MASATOSHI SHIMA

An Interview Conducted by

William Aspray

IEEE History Center

May 17, 1994

Interview # 197

For the

IEEE History Center

The Institute of Electrical and Electronics Engineers, Inc.

and

Rutgers, The State University of New Jersey

Interview: Masatoshi Shima
Interviewer: William Aspray
Place: Tsukuba (Shima's Office), Japan
Date: May 17, 1994

Aspray: Could you begin by telling me something about your early life, what your parents did, when and where you were born, and your early education?

Shima: I have no interest in that area. There is no relation between my parents and my career.

Aspray: I see. The reason I ask the question is that when historians and social scientists are studying people, they try to obtain information on many different people so that they can do statistical analysis. If you'd rather not talk about it, we don't need to. Could you tell me about your formal education, your higher education?

Shima: Yes. I wanted to study organic chemistry. I went to Tohoku University. When I graduated from the university, the business of chemistry was in decline and I was not able to find a good job. I talked with my professor, and the professor suggested that I go to a software company. I joined up with Busicom. They asked me to work in the software area in scientific programming. At that time Busicom Corporation had three kinds of business areas: one was called mechanical calculators. The second business was desktop calculators. The third business was computers. They were selling Mitsubishi mainframe computers, and they also developed many kinds of software for Mitsubishi, including operating systems and applications software.

Aspray: They were mainly devoted to business applications?

Shima: Both business and scientific. They were also selling business computers in Japan--what we call office computers--which were imported from France. I wanted to do something related with science, but the majority of software was for business. I didn't like it. At that time, the languages, which I was able to use, were Fortran, COBOL, and Assembly. The memory size was only 64K bytes, which meant it was pretty easy for me to do my job. But I didn't like the easy job. And I didn't find business applications interesting. So I asked the company to move me from the software area to the hardware area. As I told you, they had a desktop calculator business, with a factory in Osaka. After six months of software's training, I went to the Osaka factory to join the desktop calculator group.

Aspray: Had you had any experience either in software or hardware before you joined the company?

Shima: Nothing. When I learned organic chemistry in Tohoku University, it was just the time that people introduced the mainframe computer for analyzing the structure of organic chemical compounds.

Aspray: Such as the structure of proteins?

Shima: Like that, yes. Therefore, in the university I had only heard of computers, Fortran, and such things. But I had never studied computers, electronics and so on. Just before I went to Osaka factory, I bought only two books, which I studied. One was <u>Digital Computer</u> by Dr. S. Takahashi, and the other was <u>Logical Mathematics and Digital Circuit: Introduction to Automation</u> by Dr. K. Udagawa. I studied those two books a lot for about three months.

Aspray: I see.

Shima: I also studied about one month of a course related with the automation, which is related to something like logic. I thought it might be enough for moving to the hardware area. I moved to the desktop calculator group in 1967, probably in November or perhaps December. That was quite an important time for me. In 1966 it was the era of the transistor for desktop calculators. People joining the company in spring 1967 had to start from transistors. But fortunately, the technology was moving from discrete to integrated circuits such as diode transistor logic (DTL). That DTL was introduced into the desktop calculator application in 1967.

In the Osaka factory, I joined the desktop calculator's developmental factory. When I joined this group, they had almost completed the logic design, using TTL and a shift register as memory. My responsibility was to build the engineering sample and also develop the new keyboard and its controller. Through this job, I got to really study desktop calculator logic designs and also got much experience on p.c. board layout, logic partitioning, system bus, system implementation, and real time control. Especially, p.c. board layout experience became a great help when I designed the logic of the 4004 CPU. I wanted to make 4004 logic schematics, which would be useful for layout plans, since Intel was not able to hire an experienced mask designer.

Aspray: The previous machines, had these been vacuum tube machines or had they been electromechanical?

Shima: They never used vacuum tubes. They started from relays, but that was quite a big box. Mainly, they used that hardware for billing machines. As you know, the electrical desktop calculator was invented in England. The company's name was Anita. Then many Japanese companies imported and looked at the machine thoroughly; it looked nice to them. Everybody copied that one, improving the machine, of course.

I was quite lucky to move to desktop calculators because I didn't need to study transistors. I was able to start from the integrated circuit. What I had to do was study logic. But the desktop calculator's DTL device number was less than 200. This meant I was able to memorize all of the logic in my brain. The desktop calculator itself was like a computer system. That is it has a main processor, memories, display, and keyboard. Sometimes it also had a card reader, for example an IBM card reader, for storing the application program. Sometimes it also had a CRT to display internal accumulator and memory states. This meant I was able to study both systems and logic. That was in 1967 through 1968. At that time almost no foreign company was manufacturing desktop calculators for business use. This meant that Japanese companies did OEM business, like personal computers today. Busicom sold its desktop calculators through NCR, with their brand name, in the United States. They also exported with sales companies, with their brand name, to Europe and Asia and so on.

Aspray: None of the traditional American manufacturers of desktop machines were making them? For example, were companies such as Victor Comptometer in this business then?

Shima: Victor was really imported from Japan.

Aspray: Oh, they imported from Japan?

Shima: Yes. There was only one company. Monroe. They used a very funny breaking slow serial printer mechanism, which was not competitive in the worldwide market. They used it instead of impact line printers. OEM business meant that many sales companies asked Busicom, "Your product looks okay, but I want you to change the such-and-such." Especially as there were two different ways for calculating: one method is like the United States method. It looks like an adding machine. For example, two, plus, three, plus, then hit the total key. But in Japan there was a different way: Two, plus, three, then hit equal key--using the algebraic method. In the 1964 Tokyo Olympics, Seiko developed the impact line printers for recording the time records. That was very compact, high speed and reasonably priced. Every desktop calculator company wanted to use that printer for its desktop calculators--their printer model. My manager said he had developed several desktop calculators, starting from relay, transistor, DTL, and also TTL. Before he joined Busicom, he used to work for CDC as a custom engineer for computer maintenance.

Aspray: Going out into the field to deal with customers?

Shima: Yes. He knew computer hardware well. But he was getting tired of the traditional logic implementation techniques such as combinatorial logic. We call it "hard wired." When the desktop calculator's business became large, the competition of product development grew more intense. Within a shorter period of time, he had to develop different models of desktop calculators. Then he asked me, "You studied computers, software and logic. Somehow I want you to introduce software technology into desktop calculators." Then he said, "I want you to define the instructions. I also want you to make the program." This was April 1968, just one year after I joined Busicom. Then I planned to combine computer software technology and desktop calculator's, hardware, into one idea. We called it a decimal computer. I planned to use read-only memory. [Shima shows a diagram.] Here you can see three registers--entry register, accumulators, and one more register to be used for multiplication. Here you can find the two arithmetic units--full adders-- here. This is like a decimal computer's main block. Here is the program control unit. You can find out the name of a micro order. This micro order is the Macroinstruction set into desktop calculator from the read-only memory. I introduced the decimal computer's architecture for desktop calculators. I developed this desktop calculator with printer in 1968 and 1969.

Aspray: At this time did you have any contact with other people who were doing similar kinds of design jobs?

Shima: What do you mean?

Aspray: Well, were there professional meetings where you could talk to other people doing something similar?

Shima: You mean, with different companies?

Aspray: Maybe with different companies.

Shima: Nothing.

Aspray: Nothing at all?

Shima: Yes it is prohibited to talk about company secrets in Japan. At that time, the desktop calculator was a really hot product with a high profit margin.

Aspray: You talked with some of your colleagues at work, but it was your responsibility?

