Ted Hoff and Stan Mazor on their contributions to the Intel 4004

Interviewees:
Ted Hoff
Stan Mazor

Interviewed by:
David Laws

Recorded: September 20, 2006
Mountain View, California

CHM Reference number: X3774.2006

Abstract: *Ted Hof and Stan Mazor describe their work on the design and development of Intel's first microprocessor family the MCS-4 in the 1969 through 1972 period. As manager of Applications Research, Ted suggested redesigning a set of one dozen custom logic circuits for Japanese calculator manufacturer Busicom into a more flexible general purpose solution using fewer expensive packages. He hired Stan Mazor from Fairchild to help configure the architecture around a 4-bit CPU (the 4004). They describe the process of developing the logic design in association with the customer engineer, Masatoshi Shima, and working with Federico Faggin who translated their concept into four MOS silicon gate integrated circuits that today is recognized as the first commercial microprocessor chip set.*

David Laws: Stan Mazor and Ted Hoff are with us this afternoon to answer some questions on the early days of microprocessors at Intel, and specifically we're going to talk about the development of the 4004. Stan, could you give us a short description of your assignment, and what were you doing at Intel when this project began?

Stan Mazor: Well, just before that [joining Intel] I was at Fairchild, and also at San Francisco State College, where I was working on the IBM 1620, and I had a chance to work on one of the fancier 1620 computers. I heard that Ted Hoff down at Stanford had a really fancy one, a computer that you could talk to, so I arranged [delete--while I was down at college] to go down and visit Ted, and that's the first time we met. I think that was around 1962/1963. Anyway, I ended up working at Fairchild on a computer architecture and decimal floating point unit, and then I was recruited by Ted to go to work at Intel, when he was working on the Busicom project. I joined him in September and that project was underway already. [And I suppose my recent experience in decimal arithmetic was relevant].

Laws: That was September of what year, Stan?

Mazor: That was September, 1969. One of the things that was quite clever about Ted's management style is… that he posed a question to me and said, "Well, if you wanted to do a computer on a chip for Busicom, what would you do?" I had been working on a very large-scale computer, and my thoughts were really in another arena. So I thought a little bit about it, and I came back with some proposals that were pretty far away from what we actually did. I think it was quite smart of him to have me thinking about it independently, but he had already made a lot of progress by September on the 4004 design.

Laws: Okay, Ted, how about you? When did you join Intel, and how did you get involved with the project?

Ted Hoff: I joined Intel right at the beginning. I was employee number twelve, and joined Intel essentially the day that they finally had a place to open for business on Middlefield Road. My initial title was Manager of Applications Research, which meant really two functions. One was to try to help to define the kind of products that Intel would make and the other was to eventually teach customers how to use the products. Intel was founded to do semiconductor memory. Memory up until that time had been essentially all magnetic core. It was felt that there was going be quite a bit of education of the customer base if we were going to persuade them to switch from magnetic core memory to semiconductor memory. So that was how I joined Intel and what my initial responsibilities were. The microprocessor came about quite a bit after that. We had some discussions in the company about whether they should do some custom work rather than wait for these new memory chips to come out, and establish a business. The feeling was that if you do custom work, or you do a specific chip, or a set of chips for one customer, they'll be ready to take them much faster, as there'll be much less of a learning curve. And our first project was for a Japanese calculator company and the initial contact I had with that project, there was a meeting of the managers, the owners, of the Japanese calculator company in April of 1969. I was at the those meetings to find out a little bit about what they were trying to do with the way of custom logic

Mazor: If I could just interject that I was working at Fairchild [R&D] at a very fascinating high language computer, and when Ted offered me the job to work on memory systems-- now memory systems are pretty dull...

Hoff: True.

Mazor: ...and so I thought, "Well, this is a fascinating company, and Ted's a good guy to work for, but it's gonna be awfully boring working on memory systems. So it was kind of ironic that sometimes you-- I "You do the right thing for the wrong reasons, or the wrong thing for the right reasons." This turned out to be the high point of my career - working for Ted on the microprocessor, which was a surprise to me when I got there. A wonderful story.

