish off a little bit about the process of defining it? And then you created a specification of some kind? And this was published early '78? Is that correct?

Wharton: January of '78, yes. I should maybe go into a little more detail. Henry's referring to Wharton's architecture. The main thing that I did was introduce the idea of separating the processor capabilities from the IO capabilities. Until that point every processor Intel had introduced had made changes to the earlier processors' instruction sets.-- and this was the right thing to do. I'm not trying to criticize that. Figure out what the hardware is you can cram onto a chip because that's the engineering challenge. Chips had very limited transistor budgets and you had to make tradeoffs on what to include and then invent an instruction set that would give you access to those features. So Intel had the 4004 4040, 8008, 8080, 8085, 8048, 49, 41, 42, 21, 22 and every one of those processors had a slightly different instruction set in some way. Because they had different types of registers, different types of IO capabilities the instruction sets were tweaked for each. The Intel philosophy had been until then that, I think, on each of these processors the first eight bits of any instruction, called the opcode, told you everything that was going to happen, what the instruction was going to be, what the addressing mode would be, how to figure out the variables that you were going to use. And then a second byte, if there was one, would be just the data you would need for an operation or maybe the address you were branching to if you were jumping to a different part of the program. But the result of that was that in the 8048, for instance, fully one third of the possible op codes, there are 256 possible op codes in one of these machines, fully a third of them were devoted to these IO devices. Start the timer, stop the timer, start the counter, stop the counter, enable interrupt, disable interrupt, read the interrupt pin, read a test pin, read another test pin, set the carry, set a test bit, clear a test bit, test a test bit, et cetera. And that was just using up a whole lot of precious op code space. The key proposal in that first specification was that it's all this IO stuff that's causing confusion. That's where we're trying to debate what to include. Can we afford to include another timer and so forth? And that's just going to get worse because as time goes on we're going to be putting more and more stuff on. So let's define an architecture that does all of the arithmetic that you need, reading variables, incrementing variables, adding variables, saving variables, all that sort of stuff. But then have a whole big block in which IO registers could be placed. And we don't have to know what those are now because we're going to use generic instructions for every interaction. We'll just read a certain something or write a certain something or add a certain something or set a certain bit or clear a certain bit. And that provides a clean interface and then the IO designers can figure out what that bit does in terms of how it's connected to the outside world. Or how it's connected to internal timer logic or counter logic or A2D converter logic. And what that allowed, then, was a better product specification. It meant that you could define and actually start designing the whole CPU without even knowing yet what the registers were going to be. You didn't have to decide if it was going to be an 8-bit UR or a 9-bit UR or whatever. Because as time went on, you could fairly easily appropriate some of the unused bits or some of the unused registers in order to augment the functions that you were starting out with. I think the first spec that was put together in a three-day weekend in December was basically defining this schism and proposing a way to design an instruction set that would have instructions of certain formats and certain capabilities and keep each of the capabilities of the 8048. Each of the 8048 instructions was retained. Each of the addressing modes was retained. Each of the registers was retained. But it was supported in a different manner so that each of the special purpose instructions that we were starting from became one instance of a generic instruction in what we were going towards. And what that meant was that the products were not compatible in terms of being able to use the same ROM codes but they were compatible in terms of being able to take the same assembly language programs. Because each 8048 instruction would map directly on to an 8051 instruction performing the same function with the same side effects. And that would ease the transition for the existing customers that were too precious to abandon at that point. The techniques I was using to get to that capability were, in some ways, rather crude, rather wasteful. I was limited to thinking about the problem, perhaps, too restrictedly. And it led to some instructions that were inefficient. During that month of, well, Christmas, New Year's, first half of January, I worked with Gene Hill quite a lot and what he was doing was evaluating the capabilities but coming up with more clever ways to implement them. Gene had some ideas about what was impractical because of the amount of transistors it would use versus what would be easy to throw in. Because there was logic already available that could be made to do double-duty that I had no insight into. Gene came up with a proposal for taking the second byte of instructions and using a few bits of that byte, in order to increase our flexibility. And that would save us from having to go to a third byte which meant things would run a lot faster and use less code. So Gene's input was to refine the implementation of the basic instruction-set programming model. And then it was in mid-January of '78 that things were sort of coming to a head. There had to be a decision which path to follow. As I recall that had been the goal of the previous quarter but we had slipped that a couple of weeks. We didn't want to slip that too many weeks. And so in January at the next meeting of the planning committee there was going to be a decision made. And so there was going to be a debate about whether to follow the simple path or an intermediate path or the new path. And that was the January 17th specification that has still hung around.

Laws: Did you have some particular customers in mind that you wanted to make sure that this part would service or was it more general applications?

Adams: Well, we had a range of applications that we were targeting, you know, with the new product. And we had several customers within each of those that we had been able to talk to out of their experience from the 8048 and 8748. But one thing I'd like to come back to, if we could, is this was a decision on the architecture and a go-ahead decision shortly thereafter for developing the whole single-chip microcomputer itself or microcontroller as we renamed it before launch. But you have a dilemma when you have everything on the same chip, the CPU, ROM, RAM, IO and so forth, when it comes to how do you do development systems for such a device. And Hank and John had mentioned earlier here, and Bob, the bond-out chip as part of the strategy. It became clear that what we needed to do was to provide a mechanism that we could have a very efficient and yet elegant development system capability if we were going to get the design wins to get a lot of people using this new chip and new architecture quickly. And when everything's on the same chip, you just can't get to the interactions between the CPU and the memory and the IO unless you have very, very tiny little fingers, to put a probe on there. The bond-out chip allowed you to bring out signals that were needed for developing an in-circuit emulator that only Intel could provide because any third-party supplier would have a hard time as they wouldn't have access to the internal workings of the chip in the way Intel could. This also gave a real-time emulation capability and a better way to develop applications very quickly in control environments that are often very tightly time constrained, much more so than in a lot of data processing applications. But this led to also being able to get to market more quickly with what we called an 8031 chip, which was a version that would work with external EPROM for ICE, and then bread board with EPROM. And then going into the single chip with the 8751 [EPROM version] and eventually leading into an 8051 [masked- ROM version] when you'd have enough stability in your code to freeze it into a masked-ROM set. So this bond-out chip became a very key part of the whole strategy. Bill Wong might discuss more of the development systems planning.

Laws: I'd just like to ask Henry a question. Henry, was this something that came out of the process you went through in the 8748/8048, the recognition of the value of the bond-out chip?

Blume, Jr.: Well, I'll let Bill comment on that. But I think what I would like to add is we always had arguments as to what percent of ROM's, what percent EPROM's. In other words what percent of our shipments would be programmable. And this number got higher and higher because the customer didn't get his code right the first time, he wanted to have some ability to make changes. And, therefore, we liked the EPROM. But we didn't really see that the ICE was central to the initial 8748. We thought it was only a thousand bytes of Harvard architecture. You could get your code right by writing it down. We didn't realize how important the ICE was. And we thought that you would do development on EPROM's. It turns out you do developments in ICE. You prototype in EPROM's. Once the user has his code right then he wants to make EPROM's for his next 10 or 50 units, to get them into the field, to see what adjustments he has to make, reprogram the EPROM's and make some more adjustments. But we [later] appreciated just how important the ICE was and just how much of a competitive advantage it gave us.

Laws: So, Bill, at what point did you get involved with the project in order to [allow customers to] start taking advantage of this technology.

