Oral History of Stephen “Steve” Bisset

Interviewed by:
Paul Sakamoto

Recorded: July 30, 2014
Mountain View, California

CHM Reference number: X7240.2015

Paul Sakamoto: Welcome to the Computer History Museum. Today we're speaking with Steve Bisset, founder of Megatest Corporation and currently CEO of Terrajoule. We're going to ask him about his time in the semiconductor test industry, and how that influenced the development of the microchips with which our culture has become so co-dependent.

So to begin with, Steve, why don't you tell us a little bit about where you came from, your early life, and how you managed to make it over to Silicon Valley and the US in general?

Stephen “Steve” Bisset: OK. Well, I was born and raised in Australia. Ended up in Canberra. My father was a civil servant. I led a very normal, boring, suburban life. It was very “middle” middle. And so it [Australia] was an explorational culture, an adventuring culture. And when things are very boring and you live in the bureaucratic capital city, you do have sort of an urge to get out to see the world. So I could have had an easy life if I'd just stayed there. But no [I didn’t].

I graduated from high school 1968 with my friend Richard Swan, who is now also in Silicon Valley. We had a light-show business and a surfboard exporting business going. I did not have any interest in going to college, but I did want to get over here [to the US]. This is exporting surfboards to the USA. So I did want to go over here and do some sales, because I wasn't too sure how it was going at this end. There were a couple of guys over here. There was a girl I met in high school. So I needed to get over here. I didn't really think about going to college that much, except that I needed a visa to spend the summer here, and it seemed like the easiest way to get that was to apply to college.

So the girl's mom, who was a professor from the University of Michigan, kept telling me to do that. And I said, “Well, where should I apply?” She said, “What subjects do you like, young man?” And I said, “I don't know—math, science.”

And she said, “Well, apply to Caltech and MIT,” which I'd kind of heard of, but not really. So I went down to the [American] embassy and did the tests and got my student visa and spent the summer of '69 in a van full of surfboard samples hitting every surf shop from Maine to Key Largo setting up distributorships. For various reasons that [i.e. the surfboard business] wasn't going to work out.

And Caltech sent me a card saying that since I played rugby, which was kind of close to football, and they were really desperate for football players, they would give me free room and board for two weeks if I came and joined their football team. So that was actually the thing that caused me to actually go to Caltech, instead of actually just using them as a way to get an F1 student visa so I could spend the summer selling surfboards.

And then it turned out to be a pretty cool place. And eventually I took some courses from one Carver Mead, who was extremely inspirational and influential to me. I learned about Silicon Valley. I got the smell

of it. I could feel the swirl, the vortex of all the innovation and the energy that was flowing into it. And I kind of went, I want to go there and be part of that. So that's how I got to Silicon Valley when I graduated.

Sakamoto: That's fantastic. So it was really just a series of non-planned events.

Bisset: There was no master plan.

Sakamoto: No master plan. All right. And actually, did you even have a vision of Silicon Valley, or the technology culture here, before you left Australia?

Bisset: Oh absolutely not. Never heard of it. And I chose Caltech over MIT, because they gave me a scholarship I could afford. And I thought it was close to the beach. And I had visions of Annette Funicello and all that. And we actually go to Pasadena and its completely different. But it turned out to be an awesome school from an educational aspect.

Sakamoto: Wow. So then after you graduated Caltech, what was your first job? Can you tell us a little bit about what you did there?

Bisset: I had worked the summer of my junior year at Hewlett-Packard for Charlie Trimble developing a pulse generator product. And I was very impressed by both the company and Charlie. So I went to work for Hewlett-Packard for their IC design group. And I worked there for six months. And it was incredibly boring.

It was just a division in a great company that at the time didn't know what they were doing. They had an out-of-control process [to fabricate their integrated circuits], so I begged them to invest in a process monitoring system, which I developed. So it was kind of an aimless six months there.

But I'd really wanted to work for Intel in microprocessor development, because that's where the extreme excitement was at the time. And they had offered me a job doing analog circuit design for memory chips, which didn't float my boat at the time. And then six months later they rang up and said, “Do you want to work in microprocessor engineering?” And I went yep, and I went over there.

Sakamoto: So it sounds like they perhaps had initially recruited you out of college, but you had chosen HP instead.

Bisset: Right.

Sakamoto: Interesting. So tell us about your time at Intel. What was your job function there in the microprocessor group?

Bisset: Let's see. I have a claim to fame in Intel. I discovered the 8080. Which means that my first day on the job, they hadn't gotten it to work yet, but they'd had a mask error, and they had done a new run. And they had a pile of packaged chips. And they said, “Here, kid. See if you can find one that works.” So I put them in [plugged each 8080 into] this little switch box with LEDs and toggle switches and went through and checked out all the machine code. And I found one that worked. So I discovered the first working 8080. I found it.