Shima: Yes. But you can see how it was. Usually, the people had only one experience: software or hardware. Fortunately, I got the software experience before joining the hardware group. It's relatively easy for me to integrate the decimal computer's instruction architecture and desktop hardware architecture. By the way, probably in late 1968 or early 1969, Sharp developed the LSI desktop calculator, which was developed by Rockwell. At that time Sharp designed the logic and provided logic schematics to Rockwell. Then Rockwell did a circuit design. It was implemented with only four chips--including a clock. Everybody in desktop calculator companies were shocked.

Aspray: Getting down to that small number?

Shima: That's right. Then we got a contact with a consulting company in the United States, in order to research who would be a candidate for Busicom to ask the development of LSI for the desktop calculator. Also we had to start to study semiconductors, and logical implementation techniques with LSI. Therefore we collected technical information and we at the same time contacted American consulting company. Finally we chose two companies: Mostek and Intel. At that time in Japan there were two factories, one in Osaka, one in Tokyo. There was one big problem with the union. Since Busicom started from the mechanical calculator, they belonged to the metal working industry union. They are quite strong. Every year, Osaka factory had to stop production activity during the spring labor offensive. Then, Busicom established Tokyo factory in Tokyo, which does not belong to the metal working industry union. Busicom's strategy was to make small desktop calculators, but with high volume in the Osaka factory. But on the other hand, in the Tokyo factory desktop, their main interest was the high-end business application area, covering business calculators, scientific calculators, office equipment such as billing machines, and teller machines. From the top end to the business desktop calculators, there were three categories: the office computer area, the scientific calculator area, and the business desktop calculator. Once again, I was asked: "We wanted to develop the general purpose LSI to be used for not only desktop calculators for business, but also for scientific calculators and billing machines." I started its project in late 1968 just after I completed the desktop calculator with printer using the decimal computer's architecture with programming technology. However, its new project was not such an easy job.

Aspray: Yes. Did you also anticipate that these chips would be used in future products as well as the current line?

Shima: That's right. For example.... [Searching through papers and photographs] Here is a desktop calculator, which I designed with the 4004. This is a teller machine, which looks like a cash register. This is a billing machine. This also used the 4004. The main purpose for developing general-purpose LSI was that it will be utilized not only for desktop calculators, but also for billing machines, teller machines, cash registers, and so on. Then the Tokyo factory contracted with Intel, and the Osaka factory contracted with Mostek. In the case of the Osaka factory, they brought the logic schematic, actually like a circuit schematic, to Mostek. Then Mostek developed the desktop calculators LSI--with two chips at first. Later they converted to one chip.

Aspray: What were the characteristics of Mostek and Intel that made you choose them?

Shima: It is quite simple. Sharp had developed a very tiny low-end desktop calculator. We were not interested in such a simple, funny specification. Busicom aimed at a more professional desktop calculator for business application and this had to bring far better specifications in the desktop calculator. This meant we didn't want to utilize the metal gate MOS. We wanted to use a high density silicon gate MOS, with which we will be able to expect much higher integration. There were only two companies, Mostek and Intel, who had silicon gate MOS technology.

Aspray: I see.

Shima: In June 1969, I went to Intel. Actually, there were three people from Japan, from Busicom: the project manager, Mr. Masuda; also the senior design engineer, Mr. Takayama; and I was the youngest engineer. I made the product proposal to Intel. As I explained, I had already developed the stored programming technology with the decimal computer architecture. Then, I generalized its idea, using seven LSIs. I made some important area logic schematics to show to Intel. We didn't complete all of the logic schematics in Japan. But we completed 80 to 90 percent. We thought it might be enough for explanation. We planned to make a final logic schematic after we discussed it with Intel. There were two people from Intel. One of them was Ted Hoff, and the other guy was Stan Mazor. Stan was just a programmer, but Ted Hoff had a good background. He had studied computers, and he had also worked at Stanford Research in the computer area. He had good abilities in many, many areas, including computers, software, circuit design, logic design, and simulation. In the beginning of the meeting we thought that once we showed the logic schematic to them, they would understand what we wanted to do. But after several meetings, we found out that they didn't have any logic engineers to understand our logic schematics and they didn't have any circuit engineers to convert our logic schematics.

Aspray: So they knew about memory chips, but not logic chips?

Shima: That's right. Also they didn't know about desktop calculator themselves. For example, Seiko's printer was a line printer. They had only one desktop calculator, a Monroe, and it had serial printer. Therefore we had to explain what these desktop calculators were, what the line printer was, and how to control them. Also, we had to explain the function and control of the keyboard, display and card reader. It was quite a difficult job for us to explain because they had never seen them. After a couple of months of discussions, we were not able to reach agreement. Ted Hoff asked me many questions. I brought some other things. For example, this was the flow chart for the desktop calculator's program. Many times I explained the desktop calculator's program and the macroinstruction set. He had a lot of questions about this flow chart, then and he asked me, "why don't you explain the instruction set?" I explained the list of macroinstructions for the desktop calculator, showing the hardware block diagrams and flow charts and program. But this is based on the macro-level decimal instructions.

Sometime in August, after the deadlock, Ted Hoff came to my office. Always he had a very inexpensive Sharp pencil [what we call in the U.S. a "mechanical pencil"], costing about a dollar fifty. On the top of the Sharp pencil was an eraser. Always he brought that Sharp pencil. He said my basic idea is such-and-such. But he jumped into the room and said, "I found some ideas." But it looks to me there were almost no new ideas. He showed me three boxes, one for the four-bit arithmetic unit, one for four-bit general-purpose register unit, and one for the address stack unit. In the address stack unit, there was a four-level 12 bit address register because I told him in discussion that three levels would be enough for subroutine for my calculator's applications. Then he said, "I have just one program counter." One program counter with a three-level address stack would be enough. His basic idea was that rather than implementing a programmable n-digit macroinstructions, let us make your macro-instructions by simple one-digit instructions using a program. This one digit meant four bits. I thought that looked good. But he didn't show anything [in writing] how to do it. Therefore, it was quite a difficult decision for us.

Aspray: That is he didn't back it up with a paper that presented these ideas?

Shima: Yes.

Aspray: He just told you about them?

Shima: Yes. He showed us some blocks. But, as I said, just three blocks, four-bit arithmetic, general-purpose register block, and an address stack. Also on that chip there were many I/O's. I asked him, what are these I/O's? He said, "you may need many I/O's for the keyboard." I asked him how to do it. He said [gesturing]...

Aspray: "I don't know."

Shima: Yes. At that time I had a problem with this. I want someone to interpret this one. The main problem was that Intel wanted to develop only one LSI. Thus Busicom has to use the standard random-access memory for data storage, the standard read-only memory for program storage, and use TTL for the system interface. Still, there was the problem of how to control the keyboard, the display and the printer. I said, it is a problem, but the main reason why we wanted to contact with

Intel was to make a desktop calculator with only LSI components. We didn't want to use a TTL at all. Then, the discussion started among Ted Hoff, Stan Mazor and me. Stan Mazor's role was quite important for finalizing the 4004 idea. He acted as a go-between Ted and Hoff and me. He assisted me when I wanted to explain my problem and ideas to Ted Hoff, also he interpreted Ted Hoff's ideas for program examples to me and so on.

That discussion continued for about three months. One day, Ted Hoff brought the instruction set list for a 4 bit binary processor. Then, many problems surfaced. There were six major problems. Ted Hoff's basic idea was good, but without solving those problems, it was impossible to his ideas in the real application. Also, standard SRAM and ROM were used in computer applications and they were not used in consumer application area, and also they were quite expensive.