Wong: When I first joined the microprocessor development system group, I was in charge of fixing a lot of problems with the ICE 48. What happened is at that time we find out a lot of customers had trouble getting the product into production. They ran into emulation problems. When they tried to get the code to production, when they started to use the ICE modules, it's not running in real time. And then we asked why? Why it's not running in real time. It turns out when they design a single-chip architecture, a lot of address bus and IO multiplex. And you have to recreate a port to emulate the actual product that's working. By doing so, when you're in a real-time environment, say, like you have a microcontroller controlling a motor, you can't just simply recreate the port and try to stop it and figure out what's going on. Because all the instructions are already gone, I mean, [because its] real time. And you can't wait for reproducing the IO ports. So there's no mechanism. Like previous ICE module you can stop putting break point into address and debug the code. So that create a lot of problems in the field on the ICE 48 although it's a 1K code, ROM size, but customer has problem going to production. So then at that time, thanks to some engineer, working with the silicon groups, Homer Gee was one of the key engineers, they starting defining from the development system standpoint, when the user start to develop the product, they can do it right away with the ICE module and the development system. When they start to run the code, you can plug into the socket and start doing the IO control and you can debug it in real time. So what you need to do is you have to bring out the address bus and the data bus separately. That means you have to create a bond-out chip which is bigger than the normal microcontroller. And then you also have to inject the breakpoints circuitry into it so you can stop the counter and you can look at the address and then start to do step by step, single step to do the debugging. So that is how that product was created and it was the first in the industry to come out with the bond-out first before you come out with the microcontroller. And also it set the standard. From there on, most of the single-chip manufacturiers came out with a bond-out [option]. You got to figure out how the user can do the real-time debugging because you're talking about embedded application. Things going in real time. If you stop, if you don't catch the right problems then you cannot debug the problems. So you help the customer get the code into EPROM and get in production, then you can easily get into ROM code as ROM code incurs minimum charge in large quantity. It's a big investment with customers. So the step is to do the development system on ICE modules, get the code correct, burn on EPROM in real time. You try it out in pilot run production quantity before you commit to, hundreds of thousands units per month type of volumes. So that concept shortened the design cycle from user standpoint and then start to spread out in terms of more users. And the architecture became very easy to use with the tools. And that became a very important [part of the] process of getting customers to adopt new architecture or new products in that area. I think right after that we put together a lot of tools, for example we come out with a, what you call a, portable device. It's like an instrument. Where before the development systems involved disk drives and printers and all that. Now we come out with a, like, a IBM portable [PC] device where the ICE module plug into it, then you can take it to the field to do real-time debugging. That was another tool that came out to support these products. So I think the ICE 51 is probably the most successful ICE modules that were built and then from there on with the 186, 286, the ICE became a standard tool. And in the development of the microprocessor, we need to have that kind of a bond-out circuitry involved.

Laws: When did you start work on the ICE 51? Early '78 would it be?

Wong: The ICE 51-- right at the same time as the 51 instruction set was defined and also the spec.

Laws: So it'd be early '78.

Wong: Yes.

Laws: And how many people were involved, yourself and--

Wong: I was in charge as the product manager and Homer Gee was one of the ICE module design engineers. And then, of course, involved with the software, because we develop what you call PLM 51 which is language that doesn't require operating system to run the code. And then we'd come out with assembly, reverse assembly, and then also convert it. From '48 code convert it to' 51. So all the codes developed for the '48, you can easily convert to 51 and then you can, help the customer migrate it to new products easily. So that was one of the ideas - to migrate customers to faster, better microcontrollers. That was the concept. So it was a total development support concept. It's not just come out with the fast architecture, best chip. But we wanted to make sure the customer can get to the product first because that's the goal. How do we get the customer to mass production with a masked ROM product.

Laws: And it was a revolutionary approach to the market.

Wong: Exactly, it was very interesting because right after that, we started Project Crush. We were part of the whole team that go out and promote the whole architecture for Intel. So it was part of the whole-- we think not just the chip architecture, but how customer use it, how easy they can use it, how they can migrate from one to other easily. And that's the concept. The total development support solution.

Laws: And when did you have the first field version of this ICE available for the FAE's to start demonstrating?

Wong: I believe it was, like, close to, what, four, five months after the bond-out chips? I don't know exactly the time.

Wickersheim: Well, the first one didn't work so [it must have been] the second stepping

Wong: The second stepping and then gone through that.

Laws: Okay, so it'd would have been '79.

Wickersheim: Well, it was probably early '80--

Adams: Yes, it was early 1980.

Blume, Jr.: Bill, you bring up something. Was there any patent disclosures or any patents filed on this?

Wong: That's the thing that I don't know. I was still trying to figure out was that ever put together?

Blume, Jr.: I think we had a couple of patents. Dave Buddy, Dave Stamm and I had a couple of patents. But the idea [of the company] was that you [should have] a patent with every chip. By this time the industry had cross-licensed and people just didn't bother with patents and yet I think this [bond-out concept] is highly patentable. It's was not obvious to everyone from [the current] state of the art. It was a major move.

Wong: If you look around nowadays, at all the microcontrollers, without that concept I don't think it would be very easy [to be as] successful from the time to market standpoint.

Laws: So Bob some time in mid-1978, you're presented with an architecture that you had to implement. That you had to do in three flavors, or think about three flavors. There's the ROM version, the EPROM version, the bond-out version. How did you go about making these decisions? How big was your team? What was the approach?

Wickersheim: Okay. There were five design engineers assigned to the design: myself and Marty Polowski, Steve Sample, Greenie Van Buren, and Chip Lau. Fortunately, Marty and Steve decided to move to Chandler to follow up on the design. Greenie and Chip stayed. We were allowed the ability to transfer to another department if we really didn't want to move to Arizona. So I was fortunate that at least two of the four engineers were willing to move. Each of us were assigned a different area of the chip to focus on. My principal area was the PLA. Steve Sample developed the logic for converting an opcode into a set of logic that controlled things on the chip. And I took that and then did the circuit work to turn it into something that would decode the opcode and activate timing signals to work with the other circuitry on the chip. And I also looked at the interface between the logic on the chip and either the ROM or the EPROM. As I said before, it was fortunate we didn't have to focus a lot of time on developing either the

ROM or EPROM circuitry. Just leave spaces on the chip at least for the ROM version. The EPROM version was much larger than that but required a modification to the layout to push one side out far enough to support the EPROM core. And in addition to the five design engineers, there was a team of five or six mask designers. The mask designers were the ones that would take our circuit schematics and then convert that into geometric data, the layout data, that was finally used to make the mask for the manufacturing process. And at that time the mask designers were just starting to use computers to digitize the data right on the screen as opposed to drawing mask data on large Mylar sheets. But the circuit schematics were still hand drawn. So by the time we were finished with the design, we had a stack of E-sized data sheets with all of the hand drawings of the schematics of the circuitry for the 8051. -- We'd hand these out to the mask designers and they'd digitize [them]. In retrospect, I think one of the problems that occurred because we did move the project [to Arizona] before it was totally complete, is that the other four engineers left before we were able to do all of the final checks. There was one last check, which was basically to have a mask designer describe the layout while the circuit designer marked it off on a schematic to make sure every transistor was in there because there could be some errors in looking at the circuitry while creating the layout. And that, I think, was not done as well as it could have been if we had all stayed together as one team. The other thing that we found was quite useful in doing the design, to aid the mask designer, was to draw the circuitry in kind of a spatial layout like we expected the mask designer to draw the geometric data. That gave them, a jump in trying to compact the layout into the smallest possible size. So one of the big final tasks I had before I moved to Arizona was redraw the central heart of the chip to make it more dense so that we could keep the chip size as small as we could. One thing I mentioned about the marketing group and John Wharton [is that] the design team was informed rigorously about the [status of] other companies, like Zilog and Motorola, who were actively developing single-chip microcontrollers and could we do it as fast as we could?

Wharton: Pressure!!

Laws: I can imagine.

