Oral History Panel on the Development and Promotion of the Intel 8080 Microprocessor

Participants:
Steve Bisset
Federico Faggin
Hal Feeney
Ed Gelbach
Ted Hoff
Stan Mazor
Masatoshi Shima

Moderated by:
Dave House

Recorded: April 26, 2007
Mountain View, California

CHM Reference number: X4021.2007

Dave House: Welcome to the video history of the MCS-80, or the 8080 microprocessor. We have with us today the team responsible for developing those products, and I'd like to start out, first of all, to introduce myself. I'm Dave House. I arrived just before the 8080 was introduced at Intel, so I was not part of the development team. But I'll be the MC today, and I'm going to ask some of the new members to our panel to introduce themselves and give their background.

Ed Gelbach: I joined Intel in late, mid-1971. We had developed the microprocessor at that point. However, it was being used generally as a calculator type chip, and we were looking for ways to expand it. Prior to joining Intel, I worked for Texas Instruments for about ten years. And prior to that, I was at General Electric.

House: And you grew up in?

Gelbach: Southern California.

House: Southern California beach boy.

Gelbach: I graduated from USC [University of Southern California], and worked for TI [Texas Instruments] in Los Angeles predominantly. I moved to Texas for a short period of time, and then moved up here [Northern California] in 1971.

House: Okay. Steve Bisset joined our team.

Steve Bisset: I grew up in Australia, and came to choose Caltech [California Institute of Technology] over MIT [Massachusetts Institute of Technology] because of where I thought the good weather would be. That was a mistake. I got inspired to come to Silicon Valley by Carver Mead's classes at Caltech. He was one of the pioneering IC design instructors there. And we were in class projects that he had Gordon Moore running chips for us, so we started working on CPU design at school, which was an amazing experience for me. I worked for HP for six months. I'd gotten a job offer to design memories at Intel, which I was not interested in at all, and then Ralph Ungermann called me up and said, "Do you want to join the microprocessor group?" And that was an amazing thing to watch. I wasn't involved in the development of the 8080 at all. The day I arrived at Intel was the day we found the first working revision of the 8080.

House: I understand you discovered the 8080.

Bisset: I was the discoverer, yes. I found it.

House: Found it on the wafer.

Bisset: They said, "Here, push these buttons. Find one that works" and I looked at all the lights and found one that worked. But I was involved in, basically, the definition of and a little bit of design work on the peripheral chips that went with the 8080.

House: So Shima-san, you had been working for Busicom in the days of the 4004, but spent time here in California at Intel. Then when the design was done, you went back to Japan. So what happened between there and when did you joined Intel?

Masatoshi Shima: After I developed the desktop calculator using the 4004, in September 1971, I moved to Ricoh and I worked with the systems development group. I had a very good experience there that was quite useful for the 8080 development. There was the design of the interface between I/O typewriter and the minicomputer. And also, the design of the interface between 16 bit graphic minicomputers and the mainframe. And also, I designed the digital controller for high speed printers. There was one more excellent experience. It was the designing of a production tester of a magnetic drum memory, which was used for Ricoh's <inaudible> computer. I used NEC's 8 bit minicomputer, which was quite similar to Motorola's 6800. So I became quite familiar with eight bit minicomputers, the instruction set, its assembly language and also peripheral functions, such as DMA controller, timer, interrupt controller and serial and parallel port. It looks like I was ready to develop an eight bit microprocessor.

House: Good experience.

Shima: I was quite lucky.

House: So Federico, at this time you've completed the design of the 4004 and the product has been launched. The 8008 is in the market place. You're leading the design of everything but memories, I understand, at Intel. The 8008 was launched in, was it April of 1972?

Hal Feeney: April of 1972 was the launch of the 8008, the introduction to the customers. And really the introduction to what a microprocessor is, and just going through the entire learning experience at the time for the customers.

House: So we've set the stage. Federico, tell us about the birth of the idea for the 8080.

Federico Faggin: Yeah, that actually was while while we were still doing work on the 8008. It was summer, late summer of 1971. The 8008 was still in layout, but at that time, there was a provisional decision at Intel to actually go and market the 4004 and later on the 8008. So I went with Hank Smith in Europe, where we talked to a number of customers that had potential projects for the 4004 and potentially the 8008, to present the microprocessors and get their feedback. So that was at the end of August or so. And I was surprised, actually, by the amount of criticism we received, particularly about the 8008, and particularly by computer companies, like Plessey and Nixdorf. There was a certain amount of animosity even in them. Part of that, I think it was the sense that a semiconductor company was beginning to compete on their own turf.

But at any rate, that feedback was very important for me, because I realized that, from the view of the customers, they were seeing many limitations in our microprocessors, and particularly the interrupt structure. It was highly criticized and rightly so, because the 8008 had a very primitive, barely functional interrupt structure. Also they were criticizing the fact that we were multiplexing, at that point, addresses and data. We had a very small package. All the interface TTL [packages] that we needed to create a CPU using the 8008, that part we knew. It was a limitation that was imposed by a choice of package that was really not ours, in a sense. It was management's choice. And of course, they wanted much higher speed. The speed of the 8008 at 0.5 megahertz was not adequate.

So when I came back, I started thinking about how we could use the N-channel technology, which was now being developed for the 4K dynamic memory of Intel. It was a high voltage N-channel technology using three supplies, plus 5, plus 12 and minus 5 volts. And as I got to think about it, I came up with an idea of—a concept really—that we had to use a 40 pin package, that we had to put in a better interrupt structure, and the only way to do that was to have a stack pointer. And, of course, that was not an invention. Everybody knew about stack pointers. We had chosen a stack simply because the 8008 was conceived originally with serial memory, and therefore you couldn't use RAM to store the status of the machine. And of course I wanted to have N-channel technology. And then there were odds and ends. Instructions: I think I thought through about ten instructions or so. Then I wrote a memo to [Les] Vadasz in early 1972, eliciting him to get me started on that project, at the moment that the 8008 was finished, which was around March–April of that year.

Vadasz, at that point, was against that. He was against both the use of the 40 pin package, because Intel was still in the 16, and they had crossed the line of the 24 pin package, because the 1702 used a 24 pin package, but still there was resistance on the 40 pin package. And there was a resistance of doing this project, because Intel had just announced the 4004 and the 8008 and they wanted to see how the market would respond to that before starting this new project. So that whole project sat there for a while.

I also at that time identified that I wanted to hire Shima for that project. I asked permission of Vadasz to start that process, recognizing that it would take some time for Shima to arrive, given visa issues and so on. Vadasz agreed with that, because I argued that if Shima didn't do the 8080, he would do something else. I wanted to have Shima because I respected his engineering skills. So then I did a trip that I took to Japan with Ed. I actually interviewed Shima and Shima was interested in coming to Intel. Then we started the process for a visa for Shima, at the same time that I was pushing management to start the 8080. Of course, at that time, I didn't know how long it would take Shima to arrive, but the lead-in time was not Shima's arrival; it was actually the decision by management to start the 8080. That didn't happen until late September or early October. At one point, I remembered I was very eager to get going, and I went to Vadasz and I said, "Look, Vadasz, we've got to get going on this one." He basically recommended me to write a memo where I would have to have the endorsement of Hoff and Mazor, to actually re-propose the idea that I proposed six months earlier, basically, to get it going. After that, was the decision, shortly after that, to actually start the project. By that time also, Shima had agreed to come around November, so I waited until Shima came, instead of assigning a different engineer to the project. So that was the origin of the 8080.

House: I understand that there were some discussions with Ted and Stan about some alternatives, and maybe Stan or Ted could talk about the alternative approaches that were proposed for following the 8008.

Stan Mazor: Well, we were in contact with some of the guys from Datapoint, who had originally proposed the 8008 architecture. They had decided ultimately not to use the 8008, so they were building a processor out of TTL. We were selling them shift register memories, which they were using as their main memory on the Datapoint 2200 Model I. That was in fact, as Federico just mentioned, the reason that we had a stack on the chip for pushing down program counter subroutine return addresses. As we started to think about the limitations of the 8008, the three things that came to mind were the time division multiplexing of pins, which slows it down, the on chip stack, and the limited addressing modes.

So we were looking at alternative changes to the 8080. If you looked at the 8008, it's pretty hard to add some more instructions to it. The first decision you have to cross is, "Are we going to make it machine code compatible?" That is, "Will ROMs that were used on the 8008 be usable on the 8080?" And we decided early on that we would take a hit on that, that is, strict machine code compatibility would not be a requirement. On the other hand, we wanted to make the assembler language convertible, so if someone wrote program in 8008, they could convert it over to 8080 code. To be specific, the I/O instruction on the 8008 had 5 bits of device address in it, so by making that into a 2 byte instruction, we freed up 31 op codes in the instruction set.

