Oral History Panel on the Development and Promotion of the Intel 4004 Microprocessor

Interviewees:
Federico Faggin
Hal Feeney
Ted Hoff
Stan Mazor
Masatoshi Shima

Interviewed by:
Dave House

Edited by David Laws

Recorded: April 25, 2007
Mountain View, California

CHM Reference number: X4024.2007

Dave House: Welcome to the video history of the MCS-4, [also known as] 4004 microprocessor development. I joined Intel in February of 1974, so I was not at Intel at the time of the 4004, but later worked with each of these individuals in one way or another. So what I'd like to do is start out, just go through the table here, and have everyone give their background up to the point where they started working on this product. So, Stan, could you start?

Stanley Mazor: Thank you, Dave. My name is Stanley Mazor. I was raised in Oakland, California, and attended Oakland public high schools. I went to San Francisco State in 1960, and was introduced to a computer for the first time during the years 1960-1962. I took the two computer courses they had in programming. There was no computer science at San Francisco State in those days. And one of the projects that I had in school was to write an interpreter for a virtual machine called IPL-V. The IBM 1620 was a decimal machine and was to affect my career and some of the contributions I made in subsequent years. I was a lab assistant at the computer center at San Francisco State, and left in 1963 to work at the University of California, also as a programmer on a 1620, and that was the fanciest 1620 in the United States, having many special features. During that time, I had a chance to visit Stanford to meet with Ted Hoff, who probably had the second fanciest 1620 in the United States, and you could say we had a little bit of club of 1620 users. In 1964, I joined Fairchild, and I was at Fairchild Semiconductor in Mountain View from 1964 to 1969. For several of those years, I worked as a computer designer and logic design at Fairchild R&D. We had a project that was based upon Moore's Law, and the fact that we thought hardware would become free and that was to develop a very large-scale, time-sharing computer, which also had decimal arithmetic in it. And then I left Fairchild to join Ted at Intel in September of 1964.

House: Ted?

Ted Hoff: My name is Marcian Edward Hoff, Junior -- but everybody calls me "Ted." I was born in Rochester, New York, and I did my undergraduate college work at Rensselaer Polytechnic Institute in Troy, New York. And while I was going to college, I had a summer job at a company in Rochester, General Railway Signal Company that gave me the opportunity to work with transistors, and even a little bit of core memory. I came out to Stanford to do graduate work, and got a PhD in 1962, and worked in the area of neural networks. Some of that work involved using the IBM 1620 computer. That was really my first computer experience. Later, we replaced it with an IBM 1130, and I continued working, after I got my degree, as a research associate. And then I was approached by Bob Noyce, who told me he was starting a company… I had been interested in where integrated circuits were going, and had actually done some work at Stanford at least looking at integrated circuits and so I felt that the next area was memory. When Noyce interviewed me, he asked me where I though the company should go, and I said, "Memory," and it turned out that's where Intel was going. So I left Stanford, and joined Intel in 1968. My job there was called Manager of Applications Research. It was mostly to look at how to use the semiconductor memories, how to help customers use them, and to also help define what the products should be. So that's how I got started. Intel was going to do this development of what were considered proprietary products that would take some time to design in, to persuade computer companies to switch from the core memories they used at the time, and so Intel was open to the idea of doing some custom work. Our first custom project was to be done for a company we knew as Electro Technical Industries. And that's where I first met Mr. Shima.

House: Thank you, Ted. Shima San, could you give us an update on your background before you started working on this product?

Masatoshi Shima: My name is Masatoshi Shima. I was born in Shizuoka in Japan. I graduated from the Department of Chemistry at Tohoku University in Japan in 1967. In April, 1967, I joined the Busicom Corporation. I started to work on a program at the computer division, and I learned several programming languages, such as Fortran, COBOL, and assembly language on Mitsubishi’s computers, MELCOM 3100. And six months later in October, 1967, I moved to the desktop calculator division, and developed a desktop calculator that was implemented by hardwired TTL logic. The design of a printed circuit board became quite useful for the design of the 4004. In those days, all [companies] in business of desktop calculator grew up very rapidly in Japan, and Busicom started to look for the new logic design methodology, and methods that would be suitable for OEM business. Then in November, 1968, I developed new desktop calculator with printer. In that development I introduced the program logic method using the decimal computer and read-only memory for program storage. The data types were n-digits decimal data and five-bits binary data.

House: Thank you! Federico!

Federico Faggin: Yes, my name is Federico Faggin. I was born in Vicenza, Italy, where I went to a technical high school. So by the time I was 18, I graduated. And my first job was at Olivetti in the R&D laboratory near Milan. And there, I was fortunate enough to co-design a small, experimental computer, and build it. I had four technicians working for me. We built this machine-- several hundred printed circuit boards, 4k core memory with 12-bit words. That was an invaluable experience for me understanding what was going on in the computer field. That computer, of course, used transistors in 1960/1961. Integrated circuits were not really available yet. And I used germanium transistors, which were very, very slow. And yet the computer was fully functional, and it was an interesting background for the 4004 later on. I then went to university, where I studied solid state physics, with a doctorate in physics. And I then worked for what today is called S.T. Micro -- in those days it was called SGS Fairchild. And I developed there process technology for MOS integrated circuits, which was the first time that S.T. Micro had that technology. I also designed a couple circuits using that technology. Then in early ’68, I came to this country working for Fairchild Semiconductor, in the R&D laboratory because of the connection between SGS Fairchild and Fairchild. There I was fortunate to co-develop -- I was the principal developer and the project leader-- silicon gate technology. , This was the first technology using self-aligned gates that really revolutionized MOS integrated circuits. Because with that technology one could have three to five times the speed of the older metal gate technology. The devices were smaller-- about anywhere from 30 to 50 percent smaller, and much more reliable. So that technology was used on a commercial integrated circuit that was introduced in late ’68, early ’69, called the Fairchild 3708. That was the first silicon gate circuit that was commercially available and sold. I designed the circuit and I actually build them in the R&D fab. So with that experience, I was the natural person to join Intel to lead the project of the 4004 in early 1970.

House: Hal?

Hal Feeney: My name is Hal Feeney, and I was born and grew up in Davenport, Iowa. I went to the University of Notre Dame for my college training, and it was at that point that I was introduced to computers while working on my electrical engineering degree. From Notre Dame, I went on to graduate school at Stanford, and in the summers I worked with Hughes Aircraft in Los Angeles, California. It was there that I was first introduced to MOS technology. At Hughes, most of the activity was on military-based systems, or satellite-based systems, but we were doing some designs, and I started working with MOS technology prior to the time that it became [available] for large-scale integration. There were gates, flip-flops, NAND gates, NOR gates, available in the 1965 timeframe. We were using those to build up an initial digital version of an analog computer, a digital differential analyzer [DDA]. And I was working on the test equipment for [the DDA] when I was first introduced to MOS. From there, I did additional graduate work after Stanford at Notre Dame, and then came to work for General Instrument Corporation in Los Angeles in 1968. GI was, at that time, the leader in MOS technology, although it was a very, very small business from a nationwide point of view. GI was doing some custom work and some standard chips in MOS technology with probably in the ballpark of 500 to 1,000 gates on a chip at the time. That was my introduction as a design engineer to doing individual chips. I did about three or four chips at GI before leaving and joining Intel in 1970. In 1970 Intel was starting work on the 8008, Intel's first 8-bit microprocessor, but due to some customer problems, we slowed down the activity and I spent time on several other projects - one was working with the testing activity on the MCS-4.

House: Thank you, Hal, and thank you everyone. So this all started out in Japan in the calculator business. So if we can ask Shima San to tell us what stimulated the decision to develop a general purpose LSI family, and how you came to select Intel to work with.

Shima: In 1969, Busicom decided to develop a general purpose LSI family that would be used for not only a desktop calculator, but also a business machine, such as a billing machine, cash register, and a teller machine. Also Busicom wanted to integrate a lot of function on LSI. Therefore, Busicom choose Intel as a co-developer, because Intel had already developed an advanced high-density and high-performance silicon gate, MOS process technology. The provisional agreement was concluded on April 28th in 1969. However, Busicom didn't disclose to Intel about the application of a business machine, because it was the confidential matter between Busicom and NCR Japan.

House: How many people were on this team in Japan, and who was leading the team? Tell us a little bit about that.

Shima: There were three engineers from Japan: the project manager, Mr. Masuda; also the senior design engineer, Mr. Takayama; and I was the youngest engineer.

House: How did you get involved in the project, and what was your role in the Busicom team?

Shima: In Busicom, my main role was to develop the decimal computer's system architecture, instruction set architecture, and application program. In the beginning, I planned to develop nine kinds of LSI, but later it changed to eight. Those were two LSI for decimal computer; ROM LSI for program storage; and shift register LSI for data storage. Then keyboard display controller LSI, printer controller LSI, and print buffer LSI. I. And then we visited Intel on June 20, 1969.

House: So when you presented your proposal to Intel, Intel saw some problems or had some other ideas. Can you tell us how you tried to address that problem?

Shima: When Busicom showed to the Intel all of schematic of general purpose LSI, it was not accepted too well, and we were told that there were too many kinds of LSI to be developed, and most of them are too complicated, and too many transistors. We had trouble. Therefore, in order to solve the trouble, I showed the decimal computer's system architecture, its instruction set, and desktop calculator's program. This idea was readily well accepted. However, Ted Hoff told me, "The decimal computer with micro level instruction set was difficult to implement, and also there are not enough flexibilities."

House: Okay, so now let's turn to Ted Hoff, and tell us about your experience of your first involvement with Busicom and their proposal.

Hoff: I had attended some of the meetings that led to the initial contract to develop the LSI, that were held in April of 1969. So I had an idea of what some of the cost targets were, but I had not seen any specific description of the LSI, and I did not see that until Mr. Shima and his team of engineers came over in June. In looking at the design that they were bringing to us, especially because Intel was relatively small, and there were relatively few people who were capable of doing MOS LSI design, I started to have some concerns that this may be more complex than we had expected. And another issue, if I remember correctly, some of the packages used relatively large numbers of pins for the time, and our silicon gate process would have to be in ceramic packages, because there were issues of the process and the impact of moisture on it. So we had concerns that that packages used for some of the initial proposals of that LSI would make it difficult or impossible to meet the cost targets. I expressed some of these concerns to Bob Noyce, and he said, "Well, if you have any ideas to simplify the design, go ahead and pursue them."

So starting with the design that Mr. Shima and his team had, I started looking to see if I could find some way to simplify it. There were a couple of things that came to mind. One was that the memory to be used in the original design was shift register memory, which was dynamic and used six transistors per bit. And Intel was starting to work on the three transistor dynamic RAM cel l-- that would be three transistors per bit. that would be one way to cut down on the transistor count. But the dynamic RAM seemed like it had another advantage in that it would allow you to simplify the control logic. In other words, if you work with relatively small chunks of information-- which you can do with a dynamic RAM, but it's much harder with a shift register-- that suggested that maybe there would be a way to simplify the instruction set itself. And one way to do that would be to go to a more primitive instruction set, and make more use of sub-routines, and then essentially write routines that would simulate the more complicated chip set. I started pursuing some of those ideas in the July and August timeframe of 1969.

