Oral History Panel on the Development and Promotion of the Zilog Z8000 Microprocessor

Moderator:
Michael Slater

Panelists:
Federico Faggin
Bernard Peuto
Masatoshi Shima
Ralph Ungermann

Recorded: April 27, 2007
Mountain View, California

CHM Reference number: X4022.2007

Michael Slater: We have with us today [April 27, 2007] four people who were involved in its [Zilog Z8000 microprocessor] creation: Ralph Ungermann, Bernard Peuto, Federico Faggin, and Masatoshi Shima. We've heard about the backgrounds from Shima-san, Federico and Ralph in the previous tape [oral history by the Z80 team], so we'll start with Bernard. Could you tell us about your educational background, your experience before you came to this project?

Bernard Peuto: Yes. I was born in France where I got an engineering education in radio and in computers in 1967 and 1968. I came to Berkeley to do a Ph.D. In 1969, I had my Master of Arts from Berkeley in computer science and I passed my prelim. I went back to do my military duties and then I came back and got a Ph.D. in computer science in 1974. My dissertation was about memory protection, which will come back as a subject later. As my first job I joined Amdahl Corporation from 1973 to 1976. The reason I joined Amdahl Corporation was that Charlie Bass was sharing an office with me when he was an assistant professor at Berkeley and I was a Ph.D. student and Charlie Bass had a good friend of his that was working at Fujitsu so through that connection I was hired as a computer architect at Amdahl Corporation.

I spent roughly three years there doing computer architecture and specifically I did a lot of work in instruction measurement: average length of instructions, execution, code density, and speed associated with instruction. In fact, I published a well known article with Len Shustek on instruction measurement which was considered to be one of the best papers of the [IEEE Computer] architecture group in the last 40 years of architecture [publications]. Then I was asked to join Zilog where another connection of mine was there. Dean Brown was working with the people at Zilog. and Charlie [Bass, one of his friends,] had just joined. So basically when the issue of finding a computer architect with mainframe experience or computer experience came up, they contacted me and I decided to join them.

And the reason I decided to come was terribly simple. If you were working at Amdahl Corporation you were working with an established instruction set: the360. Joining Zilog was giving me the chance of defining another architecture and I was certainly hoping that the architecture would last as long as the 360 architecture lasted. There's a small and interesting story. When I was at Amdahl Corporation it was my first job as a computer architect so Gene Amdahl was God. He was not only the president of the company but he was also the guy that had designed the architecture of the 360. I had just handed in my resignation and he wanted to have a talk with me trying to convince me to stay there. And we met in the bathrooms and then we had the conversation and he basically told me, "Take this job." Go west, kind of. So I took the job. And at Zilog I was hired to be a computer architect to do the next generation. I worked at Zilog from 1976 to 1982. At the beginning, I was the computer architect for the Z8000, the Z8 and a few other things and then I became a manager of component architecture and director of component design engineering. So that's how I joined Zilog.

Slater: Great. And, Federico, by this point you had a sizeable company on your hands, so you were really in a CEO role I assume. Can you tell me about how involved you were in the actual design process and who the other key players were?

Federico Faggin: Well actually we need to go back to say the end of 1975. At the end of 1975 we were still eleven people but it was clear to me that we had to worry about, that I had to worry about, the next generation microprocessor, which clearly had to be a 16-bit machine. It was clear to me that the 8-

bit architecture had run its course. There was not a lot more that could be done with it and we needed a more radical design than the evolutionary design that had occurred from 8008 to the 8080 to the Z80.

I felt not qualified to be the computer architect for this machine,so I asked around if people knew a good computer architect coming from the computer industry. Charlie Bass recommended Bernard Peuto and so I interviewed Bernard toward the end of the year, Ralph and I did that, and we liked each other and gave him an offer to come and lead the development of the architecture of the Z8000.

My plan was that when the architecture was done, then Shima would be free of having done the characterization of the Z80 and then he could pick up the task of implementing that architecture. So that way we would regain some of the time that I felt we had lost in starting Zilog. As I said, I left [Intel] in October 1974 but we really didn't get going in full force until June of the following year. So my involvement [with the Z8000] then was really only in the beginning, to start this process and to be involved to some extent in the early discussion for the architecture, establishing the basic direction, but then fundamentally it was Bernard's work and then later on Shima joined and together they defined the next level of detail of the architecture to be implemented.

Michael Slater: And, Ralph, what was your involvement in this?

Ralph Ungermann: Well, I was responsible for a number of parts of the company and that included this next generation. I was amazed at Federico's decision to make a new, better product way before we even have the world's best product, but I was convinced that it was the right thing to do. So I helped in building the team. I found Charlie Bass and helped put together the software team and sat through and made some contributions to the planning process.

Peuto: If I may, I think those guys are a little too modest. Quite frankly at that time I don't think there was any other company that was as broad minded about software, hardware, architecture and the need to take advantage of the experience acquired with existing computers. At the same time, to me, they were people that allowed me to reframe it in the context of microprocessors because the microprocessor was not a mainframe. I think their contribution was pretty enormous because we were having lots of meetings and the feedback was what helped me do a much better job. They understood what I was talking about. It was a pretty extraordinary environment [in which] to design an architecture.

Slater: So were there other key contributors on the team besides the people we have here?

Peuto: Yes. It happens that we were for the first—I started in January,1976. The first major meeting we had was on March 31, 1976 and in attendance was Federico, Ralph, Charlie, Dean Brown, and then later we had a meeting with a high frequency between March and essentially October. At the beginning, the frequency was like every two weeks and then toward the end it was like every month. Then later Judy Estrin joined us and then once the architecture was well-defined Shima and I started working together. That was around August 1976.

So the massive contribution, in my opinion, that Ralph and Federico had was in that period between March and July because I was told, “You have three months to do the architecture.” At that point, we had essentially the major principle had been defined, and then from July to October they were still participating but then it was a lot more Judy Estrin, Shima and myself.

Faggin: At that point, the basic architecture was defined so we had little to contribute. Certainly I didn’t have much to contribute at that point anymore.

Peuto: So one of the contributions that Ralph was making, which was very valuable to me, was a marketing contribution. I had a lot of conversation with him about market and issues of size of chips, cost, margins, lots of things like that that are critical to the business that we were in. He and I had a lot of conversation on that. Ralph also had a lot of vision when it came to peripherals. I mean, those guys discussed it this morning on the Z80, the kitting issue and the need to have a complete family with peripherals and CPU. Those were things that I was really taught at the time and started to understand their consequences.

Slater: Who were the customers that you had in mind for this product? Were there other 16-bit microprocessors at the time?

Peuto: I think—No, there was none. By the way, one of the things I found out in my notes that I find quite amazing is that I had a conversation with Ralph and Federico probably in February with each one of them saying, “Tell me what you want. What do we need to do?” And they both told me the same story. We are not going to be the first to market. We have three months to decide on this architecture and they were absolutely right. It happens I think that the chip that they feared, that took a lot longer to deliver—we took a lot longer to deliver than we might have thought. Basically it was a wake-up call on the fact that we had to do better than any expected competition. So that pushed us a lot in the direction of a fairly sophisticated system.

At the same time, the other guideline was that—they had done something that I find pretty extraordinary—they had decided that software was very important, so Charlie was trying to have a system implementation language. We ended up calling it PLZ SYS and we were trying very hard to have an architecture that would allow a compiler to run well. So we talked to a consultant. We talked to people and so the software component, the development system component, the peripheral component were all added to the spec for us to say, “Is this specification allowing us to do a better job?”