Wickersheim: Once we taped out finally, and that was right towards the end of 1979, I finally moved my family to Arizona and the first task that I had once we moved was debug the bond-out chip and then start working on actually putting ROM and EPROM circuitry on to the chip. So I made many trips back to California to interface with the EPROM group. The strategy was to have the EPROM group do the design of the core layout and then we would interface that with the right circuitry to talk to the EPROM.

Laws: You did a tape out of the design in Santa Clara?

Wickersheim: It was done in Santa Clara. That was a decision made, even though most of the team and the microcontroller group started moving in July. Several of us stayed to complete the tape out there. Because the mask designers and the team was set up there with the computers and everything. And it would be awhile before Ed Kinsella in Chandler had his mask design group set up and it would have been a bad thing to transfer the layout data, partly done, and expect someone else to pick it up and finish it.

Laws: For the record here, do you recall what generation of technology was? This was NMOS?

Wickersheim: Yes.

Laws: And what lithography size? Do you remember?

Adams: The line width

Laws: Line width?

Adams: Four microns

Wharton: That sounds about right.

Wickersheim: Four or five.

Adams: Yeah, I think it was in that range at the time.

Laws: And another question, to position it in time what kind of tools did you have available to help you? If any? You had a simulator of some kind?

Wickersheim: We had a circuit simulator.

Adams: You had a number two pencil.

Wickersheim: The mask designers had a design-rule checking program which helped. It wasn't totally accurate. It could make errors but it helped a lot in verifying that the layouts were were correct and the spacing was correct. I'm trying to recall whether we even had logic simulation capability. I think we relied mostly on the circuit simulation to make sure that the speed paths within a chip were correct and could be met.

Laws: And how would you have input the data for the simulation? Teletype terminal or--

Adams: Card deck?

Wickersheim: No, no.

Laws: This was interesting time, the transition from totally hand-done stuff to the beginnings of CAD tools.

Wickersheim: We described the circuitry in a text form. And then the the circuit simulator would run and give us results.

Laws: Hank?

Blume, Jr.: Yeah, I had a couple of comments. One was as we started we design every transistor. Now some engineers at other places had learned to draw NOR gates and size the NOR gate and then the mask designer should figure out a NOR gate has so many inputs and whatever and do it automatically. So I don't know if you were drawing every transistor or doing a logic diagram.

Wickersheim: Well, I called it circuit diagrams but it was a mixture of actual transistors plus logic symbols. Where we could have a logic symbol that would reflect the circuitry we would use that. And the mask designer knew the logic symbols and what they were equivalent to in terms of transistors.

Blume, Jr.: So the mask designers had learned how to use logic symbols which meant your schematic would be on one quarter as much area. Okay, if the the circuitry we used was very dynamic, the logic simulators wouldn't work. The circuit and logic simulators would not work on the dynamic logic. You needed a switch-level simulator which was just coming in. We had a group in CAD working on switch-level logic simulators. But these really weren't working yet.

Wickersheim: Basically, we reached the point where we could actually enter logic symbols and circuit transistor symbols onto a screen and hook them together. But I don't recall that we did a lot of that on the first '51.

Laws: And the digitization was done on something like a Calma?

Wickersheim: Yeah, actually I think they were Calmas the first ones.

Laws: And then the mask was still made on a Rubylith and hand stripped?

Blume, Jr.: We developed our own master program to do design-rule check and continuity check. Then for every process you had 30 or 40 additional tables to say how the design-rule checker would interpret what you had. Some time [during the development of] the '51 series you were able to use this. I don't know if the first one.

Wickersheim: I don't remember seeing Rubulith when we were checking layouts. It was drawings--computer generated drawings of the layout.

Blume, Jr.: But the continuity checker, to match the layout with your logic diagram and--

Wickersheim: Yeah, we did not have that.

Blume, Jr.: -- design-rule checker to see that all the clearances, whatever, were right. That came into use some time in your fairly early Phoenix days, I believe.

Wickersheim: Yes. There was a constant need to improve the tools.

Laws: Roughly, how many transistors in the layout?

Wickersheim: You know, 71,000 transistors sticks in my mind but I'm not sure if that was that or another chip. Does anybody recall?

Laws: Gives us a frame of reference anyway. Okay, so you and the team completed the design. You worked very hard to get the tape out done before you left for Arizona. And you moved to Arizona in?

Wickersheim: It was right at the end of the year. Early December.

Laws: Had you seen silicon before you arrived in Arizona??

Wickersheim: No, the first-- the wafers were sent to Chandler for us to look at.

Laws: They were made in Livermore?

Wickersheim: Somebody was saying Fab 3. I think that sounds about right.

Adams: That was Livermore.

Wickersheim: Okay.

Laws: I think that gets us up the point where we're going to ask John Ekiss to pick up the rest of the discussion. But there are a couple of other areas I'd like to touch on because Bill and Henry will not be with us on the next session. Bill, you were telling us some interesting developments that came later on of working with the academic community in China to pick up the design strategy you had evolved. Can you tell us a little bit about that story?

Wong: This is a very interesting story. Back in '81 one of the major, like MIT, top schools in China was Ching Wah University. The computer science professor, I met because I was involved with developing the business in China, and he showed me an 8051 user manual, totally translated by him and also the application examples in Chinese. So I gave a copy to the museum.

Laws: Thank you.

Wong: So that's a very interesting concept because it tells me at that at that time China is modeling exactly what is the technology going on in Silicon Valley. And they looked at 8051 as one of the very key products. And then subsequently, in 1983, I had the opportunity to take Bob Noyce to China for one of the major business development for Intel. We opened an Intel office in China, working with the Chinese government. At that time we had to deal with the ministry levels and we opened up the Intel office and then start promoting Intel products. We signed a deal with the ministry for the 86 family. It's a ten-million dollar deal so that we can bring in 'the 86 architecture to China. After that, about a year later, I also put together a program for the major university in China to use the microprocessor development systems which were including 8051, like a joint partnership program. And ten-million dollars worth of microprocessor development systems and tools, probably 500 ICE modules and '51s is one part of the program. And two years later one of the university professors sent me a book, basically application notes on all the 8051 architectures, and how they can use it into real applications. So to me it was a very rewarding experience because I see now '51 is being used in China back in the '80s.

Laws: Certainly you planted some very important seeds. As you point out, the 8051 is still a favorite tool for designer systems in China.

Wong: Yeah, absolutely.

Laws: Do you have any other comments you'd like to make about being part of this team, Bill? What you learned and what you took on to later experiences?

Wong: Yes. It's very interesting. After I left Intel, I also worked with different company. I tell you, the experience I had at Intel was fantastic. I mean, we kid around, there's constructive confrontation, but we all work together, once we define what we need to work on. Although we have to work through a lot of debate because getting a bond-out chip —that was a first at Intel. I mean, develop a product for development systems. It's not for customers the first. But we did the right decision. I think that experience really, really tells you how Intel was so successful and is still very successful because of that kind of a working environments and openness. And that was very interesting. It's unparalleled in this industry.

Wickersheim: That was another policy that was quite useful, too. It was disagree and commit.

Wong: Right.

Wickersheim: During the discussion phase you could disagree all you want. But once the decision was made, then you had to totally commit to doing the decision.

Laws: Henry? Any last comments you have about your involvement in this program and what it meant for you later on?

Blume, Jr.: No. I had picked up so much more responsibility in so many other areas and Gene Hill was so capable that I was a little remote. But I do have one anecdote. I was interviewing a new college graduate from Stanford, MSEE. His name was Hoover, incidentally. And I'm coming back from lunch and he said, "What's Intel's management philosophy?" Intel's management philosophy? I didn't know how to answer it. So Andy Grove's walking down the hall. And I said, "Andy, this is so-and-so. He's graduating from Stanford. He wants to know what Intel's management philosophy is." And Grove thinks about two seconds. He says, "It's a dictatorship that believes in the First Amendment." And [after that] I didn't introduce him to college graduates, thank you.