Another idea that came to me was that the initial design used mostly binary-coded decimal arithmetic. It seemed that with a primitive instruction set, it would be possible to do binary arithmetic, and have a primitive instruction that would convert a binary result back to valid binary-coded decimal. So it would be possible to make a machine that was both binary and decimal arithmetic in its capabilities. I started to put together a proposed instruction set [and] checked to see that it could perform a number of the different calculator functions. In fact, it [also] looked like it would be possible to simulate some of the operations that were done by the other chips, like the keyboard controller chip, and the printer control chip. Around the first of September, Stan Mazor joined my group, and together we put together a crude form of target spec. Around the middle of September, our marketing department made a formal proposal back to the Japanese calculator company [Busicom], suggesting that they might want to consider looking at this other design. And if I remember correctly, we had a meeting somewhere around October of '69. Mr. Shima and his team presented their approach and Stan and I presented our approach to the management of Busicom, and they, at that point, said they really did like the Intel approach. So that looked like we were going to go ahead with our approach at that time.

House: Maybe we could ask Shima his reaction to the proposal that you made.

Shima: About the end of August, 1969, Ted Hoff came to our office. He started to explain about his proposal. And in his drawings there were three boxes, and 4-bit input port for keyboard. And those three boxes were 4-bit parallel arithmetic unit with accumulator and carry flag, and 16 sets of 4-bit general purpose register units. And four-levels of address stack, including a one program counter. And Hoff's proposal was a 4-bit binary computer instead of a decimal computer. I felt that Hoff's proposal was good, but if we accepted Hoff's proposal as it was, we had to do the project over again from the beginning. And also there were many difficulties at the time of evaluation of his proposal, because information was too little. But by mid-September Mr. Tanba of Busicom judged that Hoff's proposal looks more excellent than Busicom's proposal, because the micro-level instruction set architecture would [also] be the next generation's logical design method. As he used to be a computer hardware engineer, and in 1968, he had promoted the introduction of program logic method so in mid-September Mr. Tanba asked us to evaluate Hoff's proposal more deeply and carefully.

Mazor: Dave, can I come back to that?

House: Yes, go ahead, Stan.

Mazor: In September of '69 I had been designing a floating point unit at Fairchild, and I had programmed at Fairchild the 1620, the IBM 1401, IBM 360, SDS 920 and programmed in about four languages. Working on a high-level language computer was a very exciting project. Ted offered me a job to come to work on memory systems. And from working on one of the most exciting computers in the world to go to work on memory systems was like, "Okay, that's the end of your career, but you know, Intel's an exciting company." So I joined Ted with the idea that I was going to be working on memories and not super excited about the idea. But early on, Ted said to me, "What would you think about a computer chip?" And he asked me to go home and think about it overnight. Well, I'd been working on a giant computer, and it happened to be stack-oriented. And so his suggestion for me to think about it was completely out of my mind frame. And I came back and mentioned a few ideas having to do with a stack-based computer. But he was an excellent manager, because although at that time he had already conceived of this 4-bit processor [he did not want to] poison me in advance with his ideas. So I had very little to contribute on that day. Then he disclosed the 4-bit computer that he had contemplated. He had a separate timing chip, and I asked him, "Why is that timing chip separate?" Subsequently he thought, "Well, gee, maybe we could integrate it." So I was a bit of a sounding board, seeing it for the first time and asking questions about it. We were [located] at 365 Middlefield. It was an extremely small building, and my office had three desks in it. My officemates were Mr. Shima, and Mr. Takayama I was living with those gentlemen who were our customers and I was able to see their excellent flow charts. They had the design of a floating point unit, which is done to a large extent in program code. I also understood that they had a range of calculators that they were interested in including display and printer. One of the machines even had a card reader. I could see that they had some very grand goals for their project and they would ask very difficult questions. One that I remember well shows how expert Mr. Shima was on his product. He would ask me a question like, "How are we going to print in red when the negative result is computed?" Of course, he knew that that was a requirement of the calculator [while] we were working on the other hand on a very low-level 4-bit computer. So I was in the interesting position of trying to balance the needs that they had thought about quite a bit down to the details of printing, and our implementation. So my general role during that time was liaison in working with Ted and with Mr. Shima in finding the solutions to these problems.

House: Okay, so we got to the point where there is presentation made to the management, and the decision was made to go with the Intel design. I think it would be interesting to hear from Ted, and then from Shima-san about their experiences from that point on.

Hoff: We had the go-ahead I believe around October. My group, called Applications Research, was not considered a design group. The responsibility for chip design, particularly MOS chip design, fell under Les Vadasz. So we turned the design over to Les, and we did a little bit of, you might say, study on it to double-check to make sure that we were reasonably close on our estimates, with the help of some of the MOS design engineers. But neither Stan nor I had done chip design, and didn't feel that a project of this importance should be given to amateurs that we would represent. But it took Les quite a while to find somebody to do the next step in the project, which was the detailed circuit and layout. That didn't happen until the following April. So, we were in touch with Les during that time to see what was happening on the staffing. In the meantime, we did make a few changes. That's when we decided to integrate the timing chip with the CPU, and make it a true single-chip CPU. And we also started another project that came about from our success in selling this one. That was the project that led to the 8008. But there was also application information that we continued to develop on the 4004, and on the 8008. That is looking at these devices and seeing what we were going to need to do to be able to use them in actual applications.

House: So Shima-san, tell us your experience after the decision to go with the Intel design.

Shima: After the Ted Hoff idea came out, it took about four months to define all of the detailed product specifications. And I want to talk about what problems were there, and how we solved the problems together. Busicom team started to discuss with Ted Hoff about his proposal more deeply. And by the way, I wish to understand the difference between the idea and product. In other words, there is a big difference between the research and development. If the idea was kept of these, it will not be successful. Always the initial idea should be improved, getting feedback from the applications.

And the Busicom team found out five problems to be solved. The first problem was the lack of concept of LSI family. The first Hoff proposal was to develop on the binary processor using a standard RAM, and ROM. It required a system bus interface to the LSI. Also still in his proposal there were timing LSI, printer control LSI, and printer buffer LSI. The requirement of Busicom with the highest priority was to build the system with only LSI product. The second problem was the decimal data calculation method. In the beginning, Intel recommended us to use a method of look-up table for decimal calculations. However, it required lots of ROM, and lowered the performance. And the third problem was interpreter's programming method for application program. That used the micro-level instructions. If the sub-routine was simply used, always two bytes of ROM are required for each micro-level instruction. And fourth problem was the real time control programming method in order to control multiple peripherals, such as keyboard, calculator, display and printer. It was terrible headache for Busicom. The last problem was the instruction set itself. In the first ten days of September, 1969, the first batch of instruction set came out from Ted Hoff. This instruction set was too primitive, and only the decimal adjustment instruction set was added. And Busicom team judged the primitive instruction set required a loss of size of memory, and also it is impossible to control the multiple peripheral in the real time mode.

In September, 1969, Stan Mazor joined the team. The original proposal of Intel had been improved about two times step-by-step. At first we generated-- I generated a lot of desktop calculator's program using proposed instruction set. Then I performed a static operation dynamic operation for the proposed instruction set. By this time, Intel agreed to develop LSI family, including RAM and ROM. And also according to the first stage preparations, if the improvement was done, based on the requirement from Busicom, Intel proposal is acceptable. On September 13, Mr. Masuda, the project manager returned to Japan with result of faster stage operations. On September 16, Mr. Graham who was marketing manager of Intel, sent the data to Busicom to request the adoption of Intel's proposal. Also he strongly requested to use the program logic method for peripheral control. Also its letter said that chief development takes only six months with Intel's idea, instead of one year with Busicom's proposal. This proposed schedule became a big problem when I visited Intel in next year for logic checking. And in order to achieve a high level of performance by using the low performance processor, and a small size of ROM memory, I proposed a new instruction set, and function modifications, and a new LSI as well as getting the good advice from Intel. And I felt that Intel tried to avoid having an abundant instruction set to keep the number of transistor as low as possible. At that time, Intel set the target of number of transistor to be about 2,000. And still I believe 4004 project was not able to succeed without feedback from the system users and application program. My story may be a little bit long.

House: It's okay.

Shima: At first, in order to make the interpreters program, the 8-bit immediate data loading instruction with register in direct addressing mode was added. And secondly, the test interpreter being was added to examine the extent of external word, such as signal from printer. The definition of conditional branch instruction was modified for judging the test input. And thirdly, return from the sub-routine instruction was modified to store return address to be used for the conditional branch micro-level instruction. And fourthly, new LSI was added for the keyboard scanning and also the print buffer, and also its LSI combat 10-bit serial input data to the 10-bit parallel output data. Fifthly, to secure the flexibility, number of status register in ROM LSI was enlarged to be four. With these added instruction set, the modification, and the new LSI, it became possible to control many peripheral in their time by program. And next, in order to delete the decoder MSI for chip selection of RAM and ROM, a new instruction to designate the command line for chip selection was added. In the original proposal there was only store instruction for data transfer from accumulator to general purpose register. So load instruction, and exchange instruction were added, deleting the store instruction. If the number of general purpose register is not enough, the exchange is quite powerful instruction set. Also, it was very difficult decision to choose shift instruction or rotator instruction. Then we choose rotator instruction for the high flexibility and the reduction of application program size.

And October 14th, again, Mr. Graham sent the data for asking to accept Intel's proposal. Still I needed one more week for completing evaluations, and also Busicom in Tokyo has been evaluating new instruction set for the scientific calculators. And finally October 21st, Busicom sent the data to Intel for adopting Intel's proposal. But it took about two more months to finalize product specification of LSI. And several instructions were added in order to get the higher performance and reduce the program size. Such as like a subtract carry from accumulator or decimal subtractions; a code converter for the keyboard scanning; and clear for both accumulator and carry. And a compliment carry and no operations. Next with my best recollections, in order to avoid misunderstanding, I made functional specification of instruction set in English adding lots of drawing and the equations, but it might be in April in 1970. On December 16 in 1969, the last meeting was held with Intel. Intel's skeptic to add two new instruction set, such as a subtract carry from the accumulator, and a quota convergence for keyboard. Just I felt that Intel didn't realize how it is important to keep the problems from our side to be small. That is my comment for the Hoff's ideas. Then I returned home to Japan at the end of December.

House: Okay, so it's at that point in time when the specification is done, and you're waiting for Vadasz to staff the project. Is that right?

Hoff: That's correct.

House: Intel and Busicom are working together now on finalizing the instruction set, and making sure the application all happens. So let's hear the view from the Intel side.