Faggin: Actually it’s interesting to think about it because at that time we had not gone to market yet, so we did not have a lot more information on the market that I had certainly or Ralph had when we started the company. We started before having talked to customers, our new customers. So the market was intended to be the natural evolutionary market of the microprocessor beyond the Z80. People that had the Z80 they wanted to do more sophisticated things, but we had not targeted a market. We basically felt that we had to have the best possible 16-bit microprocessor architecture that we could make within certain boundaries of cost and time. And my hope was that it would take a year and a half to two years to actually get this product because I felt that that would have been the necessary requirement to be first in the business with this sophisticated architecture. Now there were 16-bit architectures out there.

Peuto: Yes.

Faggin: But they never really quite worked. A number of companies tried to do 16-bit microprocessors like National, RCA.

Peuto: TI.

Faggin: There was TI. But they either overshot what could be built or they under spec and so at the end of the day there was no really good 16-bit architecture on the market.

Peuto: Yes, the competition was basically the mini computers and the single CPU version of those mini computers. There was a one-chip Nova or products of that nature and they never really answered the problem because the best architecture, which Federico had talked a lot about as being rather important in the Z80 environment, the best architecture which now everybody [in the industry] has settled on, the distributed best architecture, was not really considered by those people. So the application to the business we were in with multiple peripherals [on a shared basis] was clearly not understood.

Now it was extraordinary that we took this gamble to decide that we had to have features that were computer–oriented in a context of a microprocessor design because at the time there were people that thought that microprocessors were logic replacement. The battle was raging between those two camps and I have always felt that the Z80 proved that the battle was going towards the computer side but that happened a year or two years after we finished the spec. So when we decided "what customer" [to choose], it was not by saying, "Oh, I know him" but it was by iteration and decision and the wisdom of the management that we made all the right decisions. We just basically made all the right decisions from that targeting standpoint.

Slater: But it presumably was more focused on personal computers rather than--

Ungermann: There were no personal computers.

Slater: Or computing systems rather than embedded applications given the memory management and all of that sort of thing.

Peuto: That is correct. One of the reasons we had memory management, I mean I'll talk a little bit more about it maybe later, but one of the reasons was we wanted to be sure that we would support high level language and the output of compilers. Having relocatable code, having larger address space, all of those were considered to be essential to the design. As a matter of fact, like I said we started in March to have those meetings and I came up with the first instruction set architecture on April 9, 1976. We had a meeting and then I gave a set of guidelines about "Now we have agreed between ourselves, those are the guidelines or do we want to change it?" One of the guidelines was we wanted this next architecture to provide speed. The other was that we wanted to have a flexible addressing mode. As you remember, microprocessors at the time had few addressing modes and they were not very flexible. We wanted to have a high level instruction set. By that we meant not high level in terms of compiler equivalent but in terms of "It will support the compiler well."

We wanted to have a regular instruction set. We wanted to have space compact code. I had done a lot of work on code compaction and the Z8000 was a very, very good chip for that reason. And then multiple data sizes: we wanted to have 8 and 16-bit and 32-bit and then we wanted to have, if there was a dedicated processor, like a floating point processor or some stream processor that would be a co-processor. It would not be in the main CPU. And then the final thing was at that time in April we considered segmentation to be a possibility that would solve some of the problems that we were all talking about. So when you look at this list, this is a pretty—not only by the way we delivered on all those points, but it was fascinating that in April, essentially one month after we started we had come to a certain consensus which was in this direction of "Microprocessors are used by computing engines, by computers."

Slater: That sounds like it was not an especially evolutionary design. I mean it wasn't a super Z80. It was more of a start from scratch.

Faggin: It was not intended to be a super Z80. It was intended to be a real break and an opportunity to do it right. If there is a little contribution of mine in this it was exactly to have this vision that we had to have a superior architecture. I didn't see any way that we could do a super Z80 after the Z80 and have anything that was really worthwhile.

Peuto: This is a very good question because there are two answers to the question. One is from my perspective it was amazing to have a group of people that had decided that. I would have decided that but that's a different story. And also it remained to be proven that you could decide that kind of architecture and mold it into a cost effective semiconductor. So there was a lot of constraint put on me like the 40-pin package. We had to add a 40-pin package version. We only accepted a 48-pin package version. We never wanted to have a 64-pin package version for cost reasons.

Slater: You mean pin? [Editor note: this refers to the above paragraph where the word "-bit" has already been corrected to read "-pin package"]

Peuto: Pins. What did I say?

Slater: Bit.

Peuto: I'm sorry, pins. But for cost reasons it was plastic versus ceramic, so all of those things were put as constraints on me. I spent the first three months analyzing the Z80 and proving to myself through a lot of the techniques that I'd done in my paper on architecture, proving to myself that the Z8000 would have better code density, faster speed from an architectural standpoint, and would allow you to have every instruction of the Z80 basically either executable in one instruction, or translatable. So I did a lot of work in the first three months to come with an instruction set that was basically Z80 supportive but the minute you had to decide that you had more than 16-bit for an address you could not be binary compatible anymore. So we made the break [binary compatibility] but I had strong constraints, which I proved to myself were correct to be able to have—through translation of assembly to assembly—a product that would be identical to the Z80. It's interesting that every vendor made the same decision. The [Intel] 8086 and the Z8000 made the same decision. Everyone talked about having that piece of [assembly to assembly translating] software but nobody ever sold it, so I have always found that to be one of the intriguing issues. Every piece of literature would say for every instruction in the [Motorola] 6800 you had an equivalent instruction in the [Motorola] 68000 but nobody ever wrote this rather simple piece of software that would have allowed you to do the translation.

Slater: Ralph, what kind of software support or development system support was planned for the Z8000?

Ungermann: I think I'll pass that to Bernard.

Peuto: Well, this is one of the difficult issues is that we, as has been mentioned a little before, by 1979 I think or so, we were potentially doing too many things at Zilog. The development system work and the support work for the Z8000 was lagging. In fact, it was not deliverable in time with the chip and so it happens from a management standpoint I was put in charge of it to try to fix it. And so I tried and we did some product called the Z Scan and so on and so forth, but what had been so good about the Z80 was that the software, the development system, the test board were all available together and unfortunately Zilog did not repeat that performance [for the Z8000]. And so that was one of the weaknesses that we clearly had from a market standpoint.

Faggin: Also the memory management unit was late.

Peuto: I was going to say, the memory management chip, I mean you had to believe [in our architectural decisions] us if we sold two chips, the Z8000 version segmented and the memory management unit. The memory management unit came very late making it that some people were not interested in the segmented chip because it would have required them to do some logic to fake it and there were one or two companies that did it but that was a disaster quite frankly because we had made the decision to be ambitious and require two chips, but the two chips were not shipped at the same time.

Slater: And what was the cause of that memory management unit being so late?

Peuto: I think it was a manpower issue.

Faggin: It would go under the guise of poor management. That's the best way to say it without beating the bushes.

Peuto: And, as an example, [Zilog was designing too many products], we were in the memory market, we were in the tester market, and we were in the hot Unix box market and for that market it was fascinating to me that we were selling a product but it was a non-segmented product and this was supposed to be in an environment that was a computing environment. And the kind of software tool that they did for that [Unix] box was not exactly overlapping with a need of designing for a customer using a [segmented] microprocessor so there again we were lacking unfortunately. I'm not trying to—there is a mistake also that I think we made on the Z8000 but in some sense those factors keep adding up until the point where they start becoming pretty difficult.

Ungermann: So now you see why I didn't want to answer.