Laws: Did he join Intel?

Blume, Jr.: No. But not many from Stanford did in those days. They came later, you know, they came indirectly to Intel. It was hard to recruit at Stanford.

Laws: Thank you, gentlemen. I will finish this session here. John will continue with the Arizona story.

THE ARIZONA STORY:

John Ekiss: Well good morning. My name is John Ekiss, and I will be your moderator for this next session and the final session for this journey we're on. I joined Intel in 1977, which happened to be about two or three years after the 8048 had, design of it had been initiated. My education, to divert a little bit here, is I received a BSEE an MSEE from Carnegie Institute of Technology it was called at the time. It's now called Carnegie Mellon University. And I graduated with those two degrees in '58. And then I joined one of the original pioneers in the semiconductor business, Philco Corporation. I was there for 12 years and they decided that they were going to cease operation, and I looked about the industry, and I went to Motorola who was a very fast growing concern at that time.I was there for about seven years including the time when my division worked on the development of the 6800 microprocessor in competition with Intel. I left Motorola and came to Intel in 1977 right near the end of the year. I had seen the advertisements in the electronic news for the 8048. I was very impressed with it. I came in as the Manufacturing Manager for the Microcomputer Division. My task was to make product and ship it. I had that job for a little less than a year when I was notified that-- I was asked if I would be interested in taking a division of people called the Microcontroller Operation down to Arizona for the purpose of broadening out the base from which Intel could grow their business through getting access to additional engineers to make it easier to hire, because of the Bay area, the constraints in the Bay area of manpower. Intel had that as one of their strategies and it's worked very well over time. I'd like to have the-- we've added two new faces to this group, Shep Hume and Ron Towle. I'd like them to talk a little bit about their background.

Shep Hume: Okay, I'm Shep Hume. I graduated from Georgia Tech in 1967 with a bachelor's in electrical engineering, spent a couple years in the army and came to Silicon Valley in 1970. Went to work from National Semiconductor and I did, it was 7400 series logic in bipolar, also in CMOS. So I had an early exposure to CMOS at National where they came out with a line of logic series 74C which was similar to the 7400. And I moved from there to Intersil in '73, worked for Henry Bloom who you probably saw on the previous tape for the 8048. And at Intersil I did a single chip version of a PDP8 which was one of the early architectures out of Massachusetts, I guess, Digital Equipment, and 1K static CMOS

RAM and a UART. So what I did at Intersil was also in CMOS. And I moved to Intel in 1976, Henry hired me. And I came to Intel with a strong belief that if you gave design engineers the same design rules in CMOS that they had in n-channel that they could design competitive products. I worked for Henry for-- I guess I did some odd jobs with Henry. We did a game chip for Magnavox and some early work for bubble controllers. And then in 1979 when MCS was moving to Phoenix they offered me the engineering manager slot for that group.

Ekiss: Okay, Ron.

Ron Towle: Okay. Well, my name is Ron Towle. I went to the school at the University of Illinois, graduated in 1974. I had a degree in electrical engineering and computer science, and I joined Intel immediately out of school and worked in the DRAM group at the time on a couple generations of 4K dynamic RAMs, 2104 and 2104A. That group announced that is was moving to Oregon from the Bay area and I did not want to go to Oregon, didn't even go on the plant trip and transferred into I think what was called the microprocessor group at the time, but essentially what became the microcontroller operation, working on the 8048. So I was the lead product engineer on the 8048 at that time. It was kind of funny because deciding I wanted to stay in the Bay area I think I was in the group six months or so when it was announced we were moving to Phoenix. Phoenix was actually a better location for a lot of reasons so we, of course, made the move, and worked in microcontrollers for a number of years, probably 15 years before I moved onto other things in management and general management at Intel.

Ekiss: Okay. Well, that completes our group for this second session. And I'd like to start off with you, George, and kind of give us an overview of the environment out in the marketplace and within the company for this newly formed MCO, and what did you see or remember from those times?

Adams: Well, this was in late '78 and in the early '79 timeframe that the MCO, or Microcontroller Operation, was created within the company with a total focus on this business segment. We were growing extremely rapidly based on the 8048 and the 8748. We were doubling revenue each year and showing that there was definitely a key market for this kind of technology and we could exploit that in a much bigger way. As you mentioned, we couldn't grow quickly enough in the Bay area, so the move to Arizona was planned for this operation. So in the marketplace we had competitors nipping at our heels and/or threatening with other products from various players, Motorola, Zilog and others, and we were looking to keep ahead of that. And so there's a very intense pace of development work and marketing work going on. We also were anticipating a big threat from Japanese suppliers like NEC, at the time. And so this was all going with a tight knit team here in Santa Clara, California and everything was moving quite well. And we knew we needed to expand more quickly. We needed to move the group to Arizona in this case. And so when this all came about, we pulled the group together in January of '79 or in that timeframe and said and that we're going to be moving the operation there. And over the course of the next, I think it was about a 12 week cycle, my wife Elaine and I, had gone down for a trip so I guess we definitely wanted to make the move. And then we proceeded to host groups for the next 12 weekends with weekend trips every Friday and come back on Sunday, going through here's what Phoenix is all about, here's what the housing market is, getting people lined up with real estate agents, concluding with a big barbeque out in the desert on Saturday night. And it was a lot of fun. I remember Lionel Smith was mentioned in the earlier segment, I remember a couple of times in the pool at the Point Resort at the time and number of other people. We had a lot of fun trying to recruit people. We got just about everybody to go. And at the end of that people were committed, psyched up and all of this had to happen though where at Intel you have another cultural aspect is the whole program of MBO or Management by Objectives. There was no slackening, no relaxation at all of our quarterly and monthly goals. We had to make those happen, so all of this had to happen in addition to your normal day job. So everybody got psyched up and bought into the idea. And this was in the January-February timeframe. And if you think of the weather in Phoenix in that timeframe it's a very nice place to be.

Man 1: It sure is.

Adams: It's hard to find a hotel room because there's too many people, snowbirds from the Midwest coming down. And so everybody was getting ready, selling their houses, all set to go. Then, the actual move date was in mid-July, and it's a little different there! Some people came off the plane with a little different perspective of what hot means, and they learned to reject the idea that it's “dry heat”, don't worry about it. It's still hot as hell anyways and, but in any case everyone was committed. We got into this and it caused us to need to still make the goals and keep on schedule. Intel had broken ground on a major site, a property in Chandler, Arizona which today houses many thousands of Intel employees, at the time it was a field with goats grazing on it and just a sign for the future site. We set up shop in about a half a dozen rented offices that were within a few blocks of each other over on the west side of Phoenix. Many people had bought homes in planning to be where the new site would be, so there was a lot of carpooling and things going on of people sharing rides over to the west side. I think this helped build more teamwork as well, with everybody sharing learning together how to adapt and to move into their new environment and that helped us along.

Ekiss: Okay, that's a good summary, George. Shep, I'd like to switch over to you and ask you to talk about some of the opportunities you had when you moved your group down here, and the difficultly of hiring or bringing mask designers along, and did you have enough engineers? What were those kind of problems you had, and what did you do about them?

Hume: Well, yeah --we moved to Phoenix in-- I guess between '77 and '80. Phoenix had seven floods and two of those were 100 year floods. So we moved to Phoenix with a small group of engineers, and some of the difficulties we had were just getting to work because the engineers had bought homes near Chandler where the new plant was and the temporary facilities were on the other side of the Salt River. In fact, management was on the other side of the Salt River and so management had a different perspective of the flooding rivers than the engineers did and it led to some morale problems. There's not the infrastructure in, I guess, in Phoenix. There's some semiconductors, but you don't have what you have in California, so we sort of had to grow our own. We recruited heavily. We went, I guess, between '79 and '85 we went from maybe 10 people in engineering to maybe a hundred and that was primarily college grads. And the other thing we did was Ed Conscilla , the person who was in charge of mask designers developed a class at a local community college to teach mask design so we had some of our own. And in those days mask designers were, I guess, artists. They were had to find. They worked a lot of hours and layouts were all this custom handwork. And in many cases, they made more than engineers. So basically we solved our staffing problems by growing our own people. We moved a few engineers with us and then we hired college grads.