The first problem was the lack of LSI systems. Ted Hoff and I agreed to develop not only 4 bit binary processor but also the ROM chip and RAM chip which would be connected through a 4 bit time multiplexed system bus. Quite strongly I told Ted Hoff that we needed an LSI family. At that time we thought we would be able to use a 40 pin package. But we did not realize Intel was a memory company. Intel had only two choices for packaging: the 16-pin package and the 18-pin package. Intel had never told me there would be problems in packaging. Ted Hoff knew about this restriction. Therefore, he brought 4-bit time multiplexed system bus instead of much wider bus idea that I wanted. However, when he applied for the patent, he added my name on it.

The second problem was how to do the decimal operations. Mainframes were able to utilize a huge amount of memory. They had easy-to-use binary-to-decimal and decimal-to-binary converting programs. I said that one way to convert binary to decimal and decimal to binary will require me to utilize read-only memory for program and conversion data table. It is quite expensive! At that time the maximum chip size for read-only memory was limited, which means that I was able to utilize only 2K bit per chip. Which means only 256 bytes. Quite small. Even if a 4 bit binary processor will be able to simplify the LSI hardware, with an extra ROM chip, the total cost will exceed the cost of decimal computer approach.

Aspray: Yes.

Shima: The third problem was a lack of keyboard control. Suppose a keyboard with X-Y matrix will be scanned to detect which key was depressed, whether more than two keys were depressed, and also whether any key was not depressed, it was nice to have the code conversion instruction in order to detect above mentioned status. I proposed the instruction for keyboard process.

The fourth problem was the toughest problem. That was how to control real-time operations. For example, in a simple desktop calculator; there are keyboards, printers, and displays. But I planned to bring more peripherals, such as typewriter for billing machine, paper-tape reader and puncher, and card reader. All of these peripherals have to be controlled in real-time mode even during calculation. I asked the question to Ted Hoff, how to do you do it? What kind of instructions are prepared? How to implement it by program? No idea came out of Ted Hoff. For example, today, you can use DMA, interrupt controller, much more powerful processors, high operating frequency, large memory, and so on. But I didn't want to utilize such a complicated approach. Therefore, I proposed a special conditional branch, sensing the outside external signal. That means that the external input signal (TEST input) can be one of the conditions in the conditional branch instructions.

Therefore the basic idea from Ted Hoff was quite nice. But to use his idea for real applications, the detail was not so good. As I said, lack of system concept, lack of decimal operations, lack of interface to the keyboard, lack of real-time control, and so on. The fifth problem was the lack of register indirect jump instruction. To utilize a subroutine, you always have to use two bytes. Here I showed the total number of steps for calculators was roughly 200 steps. I did not want to add an extra ROM chip, I wanted to use only 200 bytes instead of 400 bytes. Then I asked to add the register indirect addressing mode, which solved my problem. Also, this added register indirect jump instruction was able to solve the fourth problem altogether. With its register indirect jump instruction, the interpret routine was developed for the emulation of desktop calculator's macroinstruction. The conditional branch instruction with TEST condition was inserted in the interpret routine in order to detect the outside status. The longest macroinstruction's subroutine was about 1 millisecond and the timing signal from the line printer was used for TEST input signal. So every 1 millisecond, the printer's basic timing can be detected through TEST pin, and it made it possible to control the printer, the keyboard scan, and the calculation in real time manner. This invention brought the 4-bit binary processor idea into the practical application area.

At first I brought the basic idea, of a decimal computer with a macroinstruction set, a flow chart and program, and some logic schematics. I was not able to do more than that. But in the case of Ted Hoff, he was able to start from there. I started from the very bottom, like how to introduce the programming technology into calculators. I was able to reach to decimal computing architecture for the desktop calculator. In the case of Ted Hoff, he was able to start from here. [gestures high] Then he brought--finally the idea of the binary computer. He had studied IBM decimal computers, and also he had lots of experience with the PDP8. However, his experience was based on the large memory size, and there was a lack of experience in the real time control and systems.

Aspray: That had a small instruction set, didn't it?

Shima: Yes. That's right. With all of my experience, my basic idea, Ted Hoff's idea, and many really valuable improvements. Ted Hoff and I were able to realize the 4-bit microprocessor idea.

Aspray: Before you go on and tell more of the story, Intel was a fairly small company at this time. How did they view Busicom as a customer? Was this a really important business for them? Just what was their attitude?

Shima: Dr. Noyce wanted to bring out something new. He established Fairchild. As you know, before Fairchild he was at Shockley Research. Shockley's mentality was limited to research. Shockley did not pay attention to the product development, utilizing his research activities. On the other hand, Noyce's mentality was development. He developed many, many good products in Fairchild. But his mentality was limited to development. Many, many times he brought out new products at Fairchild, but he was not able to transfer the product into production in a timely way with high volume.

Aspray: I see.

Shima: TI copied Fairchild products. National Semiconductor copied Fairchild products. For example, at that time, if you wanted to use new products, you could get the sample from Fairchild for engineering models, then buy the products for production from National Semiconductor. Another example, if you get the catalogue from National, its original catalogue was from Fairchild. Somehow, National got its catalogue, then applied the National's label on its catalogue and distributed it to customers. Anyway, once again Dr. Noyce wanted something new with newly developed silicon gate technology, establishing Intel. He intended quite strongly to put energy into not only new product development but also production itself. Traditionally mainframe computer used non-volatile memory, like core memory; its core memory was quite big. Therefore he wanted to introduce semiconductor memory. It was impossible to replace all of the computer mainframe memories by semiconductor memory at once, but step-by-step he wanted to replace core memory with semiconductor memory. To do that he needed new MOS technology. He developed the silicon gate MOS at Fairchild. But the parent company was not able to find out how silicon gate technology would become important in the future. Finally Dr. Noyce established Intel and hired many engineers from Fairchild, who know the silicon gate technology well. Intel then developed both static and dynamic memory, which is 1K bit. Intel introduced a D-RAM memory into market. But when Intel developed DRAM, it took a lot of time to develop the chip itself. The SRAM chip is relatively easy. On the other hand, there was a big problem in the dynamic memory. After the chip came out, they had to study many, many things. They had to study the physics, such as how to capture the electrons under the gate for a long time. While they were studying the physics, the concept of PROM came out. Then they developed programmable read-only memory (PROM). But to use such new generation's memories like DRAM and PROM, it takes lots of time before the end-users transfer their system product into production with high volume. This meant that Intel needed money for their daily operation. Simply put, they needed cash.

Aspray: So they were then willing to look for customers to do custom work for?

Shima: That's right. In the beginning, in 1969, Busicom agreed to pay $100,000 for development of LSI. That looked like quite a lot of money for Intel. But after a couple months of discussions with me, most probably they found out there was too much LSI to be developed. But there were no logic engineers or circuit engineers in Intel. Therefore, Intel top management asked me to simplify the logic, which we brought to Intel from Japan. Parallel to that, he asked Ted Hoff to find some new way, a better way. After Ted Hoff's idea came out, I worked with Ted Hoff and Stan Mazor on how to solve the problems that I have already mentioned to you. Before the end of 1969 we were able to reach the final goal. In the final plan, I added the 10-bit static shift register to make it useful as a printer's buffer and keyboard interface. There were many improvements in the instruction set, making the RAM organization suitable for a calculator, the memory address information transfer and so on. By December, I was able to develop the key program in an area of performance and program capacity. Then I wrote the functional specification.

Aspray: No, I don't know this.

Shima: There was a patent problem between Zenith and TI regarding the one chip micro controller. This was from Ted Hoff.

Aspray: So this was Hoff's deposition?