Then we ended up with, in general, what I would call four proposals: one a very, very modest change to the 8008; the next one was a richer one; and finally, the most exotic one was quite an advanced processor with indexed addressing, indexed register, and many addressing modes. Federico and his group gave us sort of a time budget for each of these proposals, and the one that we called the “new chip" was going to be 100 man-weeks out, whereas the "enhanced 8008" would be around 22 weeks, or something like that. In light of the time budgets, it wasn't too difficult a decision to make, and we had many interactions on what those instructions should be, and it finally ended up with the instruction set that we had on the 8080.

House: Ted, did you want to add anything?

Ted Hoff: I wanted to add a few things. Stan and I were really having lots of difficulty dealing with customers who were looking at the 8008 and trying to push it beyond its capabilities. They'd come to us asking for help, and there was just a limit to what we could do. So when we heard that the 8008 would be re-done in N–channel, we were very enthusiastic about getting those problems fixed. I don't know when we made the decision to try to keep a degree of compatibility, but that seemed to really help in the long run, because then we could sort of tell the customer, "Try working with the 8008. If it doesn't solve the problem, the next generation will be coming."

House: There's a better one coming.

Hoff: "And the degree of compatibility is such, the investment you've made won't be wasted." I think that really did seem to help in allowing us to promote the 8008: having the 8080 coming along. That's one of the few products that we tended to talk about before it was available. I mean, Intel had had a policy up to that time not really to talk about anything that was in the pipe.

House: So Shima, you arrived to do the design. What was your role in the project, and tell us a little bit about the team.

Shima: In April 1972, Mr. Gelbach and Dr. Faggin came to Japan for marketing the 4004 and 8008. Then I was invited, but I felt, "It's not so easy to work in the United States." Then I told them, "If you are able to obtain a green card while I was in Japan," it was impossible, okay? But I told them.

House: You thought it was impossible.

Shima: Yes, impossible. Then about in September, some time in September, Dr. Noyce made a call to Ricoh's director in R&D and he called me. "What happened?" I said, "I wish to go to Intel." He said, "It's not so easy to work in an American company, but I will keep your position here for five years." I told my wife, "I want to go." Then I came here. But at that time, at Ricoh, there was one engineer who evaluated the 8008, therefore I joined with him to evaluate. Then I found five weak points in 8008.

One of them was, in the 8008, it used like two clocks to generate one state. It slowed down the performance quite heavily. The second problem was, as Federico mentioned, there was no general purpose stack, and also, it was quite weak for the interrupt service. The third one is it had a very limited space for memory, and also I/O. The fourth one is the instruction set of 8008 was over emphasized, to only 8 bit data type. Moreover, there was only a primitive instruction set. Also, there were very poor addressing modes, such as direct addressing mode for jump instruction, and register indirect addressing mode for accessing the data memory. There should be a register indirect addressing mode for jump instruction and also a direct addressing mode for load end store instructions. There is one more headache to using the 8008. There was only one 16 bit register to be used as an address pointer. It was really a headache. Also, there was no exchange instruction [which exchanges the data between two 16-bit registers]. That means that somehow we have to save the 16 bit address pointer to the external world. It takes lots and lots of time. And fifthly, the system bus ports are quite poor and utilized lots and lots of TTL.

There were five such weak points, therefore I knew what I had to do. Then I came here in early November 1972, and I wanted to take a little bit of rest, then work, but just after I came here meetings started, almost like two times a week. I found out, as Federico said, there were memos, but there was no written product manual. That was my headache too. Then, what shall I say? I judged the 8008 basic architecture of both register organization, such as single accumulator, with multiple registers and the addressing mode was acceptable to the so-called embedded systems, such as the intelligent terminal, cash register, printer and copier. But the function specification of 8008 was just insufficient. That was my feeling, therefore my main goal for 8080 was to develop a high performance 8 bit microprocessor, improving the 8008 greatly and also, fundamentally, solving all the problems previously mentioned.

Faggin: Of course, that was also our goal, and it was stated well before Shima joined Intel. In fact the original specification was not very far from the final specification, but of course, there was a lot of contribution from Shima and Stan Mazor, and also Ted and Hal Feeney in those early discussions, to take that original concept and original architecture and expand it, adding some more instructions and improving on that original idea. This is always what happens when you begin a process of actually implementing the device. So as Shima joined Intel, we could really get going, now having somebody that could really carry the ball. My task was also to teach Shima how to design chips with N–channel technology, which was a new technology.

All the methodology that I had developed for the P–channel device—for the 4004 and 8008—had to be upgraded for the new process technology, which I did so that Shima had all that he needed from the circuit design point of view and layout point of view. Mostly from the circuit design point of view, so that he could really in earnest start the logic design of the 8080, which started shortly after we defined, after we completed the architecture, which I would say was around early January of 1972.

Shima: I will give one example. In order to examine the 8008 instruction set, I made two programs. One was the 256 byte data transfer in memory. The other example was the 16 digit decimal data additions. Then I picked up the new 8080's instruction set, such as—let me see—load instruction and store instruction between accumulator and memory, with two address pointers of BC and DE. Next, added instructions are, 16-bit increment instruction and branch for loop instruction, 16-bit immediate data load instruction, and decimal adjustment instruction. The data transfer programs' performance went up ten times, with the same clock frequency. And in case of decimal calculations, I got 20 times higher performance, therefore I had come to be confident in the development of 8080. If we increased the clock frequency to be like two megahertz, we can expect 20 times higher performance.

House: Who were the key contributors on the team? Were there other people working with you at that time?

Shima: Just me and two mask designers. But later, three mask designers were added.

House: What was happening with competition at this time, Federico? Were you concerned about that?

Faggin: I was very concerned about competition, because obviously word of the microprocessor had gotten out. Rockwell had introduced the PPS-4. It was very similar to the 4004, to the MCS-4. Performance wise, it was roughly equivalent because they used, I think, it was 42 or 44 pin packages, so they did not have to multiplex the address and data and so on, like we did on the 4004. So they were regaining the natural loss of speed that metal gate technology had compared to silicon gate, by using a better bus structure, which was always my gripe on the early microprocessors. There was rumor that Rockwell was working on an 8 bit version. I did not know at that time that Motorola was working on an 8 bit microprocessor, but I knew of TI. TI had announced not a product, but it announced a capability of a CPU and a chip, which was essentially the 8008. We discussed this story in the 8008 oral history.

I was quite concerned. We were losing the time advantage that we had, having done the 8008, and then squandering that time advantage by not starting the next generation in time. We lost nine months, basically, just by waiting. Eventually, in the middle of 1974, the 6800 came out. That showed that our lead that was a year and a half was cut down by nine months. It was a major blow to me. But still, we maintained the lead. Part of the reason was that Motorola used a 5 volt process technology that was not ready for prime time, so their chip was very large and very slow. So it was not a match for the 8080, and that was actually what allowed the 8080 to gain market acceptance, despite the fact that, in my opinion, the 6800 had a better architecture. It was a more regular architecture. It did not suffer from this patch over the 8008 that the 8080 represented, and it had a better addressing mode. It was a nicer machine.

House: What process was the 6800? Was that metal gate or silicon gate?

Faggin: It was silicon gate. There was basically no microprocessor where metal gate succeeded to speak of. Only with silicon gate could the competition match the performance of our processors. Otherwise they would have two or three times less speed, even with the same design.

House: Even though some tried with metal gate.

Faggin: Yes. But the design of the 8080 was essentially carried out by Shima with my supervision. Shima would come from time to time to ask for help, or ask questions related to circuit design or layout or what have you. But basically, he handled all that design by himself. He managed that project by himself, and also the following debugging of the chip. He can tell us about how that went. The 8080 really represented a turning point for the microprocessor industry because of the speed advantage that was just mentioned. 10 to 20 times faster is really a head–turning kind of improvement.

House: Shima, as you were designing this product, what kind of tools did you have? What kind of CAD capabilities?

Shima: At first, as Federico said, we were able to utilize a 40 pin package, which means, as Federico said, it became possible to have the 16 bit address and 8 bit data bus, and also several necessary bus control signals were placed independently. Also, it was possible to send out processor status information through the data bus at the beginning of a memory cycle. Also, it did not drop any performance. The biggest advantage was the maximum power consumption with a 40 pin package was 1.2 watt. That means I was able to utilize about 1 watt, and it became possible to integrate lots and lots of transistors.