Ekiss: Well, our next speaker here, Bob Wickersheim, is one of those engineers who decided to move to sunny Arizona based on his observations he made in January and February.

Wickersheim: That was a good trick bringing us out in February.

Ekiss: And he had the problem of getting that 8051 over the final hurdles to get it into production working with Ron Towle who was the product engineer. Why don't you tell us about those times and how you approached those problems and what happened?

Wickersheim: Well, since the decision had been made to do the tape out in California, and at the same time the decision was made to move the team, it was difficult towards the end. It would have been useful to have had our whole design team there doing final checks on the circuitry and on the layout. A certain amount was done, but we could have done more. But there was urgency to get taped out, get moved and get settled in, in the new place. So it was no surprise when we got to Chandler to find that this first stepping didn't work. However because it was a bond-out chip, and allowed us to bring other signals out, it gave us ability to do a little more debugging than we would have if it had been, say, the ROM chip or the EPROM chip. That helped, but we still had to go through how many steppings before we had something that worked completely, D stepping?

Towle: Yes, three or four steppings I think.

Ekiss: Three or four stepping until you got the...

Towle: Until we could get the systems group...

Ekiss: One that was shippable to a customer. And how long did it take to get to that final working silicon?

Wickersheim: Seventy-nine, '80.

Towle: What the bond-out?

Wickersheim: Yeah.

Ekiss: Well, first the bond-out and then the finished 8051.

Towle: Wow. I seem to recall the 8051 being in production in the late '81, '82 timeframe. I don't know, George, if you recall when we were first in qualification...

Adams: First they announced that we were sampling in 1980, but it took that time for people to get designs done and for us to ramp the production stepping, and that was up to the D, F or G stepping, I believe, by the time it was really tuned for high volume.

Hume: Well, there's two questions here. One is the 8051 and the other's the bond-out chip.

Adams: Right.

Hume: So the bond-out chip was available earlier than the 8051 was.

Towle: Yes. I mean, the bond-out chip was always a very low volume product. We first sampled it to the systems group in a fully functional form, I think, on the D stepping. That was probably in '81, if I recall correctly. The ROM product, the 8051, followed that, I think, by at least by a year in terms of its production startup, it's fully function version.

Ekiss: And when did the 8751 come along, because that was a big source of our early revenue on that product line?

Towle: My recollection is it was almost simultaneous. The designs were very similar. It was just dropping in an EPROM array as opposed to the ROM array.

Wickersheim: It was advantageous having the EPROM group actually design the EPOM core circuitry. That relieved us the burden of doing it and they were experts in the fabrication funnies about manufacturing EPROMs. So while we were doing the debug of the bond-out chip, I think in parallel we were actually doing the ROM and EPROM versions. The EPROM was more straight-forward, but the, or sorry, the ROM was more straight-forward, but EPROM was difficult because it was much larger, and one of the items that we had to do is change the layout of the chip a little bit. It was designed so that we could push out one side of the chip to accommodate the EPROM, so that work had to go on at about the same time that we were busy debugging the bond-out chip.

Hume: One of the things that happened with the 8051, the 8748 and the 8048 were two independent layouts, and because a lot of energy goes into the geometrical layout of the part that takes a lot of extra energy. With the 8751 we negotiated-- the culture at Intel at that time was the static RAMS were developing one process, dynamic RAMS were developing another process and EPROMs was another process. And we were trying to straddle these processes. We were trying to get one mask set, one geometry for, the logic to work on two processors. And that's a little like trying to convince Ford and GM to use the same engine in a car. And we were successful with that. And actually I think microcontrollers, was one of the early voices trying to integrate memory and logic. I think there was another voice at Intel at that time who actually argued that microprocessors should have their own process so there would be an additional process, and we were trying to integrate these things. So we argued for what we called process synergy. We wanted the static RAM process where we generally ran our ROMS to have consistent design rules with the EPROM process. I think that was a big step forward, and a major saving in engineering. One of the concerns with the engineering group when we moved to Chandler was we're going to spend all our time doing cost reductions, because that was one of the things we started with, with the 8048 was a cost reduction and we could see that happening again with the 8051. So we had to find a solution to that too.

Ekiss: Let's bring John Wharton into this conversation. John was one of the people that decided that he would stay in Santa Clara, but he agreed to continue to work with the MCO group, which was now located in Arizona, for a year in order to backstop and provide the rich knowledge that he had in this project. What kind of problems did you work on in that year's period and how did you like your long distance support activity?

Wharton: Well, first to clarify I decided-- I did decide to move to Chandler before I decided not to. I liked the Phoenix facility a whole lot. I liked the group I was working with a whole lot. I liked the product line a whole lot. I was looking forward to pursuing this whole thing. I was married at the time. My wife also worked at Intel. We actually had gone so far as to buy a house in Chandler, put down the 20 percent down payment and so forth. The problem was that my wife worked for Intel in the Customer Training Group, I worked for Intel in Chandler and we just realized that it made more sense for me to stay in the factory where there would be other alternatives within the company to retain my seniority, so to speak, and her to stay where her company-- she was less likely to find something in Chandler than I would be to find back here, so that's why I stayed behind. The other thing is that the move did cause a certain amount of confusion and disconnect and we hadn't gone through that before. So my boss, Lionel Smith, and George Adams, with great patience, decided that it was okay for me to stay in Santa Clara, but stay on the MCO payroll. And I would take on the title of liaison which meant that I spent about thee weeks a month in Santa Clara working out of the office I'd had all along, and then for a week or ten days each month I would fly down to Chandler and live out of the Point Resort and enjoy life down there, meet with my boss and see what was going on with those people. During the time that I was in Santa Clara there was certainly a lot to do. I was working on writing application notes, which is something that doesn't need a whole lot of interaction. I was working on a register simulation model written in a high level language PLM. I was preparing the first ROM code that was going to be used both to check whether the ROM process worked and also it would be used for demonstrations. There were meetings that had to be conducted with the systems group to interact with how the software was progressing with the systems group, Bill Wong, about how the circuit emulators were progressing. There was also, going through my notes I hadn't really remembered this, but I had a lot of presentations to be made. If a customer was coming into Intel to get the line up on 8086s and automotive chips or whatever as part of the day long package they would typically want to get a presentation on what the low end was doing

Ekiss: The low end being the microcontroller...