Shima: Yes. For example, this is the functional specification of the 4004. I wrote its functional specification, which is called “User’s Manual”. Then Intel added something, which they called the 4000 Family Functional Specifications. This drawing is my handwriting.

Aspray: I see. All of these diagrams at the back of the depositions?

Shima: Yes. Here are some block diagrams for each unit. This is the random access memory, read-only memory, and shift register to be used for I/O's.

Aspray: So the 4001, 4002, 4003, and 4004.

Shima: Yes. That's right. These are functional specifications for each instruction. My English was quite poor. Therefore, I had to explain things with drawings in order to avoid a misunderstanding.

Aspray: I see.

Shima: We finalized all of these things by the end of 1969.

Aspray: Work that was done by Hoff and others at Intel under contract to Busicom, anything that was a patent able idea was Busicom's patent?

Shima: At that time there were several problems: Busicom told Intel that Busicom was going to pay the development charge, which meant that Intel was not able to write a patent because Busicom was going to pay for the development charges, keeping the rights to manufacturing. Also, everybody at that time thought we just brought the technology from the computer area, and there was nothing new but that we just picked up some techniques from the computer area and some from the desktop calculator areas, then integrated them onto a chip. I keep telling you, Intel was a memory company. Their main expertise was in memory. Therefore, in the case of memory, even for very small things, they asked Ed Neer to write patents. But in the case of logic, it was considered a minor thing for Intel. Therefore, they had never thought how important it would be. They never pushed Intel engineers to patent things. They never pushed Busicom to patent.

There was one problem in Intel. Ted Hoff had some basic ideas. What he documented were three boxes in processor, the first cut instruction list and a 4-bit time multiplexed bus. However, he neither had worked on details nor showed the details. In order to bring the rough idea a usable stage, I worked on all of the details, documentation, and logic implementation of 4004 microprocessor. Everything Intel will patented it, had to use my ideas.

Aspray: I see.

Shima: Also at that time, in the case of the United States, rather than writing a patent, companies found it better to develop a functional product as soon as possible. That is a much stronger way compared with patenting. Many times I was told by Intel that if an idea came out you should develop the product first, then bring the paper to the IEEE, then write the patent. Therefore, product, paper, and then patent. That was the priority. But after 1975 or 1976, this pattern changed: write patent first, then write the paper, and afterward make the product. Motorola did it in such a way for 6800. Thus the paper was written two years before the real 6800 sample. But in the case of Intel, after we made the chip, we then sent the functional specs to the customer. For a limited number of customers we sent functional specifications. But in the case of Motorola, before the product came out, they sent the document to the customer, saying there will be such-and-such a new product.

Aspray: So it's just a marketing scheme in a way.

Shima: That's right. The success of 4004, I think, came from very many things, e.g. my decimal computer's idea, and the concept of systems by LSI's printers. I also had the background in software. I also knew all of the desktop calculator's logic. That included a real-time I/O control. Ted Hoff had experience in mainframe computers and decimal computers. We were able to integrate those experiences, backgrounds, and ideas together. Also at that time as I told you there was one more engineer, a software engineer, Stan Mazor. He was not able to bring out any new good ideas. But he was able to transfer my ideas to Ted Hoff, and Ted Hoff's ideas to me. He did this quite well.

Aspray: I see.

Shima: Without Stan Mazor, we would not have been able to reach that four-bit idea so quickly. When I explained something related to software to Stan Mazor, he interpreted what I wanted to do, what program I sought, then he told it to Ted Hoff.

Aspray: I see. That's a very useful function he had.

Shima: That's right. But unfortunately, by the end of 1969, Busicom and Intel, were not able to reach a final agreement, and we had found out we needed a few more months.

Aspray: What were the problems?

Shima: Just the right conditions, development changes and so on.

Aspray: I see. Just typical business problems?

Shima: Yes. Then I came back to Japan to finalize the document, and to finalize the program. For my part, the product was desktop calculators instead of LSI itself. Therefore, I made a lot of programs. There were several choices for the instructions modifications. The data exchange instruction between accumulation and the general purpose register was chosen instead of data move from the accumulator to the general purpose register. I finalized all of them by the end of March 1970. Then I visited Intel once again. That time I was told from Busicom that my main job was just to check what Intel was doing. When I went to Intel, unfortunately, I found out they had not done anything! One engineer, Federico Faggin, had joined Intel one week before I went there. Ted Hoff said, "I'm going to do some other things. Therefore, Federico will do everything for you."

Aspray: I see.

Shima: But I didn't know Federico had joined Intel only one week before I came. I said, "Show me the schematic." And he said, "I just joined Intel. Tell me what is a 4001, what is a 4002, and so on." We fought it quite strongly because Busicom paid money and Busicom sent all of the detailed documents. At that time when I went to Intel, nothing had been done. I was mad! But finally we decided that I would do the 4004's logic schematic, and Intel will do all of the other jobs. Because I was not able to do both a circuit design and also layout. What I was able to do was to implement logic and generate the test pattern.

Aspray: I see.

Shima: Then I asked him how to draw the logic schematic to be usable for circuit design. He said, "It is relatively simple, MOS layout is in only two dimensions." Nobody was able to use three dimensions for semiconductors. Suppose you placed metal this way (Y direction), you have to bring the signal from this way (X direction), to this way by polysilicon or diffusion. Simple. Then place transistor. That's all. Fortunately, I had lots of experience in layout of PC boards. Therefore, somehow I had a very similar experience to semiconductor layout. Thus, I was able to make a logic schematic very quickly. I will show you two things.

This is the 4004 logic schematic, and here is the general-purpose register. This is the four-level stack with a one-program counter. Here are 16-sets of the general purpose register. This is the main logic controller including the instruction register. I made up the logic schematic diagram this way. This is the chip photograph; layout is exactly the same as logic schematics. Once the logic schematic was done, it was relatively easy to do the circuit design because you can see what material has to be used, and also how much length. It is easy to do not only the circuit design, but also the layout. Federico did a good circuit design, introducing many, many new techniques, such as bootstrapping. We were able to make all four chips within one year, with only one mistake in the memory area.

Aspray: That's very impressive.

Shima: After I made a logic schematic, I did two kinds of jobs: one was with the logic simulation, which tool was developed by Ted Hoff. But there was one restriction. We paid $100,000. The budget for logic simulation was $5,000. Therefore, it was quite simple. When we reached $5,000, we stopped the job. At the same time, I sent all the schematics to Busicom. Busicom made a breadboard. They also found one logical mistake. But that logical mistake was also found by the simulation just before it was found by breadboarding. Therefore, we were sure the logic was good. We were just waiting for the chip to be made. After I finished the logic simulations, I developed a test program with a testing company whose name was Pacific Western or Western Pacific. They had the small testers which used a drum to store the test pattern. There I found one mistake. Some questions arose: how to test internal operations such as a carry flag, keeping the tester program to be minimum. Then I added one function on the 4004 processor. There was one time slot, which I had never used. At that time slot, the contents of the conditional flag and the contents of the accumulator were sent out through the 4 bit multiplexed bus. That improvement simplified the test programming work significantly. Then I finalized all of the schematics.

At the end of 1970, I came back to Japan. But before that, I had to visit NCR and the European sales companies, to do marketing for next generation desktop calculators. I asked Busicom for three days for each trip from one place (San Francisco) to the next place (New York). I needed one day for discussion, one day for writing the trip report, and one day for moving. The company said okay, but I did everything within two days. I had one full day for sightseeing! Then after I came back to Japan, I made the development system. When the development system was complete, 4003 shift register and 4002 RAM chips were fully functional. But I was not able to use 4001 ROM. Then, I made a special memory board to emulate the 4001 ROM function. I picked up the standard static memories, and then I made the same logic as read-only memory. I used an IBM card reader for loading the program. I made a set of development systems. Then I waited for the 4004.