But I was hired in that group by Ralph Ungermann at Intel. And it was Federico Faggin's group with Masatoshi Shima. And the 8080 was just hitting its stride. They still had all the 7400 series chips around it [to interface it to the rest of the system]. And they had kind of taken part of what you could do [with a large number of 7400 chips] and gone

[SUCTION SOUND]

with the 8080 [miniaturized it with a programmable CPU]. And there was the rest of the stuff. [The 8080 still required a lot of discrete 7400 series TTL components to be operational as a system.] So I was charged with defining and architecting and, in theory, designing and building various chips that would surround the 8080 and make it more useful [e.g. easier to design into more applications].

And actually, in my 18 months at Intel, I believe I either was the primary person or an assistant in the definition of seven different chips in the Intel 8080 family—from interrupt controllers to floppy disk controllers—all sorts of stuff.

And that was a really cool time. I totally enjoyed it. I didn't leave Intel for any reason that was negative about Intel. I was having a thrilling time. I just had this other compulsion to do something else. And Intel was just this whirlpool of really smart, energetic, creative people that had just been drawn in. It was a special place and time there. And I had several bosses while I was there. But it was a very, very stimulating and productive time.

I never did get to design a chip. I mean, I went to school. I wanted to do a Rubylith, and to go from definition to architecture to layout to design, and debug and all that. I never got to do that, because it turned out they did not have a lot of engineers who could do product definition. Who could sit down with customers and try and tease out of them what they really wanted to do with it, and what was important and what was not. Then kind of invent solutions and sort of iterate that, figure out a useful thing to do that you could do [i.e. that was feasible within the constraints of Intel’s technology] that would solve their problem.

I guess I had a knack for that [product definition] process, and they had a lot of people who could do chips, but not a lot of people could figure out what the chips should do. So I kept getting pulled off projects to start new ones. And eventually I decided it wasn't my destiny to actually ever complete a chip design. So I didn't. But it was a really good time.

Sakamoto: And I think one of the interesting things would be that at the time you were working amongst a bunch of people who would later become real halcyon or key figures in the whole world industry of semiconductors—kind of giants if you were. And yet at the time, was it obvious to you that they were destined to be giants? Or were they already—had the status?

Bisset: I have to say at the time, it did seem that way to me. I felt like I was in the company of—I don't know if I would use the word "giants," but certainly that was a period in Intel where new engineers got to have lunch with Gordon Moore and Robert Noyce and Andy Grove. And they would ask you what you think about stuff, and you could rant off with your ideas, and they would listen. And they seemed—just the aura of them in the room—were like giants. But also like Federico Faggin and Ralph Ungermann and Masatoshi Shima and various others on the manufacturing side whom I got to know well. So yeah, it felt very special at the time, in the moment.

Sakamoto: That's amazing. So then you think about this amazing environment—and you could recognize it was amazing at the time—and yet you were drawn to go do something that was not designing chips. It was doing something else entirely. And what was the inspiration that pulled you out of Intel to go do what you did?

Bisset: It's kind of the entrepreneur disease. I had caught it early in life. I have no idea why. My dad was a civil servant. I didn't have anybody in the family who ran small businesses or anything like that. Just it seemed like starting—inventing things and starting companies to develop those things—was the way to be useful. And engineering, sales, marketing—it's all about being useful. And that just had an appeal.

I do not know where it came from. But certainly I was involved in lawn-mowing businesses and paper routes and light-shows and surfboard exporting, and really wanted to do that [a startup]. And I wanted to do something in technology invention. It's really easy to invent things. It's really hard to get a business off the ground that exploits those inventions. But that's where the adventure is.

So actually in my second job interview when I actually went to Intel, Ralph Ungermann had said, where do you see yourself in five years? And I said, well, I see myself running a startup company. And I'm going to go start one as soon as possible—which may not have been the right thing to say in a job interview, but he hired me anyway.

And then, 18 months later, I did it. But even before I went to Intel, I had a similarly inspired and motivated friend and roommate at the time—Howard Marshall—who was also a fellow school alumnus. I hadn't actually known him at Caltech. He had the same idea. So we had been trying to think of ways to start a company.

So it kind of was backwards. It wasn't the usual thing where you're working at a company, and you see an idea in the company, and the company won't support it, so you go spin it off. There were a lot of rumors like that [about] Megatest that people just made up.

But in fact we quite consciously had spent our nights and weekends for about two years just plugging away at different ideas—get an idea, it seems interesting, we'd think about how to do it technologically, we'd do some design, we'd do some costing, then we'd go look at financing and sales channels and competition. I don't know why. All that stuff just seems pretty fundamental and logical.