Wharton: Yes, low end microcontroller 8051. And so I would be onsite in order to be pulled in to give presentations in Santa Clara which would have been more difficult from a long distance. Something I'd like to back up a little bit. We're not sure whether it was the third stepping, the C stepping or the fourth stepping, the D stepping, at some point we had fully functional chips that we were able to show off and so forth, but whatever that number was the one stepping before that was a very largely functional product. Product rollouts sort of take place in phases. You start off hold tight to the chest, and then you let the other employees know, and then the field sales force, the field applications know, and then we introduce it to the world. And there's maybe what, a three month interval between steps for masking and changes and tape out and all that? The stepping before the one that was fully functional had a handful of flaws, the most serious of which was that it could not do any conditional branches. And if you work with computers you know that that's a big part of what they do is making decisions. Ironically,that seemed like a fatal problem. One of the application notes I had written was trying to push up a capability of the 8051, sort of novel in the industry, called the Boolean processor, which George Adams had referred to earlier. And basically this was sort of a new concept. If you're dealing with switches and lights and turning motors on and off and so forth, setting pins and clearing pins individually, reading switches individually, you don't really need more precision. Eight bits is not just enough, but it's way more than enough. What you really want to do is just check whether a certain pin is high or lower or set a pin high or low. And the 8051, because we were able to open up space in the opt-code map, we threw in new instructions that were optimized for that capability. And you could do a very complicated task, take a very complicated logic diagram, reduce that to inline code and execute that without doing any conditional branches, because you're computing based on moving bits around rather than changing the control flow. So we had actually written an application note in detail, 30 pages, with a lot of examples of long straight line code that did not involve conditional branches. And it turns out that all of those ran just fine on the stepping that was one shy. So we were able to proceed with the schedule for talking to the field applications engineers to give them the part, to give them the app notes that were all preprinted, show them this application running on a demonstration board, and I suspect made probably one for each sales office so that they could carry that around and show their top customers.

Ekiss: So you had-- what you had done then was to really advance. You didn't wait around for the de-stepping to come along. You used the C stepping and worked around the problem through the mechanism of your application note and advanced the time at which we got into production.

Wharton: It was just a very lucky coincidence because it didn't change the production so much as it allowed sort of shifting the introduction maybe three months, the introduction internally within the sale force. We were able to actually give them something that looked like it worked and they didn't know any better because it was the exact program that we had already documented in the application notes.

Towle: It was a fairly common practice then to send samples out to the field with a list of errata if you recall. So we always had...

Ekiss: I recall. I don't know if we want to go into the very famous incident that I would mention, but that was the Chrysler.

Hume: We will. We will.

Towle: Oh, yeah. I remember that well.

Ekiss: Let's move on.

Adams: John?

Ekiss: Yes.

Adams: Shep, Bob and John have mentioned things about all this working with all the different groups in Santa Clara. You had to work with the memory process design group, with the CPU process people as well. And we had John Wharton still in Santa Clara working in a remote mode. I think it's really important toset the time context here again. And this was 1979. There was no internet. There was no instant messaging. No one had PCs. A couple of people had Apple IIs, but they were only using them on their own. We had to communicate by people typing on IBM Selectric typewriters with orator type and foils and you have to jump on a plane and schlep those back and forth between Phoenix and Santa Clara. So and then as Bob mentioned the design, a lot of it was still on pen and paper red lining or yellow lining, checking off different schematics. So if the audience is thinking in terms of, "Well, working remotely, that's no big deal, that's routine," it was much, much more challenging to do it in that era than it was today. There were also no cell phones, no PDAs, no Blackberries for people to connect with each other.

Wharton: George, as you're saying this, what I'm thinking is thank God we didn't have cell phones, and the internet, and web browsers and fax machines.

Adams: Yeah, we could actually get more work done, that's right.

Wharton: Yeah, our key results would have gone to hell!

Ekiss: There's an upside and a down side.

Adams: That's right.

Wharton: Absolutely.

Ekiss: I remember in the timeframe of roughly late '80s period my son was working at Intel and he would tell me, this is after emails became in large use, especially at Intel, and he would tell me how he was getting these massive emails sent to him and they were addressed to one or two people and then there'd be 200 other individuals get copies of it. So that would certainly slow things down a little bit. But let's move on here for the moment. George, I'd like to ask you, by the time we got around to the D stepping, and everything was hunky-dory, and we had the 8751, and we had the design tools, did you sense some upward movement here in the acceptance of the product and excitement out in the industry?

Adams: Oh, absolutely. I think having the architecture become recognized, some of the new and novel things that it would do for control that had been built into the architecture, the ability to do the in circuit emulation and bread-boarding all came together in a way that really for the first time people could do many, many different designs with one architecture. In the past, each single chip microcontroller at the time was compelled to do different architectures, different instruction sets with each permutation, add more complex instructions and so forth. The 8051 allowed you to standardize on one architecture for a multitude set of design tasks. And so we were marketing that to everything from printer manufacturers or computer peripheral manufactures, to some automotive applications, to organs, people that were making electronic organs and electronic music synthesizers, and that proliferated very widely. That helped accelerate the growth and role for the group in Intel's evolution.

Ekiss: Ron, I'd like to go back to the D stepping and how that worked. What was the methodology you were using to test in production and how did you characterize the parts? Did you have some methodology that was in place in Intel for how to handle that?

Towle: Well, truthfully the methodology wasn't very well refined, but actually I'd like to paint a little broader picture of the environment at that time too to kind of segue from some of the other things that have been talked about. Like I said, I joined the microcontroller group and worked on the 8048, and it was announced, I think, late in the year of '78 that we were moving to Arizona. And we did our wonder visits in January and February of that year. The group I worked in, product engineering, most of the people took the trip because everybody, you know, the word was getting around that this was a really great trip and George was putting on a very good time. But I forget the decision deadline, I think it was maybe February or March when we had to say yes or no whether we were coming.

Wharton: Before it turned hot.

Towle: Yeah, before it turned hot, and I was the only one from the organization of about, I think it was about eight people that said, "Yeah, I'm going to come to Arizona." I eventually convinced my electronic technician to come with me, so the two of us actually made the trip and arrived in July. It's kind of hard to capture, I think, the spirit at the time, but while the idea that we had much of the MCS48 family ramping in production at the time, the 8048, the 21 and the 22, none of the experienced engineers on that were coming to Arizona. And in those days of Intel we, product engineering, had broad responsibilities, but at that point in 8048's lifecycle we were really focused on the cost reduction and ramping volume, and high volume manufacturing. And we actually did a lot of the physical production testing, wafer probing and unit testing. We did have a plant overseas in Penang, but we were still doing physically a lot of that work in the building. And support for that was like the number one priority. I mean, I think you talked about getting beat up because we couldn't supply enough 8048s. So it was a mad scramble and dash, but at the same time with this group growing it was a very exciting period. It was almost like being at another new startup company in a way. And that was a very energizing thing.

Ekiss: Let me just say it looked like it. We were in all the little buildings on the west side of Phoenix.

Towle: Yes, it was very much what I would call guerrilla engineering. I mean, there was no support infrastructure in Arizona. And from a manufacturing sense, I mean, if you smashed your probe card there was no probe card repair group in Arizona. You had to figure out how to get things done. So fast forward maybe a year I was scrambling to get the 8048 responsibility transferred over to other people, because I knew the 8051 was coming. And for me, although it wasn't good for Intel, it was a sign of relief that the first steppings didn't work because that means I didn't have to deliver samples and I didn't have to have the test capability ready to go. The 8051E itself...

Wickersheim: You got a reprieve!

Towle: Yeah, thank you Bob. Presented a lot of challenges, it was in a custom design package and that made things even worse for us because there was no standard tooling available to interface this product to any of the test systems we used. Now traditionally we had used a machine made by Fairchild called the Sentry Tester, Sentry 20 in our case, and for higher volume products we were moving towards the machine made by Megatest, guys that used to work at Intel started this tester company, but that required a lot of custom design of a logic board, a personality board it was called, and that took quite a while. Since the 51E, the bond-out was in low volume production it didn't justify making that big investment. So we struggled with that custom package, struggled to try and get all of the quality and reliability data and feedback so that when the actual ROM version, the 8051 product that would be designed into systems came out, it would work well. But that was a significant challenge, again, with no tools. I mean, I remember we hand wrote, hand coded the vectors, typed in thousands of ones and zeros by hand kind of thing. And we did our best to try and use the tools on the tester itself to learn the other code, but it was a real, like I said guerrilla engineering effort, a real, real challenge. And so eventually when the 8051 moved to very high volume we worked with a company called Teradyne to get them to craft a logic tester that was well suited for microcontrollers. And that was a very successful effort. It enabled us to get the test costs down dramatically on that and that became a standard across Intel for testing microcontrollers.