Mazor: I think it's interesting, the synchronicity in your life because I had worked on an interpreter in college and then I had been working in decimal floating point on the IBM 1620, and then on the Symbol computer. So here I was again in my third project working for Ted, and working with Mr. Shima doing decimal floating point. One of the problems I had was that I had a customer who would like to look a [at the] machine as he originally saw it in his flow charts, as being a higher level machine, but we had made a dumber, simpler machine underneath. So the idea of an interpreter came up. Simply, in about 19 bytes of 4004 code, we could write an interpreter, which would, more or less, present the view of Mr. Shima's original machine that was capable of doing 16-digit add, and 16-digit shift left, which we implemented with sub-routines and a 4004 code. The linkage was a single byte, because you could pick up the byte, and use it as an address. And one of the enhancements we made to the 4004 CPU was the ability to fetch data out of the read-only memory, and then to do an indirect jump. That was what allowed us to do an interpreter [for our customer].

But Ted had a number of innovations that I want to call attention to. The first was that the on chip stack was needed, because we were running programs out of read-only memory. Ted and I had seen that on the IBM 1620, which had a sub-routine return address. I think people subsequently have missed the significance of the stack - the fact that Ted was working with three transistor dynamic memory cells with built-in refresh. Also it was kind a joke-- we had 16 four-bit registers. I mean, that's a lot of registers inside of a CPU, but of course, being only four bits wide would sort of make you chuckle. And another important point with 16-pin packages is that there was very little room for communication control lines from the CPU to the ROM and RAM. Another part of the architecture, which is sometimes lost on people, is the distributing of the op code decoding logic so that a ROM or RAM was looking at the same bus. When the instruction was being fetched at the CPU, the RAM could say, "Oh, that's an instruction that I'm going to perform" rather than the CPU having to tell it [the RAM] that the RAM recognized it automatically. And finally, Mr. Shima came back with at least four changes in the instruction set based upon the detail coding that he had done. He mentioned the keyboard process, and the jump instruction. So we got it very far along [into the project]. I would say [we were] quite a bit influenced by the PDP-8 architecture, because we had a lot of instructions that looked similar to the PDP-8; a few ideas from the IBM 1620; and then some of the finishing details based upon Mr. Shima's experience.

House: And that brings us up to the point at which it became time to develop the chip. Ted, do you have any more comments?

Hoff: Well, no, I think there were just so many things that seemed to work in our favor. There were just little things that seemed to fall into place. To give an idea, we used a three-step transmission cycle to send out the address to the program ROMs, and the size of the ROM was 256 bytes which takes eight bits of address. So the first two nibbles that go out go to all of the ROMs, and each one can be making access to its data, and then the third nibble that comes along tells which ROM gets to go on the bus. That way, we didn't have to sit around and wait for the ROMs. In other words, by sending the bytes low order, middle order, and high order in that step cycle, the ROMs were ready with data, and whichever one was selected then put its data on the bus. So there are all sorts of little details like that that just fell into place.

House: So now I understand. Some time goes by and Intel engineering is busy doing other things, and not performing according to the contract in terms of delivering these products until Les Vadasz hired you, Federico, to come in and take it over. Maybe you could tell us your experience when you first arrived and how you got involved and then go on into the story of the actual development of the product.

Faggin: Yes, well, in early 1970, I was still working at Fairchild, I wanted to move from primarily a process development activity, which is what I did at Fairchild, developing silicon gate technology for P-channel, N-channel, and CMOS. I wanted to actually design integrated circuits, complex data circuits using silicon gate technology. So I contacted Les Vadasz early in 1970, because Intel was the only other company other than Fairchild that had silicon gate technology in those days. Not much happened until March timeframe. And then he was in a hurry for me to join the company. At that time I was having -- my wife was having -- a baby, our first daughter. I remember that one day I had to cancel [an interview] because my wife was in labor, and Vadasz was mad that I had to cancel an interview with him because we were having a baby. <laughter> So anyway, so eventually I joined Intel in April the 3rd, 1970. At that time I met Stan Mazor. Stan actually gave me the background of what work was done, and he gave me a bunch of material that had accumulated, specification of the CPU, of the entire family, a bunch of random circuit designs that were probably done in the design group, that actually were not very useful as I learned more about what was to be done. My first task, of course, was to figure out what my project was, but then the next day, or the day after, Shima was supposed to come from Japan, and check on the progress of the work. At that time, I didn't know they were late. Nobody told me they were late. And so Stan and I went to the airport to greet Shima. Shima was tired but still wanted to come to work immediately. He said, "I'm here to check!" I said, "Oh, oh, okay," "Great!" "Just come over." So Shima and Stan and I walk into the office and Shima says, "Where is logic of CPU?" And I said, "Well, here's what I got from Mazor yesterday. This is all the material that I have. Take a look, that's all I know at this point." I think it was an hour later, Shima comes back and says, "This is just idea! This no good! Now late! Now late! You bad!" <laughter>

House: Okay, maybe Shima wants to comment?

Shima: After I returned to Japan, I took about two months to develop the program of desktop calculator with a new printer. And I made sure that there was no problem in the functional specification of the 4004. The formal agreement was concluded on February 6th in 1970. It said that Intel shall exercise its reasonable best effort within its capabilities. And also Busicom to join in product development as requested by Intel. And on March 20, 1970, Busicom sent the desktop to Dr. Noyce with a diagram of each LSI with explanations, and a disk of instruction set. And also its letter described The following: Busicom requested that Intel substantially complete circuit diagram by Shima's visit on April 7th. In order to complete the check of all circuit diagrams by April 25. Also Busicom requested that the production of 4002 RAM and 4003 Shift Register start at the end of July, because the breadboard will be built at Busicom at the end of July. Also Busicom requested Intel to substantially complete the production of the 4004 by the end of October. This, Busicom's proposed schedule was based on the letter from Intel on September 16, 1969. And then on April 7th, 1970, I visited Intel again for checking the logic schematic. And when I requested Faggin to submit the logic schematic, he said he just joined Intel. Then there is not any logic schematic at all, but it was unexpected, because considerable progress had been expected. I was so upset! <laughter>

Faggin: Thank you, Shima. Thank you! <laughter>

Shima: Yeah, like Faggin said that I next violently agreed, and protested by a strong tongue! <laughter> Then when Busicom's lawyer came to Intel, and we requested to submit the development of a schedule, it was found out the second engineer is scheduled to be hired in June for 4004. Since Busicom planned to develop the 4004 system hardware emulator for early stage functional verification at the end of July and variations. Therefore, Busicom proposed that in order to make progress on 4004 CPU development in schedule, Shima will design 4004 logic, including the logic simulation, and CPU test prototype generation, and layout checking.

House: So what happened next?

Faggin: So what happened next is that I had now this task where I am essentially six months late the day after I start. I have four chips to develop, an angry customer, and basically Intel had no experience whatsoever in designing random logic circuits, particularly with silicon gate which is a new technology, that requires a new methodology, a new way of doing it. Intel was a memory company. They had done memories before, and those designs are very different from random logic. I was on my own. Vadasz, my boss, was completely absorbed by the memory activity. And the application group, it was not even their job to help me, so basically I was on my own. Of course, I tried to do my best to make the customer happy I felt that even if I had nothing to do with it, Intel was late, and the customer was right in complaining.

Eventually, I was able to calm him down and say, "Shima, if you help me, then we can probably reduce the delay in the schedule. And if not, do whatever you want, but there is nothing I can do to make up the time." Particularly since the original schedule was already wrong, because in the original schedule the layout time of the 4004 at four or five weeks, which was the layout time of the memory. Therefore, there was no way that it could be done. It tells you that Intel was not a random logic company. They didn't understand what it took to do a random logic circuit. Now, I had to give a little bit of background here, because it's important to understand the state of the technology in those days. Basically, companies like Fairchild or Rockwell Semiconductor, or GI that were in the random logic design with MOS technology had an entire methodology. They had computer programs to do the logic simulation, the circuit design simulation. They had building blocks that were pre-characterized. They had examples of layouts, how you do different types of circuit. They had the experience of how to determine how long it takes to do design, also how many transistors it would take to do things, and so on. All of that.

In addition a company like that would have, for example, testers for characterizing the product. They would have production testers for those products. All that infrastructure. There was none of that at Intel. None of that! In addition they were attempting to do a two-phase design without a bootstrap load, which is a no-no. You cannot do that. And of course, in those days everybody knew that a bootstrap load cannot be done with silicon gate technology, because you cannot make a capacitor with silicon gate technology, the same way you do it with metal gate. And so the entire approach was not possible. You could do, fully static designs, which would be fine, except that they take two to three times more transistors.

House: Space and power.

Faggin: That's right, more than conventional dynamic design. Two-phase designs would require a bootstrap load. Four-phase designs, which actually were the ones that were used for random logic circuits in those days, because they had the best speed- power product did not require bootstraping. Although in metal gate you could do a bootstrap. That's why Rockwell, for example, had four-phase designs, and that's how they got their calculator business. So Intel, with a two-phase design, which is an effective design methodology, did not have the bootstrap load. Fortunately, I had developed the bootstrap load at Fairchild just before coming over and had tested it out to see that it actually worked. That required an invention, but it was really more an understanding of the physics of how devices work, because in a bootstrap load what you need to do is to always have a virtual junction, instead of a physical junction. The other thing that immediately was clear to me was that what was needed was the direct contact between buried poly-silicon and the junction. That was another invention that I made at Fairchild back in '68, as a matter of fact, and Vadasz knew about that. That invention would allow you to have roughly half the chip size that it would take with metal gate to do the same function. That [invention] was essential to get the chip size of the 4004 to the level that was required to meet the economic side of the business. So my first task was to understand what I was to do. I found that there were a couple of mistakes in the architecture. I fixed those.

Then there was another problem. There was a power-on clear. Basically, you had to have the chip coming up in a predetermined state. Nobody knew how to do that. It was important because it would require another pin, and there were no extra pins because we were using all 16 pins. So then I came up with another invention. In fact, we got a patent-- Intel got a patent-- on that power-on clear circuit, a specialized flip-flop that always is assured to come up in a certain way. So then immediately I started working. I put a schedule together that to my best of my knowledge at that time could give an idea to the customer of when we could have devices. The best that I could do is to have first silicon of the 4004 in December of 1970. But I couldn't do any better than that. Even that would have required the chip to work first time, which is a tall order for a complex chip like this. Then I decided to start by doing the 4001 first, then the 4003, then the 4002, then the 4004, so that it would be a warm-up. The 4001 and 4002 had a fair amount of logic so that would require me to develop the methodology in a more gradual way, more organic way. And I could have all the pieces together for the time when we were required to do the 4004. And so I did the design of the 4001. Shima checked it.