Secondly, the newly developed NMOS process technology for DRAM would give me high performance and high density. It's relatively easy to design for a two megahertz product. The third one—this is good and bad. The digitizer for the layout was newly available, and it digitized the 100-times hand layout for the mask making. Therefore, it merited to use it. However, when I started the project, I didn't notice the danger or the pitfall yet. This pitfall caused a big problem five months later. I will explain it later. Fourthly, an in-house DEC 10 computer became available for the circuit simulations. But I have to use it from the terminal room, not from the engineering room. However, the DEC 10 became an excellent teacher for the N-MOS process and also circuit design. In fact, when I asked the DEC 10, the DEC 10 clearly answered to me right away. That is the best one for me, a good teacher. Better than Intel University.

House: I understand there weren't a lot of other engineers using the DEC 10 at that time, so it was sort of your personal computer.

Shima: Yes, it looked like it, yes. Almost nobody there. Only a memory guy and I were there.

Mazor: That in house program that Ted and I wrote, we called it PULS, and this was prior to Spice being available on the open market. The DEC 10 was a pretty powerful machine and the only peripheral we had on it was a teletype.

House: You're kidding.

Mazor: Yes, so we would run circuit simulation 16 hours a day and get a few lines of output, and you could say, "The circuit works," or, "It doesn't." We also did, at that time logic simulation, and I think Shima was also able to do some logic simulation as well.

House: So are there any challenges or problems that you faced meeting the design goals?

Shima: As Federico said, we had to develop the 8080 quickly. It took only 16 months from the project start to wholly working silicon. The project for the 8080 started in November 1972, and it was completed in January 1974. One month for product definitions and eight months for chip designing, from logical design to tape out. It took about five months for mask cutting, mask making, wafer process and debugging. At that time, it took about three months from mask cutting to the wafer out. It was a long time. Can I talk about the major challenges?

House: Sure. Yes, please do.

Shima: There were so many challenges and problems in the 8080 development. At first, as I said, there was no recent product detailed manual when I came here. Secondly, it was unbelievable, but only I was assigned to the 8080 microprocessor project. So there was no chance to perform both logic simulation and hardware emulation.

Then we talked with Seikosha. Seikosha is the parent company of Epson. They were a customer of the 8008, and they decided to build a hardware emulator for Intel. But unfortunately, it was completed just before the wafer out. But they supported my work to generate the test vector. It was quite a big help for us. Thirdly, the process for making semiconductors had been changed from PMOS to NMOS. Therefore I had to learn how to use NMOS, and once again from Federico. I was not able to utilize the database of the 8008, because the 8008 used two clocks for one state, and also it was a PMOS process. Fourthly, the schedule was so tight. In the middle of December 1972, Federico told me, “I will hire two mask designers on January 2nd.” “Oh gee,” I had been left only two weeks before the layout design.

In December 1972, there were too many jobs that had to be done concurrently. Those were: writing the functional specifications and handbook generation of circuit design, logic design certification, and also circuit design with a layout plan. Then, as I said, there was no time for hardware architecture design, and also chip plan. That was most dissatisfying through the 8080 development. I had to look for a new design approach. At first, I decided not to use any tricky logic or circuits with high difficulties. Also, nobody knows, but I gave up the performance and also input/output characteristics.

Faggin: I want to notice one thing. Here he is the customer, before, for the 4004. They wanted me to do four chips in six months. Now he's designing a chip and he's complaining that he has only 12 months to do it!

<laughter>

Shima: Firstly, both the 4004 and 8008, always there were two engineers: one for logic, one for circuit. But this time, the number of devices went up twice. [The original development schedule of 4000 series was proposed by Intel, not by Busicom. Actually, it took 6months for 4004 logic and circuit design. In case of 8080, all of logic, circuit and layout designs were completed within 6 months.]

<laughter>

Faggin: That's not true. He <Faggin points off screen at another panel member, not Shima – Hal Feeney?> built the 8008 himself with my supervision, exactly like the 8080.

Shima: Really?

House: How many transistors in the 8080?

Shima: I estimated about 5,000. Secondly, with the foreshortening of the design period, I tried to design the logic by thinking about both circuit design and layout plan together. Thirdly, I decided to set up about 15 to 20 percentage of margin for circuit design. As a result, layout was not so good. However, the 8080 was able to achieve the initial goals of performance and chip size. Fourthly, I tried to keep at least one month of stock at each design stage for the next stage of design. May I talk more?

House: Yes, please.

Shima: I started the logic design from the register file unit, which includes a program counter, stack pointer, address update logic and address output latch logic with driver. The layout of the register file unit was done quite well, because layout plan was done before the layout start. But after the register file unit, I designed the 8 bit arithmetic unit. As I said, I didn't want to use any tricky logic, therefore simply I used 4004's ALU with 8 bit length and additional logic operations. But it took lots of time for circuit simulations. At this time,. I was able to secure a stock of about two months for the layout design. By this time, all of the control signals of the data path were decided.

The next logic design was the instruction execution control. The logic design method with a state machine was quite suitable for the small system, such as 4 bit and 8 bit microprocessors. Also, I had enough experience on its logic design method. However, it was expected that the number of transistors in the 8080 would be about 5000. Also, the number of equation for instruction execution control will be about roughly 100. It was not so easy, but I found one good way, in order to avoid logic mistakes. I was not able to use a logic simulator [due to lack of manpower]. Somehow I had to find out some good way to make the logic. I generated a table of all of the instruction set. That table looked like today's Excel table. All of the instruction names were placed to the vertical side. Then the op code, all of name of machine cycle, and name of state cycle were placed to the horizontal side. Next, all of the data transfers for each instruction, all of address information, all of operations, and all of activated control signals were written in each box. After the optimization and verification by manual means, all of the control signal logic equations were made. After that, all of the control signal logic equations were compared with the table for the verification. I repeated it. I repeated that verification until there was no mistake. When I found no mistake twice, then I stopped the verification.

The biggest difference between the 4004 project and the 8080 project was the timing of layout start. The layout of the 4004 started after all of the 4004 logic schematics were drawn. But on the other hand, the layout of the 8080 started only when the register file unit was designed. Today, as you know, the data path unit is composed with register file unit and arithmetic unit together. However, I didn't have enough time to clearly think through the details of the data path unit. Therefore after 8080, for the 8085, 8086 and Z80, the data path was drawn quite well. Month after month, the logic design and circuit design became busier and busier. It became harder to find time for the layout plan of the ALU. By the way, the digitizing system was newly available, and quite often, before the connectivity check, the layout was digitized and thus it was getting quite difficult to manage the layout. I needed more time to verify and modify the layout after digitizing. Finally, I placed–how shall I say it–my family mark, my family crest on the metal mask, and all of the design works were completed on August9 in 1973. I wanted to take one week of vacation, but [I was not able to take the vacation.] I will talk about that one later.

House: So now the design is completed. It’s not rubylith–there's real tape out now, I guess.

Faggin: Yes, it's real tape out.

House: It's real tape out and we go into mask making. What happens next?

Mazor: Well if I could just interject one thing, as a logic designer and architect, we can lay out the register pairs, BC, DE, H and L in about two and a half minutes in terms of a little rectangular box. <laughs>

House: On a piece of paper.

Mazor: On a piece of paper, and the instruction set that said, "We want to exchange these two registers or we want to be able to do an add." So there's a hell of a difference between the short time it takes on the specification time, than it takes Shima a very long time to implement it. I have the highest respect for that translation process.

House: So what happened at this point in time?

Shima: Next I designed the functional tester with Intel's high speed SRAM [Static RAM] for debugging. Next Paul Metrovich completed his tester for me, and all of the preparations for debugging were completed. The long awaited telephone call came from Tom Rowe of the Mountain View factory. It was 2PM, Friday, in the middle of November, 1973. I was so excited and I kept saying to myself, "There is no mistake." When the 8080 wafer was installed in the wafer inspection microscope equipment. And next a wafer probe was quietly lowered and contacted to the wafer. It was the start of debugging. Maybe at that time, Federico was there. You were not there?

Faggin: I was leaving you alone. I know how you feel.

<laughter>

Shima: Next, I turned on the power supply, and pushed the reset button to strat up, but nothing happened, exactly the same as Federico. I tested many, many dies and wafers, one after another. Still nothing, nothing happened. Then I decided to check the internal clock driver. I checked the clock drivers layout many, many times. There shouldn't be any mistake. I picked up one micro–manipulator, then I touched it to the clock driver. Nothing happened. The data I later found out was that there had been a careless operator mistake in the wafer fab, but nobody believed me, because I'm an amateur in semiconductor processing itself, I’m not a professional. <laughs> It's very hard to convince them, "You made a mistake." But the second wafer was made from the back-up plan with red tag. It came out in the latter half of November.