Laws: At Fairchild were you working in the Digital Systems Research group for Rex Rice?

Mazor: I was working for Rex Rice back at Fairchild, yes. One other small point that we'll get to and that is that Ted already had the idea of the microprocessor, and of course he'll describe, I'm sure, that it was in somewhat conflict with the Japanese design. But the Japanese guys, two of them, they were my office mates, so I had to live with them everyday.

Hoff: Complaining about this other guy. <laughter>

Laws: So what was the reason for proposing the microprocessor solution for the Busicom project?

Hoff: I had attended that April session. I had a pretty good idea what the cost targets were. Finally it was around, I think, mid-June of 1969, a team of three Japanese engineers, including Masatoshi Shima, came to Intel to transfer their design over to the Intel design team. I got the assignment to act as liaison for them, with no design responsibility. Just help them find the right person to talk to within the company, whenever they had a problem. But I was curious, and I'd been playing around with computers for some years, and I was quite curious about what went into a calculator. They were talking about a family of calculator chips that would be used to make a number of different calculator models, varying considerably in and memory capability, and in processing capability. So I looked over their shoulders as much as I could, and got a little concerned when I saw what it was they were expecting us to do. They had quite a few chips, and Intel had a very small team of MOS designers. I think there were going to be something like ten or twelve different chips by the time the family was done. It used serial binary-coded decimal arithmetic and read-only memory to customize the chipset for different models of calculator. But in looking at it, it seemed to me they weren't making very effective use of the read-only memory. So my first thought was, "If we simplified some of the instructions that they were having their system execute, and move those into read-only memory subroutines, maybe we could cut down on the number, or the complexity of the chips that they were doing."

Another aspect of my job was working with some of the designers who had been looking at early dynamic RAM, and had gotten involved in that. So I had a feeling that maybe dynamic RAM might be more effective than shift registers for the storage of the calculator. If you use dynamic RAM instead of shift registers, there are a couple of advantages. One is that you don't have to wait for your data to come by, you can simplify the control logic quite a bit, and (at least using the dynamic RAM cell that we were playing with at the time) you only use three transistors per bit of storage, rather than six. So that was another aspect of saying maybe we might want to modify the design to make it more cost effective. The last thing was that they were talking about using 40-lead packages, and I think our cost targets were on the order of $6 a chip. And at the time the people over on the packaging side were telling me that 40-lead packages were going to cost us like $5 a pop by the time we had them put together. So we never could have offered the products at the target price if we had to use that type of package. So another thing was to look at, "Is there something we can do to cut down on the number of interconnecting leads. The more you more you compress the logic into one space, the fewer external leads you need. So that was another reason for looking at ways to simplify the logic. So those were some of the motivations.

Mazor: If I could interject. Also, they had done floating arithmetic design. And they had flow charts, and the primitive instructions that they expected that the underlying computer engine would be capable of doing was like a 16-digit addition, 16-digit subtraction, and then multiplication and division were done conventionally by adding and subtracting and shifting. So they had assumed that the processor was a pretty smart processor, but they were going to do the floating point arithmetic under raw micro-programming control.

Laws: Stan, at what point did you begin to interact with the ideas that Ted is coming up with?

Mazor: Well, when I joined Ted in September [1969], he was pretty far along with an alternative design consisting of a ROM chip, a RAM chip and a CPU. And the CPU working on 4 bits. I have to say that in the state of mind in computing [of that time], if you had proposed a 4-bit computer people would kind of laugh at you. If you think about it today, we had 8-bit computers, and then 16-bit computers, then 32-bit computers. So similarly, in the minicomputer domain, a 4-bit computer would have seemed pretty small, and people would tend to laugh at you. But the application that we had was, in fact, working on a single digit at a time. And so when I joined Ted, he had one other chip that was proposed, which was a timing chip. Since I was new to that project, I raised the question, “Well, why is there a separate timing chip?” And ultimately Ted thought about that, and integrated the timing logic inside the CPU, so we ended up with a three-chip design.