<laughter>

Peuto: You mean you were in charge? My answer to that is genuinely not the fact that I blame you guys for having done too much. Quite frankly I blame Exxon. Exxon essentially choked us with money. They basically gave us too much money and too many directions, which we then kind of went into and in some sense there are times where you have to refuse and that's very hard to do when somebody gives you dollars. But the reality was the reason we were doing too many things is because we could afford to do it because Exxon was kind of giving us the check. That's my personal view. The elephant [had grown too] complex.

Faggin: It's not one reason, like those things have never one reason but we were under pressure by Exxon to do more. It was a typical question that I would get at the board would be, "Well, could you use some more R&D money" because they saw the potential of building more rapidly a large company. And in general my answer would "No, no. I think that's all that we can do, okay." But already we started doing too many things and so it isn't so much that we then accepted the lure of more money later on. We made decisions to start early on too many fronts and it would take a lot more resources than we could muster to actually complete all of the stack. And so I have to take—I was CEO at that time. My primary responsibility was to see that that would not be done, so I'm the one responsible for having started too many things because there was a sense that we could do it and that sense that we could do it in fact was deceptive. We then got caught. Then we had way more things to do than we had the resources, because it isn't just money. You have to find the right people. You got to hire them. You got to train them and all of that. So it was, in my opinion that was the most important area and I take full responsibility for that.

Peuto: Yeah, going back to the architecture and the interaction with the market which I think is what's interesting, I was mentioning that in April we had this spec which we all agreed on and so between April and July I fleshed it out and the way I fleshed it out was that I spent a lot of time on compatibility with the Z80 meaning looking at every Z80 instruction and looking at the corresponding instruction for the Z8000. I did a lot of measurement and then [was helped by] Judy a lot, also with myself and Charlie we did a lot of code segment. By segment I mean a little snippet of codes to verify the architecture was working. I was going back to my notes on that and I was kind of impressed how sophisticated, if I may say so, we went into verifying that it would work, that it would be better, that we would meet our goals.

So by the time in July when we had a pretty clean spec I started working with Shima a lot about "Is it feasible?" "Should we change things," and so on. We had done a pretty significant amount of benchmarking and so on. So, Shima and I started. Shima got that spec around, in fact exactly on 11 June 1976 and then he and I started really working together constantly in August. And one of the great surprises that I had was that we had come up with a spec that I thought was a very good first pass with each instruction explained, the architecture explained and so on, but it had never been tested. Is it implementable? And this is partly the genius of Shima that you suggested an idea to do and he always delivers on it. But it's also the fact that we started with good constraints from both Federico and Ralph and the thing was very realistic. And so we managed to do a lot of work and still needed to do a lot of work but it was starting to be more an issue of implementation architecture than in the architectural issue itself.

Then the last big change that we did was a very fascinating change, which is the spec up until October was a spec in which we had eight registers only. It was regular architecture with eight byte registers and a couple of byte registers made a word register and [a couple of word registers made a] double word register and we [could process] 8-bit, 16-bit, and 32-bit data, as much as our friend from Motorola ever had, but their marketing suggested that that was not the case. And so we had only eight registers up until October 1976, where a lot of the work that Judy, I, and Charlie and some other guys were doing in testing the instruction code proved to us that we should try to do better [register wise] and Shima told us the miracle that there was enough silicon to have twice as many registers.

Remember there were two banks of registers so we were having a significant amount of register space in that architecture. And at that point we changed it to 16 registers, sixteen 16-bit registers, sixteen 8-bit registers and half of that, 32-bit registers. And that required, and this is one of the things that Shima was talking about, that required us to reorganize the operant codes and all of those things and so we reorganized it at that point and from that point on everything was stable. And before that, except for this sixteen 16-bit register concept, things stayed very, very stable. I'm amazed when I look at my notes about the July document that there are not very many things of significance that changed. Now that doesn't mean we didn't do a lot of work because, for example, we wanted to have a multi-micro feature. We planned to have a co-processor feature. So those features were indicated but they were dramatically dependent on implementation for them to become realistic. So a lot of the work that Shima and I did was to go in and [define these featrures further. We were not dealing with add instructions] an add is an add, and that is simple, but in some cases we really invented things that had been sketched architecturally but were not yet proven to be implementable.

Slater: Shima, you were working on the Z80. What led from your perspective to the decision to get involved with the Z8000?

Shima: You may have some objections but in Zilog there are lots and lots of noisy things. With my memory there are three different proposals for Z8000. Only one of them is the true one and one said just to enhance the Z80. The second proposal was to develop the general new generation 16-bit microprocessor which had abundant and flexible instruction set with expandability. The last one is the high performance 16-bit microprocessor which Peuto just mentioned.

Slater: And what led to your involvement in it? How did the product definition process go in terms of your involvement?

Shima: I joined the Z8000 project in July 1976 and however the confrontation in three proposals was not solved until the autumn of 1976. By that time, I had already prepared the standard cell for logical design and circuit simulation tool and circuit design handbook. It became a time limit for decisions. And finally it was decided to develop two kinds of 16-bit microprocessors on the same die by the bonding option: a simple, 16-bit microprocessor and 16-bit microprocessor with segmentations. However there was a difficulty to introduce a system stack pointer because there were two opinions in Zilog. Many ideas were introduced by Dr. Peuto. However, the functional specification was written for software engineers, but not for the hardware engineer. On top of this there was no document that could describe the detail of the instruction set with instruction code. And also this instruction set architecture allowed user to select any combination among the instruction, addressing mode, and the data type that means it required 24-bit lengths for the instructions. And also unfortunately at that time Zilog's semiconductor process was about half a generation behind Intel's process and according to the circuit simulations I estimated that the maximum operating frequency is about 4 MHz compared with 5MHz 8086 and all of those made the development of the Z8000 very difficult.

Slater: So how did the instruction format get decided? Is that something that you designed or Bernard did you?

Peuto: Well what happened is I concentrated on looking at a 16-bit instruction format: where the [op code] fields were, what was the field for, byte versus word, where were the fields? [I was working at the logical level, not assigning specific instruction bits to these fields.] I also defined the instructions and what they did. The op code bits themselves that tell the hardware that it is an add versus a subtract that was not decided at that time. And Shima did that and, in fact, he did a great optimization to minimize the decoding logic.

One thing that was made as a decision very early in the process was we would have variable length instruction and thus there was an escape mechanism that says this is a short instruction versus a long instruction. And so we used Huffman type encoding to make it work.

I don't have the same memory as Shima when it comes to multiple options. I don't fully understand what he's alluding to because as I remember it from my perspective we were doing the Z8000. There was no other option. We made the decision that was a constraint that Federico pushed for, which I think was a good one. We made the decision to have a 40 pin and 48 pin, but I only worked on one spec which had both.

Shima: That is what I'm saying, 40-pin for the simple 16-bit microprocessor, and 48 pin for the high performance 16-bit microprocessor with segmentation.

Peuto: Oh, that's what you mean by two versions?

Shima: Yes.

Peuto: Oh, I see, I'm sorry. I'm sorry. I had misunderstood.

Shima: Yes that is really difficult.

Peuto: And I, by the way, agree with you because it was a bonding option but there was a lot of logic that needed to be there. Now one of the reasons we did that was because basically we were entrusted that somebody would develop an application in the 16-bit addressing environment would have the ability to port it to the segmented environment with no change in the binary. So what we did is we did this architecture. We had those two modes. That was because we felt, and this is clearly Federico and Ralph's inputs on this, we felt that we couldn't have a system in which only the high end features were available and there was no low cost version, so in some sense the non-segmented was the low cost version.

Faggin: Yes, I was involved in that decision because the issue was that if you had less than 64K of memory, which many systems in those days had less than 64K of memory, why do you need an MMU? You don't need that. And so you need a trimmed down version that you could use as a single chip instead of a two-chip solution. The fully fledged machine required two chips.