Fortunately, after three days debugging, it was found out there were only three small mistakes. All of them were mistakes of connectivity in the layout. One ground line in the address update section was not connected with the main ground line. Next, a five volt power supply line was used, instead of a plus 12 volt power supply. The last one is that a two input NAND device was used instead of a two input NOR. At first, in order to fix them, several metal lines were cut by the micro–manipulator. It was quite an interesting job. And next, two signals were picked up by two micro–manipulators, for generating the right signal externally. Then this output signal was returned to the wafer, also with micro–manipulators. Also, I fixed the plus five volt power supply and ground line together, and totally, seven or eight micro–manipulators on the checking box. It took almost two weeks to debug the 8080 wafer. Fortunately, there was not any mistake in the logical design, and all of the mistakes were fixed. The third wafer came out in January 1974. There was no trouble in the performance too.

Then I went to Japan for marketing and a technological lecture meeting and also vacation for about two weeks. But there was so large an audience in the meeting. I supposed there was about a 500 person capacity. There were 600 people–so many people. The 8080 was evaluted quite well. I was quite happy. But when I came back from Japan, I found out one big serious trouble–big trouble. It was the internal main ground line had a narrow width, which was allowed only low power TTL. Even with low power TTL, we had trouble, but it sold quite well. After the correction of the ground line, we changed the product name to the 8080A.

Faggin: Making virtue out of necessity.

Shima: I enjoyed the 8080 project quite well. But I want to know how well the 8080 product sold.

House: So maybe we can turn to Hal and Ed for that. But first, let's hear a little bit about Steve Bisset's experience: the person who discovered the first 8080 by running your test box, I understand. But also his recollection of the creation of the supporting chips, the chips that went around the 8080 that was part of the overall MCS-80 solution. So Steve, what was your experience and involvement in the project?

Bisset: Well, what I wanted to do was design chips—what we were taught to do in school—and I wanted to do what Shima was doing. It was a fun thing: specs and logic, layout, masks and all that sort of thing. I was at Intel for a year and a half and I never got to do that, all the way through to the end, because, I guess, there weren't too many people who could draw up nice, well defined specifications for chips, so that's what I ended up doing.

The first three months were really helping Shima with his test boxes and characterizing things like the ground line bounce and so forth. But then these things were starting to sell, and I guess the chip was what it was, and nobody was in a hurry to change that, but they were interested in trying to make it easier for customers to put all the stuff around it. It started with the bipolar process and it was the clock driver. A couple of guys over in the bipolar design group, I think it was Roman Rycerz and maybe Howard Gage? I don't know. They had this 8224 and 8228 that were designed to take all the massive transistors and external logic and TTL and actually make it kind of a kit that you could plug in and make run. We did a lot of fiddling around with clocks, to try and squeeze more performance out of it [the 8080] without requiring the customer to do different stuff.

I think the evolution of anything where it's changing, is you find out there's a whole lot more kind of infrastructure things that had to happen, and testing was one of those big things. The box that we had—the 8080 I think was more complicated than the previous chips that those boxes had been used for, so there was a lot of soldering, and a lot of bad solder connections, and we had all that kind of problem. We definitely had issues about figuring out whose job it was to build testers: engineering or the production department. We had issues about how to get things transferred so you could kind of get on with the next job. But anyway, we fought through all of that. Then I was just in a position—I don’t know how this happened—but somebody decided we needed [peripheral controller] chips.

We needed a counter timer chip, a interrupt controller, a DMA chip. We already had the parallel I/O chip. So I got involved with those and had to try and find people in sales and marketing and say, "Well what should these things do?" Nobody really knew much, so we just used the engineering approach of just trying to pack everything in the smallest space we could and hope that it was right.

Something that I found interesting is that most of these chips then got handed over to other people, and I think the farthest I ever got with the design of a chip was the 8259, for which I did the global layout and all the logic design. Subsequently I heard after I left there, it was the only one that worked in any reasonable amount of time, because we simulated it fully, which was done on an E-size piece of schematic with little tenth inch squares, and had about 100 nodes on the chip. I don't know how many clock cycles it was, but you'd clock it, and you'd do all the logic calculations all through the chip, decide what every next state of the node would be, and found a couple of mistakes like that. I think basically what Shima did was the same technique. Some hard work in that area kind of pays off in terms of long schedules. One little mistake costs a lot of time when you're doing the silicon.

House: So the 8259, was that the DMA? No, the interrupt.

Bisset: Interrupt.

House: The interrupt controller.

Bisset: I actually have no idea how well any of those chips sold.

House: Actually, they sold quite well. The 8251 was the UART [Universal Asynchronous receiver transmitter]. Who designed that?

Feeney: That was Grant B. Allen. [Shima edit – the 8251 was designed by M. Miau.]

House: Grant B. Allen was the lead designer of that. The 8253 was a timer counter. Who designed that?

Bisset: I did the logic and the spec, but I'm not sure who actually implemented it.

House: You don't remember. The 8255 was the parallel I/O.

Bisset: Peter Salmon, I think, did that one.

Shima: Peter Salmon, yeah.

House: The 8257 then was the DMA controller.

Bisset: It was that Chinese guy.

Shima: I know.

Bisset: I can see him, but I can't—

Shima: He used to work under Ted Hoff.

House: Chinese guy that used to work for you that designed the 8257?

Hoff: There's one part I remember, an 8155 I believe, which was an EPROM with I/O.

Shima: That's the parallel I/O.

House: 8255, it's been a while.

Bisset: 8755?

Hoff: It's some kind of memory, I thought, some combination of memory—

<overlapping conversation>

Hoff: One of them. I'm not sure I have the number right.

House: There were a number of parts I remember that wound up by being part of the solution. Although I joined in February of 1974, I really worked on memories until 1976. I started with microprocessors when I became marketing manager for microprocessors in 1976. But those support chips were an important part of our sale at that time, because our competitors didn't have that capability.

Bisset: Yes, we certainly cranked through the spec [specification] phase of those pretty quickly. Then it got really interesting. We got engaged with a real customer, which was kind of thrilling to me, because we had people who were building systems and you could ask questions about what it should do. They actually had examples they could cite to answer these questions, as opposed to the previous process, which was sort of imagining what it might do and trying to anticipate that. So we got heavily involved with these guys from Olivetti. They wanted a floppy disk controller, and a serial communications controller: SDLC, HDLC.

House: Yes, the 8271 and the 8273. The 8271 was the communications controller, and 8273 was the disk controller, I think. One or the other.

Bisset: Yes, and what we decided there—and in a way, I'm not really sure what the architectural lesson was, but it's kind of a possible misapplication of good lessons—but we decided instead of doing two separate custom chips, that these were both chips that had a lot of interface to 8 bit processors, but they were all manipulating serial bit streams in I/O. So we thought, "Why do two custom chips?" "Let's save time and do a general purpose chip that could do very sophisticated things with any serial I/O data streams and convert them internally into a group of 8 bit registers, and interface in an intelligent way with the processor." So we had this thing called a bit processor and a byte processor, and we used the architecture that Carver [Carver Mead] inspired us with at Caltech using PLAs to control all this—so it's, you know, fast, using parallel logic to make quick decisions. You could get the speed out of it that you needed to do these things.

I think the interesting lesson—again that was another one that was a set of chips that was defined, and then I left but I kind of kept a little bit in touch on that. I think that took a long time to get done, because we were saving time, in theory, taking two custom layouts and converting them into one general purpose task. But it turned out you had to kind of have a development kit and development software and all the bits and pieces that went around it, and that caused the project to drag out.

Feeney: Another computer.

Bisset: It was another computer, and whereas that paid off enormously for something like the CPUs, which were fully general, something like these, where they were somewhat general, but serial bit manipulation was I think too narrow [of a market] for that to pay off. Probably in retrospect, that was a lesson mis-learned or misapplied. That was a learning experience.

House: As I recall, those two chips were fairly large.

Bisset: Yeah. Too general.

House: And fairly late—

Bisset: Yes.

House: —to the marketplace. We had lots of issues with them.

Bisset: Yes. I think, in retrospect, it would have been easier just to crank them out in a fully dedicated, custom manner, just to be done with it.

House: Sometimes the elegant solution isn't the best solution.

Bisset: They were very elegant solutions, but for those you need to have big, broad markets that would make them pay off.

House: So Hal, you'd moved over into marketing. Tell us a little bit about how you evolved from the engineering to the marketing, and about your activities there and the launch of the product, customer interactions, etc.