Ekiss: And it allowed us to have good test capability when we finally hit the million units a month number.

Towle: That's right. It was very capable of that.

Ekiss: In 1984 sometime.

Towle: Yeah.

Hume: And he would drag me over to look at DRC errors to prove his point. And the culture at that time-- the design rule checker could check widths and spaces pretty accurately, but there were a lot of conditions it couldn't quite comprehend. In other words if AB and C are so then the design rules is this, but if EF and G are so the design is that, and if the design rule checker wasn't sure it would create a flag. And so we'd go down and look at thousands of errors. And Ron was sure because there was thousands for errors there that there must be a real one.

Towle: I think it was tens of thousands, Shep, something like that.

Hume: And I couldn't convince him otherwise. So in desperation we changed the culture there in Chandler. I don't think it ever went company wide, but we actually-- and it took a while to get the engineers to change and the mask designers, but we said, "If it causes a flag you can't draw that way." Even though it's a little bit smaller, even if you can take a little bit of advantage of that design rule, if the design rule checker can't guarantee that it's right you can't do it. And so we went from having thousands of flags to having zero. And that whole...

Ekiss: Let's move on beyond that, because I-- oh, go ahead.

Towle: I just want to make one comment on Shep's effort there.

Ekiss: Go ahead.

Towle: I think I could be wrong on this, but I do believe on that level of technology that the 8051 produced in Fab 4 was the first fully functional wafer ever produced by Intel. So I don't know if there was a direct correlation of getting rid of all of the design rule violations, but a hundred percent yield on a wafer and running final test yields in the 99.5 percent range was pretty darn good for a product in those days.

Ekiss: I like that. I liked that part.

Wharton: I never heard that before.

Towle: Yeah.

Ekiss: Well, I think we've reached the stopping point here. We're going to have to change the tape and we'll come back and finish this up.

Towle: All right.

Ekiss: I'd like to go back to two topics that Shep raised early on in our discussion here. The first one was cost reductions and how they're needed by the business and they stress some portions of the talent we have in the engineering staff. And the other one he mentioned was his real interest in CMOS and how that came into play at Intel and MCO in particular.

Hume: Okay, well cost reductions, probably MCO was the first group that was trying to cost-reduce an existing microprocessor. I think the others tended to play out on the particular process they were on, but. And engineering was always behind, I think, at least my experience was we were always behind schedule and so there's a lot of pressure to do these cost reductions. So what we did was we had engineers working on re-laying the part out for the new process, because each process was different than the one before and each set of geometry rules was different, so it was a new effort. What we did was we took the old 8048 and we performed an algorithm where we made some parts bigger, some parts smaller, and then we shrank the whole thing by 15 percent. And we did this-- technology development was certain it wouldn't work, maybe not quite certain but highly skeptical that it had any possibility at working, and so we did this algorithm. We taped it out and it worked first time. And we argued from that point forward that Intel should leverage its investment in masks by providing a path for cost reduction with technology. Up until that point each new technology was designed around a particular memory cell, a static RAM cell, or a dynamic RAM cell, or an EPROM cell. And actually Intel picked that up pretty quickly. They went from not believing it would work to the next generation for the 8051, which I guess the first generation came out on a process called 414, but-- 411. The next generation was 414, which was a shrink. And in fact I think we decided at that point, I think that was a 20 percent shrink, that that wasn't big enough. That was too much effort even doing that, and I think they went to bigger shrinks in the future. So that was the beginning of the idea. We were a very early voice. I'm not sure there was one before us at Intel for the idea of a shrink path for microprocessors. The other thing John mentioned was CMOS. I've always sort of believed in CMOS and then when Intel developed a CMOS RAM we began a CMOS version of the 8051.

Ekiss: Yeah, but something else went on there. Management said...

Hume: Well, management was quoted, I think, as saying, "CMOS is a Cadillac where a Chevrolet will do," so there's a lot of resistance in the company to CMOS. It was not viewed, I mean, microprocessors weren't going into low power applications where microcontrollers were, and there was a view that the area penalty using CMOS would be so large that the cost would just be prohibitive. And both of those perceptions, I think, are false. The area penalty using CMOS is not that great, and ultimately the microprocessors had to come over because the power budget got so high they couldn't live with n-channel. So they stayed with n-channel as long as they could and then they came over and joined us. So we were an early voice for CMOS at Intel as well.

Ekiss: That's why I was selected to be the czar for CMOS.

Hume: CMOS, yeah. And normally you sample a product and you have it working before you get customers involved, but in the case of the CMOS version of the 8051 we actually told a customer long before we had it out that we had one coming.

Ekiss: Who is "we" that actually told the customer.

Hume: George, was that you?

Adams: No.

Hume: Somebody told Chrysler that we had this product in line and we didn't make the model year. And they ended up instead of a high tech I think it was going to be some sort of trip indicator.

Ekiss: Trip Computer.

Adams: Trip Computer, yeah.

Hume: Computer, right. They had a low tech tissue box in that slot in the car.

Ekiss: And I was assigned the task of going to Detroit and falling on my sword in a big meeting with their VP of engineering, and it wasn't real pleasant.

Hume: But CMOS was an uphill battle at Intel, but I think it's kind of the way the world is now.

Ekiss: Yeah, and obviously it prevailed as we know in hindsight.

Wharton: When I was going through my old, old files, George, I actually came across a letter that you had written to some large company, it may have been automotive, cc-ing me, giving them the introduction plans for a chip you called the 805C. That was the first I had heard of that.

Adams: Oh, interesting.

Wharton: So we can check what the deadlines were that you promised.

Towle: Ah, so now we know why that...

Hume: It may have been George, huh?

Wharton: I might have that memo right over here in my case.

Adams: Aggressive market development, yes.

Ekiss: Okay, well let's move on here and let's go to kind of the last presentation we're going to have here, the topic of this marvelous thing that we all wrought the 8051. Why did it last so long? Do you want to talk about that, John?