And then I started the layout. By the way, I had no layout people. In fact, we got a delay of at least two or three weeks to find somebody that would do the layout. Finally, they hire-- because in those days personnel will hire the person for you-- I think you remember. So they hire this layout draftsman called Rod Sayer from Lockheed. He was a mechanical draftsman, never seen an integrated circuit, never even seen a transistor. So I basically had to- do the layout of the 4001 a 2000 bit ROM. I had to draw by hand all of the building blocks and give them to him. He would simply copy it into the Mylar. So the 4001 layout was done-- the actual logic design took about a few weeks and the circuit design a few weeks. I developed a method where, instead of doing circuit simulation, that would take a lot of time, I had a method where I used graphic design that would allow us to size the transistors without spending too much time. I also determined all the fundamental rules for the logic design, for example, no more than three levels of gates, you know, all the basic elements that would then constrain the design in a certain way so that you could actually do it effectively. And so the 4001 layout took about a month, and then of course, [came] the checking. Shima helped me throughout all the phases of checking in those early days. And then we had to create the Rubylith. As you probably remember, you put the Mylar with the entire design on a cutting table, and then you put a layer of Rubylith, which is a fine layer of red Mylar over a base of transparent Mylar, and then on a [backlit] cutting table you cut traces, and then you peel off [the unwanted material]. And that is a very laborious process prone to errors. Not only is it prone to errors when you do it, but also when you transport the Mylar. Sometimes a piece of red Mylar would fall off. When you had checked and it was fine but then in transportation it got corrupted, and then you find out when you have a problem in the mask. So it was during the design of the 4001 and 4002 that it was apparent to me that the less translations we do during the design, the less errors we're going to make. It was particularly obvious as I got involved with these [high] complexity circuits. So I decided that instead of doing a separate logic design, and then a circuit design, to do logic and circuit design together in a single sheet. But also with the notion of the layout so that to the extent possible, you try to put the lines and the transistor placement in this sheet as close as you can guess to what it will be [on the final layout]. Obviously, you had to do the overall planning of the chip ahead of time, so that you know roughly where the various blocks are.. It was during this period that I refined the methodology of doing these type of designs. So the 4001 then came out came out as a chip in October. It was close to the time that Shima left for Japan. And in the meantime they got a version of the tester ready, so that I could find out if this thing worked.

Mazor: Can I interject for a minute? Intel was very small at that time, and Federico was my officemate, so after Shima's arrival the three of us were there [together]. And Ted and I are in the meantime working on other things, the 8008 and so on. But I distinctly remember, and you can help me with this, that Elvia, Federico's wife, went to Italy in the summer, probably with the new baby, and Federico was working till midnight. There was no wife at home, and this was a project that was very absorbing. So he put out an enormous effort during that time. However, being up till midnight, he would sometimes come in a little bit late, say 9:00, in the morning. And at the time, the rule at Intel was you get there at 8:00 a.m. So this was an interesting contradiction.

Faggin: Yeah, but anyway, so the first chip to actually be completed was the 4001. As the layout of the 4001 was proceeding, I started the design of the 4003. The 4003, by the way, used a new type of flip-flop, because it's a static shift register, and is an idea that I patented in Italy when I worked for S.T. Micro back in '67. That basically uses fewer transistors to do a static flip-flop. And this was a simple chip. It was a chip that the layout took a couple of weeks and it was done. Sort of in the spare time. And then after this was done, the design of the 4002 occurred. Shima checked all along. The 4002 layout started while we were doing the rublith cutting on the 4001. So basically we were doing pipeline designs, designing four chips in parallel, slightly offset one from the other. , From April to December we designed four chips, of which three were LSI in basically eight months. This was not the easiest thing to do given that a typical chip like a 4004 would take normally a year by itself. I see that Shima wants to say some words.

House: Tell us about the challenges that you faced during this time, Shima.

Shima: The most difficult logical design goal was to use a small number of transistors. At first, a detailed function block diagram of 4004 was drawn before its detail logical design. And also the interfaces between the function units were clarified. Because it's work inference, both chip size and performance acquired heavily, I repeated it several times until I was convinced. So as not to waste one transistor, I started the detail logic design. There were four units, and two sections, such that lots of transistors are needed, but design is relatively easy. Those are the general purpose register unit with two-bit refresh counter, the address stack unit with two-bit stack pointer, two-bit refresh counter, and four-bit address update logic. Next is the instruction register, timing unit, and the command control section, and the system bus control section. Since a three-transistor DRAM cell was used for both the general purpose register, and the address stack, the address update logic was placed between address stack unit, and internal system bus. Then, the contents of address stack were read out at Phase 1 time. Then updated address was written back to the address stack at Phase 2 time. And in order to simplify the testing, I added logic that the contents of accumulator and the carry flag are sent out through the 4-bit external system bus at the idling time. Then its total number of transistor became 1,139. The remaining logics were a four-bit arithmetic unit, and an instruction decoder/encoder, and the logic of instruction execution controller.

The basic idea of four-bit arithmetic logic was proposed by Ted Hoff. And I tried to use the instruction encoder for avoiding the random logic. I kept thinking about those logics day and night. The number of transistors was 329 for arithmetic unit, and 769 for instruction decoder/encoder and execution controller. Then finally, with my counting method, the total number of transistor of 4004 became 2,238. It had been approaching to the climax of the 4004 logical design. There were only one circuit engineer, and two master designers. And I started to make the logic schematic without the size of the transistor in order to use its schematics for the planning of circuit design, and layout design. I picked up the three A2-size of paper, then considering the interface with each unit at first, the general purpose register unit, and address stack unit were drawn on the paper at right side. And next the arithmetic unit, the command control unit and the timing unit, and the system bus control section were drawn on the paper at left side. Then the instruction register/ decoder/encoder and instruction execution control unit were drawn at the center of the paper. As a result, the 4004 layout became almost the same as the logic schematic. It took about three months to design all of the 4004 logic, including the logic simulation. Fortunately, there was only one mistake, and this mistake was found out by both logic simulation and hardware emulator, using different test vectors.

House: Very good.

Faggin: So we are in this process of developing these four chips, one after another. After the design of the 4002 was checked, and while the layout was going on, that's when Shima started doing the detailed logic design of the 4004 using a lot of the building blocks that were developed for the 4001 and4002. Particularly the memory and the memory refresh. All those parts that Shima mentioned.

House: So after the April when you started, did Shima come immediately to live-- he just stayed?

Faggin: Two days later, two days after I was hired, yeah. He never went back, yeah.

Shima: I was captured.

Faggin: Through October, yeah, through October he was...

House: Captured.

Faggin: Yeah, that's right, six months. So of course, Shima didn't know how to design integrated circuits, and he didn't know transistors or any of that, but he learned very rapidly.

Shima: I knew a little bit.

Faggin: Yeah, he knew a little bit, like a transistor's a switch.

Shima: That was enough. It was digital.

Faggin: Yeah, except for the RAM. Yeah, so, Shima, in the first few months also learned a lot through the design of the 4001, 4002, 4003. We learned a tremendous amount about how to take some of the stuff that I did on those chips and put it into the logic design of the 4004. And of course it was a tremendous to have Shima to actually help in that phase of the project, because that time was the peak of activity. I had the 4001 in mask making, and the 4003 in rubylith cutting, and the 4002 in layout, and starting the 4004 layout all at the same time-- only me to go around. So Shima also worked very hard. He would probably go home typically at 8:00 at night, right? But typically, when he would go home, I was still there. And at that time, my wife, as Mazor mentioned, she left in June, with our three-year-old daughter, and went to see the grandparents. I was very happy to see her gone, because that actually gave me a lot of time. She stayed over three months in Italy. It was a hot summer for me. So the layout of the 4004 started around the August timeframe with the memory part. It was the easiest part in a sense because it was a regurgitation of what was done earlier. And the key thing there was to-- what was keeping me awake at night was I had dreams of not being able to finish the end of the chip, because there would not be enough room to put what was required. And I had to erase and start all over again for a big chunk of the chip, because, remember, we are doing this by hand. So if you did make some mistakes, you'd have to erase and start all over again. And of course, given the tight schedule it was not an easy task to do because you have to determine this dimension first, or the other. Depending on which way you go, you determine that dimension first, and then you proceed down that way. If at the end you bulge out, then you got a problem, right?

House: It must be a rectangle.

Faggin: That's right, it must be a rectangle. So in fact, you can see that this was the part that was finished last. Initially I was hoping that I could fit everything in a little less dimension here but then eventually was clear, because also when I started this, the logic design wasn't finished, so I didn't know how much to plan for the control logic, which is this part here. So eventually it became clear that I could fit everything in this dimension. But I had sleepless night until the very end when I closed the loop around. So the 4004 layout took about three months. At the peak, we had three people working on it, and three layout draftsmen.

House: Do you recall their names?

Faggin: Yeah, there was Rod Sayer, Julie Hendrix and Barbara Mannis. Barbara Mannis only worked for about a month, and the other two were there for the entire duration of the project. The challenge in the layout was fundamentally to do it fast enough. You know, other than, of course, closing the loop. But then we had to tape out, meaning what we call tape out now. In those days it was the ruby out.

House: Ruby out, yes!

Faggin: Ruby out was basically right after-- or around the time-- that Shima went back to Japan. Shima helped me also with the checking of the ruby. It was a daunting task, because it's a complex chip, and the Mylar wouldn't fit on the cutting table, so it was in two pieces, and all that. So eventually we got into mask making.

Mazor: I want to ask Shima a question. In his book, he shows a picture of a big station wagon that he owned in the USA, and also a picture of him at a company barbecue, could you comment on that?

Shima: Yeah, when I came to the States the second time, I didn't have a TV, I didn't have a car. Therefore, Dr. Noyce told me, "You can use my wife's car, but car is located at Mendocino." And then one guy pick up me to go to Mendocino, but there are lots and lots of wineries. When I went to Dr. Noyce house, I was drunk! <laughter> And I stayed overnight. Then I came back to Silicon Valley.

Mazor: Thank you.

Shima: But I think Federico did an excellent job. It was unbelievable to develop four kinds of chips within, what was it? Eight months?

Faggin: A ha.

House: So when did the last chip go to mask making?

Faggin: It was October.

House: October. And then you had silicon...

Faggin: And then the first silicon was in December, pretty much on schedule.

House: Now we're up to October of 1970, So tell us about October, November, December, what happens next.

Faggin: Well, basically from that point on, you have to go through the process of mask making, and then the process of manufacturing the wafers. So there was nothing happening for me until the wafers came out., They came out toward the end of the year. It was right after Christmas. And so I got my first wafers, and I was ready to go. I mean, you know, the tester was ready, we got all the characterization, sort test loops, programs we're ready to test.

House: Were those [test programs] developed after ruby out?