And also they would respond to a fundamental problem that proceeding to market that if you had to always require two chips for the CPU the Z8000 would not have been accepted because the idea was that eventually the next generation Z8000, the MMU would be included in the chip itself and therefore it would become a single chip again. So that was my vision at that time and so that was an important way to do this, to solve this problem, and in terms of the design it really complicated matters only very little.

Peuto: Yeah, I think Federico is making a very good point. One of the things that we were really eager to do was to have low cost versions and at the same time we wanted an architecture in which there was enough expandability to it that it could go to the next generation Z80000 type generation which in fact put some of the MMU inside of the chip. So that was correct.

The other thing that was very interesting to notice is that we had this constraint that we had defined at the beginning of having relocatable code and having very large memory addressing. If you start looking at all those four corners that are represented by a cheap version and an expanded version, relocation, memory protection and all those, you basically had only one technical option you could think about, which was segmentation. So we will talk later about the fact that segmentation had a cost in terms of customer perception but the fact was given the constraints we've taken that was the only simple solution. It's interesting that you ask yourself in retrospect "Should I have found other constraints?" But so the need for segmentation came in great part because of the confluence of those points that I was making.

Faggin: On the other hand, Shima-san makes a very good point that the technology that he was asked to design in that time was actually behind what it should have been. And, in fact, I remember Shima coming to my office and complaining about that and I actually made sure that the process guys were developing the technology that was needed for the Z8000 by going down to four micron gates when I think it was five micron but when you were doing the original design. So I certainly responded in my power as CEO to make sure that the process technology that Shima needed was available to him.

Shima: Let me describe how I decided the Z8000 instruction Format. At first to catch up the Dr. Peuto I studied both static analysis and dynamic analysis on the 8080 and the PDP-11.Maybe you gave the book to me.

Peuto: Yes, I did and as a matter of fact this is the work that I did.

Shima: Yeah, and in order to get the high performance, high density program and the higher flexibility of instruction expansions, the most important work in chip development was the decision of the instruction format and also register file. Like, for example, in case of segmentation, the question is where are we going to put the segment number, segment register for stack pointers. And in the beginning we had some trouble. If the instruction format was well defined, it is possible to not only simplify the logic but also get the higher speed. First of all, we decided on sixteen sets of 16-bit general purpose registers in about June and that number of general purpose registers was quite useful not only for a RISC processor but also a CISC processor in order to reduce the number of memory access, and both the system stack pointer and the user stack pointer were placed in the general purpose register. Peuto said that 32-bit register was constructed with two sets of 16-bit register and the 64-bit register was constructed with four sets of 16-bit register. But 64-bit register was used for only the instruction of multiplication, division and sign extensions. Secondarily, 16-bit variable instruction format was chosen like the length of instruction is incremented by a 16-bit to achieve high performance with 4MHz operating frequency. Two kinds of instruction format were introduced. One of them was like a shorter format for both reducing program size and getting the higher speed. And the other instruction format was called as a general instruction format. Two bits of a 16-bit instruction were used for the selection of a short instruction format and general instruction format. General instruction format supported five kinds of important addressing mode. Those were register direct addressing and immediate addressing and direct addressing and register indirect addressing and also index addressing. The register addressing mode can be selected without any limitations, but on the other hand, if the register zero was selected, the immediate addressing mode and direct addressing mode are selected with some limitations. Otherwise, the register indirect addressing mode or index addressing mode are selected. Therefore, register zero was not used for the address register but it took a lot of time to convince Peuto. Now the general instruction format was constructed with two addressing mode fields and six-bit operation code field and a four bit source operand field and a four bit destination operand field. However, 64 kinds of instruction sets utilizing six bit op code field were not enough. Therefore, it became necessary to set the limit, to attempt the optimizations.

Slater: Okay, thank you.

Peuto: All those were very well known problems. One other thing that influenced me a lot was the PDP 11 which had stack pointers in the registers. The idea of having the register zero not having full indexing capability is very old because that’s the mechanism you use to say that it’s indexing or no indexing. And the two bit architecture, the two-bit field that tells what type of instruction it is, that’s the IBM way to do it and the increment of 16-bit was the IBM way to do it. So all of those were issues in which Shima and I collaborated to find the solutions in order to minimize hardware but they were all things that had been pretty much described in this pack.

Slater: What do you think were the biggest problems you faced in starting the design on the Z8000?

Shima: We talked about how we choose the instruction set. I describe how to have chosen the instruction set. This is quite important and the Z8000 spec itself was defined like in October but still I needed two more months to make a detailed decision for instruction format and the instruction itself because sometimes Peuto said “This is quite important. We need such and such addressing mode but this is not so important and you may do such and such.” It took about two months.

Peuto: Yeah, by the way, he’s describing something that I think is very important to mention is we were collaborating a lot and, as you know from an architectural standpoint, you can make tradeoffs. You can essentially deplete all the options and so a lot of his feedback to me was a feedback that was dealing with how hard it was and if it was hard we would try to assess if we had to do it or if we didn’t have to do it.

Faggin: If I may say a few words, I think that the issue here is that in any kind of architectural work there is the sort of give and take in the moment that you’re trying to implement the architecture, so I’m actually surprised that Shima takes a somewhat antagonistic view of this matter when, in fact, it is part of the process. It is actually working together to resolve the issues that are only seen when you start to implement the architecture that there is the issue. So I think we should be happy that the two of you were able to define a very good architecture and then so that you could implement it, Shima.

Shima: Let me explain. The first group is the instruction set which has both source and destination fields, and provides the data type of a byte and word. There are about ten kinds of instruction sets, such as load, store, exchange, addition, subtraction, compare, logical OR, logical AND, logical Exclusive OR, and test of conditions. And the second group is the instruction set that has both source and destination fields, and provides the data type of word, double word and multiple words. There are 16 kinds of instruction sets such as load, load multiple, store, store multiple, push, push long, pop, pop long, load address, addition long, subtraction long, compare long, multiplication, multiplication long, division, and division long. There are some instructions of load address for previously described addressing modes. The load multiple instruction and the store multiple instruction loads or stores arbitrary length of words to start from an arbitrary register in the general purpose register. It was judged that those instructions were more flexible than the push or the pop, and that was our decision. And the third group was the instruction that provided only register direct addressing mode. There are only two instructions such as 16 bit addition with carry and 16-bit subtraction with borrow because there are 32-bit addition and subtraction. And the fourth group was the instruction of which destination is the memory. There are three instructions such as store immediate to memory, push immediate to memory, and compare immediate to memory. Z8000 took the register to memory architecture. But according to the dynamic analysis on PDP 11 it was quite important to have those three instructions. And the fifth group is the instruction that has both source and destination fields, and it provides the base addressing mode, basic index addressing mode, and relative addressing mode. These addressing modes are not so frequently used. Those addressing modes are used in the limited instruction set such as only load, store, and load address. And that is the agreement with him.

Peuto: Yeah, we essentially--

Faggin: So you strongly agreed.

Shima: Yeah with the optimization. It was quite important.

Peuto: Yes, basically what he's saying is correct in the sense that all those discussions we had together they were in the specification when we started. There were some places where it was not. He mentioned one instruction that I've always regretted that was the case. We had a base addressing and a base addressing with indexing and he did mention those instruction. In those two instructions we just couldn't afford to let it apply to all instruction in the instruction set like add, subtract and so on, so we did for load and store only because that's the minimum. Since we now had 16 registers you were in a situation where you could load it from memory and then work on the register in then store it back. So all of those are the absolute tradeoff that you have to make when you are dealing with this and all of those were well documented in the spec.