Hoff: And a one-chip CPU. So that was the key.

Laws: Ted, you’ve talked about one of the unique solutions you brought to bear, using Dynamic RAM as a way to both speed up the processor and to reduce the die size. You talked about the need to restructure the logic so you could get the minimum pin count. Were there other unique kinds of challenges you faced in forcing all this complexity into a 16-pin package?

Hoff: Another one of the challenges was that a lot of the chips were basically custom peripheral driver chips. They had a chip that was meant to handle the printer. They had a multiplexed LED display, or might have been another type of display, but it was multiplex. So they had a multiplexer chip to handle that. And they had a keyboard scanning chip, which included multiple scans and logic to do de-bouncing of the key presses. And it seemed to me that if we had a powerful enough processor-- and generally, when you’re putting stuff in, or you’re putting data out, you weren’t doing calculation, so it seemed to me that we ought to be able to have the processor do more work for the interfacing of those peripherals. In that way we eliminated most of the peripheral chips. We finally settled on adding a single static shift register, which we used primarily to extend the number of drive leads that we had on the output to reduce peripheral logic. So we ended up with a four-chip set. There was the ROM, the RAM, the CPU and the shift register. One of the things that was in the original designs was that each of the memory chips had an additional IO, or output port on it. The feeling was that output ports were more important than input ports, because multiplexers were fairly cheap, and you could use an output port to drive the control pins on a multiplexer, and then multiplex stuff in through an input port. So we made it such that the read-only memory could be mask-programmed to do either input or output, and the read/write memory would be output only.

Mazor: When I was working for Ted, the assignments that he had me look at specifically were-- how to write programs to scan the keyboard, write software programs that are stored in ROM to run the display, and to run the printer. In each of those areas, he already had some ideas as to how we would do it, and some sample coding. I’d done quite a bit of programming at that time, so I was writing sample, what we called “snippets,” pieces of programming code demonstrating the feasibility. One of the problems we faced with the Japanese team was an adversarial approach, if you will.. I remember one of their favorite questions was: “Well, show me, when the result of the computation is negative, how we’re going to print it out in red,”--- because that was a requirement in the calculator. And of course, if you had a printer chip, [or] you might have that facility to print in red as built-in, but of course, we’re going to do that in programming. So part of my job with my office-mates was to take snippets of code and go over some of the features that they needed in their calculator and figure out how we do it in “software.”

Laws: So, Ted, yours was largely the logic conception?

Hoff: Yes.

Laws: You had the programming side, Stan. And the third member of the team, I understand, was Federico Faggin.

Hoff: Yes.

Laws: At what point did Federico join you? And how did you work together with him?

Hoff: Federico joined much later. There were quite a few steps. Actually I did a little bit of programming as well. One of the first things I did was to demonstrate that I could scan a multiplex display and the keyboard at the same time. In other words, I would use the same drive electronics that did digit select in the multiplex display, and use that same signal to drive a keyboard, and then let's say have four potential keys and read that value into the processor and then do subsequent processing with it to show that that could be done, and could be done at a timeframe that could maintain the display adequately and do the keyboard de-bouncing and so on. That was one of the tests I did quite early on, before we got approval to do this chip set, which I believe happened informally around October of 1969. And there was a formal contract written around February of 1970. But Les Vasdasz, who was in charge of the MOS design group had the job of staffing for this project, was the one who finally located and persuaded Federico to join the group. That didn't happen until April of 1970.

Mazor: Federico joined as my office mate, and then my principal job was liaison ---to try to explain what was in the architecture and what we had to do [in the chip]. And so having worked with Ted on the architecture, and worked on sample programs, and working with our customer, I was then working with Federico. We were elbow to elbow in a very small office in Mountain View.