Feeney: Let's look at both the 8080 and also back a little bit to the 4004 and the 8008 getting into the marketplace. With the 8080 now, this became the first microprocessor that Intel had that actually was able to get and able to take advantage of feedback from the customer base as far as, be it in Federico's case, the Europeans telling him all about all of the limitations that were on the 8008. And at the same time, we also began—and then it was a learning process, as we talked about on the oral history for the 8008 and the 4004—we were announcing and bringing to the market fairly naked devices. I mean, we were a silicon company, so we produced a microprocessor with, in the case of the 8008, 18 pins on it, and not a lot of stuff around it at that time, other than a datasheet and a user's manual on the day that it was announced. We quickly found that we needed demonstration vehicles that Ted and his group developed to show how the product could be used, getting a variety of schematics, example schematics on different applications, what the interface would be like to the processor and so forth.

Suddenly, when we had all this put together, we had this very little processor right in the middle of a board and this whole sea of logic around it. That was certainly one of the motivating factors for going and developing an even higher level of integration to take a lot of those simple functions and put them some place else. At the same time, we were also developing a new way of going to market, a new way of getting customers to look at the microprocessors, and Ed can comment on this too. That was when we set the price on the 8008 when it came out; we set the price of $120 for this 500 kilohertz processor.

We worked, and a short time later came out with the 8008-1, which was 50 percent faster and we set a price of $180 on that processor. The goal was to be able to sell our processors, provide the—we gave the data books away, and we probably gave away more data books than we sold processors. But to sell the processors, and not have them be free samples as had been the traditional case in the semiconductor industry to give away gates, flip flops and so forth, we said, "Hey, this is not a gate. This is not a flip flop. It's a complete system that we're making available to our customers." With that pricing, we got a fairly large number of customers to enlarge, in this case, being tens initially, and then into hundreds.

Using the 8008, and using the 4004, as development vehicles, we then developed the sets of boards to go around them, so that there would be rudimentary, at least development or proof of concept type systems that customers could work with. Ultimately that led into the development system business. And again, I'll refer to Ed to go through the history of that, because I kind of look at Ed's contribution as really being the father of really making the microprocessor happen at Intel, because he probably had one of the few marketing groups that I've ever seen that actually had a manufacturing function under it, and was doing development systems, and doing a whole variety of tools and things to make a product to the point where it could be very saleable.

Now in terms of getting feedback from the customers, if you recall with the 4004 and the 8008, both of those were done as custom processors for a single company. All of the initial design influence, design feedback and whatever, came from Busicom in the case of the 4004, and from Computer Terminal in the case of the 8008. So there was not a widespread look from the point of view of the customers. As we moved into the potential or possibility of the 8080 in the early days, after the 8008 was announced, as I said, we had hundreds of customers. Going back to some notes that Stan provided, we then took the 8080 spec and were able to put the 8080 spec, under non–disclosure, into the hands of over 400 customers, and got feedback from about 25 percent of those customers, in terms of their view of this new product that we had coming along. Now you've got to understand that customers are very, very effective at critiquing a product and looking at the warts on it, telling us where some of the minor problems were. Customers overall have been helpful, but not necessarily that visionary, on where a product line should go. So the visionary work that Federico, Stan and Ted did, in terms of putting the spec together, was very important, but getting the feedback from the customer was important also.

Probably the most important feedback we got was that which Federico alluded to. Customers didn't want to see change. They wanted to see compatibility with what they were doing, and that had to be compatibility with their knowledge, compatibility with software that they'd written, and just an overall thinking process. When the 8080 was announced, it was compatible with things we'd already done on the 8008.

Also, tying in with Steve's comments about peripheral chips, the one concern we had about the 8008 when it was announced was it really was naked. It had all the sea of TTL around it, whereas with 4004 we had a special ROM, a special RAM, a special I/O with the 4003. So we went to work more from a marketing perspective and looking how we could make up kits of parts that would be dedicated to the 8-bit microprocessor. And we selected special RAM memories, special EPROM memories, that could go with the microprocessor and support that. We worked with the bipolar engineering group and got some special chips—the 8212 and the 8214—to support the 8008.

That was kind of the genesis then for doing other special parts that could go around the 8080 when it was available to make it a complete product, a complete system that the customer could use. From a very selfish point of view, it helped lock the customer activity into the Intel product line. As we were looking at the competitors, and in particular Motorola, yes, we had a microprocessor that was very fine in performance, but also we had the development systems, we had the special memory chips, we had special peripheral devices that would go with this microprocessor and tie it all together, in addition basically to all of the user's manuals and other tools we had to provide for the customers. Now the significance of the 8080 at announcement time, there was a complete development system available for the 8080, a development system for both hardware and the software tools to go with it. This was the group that was under Ed's wing, with Bob Garrow doing the hardware for the development system, and Terry Oppendike doing the software for the development system, so that—

Faggin: And Hank Smith leading…

Feeney: Hank Smith was leading the program up until shortly before the announcement of the 8080. As an example–let's go through these, Dave. That's good. This is the first-–we can put this on the tape. This is the first brochure that was available for the announcement of the 8080 on announcement day. In the brochure, we showed that not only did we have the processor itself, but we had the memories, and in this case, we finally had magnetic tapes. When we did the 8008, we had rolls of paper tape that we were showing our customers. This later evolved then into actually fully integrated development systems. So the tools were here, and I noted that we set the announcement price of the 8008 at $120, when it was announced.

When the 8080 was announced—and Ed and I have different recollections, but we essentially got to the same point on this—the 8080 was initially sold for $360 for a single chip, and customers were more than willing to pay that for the flexibility that it gave them, and for the enhancement. They were ready to do it on the day that the 8080 was announced because they already had an initiation to Intel microprocessors with the 8008 product. Let me let Ed go on with his recollections about some of this, and then I'd like to come back talk about customer vignettes, because I've got one story about CDC that might be interesting.

Gelbach: One of the things that became obvious after a while is that I had seen at TI the evolution from DTL to TTL and as almost the logical next step, it was then the 4000 series. People looked at that not as much as a computer, but as a replacement for a lot of TTL. In the era that I joined Intel, there really was nothing like a microprocessor. I had had some experience. I used to call on CTC when I was at TI, so I was somewhat aware of what TI was doing. But what it ended up with was that, as we went from the 4000 series to the 8008 that was seen as the first move towards getting out of the just TTL. Then as we went to the 8080, that was sellable as the first computer on a chip.

Now the computer purists didn't believe it, but it was a great catchword for people that were in the industry, and especially young designers that wanted to get in the forefront of what was going on. Intel tells them, "Hey, we have a computer on a chip," and it takes them out of the realm of being a circuit designer, to becoming a computer designer. So the enthusiasm with which the industry picked it up was phenomenal.

Now one of the other things that we began to realize is that there were no tools associated with this. No one had any idea how to test it, how to make it work in a system. So it just became a logical concept that if we could supply the tools with it, we could charge a lot for them, number one. We could control the design, number two. It would bring the product to market faster. So we set up a group, solely because nobody else at Intel was interested in doing it. It really didn't belong under marketing or sales, but it was kind of a fun experiment. Since most of us didn't have any experience in it, we couldn't fail, because nobody else was willing to critique it even.

Feeney: Everybody else was willing to critique it!

<laughter>

Gelbach: We gave it to a bunch of young design engineers. They weren't salespeople particularly; they were designers and they did just a hell of a job, first of all, to put Intel into a whole new dynamic marketplace, and, on top of it, made a lot of money for the company. It got us out of being just—we go from a 1028 memory to a different--

Faggin: 4096.

Gelbach: No, 1102. It got us out of just being in the linear market of progressing products that became multiplied by each other. One of the other ideas that the group came up with was then, "Let's combine all the parts and put our own part numbers on them, so that other people can't compete directly. But we can name them with our part number, match the performance all around." To some extent we avoided a head on competitive deal with the memories and the EPROMs and other areas.

So it was a period where you could do a lot of really unique things, and get a lot of attention. The customers loved us. It was the golden area of, "We are more than willing to talk to you." Having been a salesman, one of the most frustrating things is calling on customers cold, when people are kind of not interested in talking to you. Intel got in every technology company, every university. Everybody wanted to talk to the Intel people. It turned out to be one of the great inventions or design successes in the industry.

House: So for the development system business, if we were trying to trace its origins, you'd have to go back to the things that were happening with Ted and Stan, with the words for the 4004.