Shima: Okay, let me talk about it a little bit more. I wish you can help me. All general purpose registers were used as accumulation register, data register and address register. Also those are used as an address register in the instruction of push and pop. Therefore, any general purpose register can be used as base register like [in the] 8086 or the frame pointer of 68000.

Peuto: That's correct. One other thing that was the case is that both, in fact, the 68000 and the 8086 had a severe limitation when it comes to some of those issues that he's just talking about and we were very influenced by the PDP 11 which could make any register be a push and pop register. And we also absolutely wanted to have generality like we were saying. Regularity is a word that people use. So, yes, this was design.

Shima: Then to provide those functions I decided to put three instructions, such as increment by four-bit immediate data to data [register] and decrement by four-bit immediate to data [register], load by four-bit immediate to data [register]. Therefore, six groups in the instruction has both source or destination field and the four-bit immediate data field which was used for the immediate data or bit address. There are 12 instructions such as increment by n, decrement by n, load immediate by n, bit operations such as bit set, bit reset, and bit test. I wish for you to describe later, okay?

Peuto: Yeah.

Shima: The bit operation is designated by four-bit immediate to data [register] or by contents of a selected register. Please explain.

Peuto: Yes, I'm sorry I didn't mean to cut you off. What I was going to say was this was a great result that we had gotten when Len and I had made this analysis of instruction frequency is that when you look at immediate distribution of the number of bits required for immediate is very skewed: the small numbers are very frequent and the large numbers are highly infrequent and you could then do then with an add. So [we implemented] instructions that were short instructions that had a small immediate field because they were short instructions, in fact, covering 80 percent of the case, it had an enormous impact in terms of code compaction. So this was part of these papers that we're both talking about which influenced us in the design of the architecture.

Slater: Yeah.

Shima: Then the seventh group has eight kinds of instructions, such as complement, negate, clear, test, test load, test and set, call subroutine, and load PSW and PC. The eighth group is the conditional jump instruction. That had a destination field and a condition code field. The ninth group also I wish Peuto to explain about this one. I will explain part. The ninth group is the instruction of rotate and shift operations. The number of rotation in the rotate instruction was limited to two, one time or two times. But on the other hand the number of shifts in the shift instruction is designated by four-bit immediate data [register] or by contents of a selected register.

Peuto: There was one instruction out of those—I mean all of these were pretty classical instructions in the PDP-11 environment and the 360 environment. But there was one instruction that I want to mention is a test and set instruction which is critical when you're sharing and we had an atomic operation, as it is called, to do it right. And I think we were the first microprocessor to do that that was not a mini computer and they did that much later. This brings again another thing that we had. We had two pins that were used to do multi processing and have two processors being able to share. That was also a design that Shima was quite influential on because we were dealing now with driving pins. But we put those instructions in too so that was very unique and unfortunately nobody seemed to need it for a while and that after a while [the industry used this kind of functionality despite the fact] they did.

Shima: It was like a several month discussion with Peuto and deciding the instruction format and instruction set and I was convinced the Z8000 can be a good 16-bit microprocessor.

Slater: So how did the market receive this? What was the competition like? What kind of responses did you get from the marketplace?

Faggin: Let me start with that. Basically, we were still in the design phase of the Z8000 when Intel announced the 8086 and so that was a blow for us a bit because we were not first with the 16-bit. It was my desire to be first. That's why I started the process even before the Z80 was done. But unfortunately it took a long time to get to silicon on the Z8000. And so we were second to market. On the other hand, as we took a look at the 8086 we became convinced that the Z8000 was far superior architecturally and also it was a much smaller chip and our spec was higher speed than the 8086, so that we can actually have a very competitive run. But at that time we had not gone to customers yet because we did not have the Z8000.

Peuto: Yes, I have in my sets of notes I have a statement made by the Z800. At one point we thought we would call the chip the Z800. I think it was Federico that says, "No, no, no, we're going to call it Z8000."

Faggin: No, I wanted to reserve the Z800 for a Z80 with an expanded address space.

Peuto: Exactly.

Faggin: That was my idea. My idea was that the Z80 was a great microprocessor but it was limited by 64 bits, so if we could extend the addressing space of the Z80 we could extend its life.

Peuto: Yes.

Faggin: But we never got around to it while I was there. Eventually they did but it was after my time.

Peuto: Yeah, from a market reception standpoint I think it was a battle between three groups. It was pretty obvious. Essentially Intel triggered the race and so the other two had to kind of preannounce. They both did. And it was a race in which I think Zilog was at a disadvantage. The disadvantage was that the Z80 had sold itself. We were a tiny, little company having made sets of decision on the Z80 that made it a roaring success and we're now confronted with two companies that were much, much larger than we are and trying to battle out [with them] what was better, what was not better.

And so they had a lot of initiative on the marketing front to position us that we were not very successful [in countering it]. I think in the case of Intel a lot of us think that the 8086 architecture was not one of the nicest architectures you could see. I'm trying to be sweet. But basically "Operation Crush" made such [impact that] it didn't make too much of a difference. And in the case of the 68000 the fact they had this banana 68-pin chip gave them a lot of credibility in pretending that they had more instructions than we did or more 32-bit or things of that nature, which we had a hard time battling. And one of our battle points could have been memory protection, memory relocation but we didn't have the MMU available. So it ended up being very dark [difficult] for us, yes.

Slater: So what were the timelines? So you started the project the beginning of?

Peuto: Technically?

Faggin: January 1976.

Peuto: 1976.

Slater: When did silicon first ship?

Faggin: Seventy-nine. Let's talk about the actual design of the chip. Then we can go into the marketing.

Peuto: Yeah.

Shima: It took about 26 months from product specification to a working sample. I joined the Z8000 project in July 1976. Product definition with detailed instruction format and instruction set was completed in January 1977 and then detailed logic design started in April 1977. The layout design started in July 1977. The tapeout was completed in August 1978. The first wafer came out in October 1978, but there are lots of problems. Then the fully functional wafer came out in February 1979. That is the story of Z8000 development.

Faggin: So it was the middle of 1979 before we had anything to give to customers.

Slater: And then the MMU was behind that?

Faggin: And the MMU was behind that.

Shima: Well the MMU was at least a year. I don't remember.

Peuto: Yeah, about a year.

Faggin: Nine months to a year, yeah, nine months to a year.

Peuto: So in some sense the edge we might have had was undermined by the fact that we were not delivering on that edge.

Faggin: We definitely blew—my original desire and schedule was really to get this thing done in two years. But, at any rate, we still came out before Motorola. We had devices before Motorola. If I look at what happened in the market where the Z80, like Bernard said, sold itself because it was compatible with the 8080, there were lots of applications for that class of machine. People understood the 8-bit world.

When it came to 16-bit people had to compare different offerings and out of the potential customers, ours were not sophisticated computer architects. It was difficult for them to see the merits or demerits of Z8000 versus 8086 versus 68000. It was not an easy task. And, of course, there was an opportunity to start fresh in the market, which in the case of the Z80 was unnecessary, so there was some hesitation on the adoption of 16-bit. In fact, it took a while before the 16-bit was adopted and the only killer application for the 16-bit micro was really the IBM PC which, of course, we didn't know was on the wings. And so had we been designed on the IBM PC the architecture probably would have won, but IBM chose Intel for a number of reasons, some of which I can talk about later on. But that really in my opinion sealed the fate of the Z8000.

Now I did a lot of work to get Olivetti to design in the Z8000. I made a couple trips in Italy to convince Olivetti management to adopt the Z8000. In fact, they had a personal computer that they used the Z8000 that came out in 1980 but that was not enough because in 1981, IBM announced their PC and basically anybody that had started with Z8000 had to basically change course and adopted the 8086, particularly since the IBM PC was an open system that allowed other people to really apply the Intel CPU.