I remember the 4004 came to Japan in March or April 1971. It took us a relatively long time to get the components because nobody imported this type of product. But finally, when the chip arrived, I tested the 4004 processor by simple instructions. Before exercising the 4004 processor, I made lots of test programs, to test such-and-such and this one test for such-and-such. When the 4004 processor arrived, I loaded a simple test program through the IBM card reader. I saw that the system looked functional. Since many people were watching what I was doing, then I loaded the entire desktop calculator program. The total size was 1K byte. Then I depressed the reset switch, and once this reset switch was depressed, all of the indicator lamps were turned on, which meant all of the output prints were floating, ready for starting. Then next I released the reset switch. Once you can see the lamps to show the address, accumulators and flag, you can guess what kind of programming was running. I judged the keyboard scanning program was running. Let us see. The system was fully functional, which meant that the chip was fully functional. All of the calculator functions were fully functional. That was most delightful and exciting.

Next, I exercised the calculator's function such that 123; plus, 456, plus then hit total, so printers printed out the result of 579.

Many people have asked me, How did you feel about 4004's completion? I never felt I did a good job. My main job was to develop desktop calculators, rather than making the LSI components.

Aspray: Yes.

Shima: After I developed the 4004, Busicom developed several office machines with the 4004, such as a teller machine, billing machine, and cash register. Its main customer was NCR. Many times people kept saying that Busicom developed the 4004 for only desktop calculators. But many times I tried to speak out against that. Busicom's primary target with 4004 was desktop calculators, but Busicom planned to use 4004 for scientific calculators, teller machines, billing machines, cash registers, and so on, telling the plans to Intel in late 1969. As I said, most of those products were OEM-based systems and no one knows what processor was used in the system. On the other hand, Intel was not able to see the microprocessor. There was one good advantage for using a microprocessor, such as the 4004: I was able to introduce new functions to desktop calculators, adding the extra program. For example, before the introduction of the microprocessor into the desktop calculator with printers, no one used input buffers for keyboards, due to the expense of hardware logic controllers. While the printing was executed, the keyboard was locked or the keyboard was never scanned. I worked with the production people. At that time there were two testing methods: one from NCR, an automatic tester using a robot, and the other one by testing people. But the testing people's key touch was shorter than the robot, I mean the key contact time. We measured the keyboard contact time, and found out that the keyboard must be scanned twice within 10 m/sec. Also, we found out that eight strokes would be enough for input buffering. Then I introduced keyboard function into desktop calculators by adding new keyboard programs.

Aspray: I see.

Shima: That was quite valuable to the customer. We received very good response from many customers and local sales offices. Just after system development engineer used the microprocessor, they found out that they will be able to add higher value, adding more valuable functions by adding only one program.

Aspray: I see. So it was well received.

Shima: Well received, yes. This means with the introduction of the microprocessor, we were able to introduce new functions. That everybody found out. At that time Busicom had some money troubles for mass production. I knew that I would not be able to have a challenging job at Busicom any more. In 1972 only three desktop calculator companies were able to move to very, very high volume with very low price. Therefore, after I transferred these desktop calculators into production, and also taught them how to develop teller machines, billing machines, and so on; in order to enter into the applications where much higher product margin can be expected. I moved to Rich Corporation. There I did many interesting jobs. In Rich there are many, many engineering groups. I joined their business computing group. They had been getting I/O typewriters with IBM electric typewriters and paper tape reader and puncher, adding some functions. They wanted to make interfaces for mini-computers. They wanted to sell their I/O typewriter for mini-computers as a replacement for teletypes, I/O typewriters.

Therefore I developed such an interface between the I/O typewriter and mini-computer.

Ricoh was importing a graphics terminal, for graphics computers: a big CRT integrated with the minicomputer. They wanted to connect that graphics computer to Hitachi's mainframes through the channel controller. They asked me to make such an interface. I had to study what channel controllers and graphics computers were.

A third job: Ricoh was selling business computers. They used a drum instead of a hard disk. They wanted to make a production tester for the drum. They asked me to make a production tester to test four drums at the same time. I said, "I don't know how to do it." They said, "We bought a minicomputer, but nobody uses it. You may use that one." That was NEC's 8-bit minicomputer. I studied 8-bit minicomputers, the CPU, DMA, interrupt controllers, and many, many other things. I also had the chance to study drums once again. Through this job, fortunately, I was able to experience all the necessary technologies for development of 8-bit microprocessor.

The last job I did at Ricoh was a high-speed printer's controller. By that time the teletype typewriter was quite popular. But its top speed was only 10 characters per second. Ricoh planned to develop much faster printers, 30-characters per second--three times as fast. Their plan was to replace the mechanical control so that partial control is done electrically, and partial control is done mechanically. They intended to integrate the mechanical and electronic control together. Then they asked me: Why don't you make a controller? They said, "We want to accelerate like this way, and decelerate like this way." It looks simple, but actually it's not so simple. I did a search of four different types of jobs, then I moved to Intel. Intel had already developed the 8008, but it was not successful. As I said, they were a memory company.

Aspray: Can I ask you a couple of questions before you go on? Can you tell me a little bit about Ricoh's place in the industry? What was its main business line?

Shima: Ricoh's main business at that time was the copier. Their second business was cameras. Then the business computer area. That development group is located in Yokohama.

Aspray: Did they have a particularly large part of the business computer market? Or were they particularly dynamic? What was it about the company that attracted you to go join them?

Shima: One day I consulted with an ex-Busicom sales manager about leaving Busicom. He knew one of the engineering directors at Ricoh. At that time, Ricoh was well-known as the fast growing company in the office automation business. I thought I might be able to have some challenging job at Ricoh.

Aspray: The other thing that surprises me: Here you are trained originally as an organic chemist, and you're now doing not only electrical things and logic design, but you're also doing controllers, which are mechanical as well as electronic.

Shima: You know, I was not a professional in all of these areas. What is your professional area?

Aspray: I was trained as a mathematician and then as a historian.

Shima: I see. Suppose you wanted to know some fundamental things related to your professional area. Can you ask such things to your colleagues or manager in the same research group?

Aspray: You have to ask in accordance with what they know and how they speak about things.

Shima: In the United States, it is relatively easy because people are very open. But in Japan, people never ask questions related to one's profession.

Aspray: I see.

Shima: Nobody wants to say, "I don't know." But I did not have such a professional reason. I never studied about such things in a university. Simply I would say, "I don't know. Tell me. Teach me."

Aspray: I see. And you did it!

Shima: Yes. That was so. Suppose I talked about the printer controllers the engineer for printer’s mechanical controller. I was not a competitor. I was just going to support him, help him, always, and so he could give information to me. For example, when I worked at Intel, I never studied semiconductor physics and manufacturing. But, I studied how to use it very deeply. Therefore, to them I was not a competitor. But after I became successful in microprocessors, many people joined the microprocessor development group. They felt I was a competitor. Also, when I developed the Z-80 I was in the mainstream, and there were lots of competitors. Everybody wanted to take my place. It was a very tough job to develop the state of art product. But, always, I had chances to develop the most advanced microprocessor ahead of competitors. Thus, all of necessary information regarding technologies and marketing information from customers came to me. What I had to do was to study and analyze them, then generalize from them for optimization. Also when I joined the desktop calculator's development group in Japan, I was not in the mainstream. I was going to support some person's job with software technology.

Aspray: Why did you decide to leave Ricoh and go to Intel?