Hoff: We started out with these little sim[ulation] boards that allowed you to prototype a design. I had some discussions with marketing, because we figured these would be useful to the customers, and they argued that they should just give them away. I was very much against that because I felt that once you started doing that, in other words, the next thing you were going to say was, "Well, we're losing money and we've got to stop giving them away." I figured, we could sell these at a profit, and it would still be cheaper than it would cost the customer to do his own prototyping.

So that is where we started, and we did a little EPROM programmer module that could be put on it so you could do that part of it. Then responsibility was transferred to marketing, and they just it picked up and ran with it. It struck me, one of the things that must have helped is that once a customer makes a commitment to buy the development system, in other words, think of this engineer in this other company. He'd better use the Intel microprocessor to justify the money he spent on that development system. So it really locks him into the Intel line once he's made the commitment to the development system. Generally they were expensive enough to require the engineer who was going to buy one to get his management's approval.

House: Does anyone recall how the evolution happened with the sim cards? I know Hank Smith was involved, and Bill Davidow came along, and then Bob Garrow and then Terry Oppendike on the software—

Feeney: The sim card was there initially.

House: An interesting story. I guess it really got started, Ted, with your group and some of the early sim boards, and then evolved to be a business that was bigger than the microprocessor business in a few years. So maybe you could comment?

Hoff: Well we started off doing the sim boards as a way to do prototyping of systems with the microprocessors, and it seemed they should be reasonably useful to customers to get developments going. With our connections to marketing, they seemed—initially there was a feeling that we should give them away, but after a while I think we came to an agreement that that was not a good idea.

And so I'm not sure who in marketing really became the main promoter. I think probably it was Hank Smith, but there may have been others. Essentially they took them over and then grew it from just a few simulator boards and some other modules we had for programming EPROMs and so on, and actually started coming out with full blown boxes with software support and all sorts of other things to go with it. Maybe someone else knows.

Mazor: I have a comment on that too. You know, we had a large scale time sharing machine in–house that happened to be a PDP-10, and to support software development we had an assembler for the 8008 that would run on the large scale time sharing machine, and that's called a cross assembler when you do that. So we contracted that out and that program was written in Fortran.

Then just another piece of background, there was a company in Pacific Grove called Digital Research. You might know the name Gary Kildall. He developed a programming language for the 8008. It wasn't that high a level of language, but we called it a higher level language, PLM-8: programming language for microprocessor, 8 bit. And then he developed PLM-80, and that was ready about in the same timeframe. So again, if you were writing in PLM, you didn't really care much about the assembler language or the differences between the 8008 and 8080, because you were writing in a higher level language.

We traded a hard disk drive to Gary for the software initially, and given that he had a hard disk drive, and if you will, a sim board, he developed an operating system that was called CPM, control program for microprocessors. CPM was a lead product of Digital Research. It became very popular as an operating system, especially CPM-80. For those that know the story, that eventually more or less became DOS, the first operating system on the PC, so it was pretty significant. So we had quite a bit going for our customer base in terms of writing either assembler language programs or in a pseudo higher level language for developing their software for their applications.

In addition, on the sim boards, we actually had assemblers that would run on those little boards. Now it meant jockeying around some EPROMs and plugging them in, but we actually had some lower level tools at the beginning, using a teletype and paper tape readers. And then when we came out with the Intellec® boxes, again it was a development system, so our customers could write their software, assemble it, burn an EPROM and plug it in, those EPROMs, into their prototyping board. So we certainly had not neglected that side of the business, and the fact that we ourselves were computer users, we were not naïve about the requirements for doing that.

Just a few other comments about 8080, very briefly. In 1964, I programmed the 360 [IBM 360]; 360 had 16 general purpose registers, and it was what we'd call a symmetric architecture. You could do anything with any of the registers. However, the IBM operating system programmers immediately specialized at least four of the registers, which means the programmer can't touch those registers because they're used in interrupts. So when we were working on the 8080 architecture, we were confronted with a similar issue.

The architecture of the 8008, we did it for Computer Terminals–Datapoint, more or less had symmetrical registers–H and L being a little bit more specialized, and we were confronted with the issue of having a more symmetric instruction set, or doing something on the cheap, which was specialized. Our choice was to make the H and L register, very special, the D and E, not quite as good, and then the B and C registers of the 8080, poorer. So we certainly did not offer a symmetric register set, and I didn't feel too bad about that in light of what even the 360 presented finally to the programmer. As an example specifically, Ted came up with a very clever idea which allowed the exchanging of the H,L and D,E registers by simply using a flip flop that toggled renamed which register was which. Now you couldn't exchange the H,L with B,C or you couldn't exchange B,C with D,E, so this was an example of a particular instruction, but it was implemented with a very low cost of a single flip flop.

I would say that one of the important decisions that we made was to provide 16-bit increment, 16 bit decrement, 16 bit add and subtract, and in fact about 18 16 bit operations on the 8080, although it was called an 8 bit machine. In contrast to that, if you looked at HP and other machines at the time, they typically had an index register, and the way you did addressing of arrays was by moving an index register along. We specifically decided <u>not</u> to put in an index registry because in the spirit of the 8008 the H, L was the address mechanism and you could do explicit addressing arithmetic. We preferred to give a general capability. Now it's interesting that when the 6800 came along from Motorola, they did exactly the opposite. They had an index register, which was very nice. You could <u>not</u> move it by anything except one. And also, you could <u>not</u> do 16 bit arithmetic, so they made the opposite decision, and of course, sometimes we're maligned at Intel for not having an index register and it's something we did purposely for giving more capabilities. I think the 8080 reflected a collection of interesting instructions that also reflected a compromise in the hardware that made it buildable and I again, compliment Shima for building it very well.

House: Back to the development systems. After some initial work in Ted's group, I know Hank Smith was involved and not too long after that Bill Davidow came along. Did Bill work for you, Ed?

Gelbach: Yes. Bill had worked at Silicon Systems. It was a division of Signetics, and had been doing some compiler work or some things along that line. Hank, who had started the microprocessor unit, wanted to move back east and went to work for Venrock. I hired Bill and he came in, and at about that time, we did start working on the development systems and selling the development systems.

Feeney: Moving along with that, with the work that Ted's group initially did we latched onto that as early as we could and were putting the circuit diagrams, how to use, examples of interface with the teletype and whatever in our user's manuals for a twofold reason. One was to bulk them up a little bit, but the other was to really give our customers an idea of an approach, or at least one approach or one thought process, for getting into using the microprocessors. And what happened with it is, that I think at some point we even gave our customers a copy of the layouts for the printed circuit boards. But the customers—again, this is all customer driven—they wanted to have us make them, us put it together for them, and I believe it was Norm Shanks and Andy Volkhart out of your group that were building some of these, and eventually they migrated over from your group to work for Hank Smith in the development system area, and mainly because the PROM programmer was available.

The PROM programmer became a great demonstration tool, but it also helped to sell EPROMs. That was probably one of the biggest convergences of product available at the 8008–4004 time, was to have the EPROM programmability there, and so by the time the 8080 came out, all of these things were at a much, much higher level, where the entire package was available at the time the 8080 was announced, as that one photograph showed. We had the Intellec–8 development system, and we had an Intellect–8 MOD–8, which was the 8008 card. You could pull that out and plug in an Intellect MOD–80, just plug a new CPU card in the box and go from 8008 development to 8080 development by the change of a card and by reloading some software. So again, each step along the way was raising the bar, and this continued throughout the time of the 8080 and then beyond to just build a larger and larger barrier for customers. One of the other-—

House: Barrier for competitors?

Feeney: Barrier for competitors, yes.

<overlapping conversation>

House: That must have been formidable: to compete against that stack of products.

Feeney: And then going one beyond that, looking at the simplistic side of it, the development systems got up to be $25,000 or $30,000 with the CRTs, with the floppy disks on them.

House: ICE, in–circuit emulation.

Feeney: With ICE, and so forth and so on. And in my group, Ken McKenzie was the product engineer at the introduction of the 8080, and about a year after the introduction of the 8080, he developed a thing called the SDK kits, which basically was a Heathkit–like card, where you could buy a printed circuit card, some parts that were shrink wrapped, a manual and a few other things, and go put together your own 8080 card, solder it up, and customers were buying those like hotcakes. Some of them were buying them just to get access, to get the 8080, because it was at times through that period on allocation. But at the same time, it was an inexpensive way in, about $1,000, to get a completely functional 8080 based card.

House: What was the transition between ISIS®, which was the operating system in the development systems, and CPM? What was that relationship?

Mazor: Well the key thing was, you had to have a floppy disk, which we had. It turns out, we had our own operating system programmers and that's what ISIS was. It was very similar. As a sideline, both Gary Kildall and our operating system group more or less used the interface of the PDP-10, which was called TOPS, so the commands that you might be familiar with, like “copy”, “delete” and “file” and so on were in common. They didn't look that different from the user's point of view, but the code was done entirely separately.