Slater: So the desire for software compatibility with IBM PCs really capped the marketplace.

Faggin: That is right, capped the marketplace for us. So the only other set of applications that were open then to us were graphics-oriented applications. For example, the Apple Macintosh was a graphics-oriented machine. Of course we didn’t know that that was going on either. But because of segmentation we could not participate in that market because those applications require leader addressing space because the data was much more than 64 kilobytes.

So we basically by having a segmented architecture unwittingly we closed ourselves off from the other set of applications that were 16-bit applications of minicomputer performance. And so what was left for us at that time was not very much. That allowed momentum to be built on the Intel side, to the Motorola side as a distant second because of their adoption in the Macintosh.

Peuto: Yeah, I think this is a very fair analysis. I also think that the interesting point is the machine that had the more limited segmentation machine is the machine that survived. And so in some sense it is true that given the market available to us, we had a product that was starting not to be complete, especially since we didn’t have an MMU yet. But it is also true that it is a strange twist of fate that the machine that had the segmentation in the worst fashion by having indirect register segment numbers was the machine that essentially dominated the market.

Faggin: But it goes to show that there is a point where architecture and design and any technical issue is not sufficient to break into the market, that being first and having the strongest marketing and the strongest momentum there is what makes a company win.

Peuto: When I talk about my years at Zilog my answer is always I learned how important marketing was because fundamentally Intel, with Operation Crush and Motorola with their ‘Push’ essentially squeezed us in some sense. There is also even more statement that can made about that because, as you remember, the story repeated itself with RISC versus CISC where people were pretending that the cleaner architecture of RISC was better than CISC and, guess what? Intel managed to silicon over all the weaknesses of the architecture in order to make it work. So in some sense it has been an incredible lesson for all of us who thought that clean architecture was really, really important and, in fact, it was not as important as we thought and, in fact, it was proven not to be as important as we thought.

Slater: Sort of a sad observation as the architect.

Peuto: Yes, it is a very sad observation and, like I said, the lesson that you may have the best gadget ever in your mind. [But] if you don’t have enough marketing muscle - forget about it.

Faggin: I mean time to market is of the essence. The Z8000 for whatever reason was one year later than what I wanted and what would have absolutely captured the market, so that is the biggest problem and that was one of my biggest requirements in the beginning of the project. That’s why I started so early to do this thing in two years.

Slater: [Shima-san what were] the problems you faced when you started the design of the Z8000?

Shima: Both lack of manpower and the lack of logic simulator became big trouble and I was able to use only one junior engineer for circuit design, layout plan and layout monitoring. The number of total transistors with the Z80 was 8,200 and the number of equations of instruction execution control was about 200, and it was expected that the number of transistors of the Z8000 is twice or more than Z80 and also the number of equations of the instruction execution control is about 500. There is a huge amount of logic that was a workload which was not able to be controlled by one logic design engineer.

Also, there are two kinds of microprocessors in the same chip. Therefore, I expected that it takes one and a half year to complete all of the design activity and also I expected lots of bugs and many difficulties of debugging. And fortunately one young, bright, and brilliant engineer, Jamshed Patel joined the Z8000 project but he just graduated from a Master's course at U.C. Berkeley and I educated and trained him asking him to generate the process file to be used in the circuit simulation and also generate the circuit design handbook. This is a good way to teach young people.

As I described before I had to use a four megahertz operating frequency. My next challenge was how to design the Z8000 with less clock compared with 8086. I didn't use the micro program logic method which required high speed MOS process and logic simulator. Let me talk about the five policies which I established. First of all, we chose the three clock for system bus. Therefore the first policy was to reduce the number of clock of register data transfer instruction to be three clock. This is quite important to get the high performance. The second policy was to reduce the number of clock for following addressing modes, the register indirect addressing mode and index addressing mode and the direct addressing mode. This is the weak point of 8086. Third policy was to reduce the number of clock for 16-bit multiplication. The fourth policy was to reduce the number of clock at the time of judging whether segment address offset is short offset or long offset. The fifth policy was to keep the operating frequency to be high as much as possible. Those are my five policies.

Slater: And how did you implement those policies for the design?

Shima: Yeah, first it is desired to decode the fetching instruction and the loading segment information before it was stored into the instruction register or it was used. And I introduced the logic of look ahead instruction decoder that is placed on the instruction internal system data bus.

There are many instructions that require high speed. This look ahead instruction decoder decodes those instructions and reduces one clock; thus, for example, the register-to-register data transfer instruction is executed with only three clocks and also any extra clock is not used for the direct addressing mode, index addressing mode and a judgment of type of offset address for segmentation. Later this technique was used in NEC's 32- bit microprocessor. Next, the dual-port RAM was introduced in the register file in order to reduce the number of clock of not only multiplication but also 32-bit operations. If the dual-port RAM was used 32-bit data is read out at the same time and also both 16-bit read operation and the 16-bit write operation are performed concurrently. In addition, 32-bit shifter for multiplication was placed between read port of register file and internal system data bus. Then 16-bit shifter was placed between the internal system bus and arithmetic unit in order to reduce one clock at the relative addressing calculations. In the arithmetic unit the execution section of calculation was separated from condition judging section by the pipeline control and it became possible to execute arithmetic operations at every clock for assuring the high operating frequency.

I was convinced that the performance of the Z8000 is higher than that of 8086 and also 68000 especially when the performance of the Z8000 compared with 8086 and Motorola's 68000 that uses microprogram. The Z8000 had no performance problem in the memory access instructions; therefore, total performance of the Z8000 became much higher than 8086 and 68000.

Slater: Thanks. So shall we return to the marketplace and some of the business issues you faced? Federico, do you want to address how the Zilog situation affected you and what other challenges you faced with the market?

Faggin: Well the 8086 was introduced sometime in 1978, as I mentioned earlier, and it was not a surprise because we were expecting something from Intel in the 16-bit world. But the fact that there was one now indicated that we were behind and that was an important factor. At that time also the company, by the way, had grown from eleven people at the end of 1975 to in 1978 we were over 1,000 people, so I mean that gives you an idea of the amount of expansion that we had for a company in a matter of a few years. That was not an easy task for me as a CEO, of course.

And on top of that Exxon Enterprises was beginning to show their card, show that they wanted to be an overarching information technology company. They started to talk about combining the various companies into a single company and more and more of my time was beginning to be spent with Exxon Enterprises in New York. So that time was a very important time of transition for the company as we were beginning to fight the battle of the 16 bits.

As I mentioned earlier, the IBM decision to go into the personal computer, which I don't know when it was held, but it was held sometime later but that eventually sealed the fate of the Z8000. But the Z8000 was I judged a definitely superior product to the competition as was appearing with the 8086. In fact, we were relieved to see the Intel microprocessor was not really as powerful as we feared it could have been.

Then sometime later, Motorola announced their 68000. At that time, we were already pretty much done with the Z8000. So we took a look at that architecture. That architecture was leaner address space and packaged in a 64-pin package. It was about this long and this wide which was not a very cost effective package particularly in the low end of the 16-bit microprocessors. We also managed to see that we were superior to the 68000 as well.