Hoff: One thing that really helped was that Federico had prior experience working on calculators and computers and so on. That was a tremendous advantage in transferring the design information. The typical MOS designer might not have been able to run with the project nearly as well, if they hadn't had that kind of experience.

Mazor: One other minor interesting point, Federico's wife was away on vacation in Italy, and here he started out with four chips to design, and three very important chips, so he was working till midnight, really burning in the hours, and did a sensational job in implementing it.

Laws: How long did the project take from the point that Federico joined you to getting the chips out into the marketplace?

Hoff: He came in around, I believe, the beginning of April of 1970. We had done some very rough sketches, primarily just to do transistor count estimates, and he said they were of no use to him. So he said he was going to start from scratch, and I understand that's what he did. I believe he had working silicon early 1971. And I believe-- you probably have to talk to him-- he had life in the CPU around the end of January of 1971. There were still a few problems, but he was able to demonstrate that if those problems were fixed, it would work. And then I think he actually had working silicon about maybe the end of February of 1971.

Mazor: One other point I'd like to mention in terms of innovation. Ted and I had worked on the IBM 1620, so we had shared experience. And that machine had a sub-routine call instruction, which we can think of as [using] a little bit of a stack inside the computer. But people sometimes fail to realize that almost all of the mini-computers were running with core memory, and all the sub-routine call instructions relied on saving the sub-routine return address in the memory, which you could do, because you could write into core. So an important feature of this processor was the push-down stack within the CPU for implementing the sub-routine returns.

Hoff: Especially if the program is going to be in read-only memory. <laughs>

Mazor: It's amazing that some of competition in subsequent years came out with processors and failed to recognize the requirement that if you're using read-only memory where are you going to store the return address [you would need an alternative solution, like our stack within the CPU].

Hoff: One of the other machines that I had had some experience with was a PDP-8. And the PDP-8 does use the core memory as the place to store the return address. And that's when you call a sub-routine, I believe the first location of the sub-routine was actually the address where the return destination was stored. So that was not a suitable architecture for our read-only memory based applications.

Mazor: I was saying later Masatoshi Shima wrote a book about his experiences on the project. It's written in Japanese, so I can't quote exactly, but I believe one of the things that he said is that when his project was terminated, and our part of it was started: "he felt it was like his ship crashing on the rocks".

Hoff: That much confidence. <laughs>

Laws: Shima eventually joined Intel, I believe. Did he continue to work on the project after his ship crossed on the rocks?

Hoff: I believe his responsibility was generating the code for the first generation of calculators that were to use the 4004 and other chips in the family. However, I believe that Busicom as a company had been in cash flow difficulty for some time and if I remember right, they never really did buy very many of the devices. I think they essentially abandoned the project. But they did have working calculators. I believe Federico, has one. But I don't believe they were that successful in the marketplace. It was after that time that he did come to Intel, and he did major work on the 8080 under Federico's direction.

Laws: Continuing with the 4004 - I understand that it was introduced to the marketplace after negotiations with Busicom to grant Intel the rights to the product?

Hoff: Somewhere in my files I had a copy of the February, 1970 agreement. I think Intel archives currently has it-- they wanted to get it and make a copy of it, so I hope I get it back, but that agreement actually had statements in it to the effect that if Intel were to sell the product to other parties, that depending on when those sales took place, there would be portions of the engineering charges forgiven. And if I remember right, the total engineering charge was to be something on the order of $60,000, and then it was scaled down from there. I don't believe they ever paid any of the engineering charges, but I don't have that information definitely, but that was my understanding. The marketing people were going over to Japan sometime around May of 1971, if I remember correctly, after parts were available and it became time to negotiate production quantities. Both Federico and I went to the marketing people and said, "You know, if you're going...,"-- in fact, I had heard that the Japanese were asking for price concessions-- I said, "If you're going to give them any concession, at least get the rights for us to sell it to other customers." And so that supposedly was done, but with some restriction that they not to be sold into other calculator manufacturers. So that was supposedly one constraint. But it turned out not to be all that important. We had lots of other applications for it.