House: Separate implementations.

Mazor: Yes.

House: Do we have anything else that anyone wants to add to the story of the 8080?

Mazor: I'll just make one point. In the op code space, there were all but 12 operation codes unused, so it left a little bit of room for the 8085. When that group came along, they made an enhancement to the 8080, again, another midlife kicker chip. So we used up almost all of the op code space, but not all of it.

Faggin: You left some for the Z80.

<laughter>

House: Or maybe the Z80 used them before the 8085?

Faggin: Yes, actually that's true.

House: I think the sequence may have been the other way around.

Gelbach: Intel also championed one thing at this point. I had started a program on application engineers, which originally came along to do a lot of the memory work, etc. But as we started to transition more towards microprocessors, we started to hire people that were experienced in the computer end of the business, or understood the microprocessor. And it was considered at that point a really exceptional job opportunity for people. I remember going to colleges and we used to get more applicants for that type position than anything else. But I think that also gave us an edge that we were willing to put money into this application group. I think you headed that for a while when you first came in, didn't you, from that viewpoint? But we were willing to structure an organization—from a semiconductor company that was unique, that all they did was work on customer applications, help them design things, in return, obviously, for us winning the design.

Mazor: As a minor part related to that, Rockwell had the alternative approach that when they were using the PPS4, they had decided, "Don't try to teach the customer how to use it," but they had a small design team. They would move into the customer, solve the problem for the customer, write the code, win the design and then go on. Our approach was almost the opposite, which was having, you know, a broad capability to educate, a broad capability to support fairly small customers with the idea that they're going to write their own code and we're there to help them, but certainly not to do their design. It's amusing that the companies took exactly the opposite approach.

Feeney: Didn't you get a multiplying effect from that? Because I know your career later on went into training, Stan. Didn't you get more of a multiplying effect from going out and teaching the customers how to do it, so that they could do one design and then move on to another and another, as opposed to just taxing corporate resources?

Mazor: Well, I did work in the training department for a number of years, and we trained an unbelievably large number of people. It's interesting, if you look at the general gestation, that Intel took the responsibility of educating the public on programming these machines, and providing development tools. About five years later, these kinds of things would show up in junior colleges and in colleges, and about four, five years after that, there were plenty of ways of learning about this product line in the public domain, and Intel no longer had to carry that burden. But in the early years, they certainly did and succeeded at it. That was another small, little business opportunity.

Gelbach: We used to charge for people going in and helping the customers, where we thought we could charge at different rates. We also had our distributors set up programs where we'd go through distribution, and they would sell the applications engineers, time–wise. So we made money from many of the different sources.

Mazor: Just on that point, I remember specifically Grove being very adamant at one point, and that's when we came out with new products. Some distributor wasn't training his own field application engineers on the new products, and Grove wanted to withdraw a discount to them, because he wanted them to continue to make the investment. I think that also added fuel to the fire.

House: Or maybe that was Ed.

Mazor: Thank you.

House: Anything else that anyone wants to add?

Hoff: One thing. I think Ed touched on it earlier: the business of recruiting young people to come to a company. Before Intel was in the microprocessor business, the memory business I think was looked at as not all that exciting. We talked about computers—it was always how big a computer you get to work with. I had presumably bright young college kids essentially walk out of an interview. They weren't interested in coming to a company like Intel, because they wouldn't have access to the kind of computer they wanted to work with. Once the microprocessor was discovered by the media, it was almost the opposite. Now they came to us, saying they'd love to have a job with a company like Intel. So it had an advantage, essentially, in attracting the kind of talent that will generate the next generation of products.

Feeney: That leads to the comments that we've had the last couple of days about the tools that Shima didn't have when he was doing the development of the 8080, and the fact that by developing the computer chips and having them embodied in systems, it's been somewhat of a snowball effect ever since in terms of getting the proper simulation tools, the proper design tools, the proper testing tools and everything else that are required for these systems. So it's been kind of a snowball effect in many ways with each generation.

Mazor: I have a question for Ed. I don't know if you want to answer this, but Jim Lally was product line manager of the MDS systems and he believed that a market to engineers was an inelastic market, and under that model, the goal was to raise the average selling price, and we kept adding features that made the development system more expensive, such as in–circuit emulators and multiple disks, so we could get a system up to about $30,000.

Now I've raised the following question in my mind: had he assumed the opposite, that it was an elastic market, and had we been able to reduce the price of the development system, make it out of plastic, and hammer the price down, would we perhaps have found ourselves facing a different marketplace than the engineer?

Gelbach: My just kind of cursory answer to that is "No." Because I don't think the price, the selling price of the development system was ever a limit to anybody that wanted to design anything. More, I think if somebody didn't have the money and they were interested in designing something, they did it without a development system.

Mazor: Let me take it a step further. The difference between a personal computer and a development system is the price. So I'm not thinking about people designing computers at all. I'm thinking about people who decided, "I want to use this machine for something else." Now granted, when we talk about the personal computer, the first things that comes to mind are the killer applications, one of which is the spreadsheet, and the other is word processing.

Feeney: But some of those were on the development system.

Mazor: Well, go ahead. Speak to that then.

House: They were on later. I remember the PDS, the personal development system, the luggable one.

Feeney: Well, WordStar was on there too.

House: Right. We sold that one, actually with CPM as an option, ISIS and CPM, and we sold a spreadsheet and a word processor, and a few other packages, but it was too late. The PC was already out of the bag then.

Faggin: But it's a completely different customer base. In one case you sell to engineers, in other cases you sell to common folks. Intel was not—

<overlapping conversation>

House: One can speculate what might have happened if in fact we had brought the price down and we had put a word processor and a spreadsheet, because I think there were Display Writers and Wang word processors that did better word processing. The thing that made the Apple take off was VisiCalc, because it had a word processor that was kind of okay, and it had a spreadsheet, which is something you couldn't get on a Wang word processor.

Mazor: But don't forget, even in our own customer base, we had engineers who were abusing their development system and using it for these other things. Mind you, we're not calling on "John Q. Public," by any means.

Hoff: Although I have to say, we had a number of people in marketing who were buying Apple computers to run spreadsheets and do things, rather than borrow an Intel development system for that kind of purpose.

House: Because VisiCalc ran on that machine.

Hoff: Yeah, yeah.

Gelbach: I'm not sure that if we'd dropped the price dramatically, it would have helped the microprocessor business, because I think the people that were interested in the microprocessor business, and had the resources, would buy the development system, whether it was $15,000 or $30,000. If they were interested in the other aspects, spreadsheets and word processing, would they have looked at the microprocessor? Why wouldn't they have gone to the Altair or Commodore?

Faggin: Also, it was a completely different set of people that started that industry, the PC industry, and they were kids. They had actually no idea what they were getting into, but the saw the potential of the computer in people's hands. Intel, including myself, I didn't see a lot of application for a personal computers early on, because there was very little to go for.

<overlapping conversation>

Hoff: The peripherals were a tremendous limit. I mean, the cost of even a floppy disc or a printer, they were outrageous compared to the cost—and the cost of memory kept you from running the big applications that we take for granted today.

Feeney: Memory on the first IBM PC was $500 for an add on of—

Hoff: For a PC alone I think I paid $4,000.

<overlapping conversation>

Hoff: A basic unit was 64K.

Feeney: But I think that does account for one thing, Ted and Stan, with the development system. At the time that the PC became very pervasive, it was at that time that the development system business at Intel dropped off, because suddenly now there was a “low cost” PC of $5,000 that did displace a $25,000 development system.

House: And people started coming out with tools.

Feeney: And the tools ran on the PC.

Bisset: And there were low capital cost alternatives. When we did the tester for the 8080 [at Megatest], and we didn’t have any money, we rented time on some computer at Kansas City, and we downloaded and stored our programs on paper tape and fed it in to a teletype and programmed it in PLM.

Feeney: So a time share?

House: Time sharing a development system?

Bisset: No, time sharing, I think it was a PDP–10 in Kansas City to compile PLM programs so that we could run them on our 8080. We wanted to test the 8080, but there was no capital cost there, so I don't know if it was something that was done significantly, but it was certainly marketed by the time share people.

Hoff: One of the significant factors, I would think, on the effect of the development systems was the fact that the IBM PC had an open architecture, and that IBM published so much information about how to do interfaces, made it available, and so many different companies started offering different kinds of modules to go into it. It always seemed like Apple kept their designs more proprietary, and—

Faggin: On the Apple II, though, that was what made Apple II succeed. They started doing that with the Macintosh. And in fact, that initially made the Macintosh not as attractive, and eventually they opened it up again, and that made the Macintosh take off.