But the market for 16 bits was in its infancy, so before the customer could really digest the offering, decide to use this new processor for new applications and so on, it took some time. So if it wasn't because of the specific IBM decision I think the turn of events could have been different for Zilog. But the fact that IBM chose the 8086 and also decided to have the PC be an open system so that other people could then have other personal computers, develop software that was available broadly and so on, really was an unforeseen event that had a major impact in the adoption of the Z8000 even if the MMU was about nine months to a year late. Because, for example, that fact did not impede Olivetti, for example, to have a PC in the marketplace before the IBM PC, but when the IBM PC came out everything changed. So that was the most important element. Very often you don't have one cause that is so dominant but at that time that was a dominant cause.

At the same time, the company was under strain and stress. I was preoccupied a lot more with Exxon than I was with the marketplace and so that was also a time of turmoil for the company. It certainly did not help. But a major reason why IBM chose the Intel was that we were perceived by IBM as competitors and I was told many, many years later by an IBM insider that Cary [Frank Cary IBM CEO] had a memo, internal memo forbidding IBM to use Zilog products because of the affiliation. At that point we were an affiliate of Exxon Enterprises.

Slater: Somewhere in here you formed a relationship with AMD right which actually marketed the Z8000 at some point?

Faggin: Yes, that's a good-- I had forgotten about that basically. Obviously the natural second source for the Z8000 would have been Mostek. Mostek was the second source for the Z80 and we were in discussion with them for quite a long period of time. They wanted the Z8000, of course, but they were unwilling to give us anything for it, so there was a point when I simply put my foot down and said, "The hell with it." They want it essentially for free just because they got a good deal with the Z80. There is not a reason to do that for the Z8000. And so we looked for another source and we found AMD interested in second sourcing the Z8000 and so we switched horses at that time. That must have been 1979 sometime.

Slater: And how was the fab going? Did that continue to work out well for you?

Faggin: Yeah, the fab continued to work out well. We had some problems with memories simply because we were not a memory company and they were used as fillers. And so essentially that was a mistake to actually waste some of the resources in that direction for the company, but the fab worked well and then in about 1978 I decided to set up a second factory in Nampa in Idaho and that factory became operational in 1979.

So in 1979 we had two factories, one in Cupertino and one in Nampa, Idaho and that factory was superb. That factory was done from the ground up using social, technical analysis of the workplace and so on. We had an incredible success. We hired people and we had—generally when you have a new fab you have high turnover early on in the fab. Typically you would talk about 30, 40 percent per year or more, sometimes ten percent per month of turnover and we had only a few percent turnover per year because of the time that we took to train people and really get them into the process and really participate effectively. The yields that we had on that fab were fantastic.

Slater: Bernard, as you look back, I mean there were a number of architectural innovations, at least in the microprocessor realm that showed up first in the Z8000. As microprocessors continued to advance have you seen a lot of those show up in future microprocessors from other companies?

Peuto: Yeah. In some sense we were too early with a lot of features that were copied later by others. Now they were copied not because we did it but they were copied because they were required for any kind of computing application. But one other thing that is an unheralded feature of the Z8000 is that we had co-processor support. We had instruction that allow a co-processor to work in conjunction with the CPU. We had multi-processor support.

And toward the end of the design I suddenly realized that with minor changes we could do the MMU and make it in a virtual memory machine and, in fact, that virtual memory version of the Z8000 was introduced around 1983 or 1984, something like that. So we basically had memory relocation, memory protection, virtual memory, multi-microprocessors, test and set instructions, so in some sense you don't want to say that we were too early but there is a little of that because those features didn't seem at the time to be required by the market.

Now they were not features that would have made the product more costly and if the market had accepted the product then they would have been welcomed features because people [could have] upgraded to them with no headaches. In fact, as you know, Intel had lots of headaches to upgrade to those features. Similar things happened with the Z80000, which we are only going to allude to. This was one of the first products which had paging inside of the CPU. In some sense the VMMU was inside the CPU and had a cache, a very small cache but it had a cache. You had to wait for the 486 to have a cache.

So all of those things were things that were consciously done in order to, in fact, to create a next generation to which people could have [upgraded to in a compatible way]. I don't think the cost was enormous, although the two chip system was difficult for us since we were late. So that's a little how I see it in terms of that.

Faggin: As a matter of fact, the chip size of Shima's design was smaller than the 8086.

Shima: Yes.

Faggin: It was a smaller chip size, so it was very cost effective. We never had any trouble building Z8000s.

Slater: Were they priced competitively?

Faggin: Oh, yeah, it was not a question of price or manufacturability and obviously definitely not a matter of spec really. As I mentioned, the fundamental turning point was the IBM selection because at that time there were not that many 16-bit applications in the marketplace. I mean people were still almost—the industry was getting microprocessors out there that were a bit ahead of the capacity of the customer base to absorb that.

Peuto: Yeah, on the other hand we've already alluded to that but I want to come back to it. One of the other problems that we had and I think I am like Federico. We see eye-to-eye on this. This was not the killer problem but it didn't help that we were so late about the MMU and we didn't have an emulation system. Software support was behind. So basically as you were dealing with a design win that was a long design win and people looked at our offering and they saw those gaping holes and so it was very hard to sell.

What was interesting was that in Europe, the head of Europe [sales and marketing], a guy named Amori Piedra did very well in getting designed in but it's because essentially he had put a team [of applications engineers] together that was very good and probably better facing the challenges that we would have faced. In the U.S. we were not successful. So it was interesting to me, I flew to Europe very often on sales calls because basically Piedra was calling on me and going to customers and we had some interesting success there.

Faggin: Yeah, that's true. And part of that is that there were percentage-wise more projects that were minicomputer class devices in Europe for some reason than here. You had companies like Nixdorf and Plessy and so on that they were beginning to actually adopt microprocessors in a big way and they were more sophisticated in a way when it came to microprocessors than the U.S.

The U.S. it was more embedded applications early and terminals and there was a more mature computer industry and so the microprocessor needed to be either much higher than the Z8000 because there was a minicomputer industry that was quite developed. But it was really wedded to their own architecture, so that was not open to us; where in Europe there was more opening for that. In fact, the first computer to use a microprocessor was a European design. It was the Mistral that used the 8008 back in 1972.

Slater: Were there any successful Unix systems using the Z8000?

Faggin: Yeah, there was a company here, what was it; yeah there was a company that was doing it. I forgot the name.

Peuto: I have it on the tip of my tongue but not.

Faggin: Yeah, there was a company that used a Z8000 for a Unix system and it had initial success I must say, but again we're talking about in the scale in which you want to sell chips. You're talking about tens of thousands of systems per year when, in fact, a successful application like an IBM PC would be hundreds of thousands a month.

Slater: So at what point did it become clear to you that this wasn't going to get the traction that it needed?

Faggin: Actually I was already gone from Zilog because I was gone from Zilog at the end of 1980. The IBM PC was introduced in 1981, so by the time—in fact, the impact of the IBM PC was felt by me personally after I had left already the company and I saw that as that's it. The game is over as far as the Z8000 is concerned.

Peuto: I have to tell, I stayed at Zilog a couple of years more than Federico and I have to say one of the most educational and humbling experiences I ever had was to be for seven years in a company in which you design a product and seven years later you can really see the result of that product design. So it was very humbling, by the way. And the other thing that was the case is when the PC was announced I remember writing a memo and the memo only said, “I want one” but there is a twist to it. That product didn’t have the Z8000 in it so it was kind of-—I mean it was my gut reaction but it was kind of a very sad statement to make.

Slater: Well, should we close with some unique memories that each of you have from the project?

Shima: May I?

Slater: Sure.