Shima: Six months after I joined Ricoh, there were two people from Intel--Federico Faggin and also Mr. Gelbach, who was a vice president for sales. They said they wanted to improve the 8008. The 8008 itself looked okay, but nobody used it. I explained to them it was because of the lack of a system concept. The concept of the 8008 was the replacement of the character manipulation function in the intelligent terminal. Intel forgot and was not able to show to customers how to solve their problems at the system level. They forgot about that in the 8008. They developed the 8008, but that was all.

Aspray: I see.

Shima: So the customer has to use about 30 TTL to make the system interface. There was no advantage in using the 8008 further. They wanted to develop the 8080. I said to them, if you get a permanent visa for me while I am in Japan, I will go there. They were quite happy, but it was impossible to get the permanent visa while I was in Japan. Finally, after four months effort, they found out it was impossible. Then Dr. Noyce made a call to Ricoh's top management: I need Shima. But that top management didn't know about my name. Who is this Shima? They called me, and I went to the top management people's room. I explained what I did in Intel. They said, "You may go to Intel. But within four years you can come back." I was interested in microprocessor development. I wanted to do something now, and the microprocessor was one of them. Always I wanted to have a challenging job because I saw what Noyce did while I developed 4004. I was watching what he did. I liked his lifestyle. I was quite strongly influenced by Dr. Noyce. Without the invitation call from Dr. Noyce, most probably, I would not have joined Intel. Finally I decided to go to Intel because I had enough background in systems: controllers, minicomputers, architecture and so on. Therefore, when I was asked to develop the next generation of 8-bit microprocessors, it appeared to be quite an easy job for me to define the architecture and improvements because they wanted to keep the compatibility with the 8008. It was relatively easy for me to see where to improve.

Aspray: I see. Do you want to talk about the design process for the 8080?

Shima: At that time, I would say Intel was a memory company. They had no one who had experience in logic design in their management. This meant they accumulated much useful and valuable know-how in physics, chemistry, semiconductor processing, packaging, and memory. But Intel had only one experience in logic, the 4004. The 8008 was the derivative product of 4004 for implementation. I will explain how I designed 8080 later.

Aspray: Was their attitude the same as other companies?

Shima: A little bit different. For example, in the case of Fairchild, they had been using CAD systems. They used the standard cell approach. Fairchild provided the list of standard cells to a customer, then the customer designed with the standard cells, afterward Fairchild used an automatic place and route. Unfortunately, it was not successful because the chip size became too high and was too costly. In the case of AMI, they had too many logic engineers and they were a so-called custom house who designed logic and circuits based on customer's requirements. In the case of National Semiconductor, they had a lot of logic engineers, but they didn't look for tomorrow's product. They looked for future products. For example, while we were developing the 8080, they did a 60-bit microprocessor. No one used a 60-bit microprocessor. They were also planning to make only one chip per wafer; that is like a research type of job. Many companies had different approaches and different opinions.

Aspray: Okay. I see.

Shima: But Intel's final decision was that the 8008's basic architecture looked okay. There were many weak points in its performance, system interface, instruction itself, physical memory space, limited usage of stack operation, power supply voltage, and so on. So the decision was that to let us polish it. It is like a jewel: but the size and quality were good, and the mistake was no one knew how to cut and polish it. I started 8080 microprocessors in development in the beginning of November in 1972.

Aspray: Were there other people working on this also?

Shima: No. For the project itself, only me. In defining the 8080, we had many, many discussions. Almost every day, there were about five people: F. Faggin, T. Hoff, S. Mazor, H. Feeny and I. It took less than one month to reach our conclusions. In the beginning some people said, "Let us convert the 8008 to new process, such as NMOD instead of PMOS." I said, "No, it is not a good idea. We have to sell the product to systems people. Systems people didn't like the lack of systems concept." Also if you compared it with 8-bit minicomputers in the market, there's no advantage for system customers to use the higher operating frequency 8008. I did not want to repeat the same mistake as the 4004's original idea: the lack of system concept. To go into the business for high performance components area, a limited usage stack was not good. In the case of 8008, they used only eight levels of stack. Without unlimited nesting capability, system people never used the higher operating frequency 8008. So we decided to throw away the internal stack and introduced the 16-bit stack pointer in the microprocessor so that the external memory could be used as a stack. Also, there were a lot of requirements for DMA functions in order to transfer a large quantity of data between memory and memory/IO. There were many, many things. There was a lack of 16-bit operations. The lack of 16-bit operations for data transfer and calculation. It was an 8-bit machine, but to access the memory, system users felt that the memory space in the 8008's 12-bit address would not be adequate any more, and they were looking for 64K bytes of memory space. In order to support 64K bytes of memory, 16-bit operation became essential.

I admit there were many, many areas to be improved. As I said, I had enough experience that finally we completed the 8080. We concluded it in early December of 1972. Then, I was told that, "We are going to start the layout on January 2nd in the next year." I said, "What! Only a few more weeks." And they said that January 1st is a national holiday, so we will be able to start on January 2nd. He said, "We hired two people as mask designers." I told them I was just then writing the final functional specifications, now being typed, and I needed a couple of months for making the logic. But they had already made a decision.

So I decided, the logic design will start from the register file. I will show you. [showing diagram] Here are the six sets of 8-bit general-purpose registers, including the 16-bit program counter, and 16-bit stack pointer. Next to register file is an address increment and decrement. The second block was an 8-bit arithmetic unit including 8-bit accumulator and flags. The third block was instruction registers and decoders and controllers to implement instructions. The remaining block was the system bus interface including DMA and Interrupt controller. I realized that it may take a lot of time to design 8080 with only one engineer's power, but I may be able to start from the register file. So, I made up this register file's logic very quickly, within two weeks. When the mask designer came to Intel, I gave him a job for memory cell design, because with my experience in 4004. I knew it would take one to two weeks to do the job in early 1970's era. What I needed was to keep some stock. Suppose there was a two-week stock for mask designers. I would be able to use two weeks to do something new. Within two weeks, for example, I completed the logic and circuit design for 16-bit increment and decrement. So the mask designer would be able to move memory call design to next section. While they probably are doing layout of increment and decrement, integrating register files, I was able to work for 8-bit arithmetic unit.

Aspray: So you're running just ahead of them.

Shima: A tightrope, yes. Therefore, if you look up the 8080's layout, the layout itself was quite bad. For example, here are the metal lines for instruction controller; here is the metal line for internal 8-bit database. And this 8-bit data bus had to have a line that crossed over the controllers. This kind of layout was quite bad. I admit that I did a very poor job of planning the layout due to a lack of time. On the other hand. The Z-80 looks much, much better and I can say, “Beautiful!” But fortunately, I was able to complete the logic design, circuit design, and all the layout by August 9, 1973. In the case of the 8080, the logic schematic was drawn by hand. The layout was also drawn by hand but the circuit simulation was done by in-house on a Pop-10. But we digitized that one with a graphics system, which means that the modifications were done through a graphics terminal. I started with two mask designers, but in the last three months, I had about eight mask designers.

The biggest problem was the quality of mask designers. There were almost no experienced mask designers to layout the random logic product since Intel was a memory company. Then a problem appeared. Intel introduced the monitoring system in layout job such that how many transistors were drawn rather than how many transistors were saved. “Saved” means the error free. At that time, there were no CAD tools to check the layout such as DRC (design rule check) and connectivity check (check layout against circuit schematic). Thus, many mask designers just placed the transistor and the wire for inter-connection, without checking the layout rule and connectivity. So I had to correct those careless mistakes, day and night, day after day.