Hoff: But the availability of so many options on the PC, and the fact that it used an Intel processor, really made it probably compete with the development system.

House: Certainly this started a series of events that turned out fairly well for Intel, whether getting in the PC business would have been good or not. But we do know that the alternative that they took worked extremely well for the company, and allowed them to harvest the profits of a very large industry.

Bisset: I think Intel had a higher profit margin than most of the PC makers.

House: Any of the PC makers. In fact, they made more profits than all of the PC manufacturers put together. In fact, it's not clear to me that PC manufacturers actually made a positive profit.

Feeney: And Intel, in the space that Intel is in, its revenues today are about half of its four or five largest customers. And as a result, it's probably captured far more value than its customers can capture.

Hoff: And it also is in the situation that Intel is a major manufacturer of a good part of what goes into the PCs. In other words, manufacturing essentially complete boards with all the logic on them, so the PC manufacturer buys a board from Intel and sticks it in the box with the power supply.

Mazor: Just touching on another issue. One of our competitors and a very good MOS house in Silicon Valley was AMI. They were known for doing custom chips. They had a lot of designers and could turn out chips very quickly. When they looked at the 8008, they thought we had done a pretty dumb job and so they decided to design a very elegant chip called the 7200. I talked to someone at AMI who said that when they finished the design, they didn't have a table large enough to hold the masks, and ultimately they were never able to produce a chip because it was too large. I think one of the things that we did, probably our focus was on being chip manufacturers and making good chips at good yield, rather than looking at top down and saying, "Can we build the best computer?" Maybe it won't fit on a chip.

Hoff: I think that I would also go to say, you can look back and say, "Boy, did we slow down things," or whatever. We were too conservative, or "16, 18 lead packages for the 4004 and 8008." But had we gone too aggressively in the other direction, we easily could have gone into something that we couldn't have been successful at even producing, and there the whole thing would have blown up. I don't feel badly that we may have been overly conservative.

Faggin: I don't think the 16 to 40 was a problem, Ted. I mean, everybody else was doing 40. I don't take that point.

<overlapping conversation>

Shima: I want to concentrate on 8080s. It's too much, too far away from 8080s. I want to finish with my unique memories.

House: Yes, please.

Shima: I became the father of twin babies on the next day of tape out. And that means I became a father of three babies on about the same day. But unfortunately, I lost one week of vacation. It was a moment in my life and I got excited, and a cheerful day came to me. That day was February 13, 1974. I got big support from Federico and also Ungermann to announce the 8080 to IEEE's ISSCC at Philadelphia. There was so many audience, huge audience. It was evaluated quite high. Since then, so many visitors came to Intel, and also Dr. Noyce brought many, many people to the lab. Then he said, "Why don't you explain? Why don't you demonstrate 8080?" And it looked like everybody became happy. Then I heard that Intel began to be convinced of the microprocessor business. After I completed all of the 8080 related jobs in July 1974, and I became, let's say, like a big brother, or I shall say a substantial person, in charge of the development of five kinds of peripheral chips. I was getting much busier, but I was quite happy to work with Bisset. He did an excellent job for the production testers and also quite unique peripherals, their functional specifications.

House: Thank you.

Faggin: Maybe you could acknowledge the draftsmen that were used on the layout.

Shima: Sure. I just forgot the names, but many thanks to everybody, yes.

Mazor: Shima, I'd like to ask Federico a question. Federico, you were working in a memory company, and the management often had the attitude of "one man and one chip", and of course memories are often—you know, you design the bit and there's a lot of step and repeat in terms of the design. Did you feel handicapped? And again, back to the packaging a little bit, that we were overly looking at it?

Faggin: There were definitely restrictions in the number of resources that I had available, no question about that. So if I had asked Vadasz for two engineers for a product, he would have said "No." It was as simple as that. I had to make do with what I had. Now of course, the 8080 was toward the end of the run for that. In other words, any new product would require more than one to one person. Also, it was difficult to have one person with all these skills that were required to design a chip, particularly if you had to define the chip, do the logic, do the circuit, do the layout, do the testing. I mean, they were getting more and more complex and more and more—

Mazor: Specialized.

Faggin: More specializations were required from people to do a good job.

Shima: Year after year, logic simulation had to be done. Also, one more difficult job, it was test vector generations. Huge amount of time. The success of Intel—Intel had good test vectors. Then you can move to 286, 386, and 486. Therefore test vector became quite important.

Faggin: Yes, it's an exponential problem almost, right? So in fact, ten years later, you needed a team of 30, 40 people just to do the CPU. Nowadays it's probably several hundred people.

House: Hundreds, yes, hundreds.

Hoff: Another area too is that in the embedded area, the addition of things like analog to digital and digital to analog conversion, interface to the real world problems, also adds additional complexity and different kinds of skills on the part of the engineering team.

Faggin: But certainly I could have used more than one engineer per chip in those days. So if that is the question, definitely we could have done things faster, had I had more resources. But it was inconceivable to ask for two engineers for a chip.

Mazor: Let's talk about the 8008 and Hal, maybe you'd like to join in this as well. If we look at the TI design, we can see that they used what we would consider pretty sloppy techniques. And in fact, it may have been as a result of using some automation, or some throwing some blocks together. So they were not quite as concerned with the total die area being occupied as well as the density of that area.

Now as we know, one of the issues with die density is "What's the probability that a defect is going to hurt something?" And when you're very dense, the chances are very good. If you have a non-dense die, as the TI chip was, then a defect there isn't going to hurt you. But it allows you to get to market faster, so there is a chain and a trade–off there, of getting to market faster with say a little larger chip and not as efficient. Again, I think as I interpret it, the designers at Intel, having been focusing on memory design, were quite concerned about packing things in, getting the density and instead, let's say, getting to market, possibly a little later.

Feeney: Well, the greatest concern at Intel was just getting it inside of the package, because we had that constraint and that just limited the maximum size of whatever we could do. And again, we were doing this for a single customer. We were not doing it for a mass market, and that's the major switch between the 8008 and the 8080. The 8080 had the benefit of standing on something's shoulders to get out there and to get to market. That's where you end up with, as we saw with this brochure, you've got a complete package of stuff that goes to market, as opposed to a naked 18 pin chip.

Faggin: Another point is that the automated design system that existed, like Micro Mosaic of Fairchild in those days. They were not even close to the density that you could do, even with the packed technology, metal gate to metal gate. So to make say 4004 with that technique, it would have been six, seven times larger. So it couldn't even be done. It wasn't an issue of doing it faster with an automated system.

Shima: Yes, at the development time of 4004, I contacted with Fairchild how much size it will be. It was quite big.

Faggin: Yes, it was not possible.

Hoff: I think one of the things too to look at, both the 8008 and 4004 were in relatively competitive situations. The calculator market was one that was considered very competitive. We had very aggressive cost targets, presumably, especially for a start up company that really didn't know what it was doing, and with an untested process and so on. Even the 8008 presumably was replacing something that could be built with, what was it? Sixty packages of TTL, which what? You paid 20 cents a package or something?

Feeney: Well it was about $1 at the time things started and fell to 20 cents by the time—in that 1971 timeframe is where the real steep drop in TTL was.

Hoff: I used to hear numbers quoted, like "Figure a buck a package, by the time it's on the circuit board and installed." So that would be like maybe 60 bucks and you're hoping to sell the processor chip for $60, and it takes a bunch of TTL to go with it to finish off the job.

Shima: The performance is quite important. If we are able to reduce, like ten percentage of clock speed, we can go. But suppose we have to keep two megahertz or 2.2 megahertz, or 2.5 megahertz, we cannot utilize automatic machine.

Mazor: By the way, I didn't mean Micro mosaic or something like that, but if you look at the TI layout for their version of the 8008, it reflects a different approach.

Faggin: No, but that was hand done like anything else, except that was metal gate. And the metal gate, one layer interconnect is a loose design.

Feeney: And they have a 24 pin package to drop it into.

Shima: I love the silicon gate technology, because of high density.

Faggin: Yeah, I mean, basically you can pack everything in, particularly with the buried contact and two layer interconnections. You could do half the size for the same attention to the layout.

Shima: Yeah, because all of logic was laid down under the metal. That's quite a big benefit.**House**: So we're coming to the end of another tape and I think we're coming to the end of our session here. I want to thank everybody. Obviously, this was a product that changed the world, and this is the team that made it happen.

END OF INTERVIEW