Wharton: Yeah, that's sort of spooky even to me. About ten years ago I was asked to give a talk at Stanford on the occasion of the 20th anniversary of the specification, January 17th, '78, so this would have been January 20th, 1998 or something. And so I put some time into doing a 20 year survey of things. Most processors are introduced, they take a year or two to be designed in, a couple of years to ramp up, they're in mature production for a couple of years and then they get displaced by something else. The 86, 186, 286, 386, 486, Pentium, Pentium Pro, Pentium 4, Quad something or other, cores, and in each case they'd sort of do a redesign. And that didn't happen here, and yet the volumes on this chip are continuing to rise. They didn't die after eight years. They kept going up and up and up. The current production rates I'm not sure exactly what it is, but they're in the order of magnitude of measured as billions of units per year still. Ten years ago-- that's billions with a "b" as opposed to maybe hundreds of millions of PC type processors. I'm not sure who's watching this video, but chances are you've never heard of some of these parts that we're talking about, and yet you probably have them in your wallet or in your cell phone. The 8051 was adopted for smart cards at one point. It was very popular in the cell phone industry. RFID tags that at some point will be appearing in every cereal box to assist in inventory is an antenna in some cases with an 8051 as the processor that's doing the communications. The issue of why the volumes are as high-- oh, and also in terms of manufacturing, at the time I did this Stanford presentation I looked into it, I found basically five vendors of significance ten years ago that were building the chips. And if you counted up the total number of products among them, different ROM sizes, different RAM sizes, different port structures and so forth, there was a little over 200 different products that were available. Two years ago I was at the Embedded System Conference and got a handout from a company that makes software tools, Keil or Kyle I'm not sure, where they listed the different products they supported. And on this brochure they listed more than 60 different manufacturers that had product lines in the 8051 family. Several of them were more than 50 products within a certain vendor. Intel had about 60. A couple of companies made more than 100. Phillips and Phillips offshoot had listed 160 different flavors of this chip with different combinations of IO structures. So that's also sort of, well, certainly very unusual in the market. I think as to why this happened, in retrospect there were probably three factors that I'll try to summarize. The first is that products have to be accepted before they go anywhere. Customers have to be interested. In this case we had included some gimmicks that perhaps made the chips more useful for the embedded control market. I mentioned the Boolean processor. We had a lot of stuff that was oriented towards real time computing. And going through my records, part of that was to be a better product, and actually a whole lot of it was because we thought a feature would be more easily marketable, unique. Here's something Zilog's not doing, let's keep track of the things that we should emphasize. So that gave customers some interest in part which may have lead to its initial acceptance. The second factor is that part of the motivation for the 8051, as I mentioned, was that with the 8048 family each of the offshoot would be a little bit different and we would have to start over. We would have to develop different documentation, a different user's manual, different data sheets, the people that did the assembler would have to go back and make some changes to change the mode. The in circuit emulator people would have to start over again. And that was killing us in the marketing group, in the systems groups having to do all this overhead. The engineers were doing miracles in putting analog circuitry onto a digital chip, but that was some finite number of months, the tail wagging the dog effect of having whole divisions then having to start over with their documentation. And what we thought was by introducing a product that was easily extended if we added a new converter we would simply send a memo to the customer saying, "You know this register that wasn't used before? Well, it's there now and it's another port, and that's all you need to know." What we didn't realize is it was exactly that overhead that was giving Intel the lead in the market. All these other companies that wanted to compete couldn't afford to do customer training classes, didn't have the profits to develop user's manuals, didn't have software teams to do the assemblers, and what they found was by them jumping on the 8051 bandwagon and building an equivalent what they could say is, "Go to Intel for the training course, go buy Intel's user's manuals, go buy Intel's assembler, go buy Intel's in circuit emulators. Do all this stuff that Intel's good at, and then when you're ready to place an order come to us because we're a little bit cheaper, or we have extra ports, or we have-- we only have the amount of RAM you need and so you don't have to pay for extra. And so it was really the first chip that other companies could compete in the chip market without providing all the support issues. And that sort of snowballs on itself, popularity leads to popularity, the market grows, more people want a piece of it. Sixty companies were in the market two years ago. Of the thousand-- together they made over a thousand different products on this brochure that Keil was handing out, and 20 of those were flagged with asterisks saying they'd been introduced in just the last two months prior to when the brochure was. I know of at least two companies that are at this point only now getting ready to introduce families of 8051 products, large semiconductor companies that are jumping on the bandwagon 30 years after the specification was adopted. So one thing is why was it accepted? Secondly, why did it expand? But the third thing is that if the products didn't meet today's needs they would have died off at some point. And in the embedded control world this is a whole different philosophy. People are familiar with desktop computers, and in desktop computers you're very interested in performance, better performance, spreadsheets, calculations, more precision, graphics, animation, presentations, rendering, web surfing, data compression, decompression, faster's always better and that's why they've always had to keep turning this crank because customers were never happy with the level they had, or they'd always be a little bit happier if they had a little more. In the embedded control marketplace what we're doing is controlling the world, interacting with world, interacting with human beings, interacting with machinery, turning motors on and off, gasoline pumps, cash registers, keyboards, cell phones, digital cameras, and in these markets what you're doing is a process that's primarily controlling items, looking at inputs, making decisions, controlling outputs, but you're doing it at real world speeds, and the real world hasn't changed much. People type about the same speed as they did 30 years ago, so if something was adequate for typewriters 30 years ago it still works. Gasoline, Moore's Law, doubling every two years, imagine if gasoline pumps had pumped gasoline twice as fast every two years. That would be a remarkable flow rate, but they don't. And so keeping up with the-- instead of trying to get the maximum possible performance, and having to keep rethinking things in order to do that for each new technology, the only thing that the embedded control world is interested in is having guaranteed adequate performance. I'm not sure that would be a very good sales brochure, "Our product is guaranteed adequate," but more than what's required is lost. If you have a chip that actually runs faster than you need it to that's wasteful in terms of, well, energy if nothing else. Batteries would go dead. It might get hotter, and so what you would do in that case is slow down a product and so you actually use it at less than its capacity. Now, the 8051 has sped up. One of these new vendors that's getting ready to introduce something this year runs at 100 million instructions per second as opposed to one million instructions per second, a hundred to one ratio, but that's still a lot less than the speed up you see in the other world. And as long as the real world, the physical world doesn't change greatly I think that this product will continue to meet the market's demands.

Ekiss: So that's a good summary of what we have wrought here with the 8051, and I think that brings our story to a conclusion unless people have any life experience that they would like to share.

Hume: I think being part of setting an industry standard is a very rewarding experience..

Towle: Thrilling.

Hume: How many people get an opportunity to do that?

Adams: It was being part of a start up team, in effect, inside a very fast moving, what later became more recognized as a very well managed overall corporation, a role model for the industry, but a start up within that environment. So it could be challenged and stretched to do things that weren't done before. Later in my career I was VP of PC products at Phoenix Technologies, and we worked with Intel and BIOS on systems, laptop power management, many different interfaces, USB, FireWire etcetera, and I was always just very nostalgic to think just how many times we ran into the 8051 in all types of different devices around the PC. And also we had started a business of licensing sales for FPGAs and for people to do macro custom logic and the 8051 was almost always the CPU or controller of choice that people were using in different designs for that. So it had just taken on a life of its own and it was very gratifying.

Hume: Yeah, I'll come back and say I remember going to Georgia Tech and talking to professors trying to get them to design the Intel system in, and at that time it was all Motorola. And it really, to see the change, I mean, it really is a standard. It's the standard now.

Wickersheim: I always get a kick when I meet new people who ask me what I do or what I did to say, "Chances are very high that a chip I worked on exists somewhere in your house.

Towle: Or multiple of them in your house and in your car. The only thing I'd like to add besides the excitement of working on a product like that was the opportunity to work with the people, a pretty unique set of people and it was very rewarding.

Ekiss: So the move to Chandler was pretty nice.

Towle: It turned out okay, John.

Wickersheim: I'd say it was good.

Hume: In spite of the floods.

Wharton: And I'd like to come back and make sure that Lionel Smith, I've mentioned his name many times, he was a very good manager, a good engineer and had a way of understanding people issues far better than I did. He did a lot of things behind the scenes.

Ekiss: I think we all remember him, and he was one of the first people to commit to come down.

Wharton: A couple of things that reflect this is, I was at a party with him once with a bunch of non-engineers and this was '77, nobody had ever heard of Intel at that point, and somebody asked him, said, "Where do you work?" "Intel." "Well, we never heard of that. What does your company do?" And he said, "Oh, we're changing the world," just matter of fact. And in later years when he talked about the microcontroller market he would say, mention, tell people that, "886, desktop computers, PCs, things like that. That's what gets all the attention, and all the publicity, and all the ads, and all the TV shows, but our microcontrollers, that's what accounts for the other 100 percent of the market."

Hume: You talk about changing the world, it reminds me of this ad we had in the early '80s.

Ekiss: That's right.

Hume: One family-- I forget how exactly. Since 1979 or something one family has been quietly...

Ekiss: One part has been controlling the world.

Hume: Quietly controlling the world, yeah.

Ekiss: I loved that ad.

Adams: That's good.

Ekiss: It was true.

Hume: Yeah.

Ekiss: Okay. I think that's it.

Wharton: Thank you, John.

Adams: Thank you.

Wickersheim: Thank you.

END OF INTERVIEW