Laws: Were either or both of you involved in going out to customers and promoting the product, and getting acceptance of this concept?

Mazor: We were extensively involved in that. For one thing, it was a pretty complicated product, and there weren't many people in the company who could grasp it, let alone talk to customers about it. So both Ted and I put on seminars and did some sales training, and ultimately made a [video] movie. So we were quite involved with that and certainly with applications for it. It was pretty obvious to us that we had a universal part that could be used for many applications. I think that about the only application that we clearly didn't think about was the home computer. We could never figure out why you'd want to have a computer in your home. <laughs>

Hoff: Another aspect of that are the peripherals that a home computer would need, things like printers, and mass storage and any of these other things were very expensive! So maybe you were paying, I think the number was $20 or $30 for a 4004, but for maybe $100 or so you could buy enough TTL to make a processor that would run ten to 100 times faster. So if you're going to have thousands of dollars invested in peripherals, why would you save $70 or so and throw away 90 percent or more of your performance? So that was one of the reasons we were not enamored of the idea of computers for the home, or even desktop computers as an application for the 4004.

Laws: So you had to find earlier adopters in some other sphere. Can you give me a few examples of some of the early applications?

Hoff: One we used to like to promote was the elevator controller. In other words, a smarter elevator. Some that came to us were gasoline stations; recording the amount of fuel, using it for inventory control and things of that nature. Basically, the small processor would be used as an information gathering device to record how many gallons were pumped from each of the different pumps in the station. And that information might be sent to a central location perhaps once a day, and from that they would know, you know, how often they had to refuel and things of that nature. <laughs> The wildest one I remember was that somebody supposedly put microprocessors on cows The idea was to record when the cow went to the salt lick and when the cow went to the drinking stand. And so to get an idea of how much salt they consumed and how much -- and then to correlate that with milk production.

Laws: Do you have a similar favorite story, Stan?

Mazor: Well, just two, and they're in opposite directions. One was someone calling up and saying, "I work for the phone company, and I'm going to quit my job to do something with this component," someone who was extremely enthusiastic about it. And someone else who would say, "I don't believe you have a computer on the chip. This must be just an arithmetic unit, or some mistake," or something like that. So they would go from stark belief to stark disbelief. <laughs>

Laws: At what point did you begin to realize what a prize you'd created?

Mazor: Well, I think just a couple of anecdotal things. When we created it, it was obvious to us, having worked in computers for years, that this was a general purpose gadget that could be used extensively. But I remember watching the stock price on the market when the announcement was made, and it didn't move a quiver. In other words the significance of the component to the Intel as a corporation wouldn't happen for years later.

Hoff: So there were a couple of things. Stan mentioned that we gave seminars. There had been a series of seminars offered by Intel Memory Systems and they told us they would send out invitations to potential customers. They'd send out 100 invitations, they'd get maybe 30 responses that would say they were coming, and 15 people would show up. So when Stan and I started going out on the road, they told us that's what to expect. And I think they booked some of the meeting halls along those lines, and we might have twice as many people show up as had given responses. We had very good turnout for those sessions.

Laws: So there was awareness growing in the engineering community what this could potentially mean.

Mazor: Thanks to the marketing campaign, we did a couple of clever things through marketing. And one was we called it "The Alternative," and the idea was that this was an "alternative" to logic design. So what we were basically trying to do is convert logic designers to become programmers.