Faggin: Yeah, those were developed after ruby out, and a lot of work was done by Hal, and also some by Shima. But mostly Shima worked on the production tape for the Pacific Western tester. So I got the wafers at 6:00 [PM] or so, so people were just leaving. I was actually happy that nobody was around, the last thing you want is for people to ask "Does it work? Does it work?" So-- yeah, too much pressure, right? And so I loaded the first wafer in the probe station. We had a probe station with probes that would go onto the pads, and then you come down with the probes into the pads like this, and then you send signals, and you test the signals. Generally, in the early debugging stage you use a oscilloscope to see if you get what you expect. And so I went down to the first die, and in the scope there is, like, floating, nothing happens. Oh, okay, well, you know. So I go to a die, floating, nothing happens. Well, okay. Another die. Same thing! And then I said, "Well, it's a bad wafer." So I took another wafer, put it underneath, go down, same thing. Now, by this time I was trembling, I was sweating. <laughter> I said, "How could I have fucked up so bad?" You know, fortunately, the 4001 worked first time. 4003 worked first time. 4002 had a couple of minor problems. So I could not have screwed up so bad, okay? So I took the wafer under the microscope. I look under the microscope, and lo and behold, they forgot the layer for the buried contact. So a good 50 percent of the gates were floating, and there was no signal, so obviously, the whole chip didn't work. So it basically wasted three weeks at that time, which was not welcome, because of course, Shima was there waiting for his chips in Japan, right? And but, you know, there was nothing that could be done at that point.

House: So you had to go back.

Faggin: And so we had to go back. Finally we got the new batch of units out in January, some time in the second half of January or so. And same thing, they came out at 6:30 in the evening, and now this time, I put down the probes, and "Aha! Signals!" <laughter> So it looks good, and I run a little test loop, and it passes, and so I was very excited at that point., Everything that I was testing seemed to be working. I kept on going on until 3:30 in the morning-- 3:30/4:00 in the morning, and I check all the major blocks at that point. And then I went home. I was exhausted. And so at that point, it was clear that we had a 4004, because even if there were problems, which there were a couple of problems, they would've been easily fixable, because all the major things were working. And so then the following week or two, I debugged the whole thing, and found those two problems. What was interesting is that in those days, we had little probes that have very sharp tips. We would go inside the circuit itself and probe in metal lines.

House: So you had no glass layer on top?

Faggin: That's right. There would be no glass layer. Sometimes just by rubbing the tip on the metal we would simply break the metal. And you know, in cases you wanted to test what would happen with less signal. So it was interesting how we were able to debug relatively quickly chips that were very complex, where very few points were available.

House: Was Shima in Japan now?

Faggin: Oh, yeah, Shima had already been gone for a long time. He had gone in the middle of October or so. And the other thing is I put a number of test points inside here on purpose, so that I could test different points where they had a little larger metal. Little metal tabs, so I could go in and test internal nodes that otherwise would have been difficult to get at to help the debugging phase. So basically at that point the only thing that made it not useful to Shima was that there was a problem with the program counter. It would only count up to 16 and then it wouldn't ripple. There was a piece missing. So it couldn't really run a program. Also it would've been able to run a complete program because the other mistake was not as catastrophic, but this one was a bad one.

House: Sixteen location program was not...

Faggin: Yeah, was not a very good idea.

House: And ran a calculator...

Mazor: It ran a calculator, too. You know, I've met naïve designers who didn't think much about testing. One of the things that these guys had done is that during the operation of the CPU, it put the contents of the internal bus out onto the pins. They were doing something more visible outside, and out-- and essentially more testable. I think it's another good experience they had, and it demonstrates good skill to do something like that.

House: Yeah, good forethought.

Faggin: Yeah, so then, of course, after debugging there was another change of the mask set. By March I shipped the first fully-working 4004 to Shima. And then I think a week or two later, Shima said, "It works!" Because he plugged it into his own calculator. By that time he had done all the software, but I'll let him tell the story of that.

House: So maybe you could tell us about receiving the first chips.

Shima: Yeah. My real goal was to develop the desktop calculator with printer. It took about four months to develop the prototype of a desktop calculator with printer, including the development of application programs, and the 4001 ROM hardware emulator. From the general idea of Busicom, it [the development] lasted about two years and three months. And in April, 1971, finally the desktop calculator worked publicly. I was so excited! And I was quite happy! The desktop calculator with a printer was constructed with one 4004 CPU, and two 4002 RAM memory, and four 4001 ROM memory and the three 4003 shift registers. I was able to add several value-added functions by program. And just after the development of the desktop calculator, Busicom developed the Cash register, the Billing Machine, and the Teller machine for NCR Japan. And let me put my comment. The microprocessor replaced the [hard] wired logic by program. After the microprocessor was developed, the functional specification of the system was decided by the capacity of memory, and the performance of the microprocessor. And I was quite happy to work with Intel. Thank you!

Faggin: Thank you.

House: And so...

Faggin: And so then after that, of course, [came] the task of transferring to production all these parts.

House: Is this when Hal Feeney got involved?

Faggin: Hal Feeney got involved toward the September...

House: Hal, maybe you could tell your experience of the 4004 project.

Feeney: Well, as Federico said, I got involved in it at the time that we were kind of delaying or running the 8008 at a low level, and got involved helping him almost any way that I could. But the main project was a very deep memory that would allow us to functionally test some of this stuff. Because again, we're sending an instruction to the processor, we're getting information back from the processor, we're comparing and so forth. And we had no tools that would allow us to do this, or allow us to have any kind of a long program. Now, at the same time, in Ted Hoff's group, Tor Lund was working on a PROM programmer. On the PROM programmer, he had a paper tape reader. The only other paper tape readers that we had available to us were very slow paper tape readers that would run on a teletype machine. As a result, we took advantage of generating programs on the teletype machine and then loading them very quickly into a very deep memory, using the high speed optical tape reader. And the best price memory we had available at the time was Intel memory. It was the 3101 bipolar memory. So we built a rack of 3101 bipolar memory, put the appropriate interface with it, loaded it up with the tape, and then we could set it up and time it and do a comparison with the operation of the processor. We would send instructions to the processor, and then also be able to have the outputs that we we're expecting on the tape. So we send the instruction and have a comparison with the bus outputs that we expected, and go through step-by-step instruction by instruction. We could either set up a program and do something useful from a program point of view, or the initial time [of testing each device] go through instruction by instruction by instruction, and insure that each and every element of the program worked.

House: It was largely for engineering debug.

Feeney: This is the engineering debug and characterization part of it. This is not the [production testing] part that would go in for any kind of production. But I was highly motivated to do this because I knew that I would use exactly the same tool in exactly the same way to go through and do the 8008 when we had working chips with that. Federico was talking about going home at 3:00 in the morning and one thing we learned about that is that there's very little Silicon Valley traffic, and you can get home very quickly at 3:00 in the morning. <laughter> With no problems at all! Yeah, and we had a lot of experience doing that. But setting up, having the test equipment, being able to go through and do almost anything we wanted to with these chips—at that point it encouraged us also that, we can play with these things, we can program them, and thinking of a variety of things that customers could do if we ever had customers outside of Busicom to use these devices. This was the job that we had to go through from March up until the following November when the product was released to customers outside of Busicom\

Faggin: Well, from the time I got the project, I was very excited about the opportunity to develop the 4000 family, because I saw, and Ted and Mazor saw, the enormous potential of this type of device. However, early on, this project was an exclusive project for Busicom, so it could not be sold to anybody else. I always felt that the 4004 was good for a number of control applications, because even in the calculator, or in the billing machine, you have a number of peripherals that you need to control. So I felt that there would be all kinds of other applications other than calculators that could use the 4004. Now at that time, the way I remember it, both Mazor and Hoff were skeptical that applications for a general purpose control with the 4000 family was going to be good. So I took it upon myself to see if I could use it in a practical application. The opportunity came when I had to build the production tester for the 4004, where I used the 4004 as the controller for the tester. And so I developed a program, I used the 1702, the EPROMs that were available then. Basically I did what a customer would have to do to use it, and learning how that could happen. And at the end of that process, which worked out quite well for me, I really tried to convince everybody that, it was a good product to go out in the marketplace. And at that point, I think, the management of Intel started getting convinced that it could be a practical product to go on the market. And so that was the beginning of a movement toward marketing the 4004. Of course, the 8008 was in the wings. That was from the very beginning considered good enough for other applications in the marketplace, but initially not the 4000 family. That's the way I remember it.

Feeney: But the 8008 was oriented toward data terminals, character manipulation, and whatever. The 4000 series was in the control function part of it. And I think at that time, management really didn't understand the difference between control computers and data processing computers.

Faggin: But also, we did not think that Busicom would allow us to use the 4000 family for other than calculator applications, right? And so if you take that out, you know, the feeling was there would be very little left for the 4004 to be useful. And that's what I wanted to prove, that in fact, that was not the case.

House: So how did Intel get the approval from Busicom to sell this to other customers?

Faggin: Well, at that point then there were a number of us pushing management, to do that. For one thing, I recommended Noyce to lower the price of the chips to Busicom, and get a release from that. That was just before I knew that he was going to Japan to visit Busicom. And that's after I talked to Shima, and I knew that Busicom was in financial difficulty, and they felt that they were paying too high a price for the chipset. So that certainly was one of the elements that helped.

House: What timeframe was that decision?

Faggin: This was probably in the May/June timeframe. Around that time.

House: Of '71?

Faggin: Of '71, yeah, '71.

Hoff: I think the impression of us not being interested in offering this as a product is wrong. Actually, we felt quite betrayed when marketing told us that they had given exclusive rights to Busicom for the set because we felt it had other applications even back in that October/November, '69 timeframe. But subsequently, I've seen a copy of the February contract, and it actually does talk about the potential for sales to other people, but with the forgiveness of engineering charges should that take place. But I, at that time, I don't think I had seen that February contract. But when marketing told us that they were going to be going to Japan, because Busicom had asked for a reduction in the price of the chipset, because the calculator business was becoming more competitive, I was one that went to marketing and said, "If you can't get any other concessions, get the right to sell it to other people." Now that was about May of '71. And they came back and said they had the rights then to sell it. And I believe there may have been a brief blurb put in-- I think it was *Datamation* around May of '71, talking about the chips. But it wasn't like an official announcement, and it just kind of went away. I went to Noyce and to marketing people asking them to please get this thing available, I think it was probably early summer of '71. I believe it was before Ed Gelbach joined the company. But Bob Noyce said, "We have a tiger by the tail here, but we are not ready to make a decision about announcing it as a product or adopting it." And I said, "Every time you put it off, you've made the decision not to announce." And I said, "Somebody else is gonna beat us to this if we don't get this out there."

House: There were some reasons that were stated by people that were on the other side for not introducing this. And maybe you might review what the opposition was.