But I was not able to use any extra engineers for logic and circuit design. So, I concentrated on the logic design, circuit design, circuit simulation, and layout plan/checking. But, no time was available for logic simulation. Finally, on August 9th I completed a tape-out. After the tape-out, I was not able to touch the design any more. Simply, I had to wait one to two months. I told my wife, I could take a vacation starting the next day for one week. But on the next day my wife delivered twin babies. Finally I got one more baby, which was called the 8080. But when the 8080 came out, the first 8080 was not functional, it was completely bad. No circuit was functional and it looked like that the chip was still born. I said there was something wrong in the process. Unfortunately, as I said, Intel was a memory company and process company. They said, "Oh, we have never made a mistake in the processing, making wafers. There shouldn't be a problem in your chip. You should look at your logic." I said, "Even if there was something wrong with the logic, the clock should work. Also I had to demonstrate my logic was not bad. It took about one week. Finally they said, "Okay, it might be our program." Fortunately, within one week they successfully made the 8080, and the chip came out. Finally, there were three minor mistakes. One ground line was not connected. To debug the chip, I needed to connect two ground lines with micro-manipulators. The second problem was, I used the process of high voltage of n-MOS, which means it required two kinds of power supplies, +5 and +12. And instead of +12, I picked up +5 in one gate. But it was just a speed problem. The third problem was in the layout. I had used a two input NOR logic. But, on the layout, I found a two input NAND circuit. I picked up the two input signals. Then I used the TTL to make the two inputs NOR logic, fed it back to the chip. Then my chip became fully functional. Fortunately, there were no mistakes in the logic and circuit design so far.

Before this time we had already announced the 8080, and the response from customers was quite good. But still, most of Intel's top management people never believed that the 8080 project would be successful. Later, they said that they believed the microprocessor would become a big business. But, I felt they have never believed the microprocessor business would be good because they never assigned an engineer to the microprocessor area. Just after the official announcement with the fully functional 8080, Intel received many telephone calls from so-called big companies. They wanted to visit Intel. Many customers came to our microprocessor lab and Dr. Noyce showed them the 8080. Dr. Noyce looked very happy with a big smile. Then I was asked: "Shima, how many people do you need?" I said, "Well...." They had what they called a small machine group with about five people. One was working on vending machines, because Intel had a PROM, which can be programmed at the customer site rather than at manufacturing. Thus, PROM logic was good for vending machines, which requires the data changing even if after it was shipped to market. One engineer for calculator chips for Russia. Three engineers for custom design products. Intel said we were able to stop the job at any time, and you may use all of five people. By this time, in order to develop the standard peripheral chips, I had already made the specification for system peripheral bus. I wanted to develop many peripheral chips such as timers (8253), interrupt controllers (8259), DMA (8257), and UART (8251) like SIO, and Parallel I/O (8255). I wanted to develop these basic system peripheral chips very quickly and timely. Most of these peripheral chips are still used in today's IBM PC. Then in order to find out what level of engineers I could use, I interviewed them. One guy said, "Oh, I have the experience for logic design, but I will be able to do a good job for circuit design." Which means he said, I cannot do a logic design. One guy said, "I can do everything, but I cannot do micro-programming or random logic." So, I said, "Okay. Let us use PLA (Programmable Logic Arrangement) which is easier technique compared with random logic." One guy said, "I can design logic but I am not professional in circuit design.” Fortunately, within one year, my group developed all of the peripheral chips.

Just after that, the oil shock came. Intel was quite lucky. Motorola wasn't because their microprocessor development approach was quite different from Intel. In the case of Intel, since the 8008’s basic architecture was good, attention was paid to improvement. In the case of Motorola, they assumed from their research that 8-bit architecture would be good enough. This means there was about a one-year gap [in the work towards a microprocessor at the two companies]. When the oil shock came, Intel had completed the development of both microprocessor and peripheral chips. Unfortunately, Motorola had just completed the 6800 microprocessor. Thus, Motorola always priced products higher due to their lower volume since they came to the market one year after Intel came. Motorola had the same mistake when they developed the 16-bit microprocessor. When IBM developed the personal computer, they needed two things. One of them was an operating system, and the other was the system peripheral chips. Motorola's peripheral chips were very expensive. On the other hand, Intel's peripheral chips were less expensive. Which means IBM had lots of experience with the Intel chip. Then IBM bought the operating system from Bill Gates and the 8008 from Intel for their first generation personal computer: the PC.

It is quite important to develop the state of art microprocessor in time. There is one more example. After the first generation IBM PC came out with 8088, IBM planned to develop the next generation personal computer. They had the choice between keeping Intel architecture or moving to Motorola. They waited for the 6820. The Motorola people promised it was coming out at such-and-such date, but it was delayed for one year. IBM was not able to wait anymore. Then they picked up the 286 for the 2nd generation personal computer PC/AT model. But once Motorola's 6820 came out in time. IBM may switch the architecture. Therefore, the development speed was quite important.

People say Motorola's architecture is clean and better than the 8080. But without real functional products, the system user cannot do anything. However, there was one big mistake in the 8080. 8080 was the first product with NMOS in Intel. Also, before this time, nobody made microprocessors with NMOS. People only had experience with memory. It means that the data output from memory was the only one. In the microprocessor, the number of the address pins was 16, the data was 8, plus control. There were many, many output pins. There were huge dynamic currents from the outside to the ground line on chip.

For packaging, I had never paid attention to the resistance on the wiring pattern between the bonding area and the lead pin. Due to the huge currents through a big resistance, the ground line inside of the chip could not keep “0” voltage any more. Therefore, I asked the packaging company to modify the package. We produced many 8080s, about 30,000 units, before that analysis was completed, because the 8080 was so successful. The sample price of 8080 was $300. There were lots of requests from Japanese customers: "If you have a chance to come to Japan, please talk about the 8080." I told the company, "Please wait for two weeks for a detailed analysis. I have been asking someone to do it now. Don't make a high volume." But when I came back from the business trip, they said there were so many calls from the customers. Then we made 30,000 units. "What?", I asked. Then I found out the problem. In 8080 we had two versions, 8080 and 8080A; 8080 means low-power TTL compatible, which means small currents; 8080A is compatible with standard TTL.

Therefore, all of the 4004's success and 8080's success resulted from the fact that I tried to develop a microprocessor, such that the system users wanted. I also tried to introduce the system level solution with microprocessors, which has brought a long-term success for the company. After the 8080's, I developed the Z80 at Zilog. It was quite easy to find out how to succeed. It had to be compatible with 8080. Low price, low power voltage, low power/current, more powerful functions and instructions for computer people, a much smaller die size, and so on.

Aspray: Why did you leave Intel to go to Zilog?

Shima: The oil shock. The oil shock took stock prices from $73 to $18. There was no more advantage to staying in Intel. Also, at that time, Federico Faggin and R. Ungerman left together to find Zilog, which meant Intel had no people who knew microprocessors logic and business. It was not easy job for me to convince the memory guys.

Aspray: How was Zilog formed? How did it come about?

Shima: They got the capital from Exxon.

Aspray: The only product was to be this new chip? That was their main business area?

Shima: Yes. In Intel after the 8080, F. Faggin, R. Ungerman and I planned to develop much higher performance microprocessors. At that time, there were so many new microprocessors from many companies. F. Faggin, R. Ungerman and I found out that all of them were not competitive. For example, one chip from Rockwell was not powerful but less expensive with high integration. One chip from Fairchild called F8 was the high-integrated microprocessor. Both of them were for controller areas, relatively small systems, which means there was a CPU and a high-integrated peripheral plus the memory: such as a system. I knew that the microprocessor business would grow rapidly. In order to take a leadership role, it was very important to provide a high performance microprocessor to system users. Thus, Intel wanted to move to high performance. But instead of moving into high performance, Intel decided to develop the high integrated microprocessor. Then they developed the 8085. We thought the approach to 8085 might not be good. For a short time it would be okay. But many system customers needed much higher performance rather than high integration. Therefore, with a combination of high performance and high integration, we believed we could take the market share. High integration alone was not good enough.