Hoff: And another aspect, both Federico and I had other tasks in the lab in terms of building test equipment, test jigs and the like. As these products became available, we were saying, "My goodness, that would make this design job or that design job so much easier!" In fact, one of the things I was building was a device that burned [programmed] EPROMs. The first one we made, we did it with standard logic, and then they changed the design of the EPROM on us. It took a long time to get the logic debugged so we decided, "This is much too difficult." So we did a second design using the 4004. Took us a while to get it debugged. We ran into some problems with a tape reader. But once we got that straightened out, we were quite happy with the way the design turned out. And as I say, they did make this change in the EPROM, and we were able to reprogram our burner for that [change] in like one day. We were in contact with one of the major producers of ROM burners at the time, and we told them about this experience, so they started looking at microprocessors as something that they should put into their machines.

Laws: I spent a lot of time working on programming PALs at AMD, and at Altera, and then at QuickLogic. So thank you very much. You saved our bacon many times by allowing us to change an algorithm overnight to get out the next morning.

Mazor: From a broader viewpoint that I could reflect that when you're in the memory business you sell dynamic RAM chips, which Intel was making, to just a few companies such as IBM, GE, RCA, Xerox. You only need one or two salesmen, and it's a commodity kind of product that gets designed in and they use a lot of it. The microprocessor was almost the antithesis. You ended up having thousands of customers. Each of which needs some degree of support, and whose applications for the computers might only be a few thousand. So it was a completely different positioning for the product and the kind of company we were [previously].

Hoff: Yes, but also on that, someone who probably doesn't get that much credit for being involved in the microprocessor is Andy Grove. It was Andy Grove who recognized that so many of the sales were through distribution and that the feedback from distributors was deficient in terms of being able to predict the production requirements that he put pressure on marketing to start to get some programs to get better feedback through the distribution chain. And later on I saw what could happen after I joined Atari. I saw what happens when a company doesn't have adequate view through their distribution chain. <laughs> Andy Grove deserves a lot of credit for that vision.

Laws: Any other thoughts before we sign off?

Hoff: Well, one thing is that people may look at the 4004 and think of it as an awful instruction set, but the constraints that we had for the calculator family almost, to me, seemed to make it fall into place. One of the things was that they were talking about up to like 16 digits in a display and were doing binary coded decimal arithmetic, so that suggests four bits for a quantity. And they'd say, "Wait, if we have a binary machine, four bits will select up to 16 locations, so why don't we get rid of the decimal part of it, and make it a binary machine?" Then I thought, "Wait a minute! If you had two binary coded decimal digits, you get a binary result in the range of zero through 19. So all you need is a decimal adjust accumulatory instruction to convert it back to binary coded decimal." So instead of having to have binary coded decimal arithmetic, all we need to do is this one little instruction, which is much easier to implement. There were so many things like that, that fell into place in the instruction set that it just seemed to write itself, almost. <laughs>

Mazor: my customer had originally proposed a machine that could do 16 digit shift left, and 16 digit shift right. And virtually he thought of it as a much smarter computer, and we built a computing engine, which was a lot dumber. So one of the things I did in software in the 4004 application for the calculator was...I wrote an interpreter that made the machine virtually look like the machine that our customer had proposed. So since I was between-- what the customer had originally wanted, and what we actually built, this sort of eased the tension. It gave him a way of looking at the machine through sub-routines and what we call interpretive code. So actually inside the desktop calculator, we had an emulator emulating a higher level machine, but written in machine code. So that was quite exotic application.

Hoff: Well, another aspect of the design, our original target was that it would run with a one megahertz clock, and that's what people in the MOS group had told us. I think it was Federico finally decided he'd have to take a little bit of a production hit if he maintained that. So he backed it off a little slower. But if we had maintained that speed, we had a decimal addition loop that was 80 microseconds per digit. And that was the same number as the IBM 1620 that Stan and I had both had experience with. So we almost made performance that was equivalent to the 1620. Now the 1620, the machine that I used at Stanford cost us $2,500 a month <laughs> in other words to lease, and that was with an educational discount, which I believe ran like a 60 percent off list price. So those machines were far more expensive than the 4004.

Laws: A dramatic example of the cost reduction achieved over of time in this industry.

Hoff: And that was just ten years.

END OF INTERVIEW