Hoff: Well, there were several arguments that were made. One was we'd be competing with our customers for memory. In other words, our main customer base was the computer companies that made large mainframe and mini-computers, and they were going to be buying Intel memory, and if we were in the computer business, we'd be seen as competing with them. And so maybe they would go to other companies that did not have microprocessors in their portfolio. I think we tried to reassure that these computers were much smaller, much lower in performance. They weren't going to necessarily compete with a minicomputer. If somebody needed a mini-computer, he's going to buy a minicomputer. He's not gonna be able to use a mini-computer to solve a microprocessor problem. In other words, the level of performance that the microprocessor had was significantly less. The other thing was, "How will we ever support these?" And the statement I heard was, "We've got diode salesmen out there trying to sell memory chips. How're we going to support computers?" And that's where Stan and I would say, "Well, we'll try to put together design aids, application notes, user manuals, and things like that." I think we could probably hand-hold maybe a dozen major customers, and the rest of them we'll try to have enough information so they can make it out there on their own. Now I kind of figured if this thing was successful, the feedback will take care of itself.

House: I understand there was an argument also...

Hoff: There was a concern that too many people who had used mini-computers would come to use one of the microprocessors, and it's offering probably an order of magnitude less performance, and that they might be disappointed. That was the biggest concern - that they would say, "I'd rather spend $80 more or $100 more, and buy TTL and build a processor, rather than buy your $20 chip and not be able to do what I need." Fortunately, that didn't happen. That was probably the only thing where we had any reservation about offering it as a product.

House: As I understand there was a concern that the memory cross would be so large, that reducing the cost of the processor wouldn't be that important.

Hoff: Well, for some problems that would be the case, because, our target when we started Intel, or actually some time after we started Intel, was to get the cost of memory to a penny a bit by 1972. You have to figure this is happening even prior to 1972 --to make this decision whether to offer this as a product. And it doesn't take very much memory. I figure, what'd'ya take, 3000 characters to store a printed page. Well, that's $300 worth of memory, approximately, at those prices. And would you really want to, you know, save $80, when you're already committed to maybe $320 or so, and throw away a factor of 10 in performance, which is what the TTL would offer you over early microprocessors. But if the application was one that didn't need much memory it would be less of a problem, and that's why we looked at things like that. We talked about things like an elevator controller. All you have to do is store a little bit of data for each floor. It's only a few bytes of information.

House: Also, I guess there was an argument about the number of computers that were sold a year.

Hoff: Well, one of the arguments I got from the marketing people was about the time I was saying, "You should get the right to sell it," said, "Look, they only sell about 20,000 mini-computers over each year. And we're late to the market, and you'd be lucky to get 10 percent of it. That's 2,000 chips a year. And they said, "It's just not worth the headaches of support and everything for a market of only 2,000 chips."

Mazor: Can I chime on here, too, Dave, if you don't mind?

House: Yep.

Mazor: Just to reiterate a couple of things that might otherwise get lost. If you needed a CPU in the system, and if you're going to have some program you needed a ROM, so the minimum system is really two chips, and that's using the index registers of the CPU as a ...

House: Memory.

Mazor: As memory, which isn't much. So typically, we expected a customer to be adding the RAM, which was an oddball amount. It was 320 bits, but that's for another reason.

House: That's for ...

Mazor: Yeah. But in Shima's case, I believe he had four ROMs in his calculator, and then later on he could do something like offer square root, or some other function by adding a ROM. So the maximum system that we provided direct support for was 16 ROMs and 16 RAMs. So the system was very configurable. Back on the point about minimum cost systems - you could build a system with two chips or three chips down in the low end. As you got into larger and larger systems, I think Ted's point would be at that point there might be more efficient ways of buying memory, and perhaps therefore having a CPU. Now another point that hasn't been stated, but it was part of the architecture that I think was worth mentioning, and that is that when you got a RAM chip it also had a four-bit output port. So for every RAM you added to the system, you have another four wires to attach to lights or switches or keyboard. So in the Busicom case with two RAMs that gave them eight output signals. But an additional feature was that with each ROM chip, you got 256 bytes of program memory, and a four-bit IO port, which was mask programmable. That is the customer, upon ordering the ROM says, "Well, in this case on this ROM chip I need three more output ports," or, "I need three input ports, and an output. So [in a] sense you're doing a custom mask for the ROM, not only is the program configurable, but also the port is configurable." So the parts were fairly general purpose to the extent that the system was flexible, both in terms that the amount of memory, and the amount of IO. , Therefore we thought the applications would be quite broad. I'll just mention a couple of other quick ones while I'm at it. And that is having done a lot of software over the years I knew that we would need an assembler for writing assembly language, because that's how you wrote the MCS-4. And I remember mentioning it to Gordon Moore, who believed that an assembler is someone who works on the manufacturing line putting it together. So I was able to throw together an assembler on a weekend by using a macro assembler. You may be familiar with high-level macro assemblers. You could create a macro for each suite of instructions. The other minor point is that the mask that we produced for our customer had 256 bits, and those bits were physical items on the mask, and you actually had to generate a way of putting down those little bits. So it was a minor support tool. We wrote a program that took the ROM data that had to be stored and printed it out.

House: And mapped it.

Mazor: And mapped it literally, so you could stick down the ones or zeroes that were applicable. So that was some of the early software support.

Hoff: There's one other point, too. When we heard that, the 4004 was not likely to be available to other customers besides Busicom, but we had the 8008 in the line, and the 8008 was supposedly going to be available to any and all customers, assuming that Intel decided to go into business, Stan and I designed another four-bit processor, with the idea that if we did not get rights to the 4004, we would have a low-end processor. And this one actually could run out of standard memory. In other words, it could directly talk to an EPROM, or a standard read-write memory. We called it the 4005. We, —Intel, never produced it, but as part of a sale of technology to MIL [Microsystems International Limited] of Canada, we transferred the design of the 4005 to MIL of Canada. And I believe they actually did produce the processor.

House: It was never introduced as a product that we know of?

Hoff: I think it was by MIL of Canada, but Intel never produced it and sold it.

Mazor: Not only that, one of the things that we were concerned about was that on the 4004 we were too ambitious, so this 4005 was a greatly simplified four-bit computer, even in comparison to the 4004. For example, it had no index registers at all, and no stack, and so it only had like two pointers so it was a very simplified four bit computer.

House: In fact, Federico got it all on the chip. Yeah, I understand there was some opposition inside even early on to this whole project also. Tell us what Andy Grove and Les Vadasz' attitude was towards the 4004 when you arrived?

Faggin: Well, I think Andy's attitude was fairly old-style actually. I remember I was walking down the hall probably one or two weeks after I was hired, and I said, "Andy, this is a great thing! It's a great project! You know, this thing is going to really change the way we do things!" And he looked at me, with a face like saying, "What're you talking about?" <laughter> No, but Vadasz was also of the same opinion, it was a bit of a waste of time. Not as strong as Andy. But they were so busy doing the memory, which was the main business of Intel, that he had no time to dedicate to this side of the business that was considered kind of a poor orphan.

House: A concern that this would distract from the basic business of making memories?

Faggin: That's right, with that the fact that, they didn't hire the people in time. And when I joined, I had to resources, and on and on and on. So it clearly was not thought by the two of them as a line of business worthy of pursuit at that time.

House: So marketing thinks that there's not enough big enough market, and that their salesmen can't sell it, and a number of these things. And management's concerned that it might interfere with the memory business. So who did you talk to on the project?

Faggin: Moore didn't seem to have much of an understanding or interest in that. At least not to me.

House: Moore?

Faggin: Gordon Moore, yeah. And then really, it wasn't until Gelbach joined the company that Gelbach having lived in a company like TI, where there was a fair amount of business with random logic and chips of that sort, calculator chips and so on. You had much appreciation of that part of the business. And so Gelbach was my entrée into the marketing side and I tried to convince him to go to market with the 4004. And so he was much more receptive, as a matter of fact.

House: When was the actual introduction?

Faggin: It was in November

House: November, ’71?

Faggin:; Yeah, that’s right. But the decision was really done in early summer, kind of June/July timeframe, at least the preliminary decision that we’re going to go out, and then probably the final decision was made in September, I would say, timeframe. But at that point, all the plan of the introduction was done and laid out, and all the documentation was being prepared, so at that point it was irreversible. So it was really September. Is that what you remember roughly? September was the irreversible decision?

Hoff: Could be. I don’t even know when that decision was made. It always seemed to me that things changed once Ed Gelbach joined the company. There was a different attitude moving ahead.

Faggin: Yeah.

Hoff: You know, you have to realize, I think that we forget, with computers being ubiquitous today, they were much rarer in those days, and not many people had experience with computers. There were a few of us on the technical side who used them in design and so on, but probably most of management never dealt with computers.

Feeney: And we were very limited on how much [timesharing] computer power we could use at any given point in time along the way, too, even in the design of these computers.

Hoff: So things have really changed since the announcement of that part. But it was a few of us who really did believe that this thing had potential, and fortunately, we prevailed in getting Intel to announce it as a product.

Mazor: I have a datasheet here, and it says on the datasheet that it’s a micro-programmable general-purpose computer, four-bit parallel CPU with 45 instructions. Now when we did an advertisement, there were many people who did not believe that we had a computer. They thought it was an arithmetic unit, or something else. And they really were offended, but I tell you when they looked at the datasheet, it was compelling.

Hoff: Another thing about it, this was marketing’s decision. Data sheets at Intel, at least up until this point, had all the electrical information in the front. And the application information in the back. And it was decided for this product the application information would be in the front, and the electrical information would be in the back. <laughter> It was moving from components to systems.

Feeney: Well, Ted, I don’t think it was quite that dramatic at the time. The whole issue was to have a datasheet. And on that datasheet you’ve got the features at the top, you’ve got the pin-out on there and everything else. There was a very distinct problem with the MCS-4, because you needed to have four pin-outs. So there was no way to get all four pin-outs on the front page, so the very first thing violated the Intel tradition...

Hoff: I do remember somebody in marketing telling me, "This was a milestone for Intel, going away from the traditional datasheet."

House: From the traditional memory datasheet.

Feeney: I mean, it's a matter of how these things are done as an Intel style, and this was part of the mold to break of Intel style. Yes, we do shift registers; we do memories. Processors are a lot different, and then, by the way, there's a [user's manual] book with it; call in and get the book that tells you really how to use this thing.

Mazor: There're other anecdotal stories. I'd like to share one. A fellow called me up from the telephone company. He said he had the datasheet. He was going to quit his job and do something with this chipset. That's how dramatic it was!

Shima: But the Japanese market, system makers were looking for the microprocessor type of device. For example, a cash register. To make a cash register, it require like 200 to 500 TTL [packages]. A huge power supply. Thing like 4000 line came out, they tried to look for the microprocessor quite deeply.

House: It was taken more seriously in Japan.

Shima: Uhm hm.

House: So now if we can turn to building out the product line. You've got some chips and you've decided to make it a product, but it's really a computer and out of this came the genesis of a systems level business at Intel, and simulator boards, and this sort of thing

House: Hal, maybe you want to talk about the marketing efforts to launch this new concept.