Aspray: What year did you join Zilog?

Shima: In February, 1976.

Aspray: What were your responsibilities?

Shima: Z80 and Z8000 for 16-bit microprocessor.

Aspray: I don't know enough about the semiconductor business to tell, but to what degree was it anticipated that the computer industry was to be your main customer? To what degree did that shape the design of the chip?

Shima: We didn't focus on mainframe computers. We focused upon office equipment like high performance printers, high-end cash registers, and intelligent terminals. For the 16-bit microprocessor, we wanted to replace the minicomputer market. Therefore, the mainframe computers were not competitors for us. Also, they were not our customers. But, mainframe computer's peripheral groups were microprocessor users, including industries like telecommunications.

Aspray: How did Zilog differ from Intel in terms of its technical philosophy?

Shima: When Zilog developed the Z80, it was quite successful because they knew the weak points of Intel, what Intel was not going to do. After Z80, they made one mistake. The original Zilog was founded by two guys, Federico Faggin and Robert Ungerman. Federico Faggin came from the semiconductors area. Ungerman came from the software and system area. Federico wanted to do the products related to semiconductors, but Ungerman wanted it related to his systems. The company did not have enough money for two businesses at the same time, and it didn't need two top managers, who fought with each other. Small companies are not able to hire good management. They should have brought more people from Intel, but they made an agreement with Intel not to. I think that was Zilog's biggest mistake. Originally Intel hired many, many people from Fairchild. They brought in lots and lots of Fairchild internal confidential documents. I had many such documents in my cabinet when I developed 4004. When Federico told Intel he would not hire people anymore, which meant he would not get the best people for the logic area.

Aspray: But maybe it was necessary. Intel learned the mistake that Fairchild had made and had threatened suit against this new company. Maybe there wasn't any choice.

Shima: At Intel the process guys took the leadership. After that, marketing people took the leadership. Maybe that was why they were so successful. Intel's success came from microprocessor alone. They did develop many, many products, but except for microprocessors and some micro-controllers, nothing was successful.

Aspray: Would you like to talk about the design of the Z80 or the Z8000?

Shima: In the case of the Z80, it was quite simple because I knew 8080 well. The biggest problem in developing the Z80 was how not to copy 8080's logic. For example, with the 8080, I used an 8-bit arithmetical unit. In the case of the Z80, I needed just compatibility for software and a little more performance, with the same clock. I was able to use 4-bit arithmetic units, in an 8-bit microprocessor. For all of the area, I needed to find a different logic and a different circuitry approach for the Z80. That was the biggest difficulty. In the case of the Z80, I was able to utilize one engineer. I also used about three mask designers from the early development phase. Everything was done in the way I wanted. In neither the 8080 nor the Z80 had I done logical simulations. In both cases, we made a breadboard and made sure of the logic. We found one mistake in the Z80's. In order to differentiate from the 8080, software people wanted to use a different mnemonic for assemblers, for instructions. There was especially some misunderstanding in locate, shift, right, left, by mnemonic. I made them the opposite way.

I joined Zilog in February, 1975. I think we completed this chip in late October. Therefore the design was done in eight months. Logic was made sure by the breadboard, and circuit was simulated by the time sharing system with CAD tools. The layout itself and layout design rule checking were done the same as the

8080. There was a specialist for design rule checking. He was not able to do a layout job. I don't know how much money we paid, but it was quite expensive. For connectivity checking, I told the young engineer, "If you find a mistake, you have to check it once again. If you find no mistakes twice, that's okay."

In 1976, when the Z80 chip came out, there were no problems. We were able to ship the sample quite soon. There was very good customer response. We had asked to make wafers for some very small companies and found out that the wafer manufacturing capability was not enough. Therefore, we asked Mostek to produce Z80 and work as a second source to make the wafer. There are two kinds of ways to make the wafer. One company says, "Please send your master plate to be used for wafer making." The other company says, "Please send the database." Mostek asked for the database. They wanted to make the mask by themselves. After the Z80 job was complete, I took a vacation. Then there was a telephone call from Zilog. We'd received the wafer from Mostek. It was not functional. I said to them that maybe something was wrong in the testers. But still they were not able to find out what was wrong. I went to Zilog and I did many, many tests. Finally I found out that Mostek made a mistake in making the master, the poly-silicon. The poly-silicon was broken by dust under the metal. I asked the engineer to peel off the metal and check under the metal where I pointed out. Nobody believed me because I had no background in semiconductors and it was midnight. I told the testing guy: "Please peel off the metal. Just look at it, and tell Mostek." Nobody believed it. But when I had looked through the microscope with light from different directions, I found a funny shadow. The next day, both Zilog and Mostek engineers found the lack of poly-silicon.

After the Z8000's came out, my wife told me it was time to go back to Japan. Our twin girls were six years old, and before they started elementary school she wanted to go back to Japan. I told the Intel people I was going back to Japan. I would not see them again. They asked, "Why don't you come back to Intel? We want you to do something design-related in Japan." I agreed. My main job was to establish a design center in Japan. It was a good, challenging job. Before the completion of the design center, I reported to the vice president. Once the design center was established, I went back to a daily management job. I had to talk with the department manager in the United States to bring jobs to Japan. I was not in the mainstream anymore. I had to report to people who used to report to me. After about six years, I didn't want it that way.

Aspray: What was the purpose of the Japan design center?

Shima: The main purpose of the Japanese design center was to develop a standard product for a worldwide market. In order to train the Japanese engineers, I developed CMOS version 8088 and 8086. But, the problem came out, Intel business was going down, and I had to get the budget from the division at U.S. It meant that no one wanted to give the challenging jobs to the Japan Design Center. The Japan Design Center became a subcontract design center for a U.S. group. There were no chances to define new products any more.

Aspray: I understand. Are there any last things you want to say at the end of our conversation?

Shima: The success of 4004-microprocessor development, I think, came from three people: Ted Hoff, me, and Federico Faggin. Without any of those persons, we would not have succeeded. For example, I was able to develop general-purpose decimal computers, architecture-based LSI. That was different from the microprocessor. If I did not taken my basic idea requirement to Intel, then Ted Hoff would not have been able to find out certain new things. If Ted Hoff's basic idea was kept as original it was not suitable for application, like the 8008, that chip would not be successful. I was not able to define the detail that brought many improvements onto Ted Hoff's basic idea, and also I designed all of the 4004 logic. Even if the specs and logic came out, without Federico Faggin we would not have been able to make the chip. More than ten years ago, when I returned to Intel, I asked Ted Hoff why he left the project. He said, "Oh, I was busy with some other things." That was true. Also he said, "I thought we could not make it." Rockwell did for Sharp, but only six or seven hundred transistors. We estimated more than 2000 transistors will be required for the 4004. After 4004 it was quite simple to develop the 8008. They copied the layout approach of the 4004 and expanded it. The 8080 was just an improvement over the 8008. Z80 was the improved product over the 8080. However, in the case of the 4004, we searched for how to replace the hardwired logic for real time based I/O control by software. I believe that the biggest invention with microprocessor was the replacement of hardwired logic by software, and was able to demonstrate the computer can be implemented on a piece of silicon. Nobody knew how to do it. Therefore, up to the 4004 it was a very challenging job. After the 4004, it was easy.

Aspray: Thank you very much for your time.