Shima: Up to the Z80 there was not a big problem in both debugging and the test vector generations. Also it was not so difficult to find the good die for the characterization. However, there are so many instructions in Z8000 it is impossible to store all of test vector for debugging in the memory of test bench anymore. Also, MOS process was getting denser and denser and also the size of the defects in masks was getting smaller. That is it was not so easy to find the fully functional die. So I connected the test bench with the system that has enough size of main memory and floppy disc. This system was able to store not only tester results but also the result of a system bus at testing. Next, I asked a software engineer to develop many programs for easier chip debugging and also easier wafer sorting. This system worked quite well and shortened the debugging time.

Let me say some personal things. At about the end of 1979, I planned to purchase a new house but my wife told me that it is time to go back to Japan for educating three daughters in Japanese language because the promised five years had already passed and she supported me quite well for seven years. And without her support I would not able to concentrate on the development of the microprocessor. Then I returned to Japan in early 1980.

Peuto: I have many stories. The first one that I want to mention is the fact that I hope I will not—I know I may forget a lot of people but it was an amazing team and we’ve already mentioned Ralph and Federico and Shima and Charlie and Dean Brown and Judy Estrin. Beyond the Z8000 MMU and VMMU, the design group was doing five or six peripheral chips and for each there was only one guy doing them. The team did them for the Z80 and they did them for the Z8000. There was Ross Freeman (who went on the found Xilinx). There were Dan Hillman, Billy Carter, Peter Ashkey and I know I’m forgetting one or two guys. And when I was managing this group, because after I had done the architecture I became the manager of the group, I had the best team of people I ever had. It was just amazing.

Faggin: Gary Presenko.

Peuto: And I was going to mention Gary. Gary and I did the Z80 and in this case the rule about the second product being always a failure was wrong because that was my second product and thanks to him, to Judy and to Federico this product is still selling today.

Faggin: The guys are still in production of the Z8 too. But in a way that's less remarkable than the Z80 is still in production because the Z8 is a microcontroller so you expect that. The 8051 is still in production.

Peuto: Yeah, yeah.

Faggin: Not at Intel but at some other companies.

Peuto: I agree.

Faggin: But the 8080 is not in production. The 8085 is not in production. Nobody is doing that so that's the remarkable thing as a matter of fact.

Peuto: You're right in the strict technical sense but this is more of a personal question.

Faggin: Yeah.

Peuto: I thought the Z8 was a very good product and the fact it's still selling is just pretty amazing. So that was one point. And then the other stories I want to mention is one trip to Olivetti. We were trying to sell Olivetti with the Z8000 and so Federico comes to me and says something like, "You have to leave tomorrow for Italy." John Banning was the head of architecture so he and I decided to go and I don't remember who else was on the team but at least there was the two of us. And John Banning didn't have a passport so it was a big thing to get a passport in an emergency. And then we fly to Olivetti. This is a three-day trip. You fly there. You stay one day. You come back. And then when we came back—and when we flew there we arrived in the town where Olivetti is which is?

Faggin: Ivrea.

Peuto: Ivrea. And Olivetti would put people in this hotel that looked like a typewriter from the outside. Yeah it did look like a typewriter from the outside. And on the inside it was like being in a boat cabin. And it was pure high quality Italian design but that was quite an experience. And then we come back to New York and there is a big snowstorm so we get stuck in the airport sitting on the ground for 24 hours to get a plane, so that was my trip to Olivetti. So I think that was amusing.

And the other thing I wanted to mention was the whole Exxon strategy. We've talked a lot about it, but basically Exxon would suck up people, like they suck you up by going there, so I was named by Federico—thank you Federico—to be the representative to a group of people that were doing strategy for Exxon. And so I would go to New York like every week taking a red eye and so on and so forth. And the strategy was we were going to kill IBM and, by the way, this even made it to the newspapers, which is why in the end we got negative reaction from our customers.

And they wanted to have a paperless society. Now in a strict sense they were really wrong because we are printing a lot more paper than ever was envisioned when they talked paperless society, but at the same time they meant in essence the web without knowing that they were talking about that, meaning something electronic. But I spent hours and hours in meetings talking about future and so on and so forth and it was a big drain. I mean it was fun on one hand but it was unreal.

Faggin: It was not our business.

Peuto: Yeah, it was not our business. Oh boy, it was not our business. And the last comment I'd like to make is the amazing overlapping path that exists, you know, in some sense. Dave House here in the room behind the camera was the guy that we were fighting with when we were at Zilog with Charlie and Dean Brown and all those people. We met again with them on so many paths, so many different paths, so it was just pretty amazing. And in that sense, I want to second what Federico and Ralph were saying this morning: it was a life forming experience.

Faggin: Yeah, it was a very special company, particularly the first three years—two, three years. As I mentioned earlier on the Z80 segment, the energy level, the enthusiasm, the quality of the people, the esprit de corps. We had lots of fun together. It was a very good time.

In a sense it was exactly the kind of company that I wanted to have and to build but it didn't last more than three years and then sort of the weight of the large organization, the weight of some of the early errors, of having gone with an investor that wasn't the correct one, began to be felt. And basically I felt ever more kind of under this weight and powerless under that condition. Also, Ralph and I got into some issues and part of that was because execution wasn't getting done. I mean we were late. We were not manning right. We were not doing things that we were supposed to do and our relationship got affected by the execution because he was in charge of execution. He was the executive VP and so all of that contributed to sap that spirit that was there in the beginning.

And so sadly the company—also chance and fate and the fact that we had a powerful enemy that justifiably wanted to bury us. The combination of all those factors and luck of the draw on IBM and because I certainly think that a lot of that was a lucky circumstance for Intel and an unlucky one for us. And all of that made it into a very difficult situation. So at the end of, the middle actually of 1980 I was getting tired. I also was drained. I mean you can imagine building a company from starting from working 80 hours.

That's where I lost my eyesight by the way in the laying out of the Z80 and then building this company. I had an ulcer after the Z80 worked, so I think it was probably April or May. I got a perforated ulcer that I didn't know that I had an ulcer. I mean that's how I was mistreating my body. I wasn't eating because I was working 80 hour weeks but that didn't affect me so much. I mean the point was that all this weight eventually was overwhelming and had I been more experienced and known what I know today, Zilog actually could be where Intel is today because we had absolutely the vision, we had the people, the market was there, and had I known what I know today I think the company would have been a very different company. And it's not a regret because I learned tremendously. I could not be the man that I am today if I did not go through this experience, so this is not sour grapes or anything like that. It's just that that's what life brings and I'm happy to have gone through that.

Peuto: Yes, I want to second that. One other thing that was so real was, like he said, the first few years when we had this incredible run that we were on was that each one of us for whatever reason at one point had to disengage. People quite naturally left to do other companies, Shima for family reasons, so we all left. And generally when we left it was kind of painful because it had been such an incredible place before, so it's kind of like as if you were paying a price for the fact it had been so good. And in my case, what made me leave, I stayed there seven years, what made me leave was the day where one guy was leaving my group. You cannot not expect that. But I realized I couldn't replace him by somebody that was as talented as he was. The minute I realized that I couldn't build the team.

Faggin: Was that John?

Peuto: Yes, that was basically John. At that point I left.

Faggin: Yeah.

Peuto: It was the end for me. I don't want to finish on a sour note.

Faggin: But it was—

Slater: Closing comment?

Faggin: I mean my closing comment is that I think actually that for five years Zilog set the pace for the industry. Zilog was for its first five years of life, maybe four you can argue, the pacesetter in the industry, what Intel later became. I believe that and I don't think that I'm immodest in saying that. I think that we had superior architecture, superior execution and superior vision to Intel, but we did not do as good at execution over the long run and so Intel at the end won. But we made a contribution to the world. Witness the fact that the Z80 was used as a teaching microprocessor throughout the world for about 15 or 20 years after its introduction.

Slater: Well, thanks very much.

END OF INTERVIEW