Feeney: Okay, well, I got involved with the marketing side of it kind of in two-fold way. I was finishing up the 8008 in engineering in the fall, or fourth quarter of 1971, and it was November 15th that we announced the 4004, or the MCS-4, [that was our evolution in [the way of] thinking [of the family]. We went from being a company that would just announce chips to announcing a complete family of products for the very first time. And on announcing the MCS-4 we called it the alternative to random logic. And with an alternative to random logic, it was a matter of replacing TTL that had been used heavily for gates, flip-flops, other slightly higher levels of integration, and be able to replace those [TTL devices] with programmable semiconductor devices. Now we had to go through several things. One, we had to present it as a component, and we talked about that with the datasheet that Stan showed. Beyond that, we had to do a user's manual, that would pull together all of the instructions and tell people how to program the device, and beyond that, it was a matter of showing them and giving them tools to use. Well, at the time of the introduction, we basically had the Alternative [marketing brochure], we had the datasheet, we had a user's manual; we printed lots and lots of users manuals and gave them out to everybody that called in and went into numerous reprintings of the user's manuals, because it seemed the customers had just an insatiable appetite to get information on these new devices. So everything was set up to bring this [microprocessor product family] into the market.

The one obstacle, or speed bump we had with this was that because we had permission from Busicom to sell the product to other customers, [Intel sold the MCS-4] with the caveat that the product could not be used in any calculator applications. Therefore, before anybody could get a sample of the product, they had to sign off on a disclaimer that the product would not be used in calculator applications. And for each and every customer we had to do this. But it also gave us very good tracking of our customers, because everyone had to be identified before we could ship any product to them. But it raised one other obstacle, and that is that we basically couldn't sell this product through distribution. Everything had to be done directly with the factory. So [there were] a lot of complications as we went along with that. But the one piece that was missing, and it came up very quickly out of some demonstration work that was done in Ted's group—a board that demonstrated the 4000 series on a regular about foot-square computer board, with standard memory, and the 4004, or the 4002s because they had I/O, with standard ROM, but also again, a feature that could use the I/O off of those devices, and then demonstrate them in a variety of different applications, interfacing with the teletype and programming the EPROMs. But I will turn it over to Ted to discuss that a little bit more, because that became the genesis ultimately for development systems that Intel would sell, and would be a very strong business.

House: I remember at that point someone suggesting that since we were giving away so many manuals and selling so few chips, that we sell the manuals, and give away the chips.

Feeney: Give away the chips, yes. Intel in the past, typically had given away lots of free samples. We made a decision at the time that the first microprocessors came out that we set prices on the microprocessor devices, and did not give them away as free samples, as a way 1) of qualifying our customers; and 2) demonstrating that there is significant value in having a processor on a chip.

Faggin: And number 3) paying for the manuals. <laughter>

Feeney: Very true!

Mazor: Just quickly a background point, and that is when Intel got started the primary memory in computers was core memory. And core memory was non-volatile. That is if you turned off power at night, came back in the morning and turned on power, presumably your program was there intact. Now in reality, anybody who worked in the computer center probably didn't really rely on that fact. But one of the objections that we had in doing semiconductor memory is we had to say right up front that it was volatile. And more or less we had to overcome the objection of the customer which is, "Well, wait a minute, you know, if you turn off power, you're gonna lose all the data." While I don't think this is really a serious issue, [but] we did have to confront it. So one of the products that was outlined that we needed at Intel was a non-volatile memory. And it was understood in the early '70s that the nitride process might give you a way of treating the gate in such a way that it would remember information when power was off. So another officemate that I had when Faggin moved on was Dov Frohman-Bentchkowsky. And he was given the project of creating a nitride memory. Well, to do that he had to study the gate-silicon dioxide interface, and to do that he had to get not much metal on the chip, and actually it was during those experiments, that he called me over one day, and said, "Look at this," and he showed me on a curve tracer, what we call a logic One, and he blew on the chip, and it became a logic Zero. So he had come through with a mechanism, which we called the floating gate for storing charge. And in this case in a non-passivated chip, the moisture in his breath could cause the current to leak out. Interestingly enough, management didn't believe in it too much, and gave him quite a bit of latitude. But in the early days of Intel, we developed this new product which became the EPROM. And EPROM became an essential synergistic product with a microprocessor because our customers would want to try out programs, and the way they would do that was with erasable or reprogrammable EPROMs

House: I remember we used to sell the EPROMs as only an engineering tool. And you'd obviously go to masked ROM when you went to production. Of course, because there were always software changes, the customers actually went into production with EPROM, which was a nice surprise for us, because we got a lot more money for EPROMs.

Hoff: I was gonna just add that, because the initial view was that they were only a development tool. I often wondered does anybody know to what extent they characterized the life of [charge] storage on the early EPROMs, because it was thought they were just gonna be used on the bench. But we found very quickly that customers were using them [for production].

House: I remember putting EPROMs on the roof on [the] Bower's Avenue [building]

Hoff: Yeah.

House: To see how long they'd-last out in the sun, to see what'd happen to them.

Hoff: But we did the simulator boards with the idea that it would be a useful tool to help develop a microprocessor application, and tried to make them duplicate what you would have when you went to the mask programmed 4001s and so on. And provided the board such that you could use the EPROM for the read-only memory part of it. Initially the marketing people said we should just give the boards away. And I always argued, I didn't think that was a good idea, because we would always be there looking for an excuse to stop giving away money. I argued that we should be able to sell the boards, and even make a profit on them, and it would be a lot less expensive for our customers to buy a board from us, which we were producing in volume, than to have to design it and build it from scratch themselves. And on top of that, this would be a tool to get our microprocessors into the marketplace, because if [the customer has] got this hurdle that he's gotta design a prototype board himself, that may be enough to stop him from using the microprocessor. So it seemed like it worked out well, and then we turned it over to the marketing department, and they just took off with it, and actually developed a full-blown development system business, which, I guess for some years, was quite a money producer for Intel.

House: Produced more money than the microprocessor business did for a lot of years. That was true. And I understand that out of that came the first sim board that had to program EPROMs. , So how did the EPROM programmer evolve?

Hoff: Well, we had two EPROM programmers. One was just a board that plugged into a sim module. But we had also been charged with building what we called an EPROM burner. And the first one we did, we worked with an outside company that did wire wrap, and we gave them a logic diagram and they had some problems. We had a hard time getting that thing to work. And about that time the microprocessors became available, and we said, “What are we fooling around with this random logic for? Let’s use the microprocessor to build the EPROM burner.” And we did that.

House: Was that done in your group?

Hoff: That was done in my group. I think we called it the 7600, if I remember the number of that. And we had one funny problem when we were debugging it. We had a paper tape reader on it, and for some reason it was reading the paper tape wrong, and we were convinced we had a bug in our program. And it turned out the driver to the paper tape reader had a snubber diode on the coil, and that was slowing the paper tape reader down. But we actually wrote a simulator of the 4004 to check it out, and the simulator said it should work. And the hardware wasn’t working.

House: What did the simulator run on?

Hoff: Well, the simulator ran on another machine.

House: Like a DEC [PDP-] 10 or something.

Hoff: Yeah. So that was one case. And then we had contact with some of the outside suppliers of EPROM and PROM programmers, and we pointed out the advantages of microprocessors. And about that time Intel switched from, I think it was the 1702 to the 1702A, or it was some change like that, that had a major impact on the programming algorithm. And we redid our burner in like one day, with just, you know...

House: Because it was programmable.

Hoff: It’s all programmable, you know. And we pointed that out to the people making the EPROM burners, they should be using microprocessors in theirs, too, and they took that to heart!

Mazor: The genesis of this product line was that we had the SIM4-01 board with EPROMs on it that allowed you to emulate your final program. But our customer was confronted with the issue of writing a symbolic program, and assembling it. So a little bit further in that gestation, we actually had an assembler written in 4004 code and stored in EPROMs. It ran on a SIM4-01 board. So the customer’s process went - typing up on a teletype and punching paper tape in source code, plugging in the four assembler ROMs on the board, running his source code through the assembler and producing object code on the teletype, unplugging his assembler programs, programming from his paper tape into his EPROMs, and plugging his EPROMs back in the board. Now this sounds like a very messy deal, but it did provide a mechanism for the customer with a teletype and this little board to create and develop and debug his own software.

Hoff: It sounds like a horrendous procedure, but anybody who had a PDP-8 and ran Fortran on it went through an ordeal that was an order of magnitude more difficult than that.

House: Oh, is that right?

Hoff: That’s correct.

Feeney: Well, I think this is kind of the prototype of software distribution where you now go out and get software on a CD-ROM, and at that time we were using the 1702s as the vehicle for distribution. But you know, watching all of this evolve, and watching our customers work with this was kind of interesting, because it was after January of ’72 that I got involved in marketing, and was doing the product marketing for all of our microprocessors for a number of years. As the customers moved from this barebones type of development environment into finally a complete packaged computer with all of the software, and all of the tools and everything that they needed, it expanded our market. But the customers really wanted to get in, work very closely with the product, and on the original SIM4-01 boards that were available, they could get in and get to quite a variety of different test points on the board and see exactly what their systems would be doing and have a very good tool for emulating those systems. And then with all of the random logic that was there, it became fairly clear that if that logic could be replaced by something [integrating the random logic], we could even make the customer experience better; and that goes to Federico where he carried things through further with the 4008, and the 4009.

Faggin: Yeah, it was obvious that people wanted a microprocessor, but standard memories also. Also as time went by memories kept on getting better, faster and cheaper and more bits and so on. And we would have to continue to redesign the 4001, 4000 bits, and then 8000 bits, and so on, so it would’ve been an expensive proposition to keep that family growing at the state-of-the-art level, so it was clear that we could have a way to build a system with the 4004 that was cheaper by developing a couple of chips that would interface the 4004 to extended memories, and those chips I designed very quickly, because they were relatively simple. But I came up with the idea, that was the 4008 and 4009 that proved a boon. A number of customers were selling boards, single board computers for control applications. ProLog was a company that was a big user of those boards for control applications, where they could interface simply with existing standard memories. And then later on, I came up with the idea of the 4040, which was an improvement over the 4004, because it doubled your registers, and you had a real interrupt instead of a flimsy test pin, and a bunch of additions. That product was intended to interface with standard memories, and to give a mid-life kicker to the 4004, which was clearly going to die some time later, because a new generation 8080, and the N-channel technology was coming up.

House: And the 4040 came out when?

Faggin: The 4040 came out early ’74, and then it probably was in production until probably ’78, ’79, something like that.

House: And that sort of gave birth to the 8048.

Faggin: That eventually was superseded by the 8048, which was introduced in ’77, ’78.

Mazor: So just let me reiterate. 8048 is a single chip with ROM, RAM, and CPU on one chip, and more or less followed in the controller applications that the 4004 carved out. And so I think Intel’s product line then divides into these microcontrollers that contain programmable memory, and microprocessors that do not.

Hoff: 4004 gave birth to the embedded controller, and 8008 to the data processor.

Faggin: Yeah, that is clearly a distinction, almost all the applications of the 4004 early on were embedded applications. Yeah, [maybe] not a single chip embedded application, but an embedded application.

Hoff: That’s one of the things, too, that the media today still seems to be clueless about - embedded control. They always think of the desktop computer as computers, but the degree to which embedded control is used everywhere is something that seems to be lost on them.

Faggin: Well, it’s at least a factor of 20:1 in microcontrollers versus microprocessors [in consumption]. At least 20:1.

Hoff: At least.

Feeney: Even inside of the computers that are sold today.

Feeney: Yeah, the keyboard controller, the Ethernet controllers, the...

Faggin: The touchpad controller; it goes down the line. There’re probably 20 microcontrollers in a single PC.

Mazor: If I could also add something about what was going on in the competition. Maybe my friends here can add something as well. As we came out with the 4004, there were companies in the semiconductor business who thought we were making a big mistake, because we were coming out with a brand new architecture, and a brand new instruction set, and that meant that the customer base had no existing software that they could utilize. So RCA as an example, decided to make a PDP-8 compatible chip, I think they called COSMAC. However, the fact that their class of machine stores the return address for subroutine calls in main memory, made it only useful for a system with read/write memory, and not with ROM. So it promptly failed in the controller market. Fairchild, which flipped over what we did, decided that what the world really needed was a bit-serial machine, and they came out with a 16-bit processor, but it was bit serial, which we had rejected. So it’s interesting that we created some ripples in the marketplace, and competition responded. I can tell two anecdotal stories, not with a high degree of certainty, but with regard to the 4005, my office mate again was someone from MIL, Ken Au who was their engineer. My role was to translate those specifications and get them stored in the logic design.

Well, when Faggin succeeded with the 4004, we abandoned our efforts. MIL went back and did that chip. But in the end we did compete in the marketplace with the 4005. I can remember a personal situation as an applications engineer, where I'm confronted with a competitive situation, and it was 4004, you know, our little son, versus the 4005, our little daughter, <laughter> and I had to argue why, one was better than the other, in this case, the Intel product. The other is that I was called upon to go to a calculator manufacturer and to give them a specification. But we couldn't give them the 4004 specification. This was Victor Comptometer in Chicago. And I did give them a specification for the 4005. Now it turns out eventually we competed in the marketplace with a very successful product called the Rockwell PPS-4. If you studied the PPS-4, I will claim you will find an awful lot of similarities between that and the 4005, and it turns out that particular calculator manufacturer did a lot of business with Rockwell when Rockwell was new in the calculator business. I think Shima can also tell a story about the 4004 going to a Japanese manufacturer and perhaps getting picked up.

Shima: One day a 4005 stick went to the Sharp. And Sharp gave that stick to NEC. Then NEC started to develop the four-bit microprocessor, and it was quite successful.

House: So what did NEC call that?

Shima: ViewCom-4.

House: ViewCom-4, okay.

Mazor: So my point is there may be more origins of the microcomputer history here than just the Intel ones.

House: Some legitimate and some bastard.

Faggin: Yeah, but there's not a question that even in 1970, '71 it was a race. I mean, really, I mean, if you look at TI, for example, we helped them it looks like. It looked like, but still they were developing an eight-bit processor which was essentially the 8008. And they came out in April/May timeframe with that product. We heard-- I heard from Vic Poor that it never worked. But TI claims that it did work. In any event, whether it worked or not, it was later than the 4004 by one or two months. That tells you how much execution was important to being first in the market.

House: Tells you how important it was for your wife to go to Italy.

Faggin: Yeah! Absolutely. I mean, if we had been two or three months late, TI would've claimed that they were the inventor of the microprocessor, right or wrong.

Hoff: By the way I did have a meeting with Gary Boone [of TI] at one point, and Boone told me that TI did not start working on their version of the 8008 until after they heard about the Intel version. They had the Intel target spec. And they were working for Computer Terminal on a three chip set. In other words, the registers that they [Computer Terminal] originally approached us for was one of the three chips of the processor. It wasn't until TI heard that Intel was going to do it all on one chip that they decided they would do a one-chip CPU.

Faggin: And that was about the time when I started working on the 4004.

Hoff: Yeah.

Faggin: In fact, a little earlier because the first few months I was working on the 4001 and 4002 and 4003, and therefore, you know there was a delay from when I joined the company.

Hoff: This was actually probably before you joined Intel.

Faggin: That's right.

Hoff: Because that spec went out around January or February of 1970.

Faggin: It took them about a year and two months or so -- a year or three months to get to what they claim a working chip, but it was a couple months too late. Otherwise, they would've been right to claim that they had done it.

Hoff: There was another interesting aspect of it, though. They did not understand our interrupt capability.

Faggin: Well, neither do we because the interrupt capability on the 8008 was so late. <laughter>

Hoff: It was a little-- well, there were ways, but the difference was that the TI chip did not change the restart instruction from a jam load to a call. We made the restart a call, so you could use that single byte instruction for the interrupt, and they did not have it. But at least in one of their patents they argue that you use the restart to implement an interrupt. Which means you just wiped out the status of the processor.

Faggin: The other thing which is important is that their chip size was exactly twice the chip size of the 8008.

House: Was it metal gate?

Faggin: Metal gate. So that's the only example that I have that you can actually point and say, "Here is A/B comparison. This is twice the size of that, and it's exactly the same function."

Mazor: However, if you want to look at it from a patent point of view in retrospect, and if we could all learn from this, and that is had we done the three instruction-set computer, we could've patented the computer, and we might have had patent rights that would be important. Whether it really is small or not, and efficient, or as good, is a different story. I want to give kudos to my associates who did such a fine engineering job, and pulled off this magnificent family that they did, and also to my boss, Ted. I have to tell you two more stories. Ted came up with two ideas using the 4004 family at work. One of them was a videogame, and he had one of his engineers, Glen Louie produce a game called Space War, using the MCS-4 set on an ordinary TV set. And so we were demonstrating a videogame. Now one of the problems in the applications department, you always have these sort of crazy ideas, and you get to pursue them. You don't get to go into that business, and I assure you when we showed a game on television to our management, they said, "Who would ever want one of those?" <laughter>

House: That's a crazy idea!

Mazor: Another thing that Ted came up with which was quite clever, at one time we had an idea for a calculator chip, a single chip calculator, and then he had the idea of putting a ROM on that that was basically pushing the buttons of the calculator to do sine and cosine and higher level math functions as opposed to simple four function. Again, some of our slide-rule carrying management couldn't really see that that one would have been an application for scientific calculators. Now again, it is the mania of an applications department to think of crazy ideas, and Ted certainly had a lot of forward-looking ideas on applications, since both of those applications came to market fruition.

Hoff: I have to say I have very mixed feelings about marketing on a number of issues. I mean, traveling with marketing people and seeing what they go through, I came away with the greatest respect; but then on the other hand, you say something like the scientific calculator, which I was discussing with the marketing people, and they concluded, "Well, the market might be for a 1,000 machines a year." <laughter> And you say, "Well, they're in the business. They supposedly know."

Faggin: It wasn't too long after the 4004 that the HP35 came out.

Hoff: Exactly.

Faggin: And so there were a bunch of other people thinking the same way.

Feeney: Well, as all of these things came together it became very much this convergence of technology and enabling functions—smaller displays and being able to have the LED displays, then liquid crystal displays. And it's a matter of just having the right elements at the right time, and I think this is what Intel had with the 4000 series when it came to market. They were the right elements to go in [a system design], and albeit, not fast enough to meet all of the customers' needs, or meet the customers' desires, but they were able to be used in digital scales, in cash registers, in small control devices. Traveling with Bob Noyce right after the announcement, he was speaking from the point of view of, "Hey, it may be more pervasive than the fractional horsepower motor," and he wasn't even close. It was more pervasive than that! But again, it's making the right decisions, and going the right path. Ted, you were talking about the TI version of the 8008, the initial spec, and the reason they [TI] were at three chips [their initial spec called them to be doing a serial machine]. And as soon as they found out that Intel was doing a parallel machine, that's what caused their move to go and emulate exactly what Intel was doing; and that's what enabled them to get a product at least into design as quickly as they did, because they knew the approach. They knew all of the clues and everything that we had decided to do.

Mazor: Can I turn a few more generalities if you don't mind, and that is Ted and I working for a memory company, and Ted was confronted with a problem, and his solution was, "Gee, I wonder if we can solve it by building memories?" Namely, the ROM and the RAM in the 4000 series. The other is we were presented with a real customer problem, and when we were working on a CPU chip, it had to solve the problem. It had to be done to make that thing act like a desktop calculator. So it wasn't creating a computer in the abstract, but creating a computer to solve a particular customer's problem. And we had a customer who knew very well what he needed, and that gave us the opportunity. He didn't change his mind from day to day. So we had a customer who knew what he needed and was able to see how to apply our products to it.

House: And so there were a number of other people that were involved in the program. Obviously, we can't include everyone, but one person we invited here today, was not able to make it for a health reason, was Tom Rowe who worked in the process side. So he's a silicon process guy working on perfecting the various processes that were used, and making sure that they yield, and we can actually make these products. And possibly there were others that should be mentioned.

Faggin: Well, in my group, Paul Matrovich was my technician, and he really helped in all phases of testing, building the testers. You know, we built a team of electronics for the tester that Hal was mentioning earlier, so that we could also set levels and detect some of the more analog kind of measurements on the chip. And Charlie Corbin another technician that came in toward the very end of the project. Then Yung Fang was an engineer that came in also toward the end of the project, and helped doing the detail characterization, taking measurements and making sure that the product was working under all kind of conditions-- temperature and voltage, and so on. And I think I mentioned the three draftsmen. Rod Sayer was there for all the chips. Julie Hendrix did the 4002 and the 4004. And then Barbara Mannis just helped about one month on the 4004. I certainly want to acknowledge them. They did a very good job.

House: Any other names you want to mention Shima San?

Shima: No, I told everything.

Hoff: We might want to mention there was Norm Shanks that did quite a few of the sim boards and things like that, which were very helpful in those early development tools.

House: And we mention Ed Gelbach. I think Hank Smith had some roles?

Hoff: Oh, yes, yes. I mean, both Hank and Ed were very instrumental in getting this thing launched, and putting together all of the support tools, and so on, that go with it. The datasheets and user manuals and getting all that stuff published.

House: Is there anything else anyone would like to add? No? Fine, okay, thank you for joining us all today.

Faggin: Thank you!

Hoff: Thank you.

Shima: Thank you.

Mazor: Thank you.

Feeney: Thank you.

END OF INTERVIEW