I think we went through about six different ideas before we came up with the idea that was the basis of Megatest. None of them were good enough. I mean, there'd be one of the key elements [necessary to start a company] that wasn't satisfied.

The engineering part was always the easy part—the invention part. Figuring out how you would sell it, why anybody would buy it, why they would choose it over the competition—that sort of thing. We just didn't find a package [of all of the necessary business elements] that fit. With the Megatest concept, the Q8000, we found a set of things that looked like it would work. All the pieces [were there]. So we said, right. Let's go.

But I will say that the idea was definitely inspired by what was happening at Intel. We pursued other ideas that were not. But you know, when I came to Intel, the 8080 had just hit. There was the 4004 and 8008. And they'd kind of seeded the field a little bit.

But the 8080 was taking off. Intel was making 3-inch wafers and selling samples of these things for $360 apiece. We looked at that and went, you know, these things, once we build a few of them, they're going to cost about $5. And you could just look at the power of what it did, and we were going, this is going to change everything. I mean, people are going to invent things to do with this—you could smell it. They'd go way beyond just replacing some 7400 chips in an existing product.

They're going to do this, and then the unit volume is going to go way up. It's going to explode. And then Intel as a company is not going to be able to test the things. I mean, we sat there and we looked at it, and this was the motivation. So because the test equipment that was available at the time was too big, too expensive, and too slow, and not thorough enough in its coverage. You could just see that this was an enabling part of this microprocessor revolution.

I think business idea number two or three was also a tester for this, but we couldn't figure out a solution to this high-volume production testing opportunity—or need—that we saw that really worked.

One of our criteria—we pretty much understood that coming from the outside as a couple of students with no MBAs, no business experience, and a couple of engineers with an idea—that if we were going to come out with a product, there was a pretty high sales threshold there. Credibility. If we were going to break in, having something that was a little bit better was not going to do. So, I mean, we had come up with tester concept version number one, that was maybe two or three times better [than the currently available test equipment]. And it just didn't seem good enough [to enable outsiders/nobodies to break into an established market].

Then one day I was going for a long walk, and just mulling it, and something clicked. And we just saw a different way of arranging the boards and the architecture and the concept. This was the Q8000 concept. I went and talked to Howard, and we started working on it, and it looked like all the pieces would fall into place—including being able to finance it.

We were students [i.e. still living on miniscule student budgets], we were sharing a house. We had put ourselves through college. We had [what seemed to us at the time] these amazing salaries. We had no idea how to spend money. I didn't know how to go out. I didn't know how to order at a restaurant. So money was saved.

And I had my initial stock option at Intel that I got to exercise the first quarter of once I got by the cliff [ie. option vesting period]. Intel stock had been thrashing at the time, and back then you could reprice stock options without going to jail. So Intel did that [repricing] constantly. So I had a stock option at $50 a share, and then it went to $30, and then it went to $18 or something. And then the stock bounced up to $80, and my shares vested, so I sold them. I got $10,000, which was my seed capital, along with Howard [Howard’s savings of $10,000].

So we calculated that it would cost us $20,000 to design, buy all the parts, build the first tester, and sell it to somebody. We were off by a factor of two. So we ended up borrowing another $20,000 from friends. We offered to sell them stock. They said no. You guys are crazy. This will never work. But we know that you can get jobs. So when you fail, we know that you'll go and get good jobs, and you'll pay us back. So they loaned us money for 7% interest.

Sakamoto: So, if I could go back for just a moment, though. You had a bunch of other ideas. What were some of the ideas, or one or two, if you can remember, that didn't pass muster?

Bisset: Well, I said there was one that was designed to solve the same problem, but architecturally and cost-wise it wasn't good enough. Wasn't going to pass muster. Another was everything was digital. So we

looked into digital tune-up meters for cars that you [would] go buy at the car parts store. That was something where we figured out how to do it. Then we figured out that other people could do it, and that the business was more controlled by the distribution channels than by whether you had a clever idea, and that we would work really hard and just accomplish showing somebody else how to do it. That wasn't going to fly as a business. So that was one.

Howard had had some ideas in the medical device field. I'd have to think about exactly what those were. And there were a couple of others. So it covered the range.

So they weren't all by any means anything to do with the job that I had. But interestingly enough—and that's a telling story—I wasn't in manufacturing at Intel. I'd been saddled with fixing a tester that engineering [the Intel integrated circuit design engineering department] had done, and I'd gotten myself out of that as quickly as possible. And that tester had some of the conceptual elements of what we did, but not all.

But it was my understanding of where Intel was going and just understanding that there was going to be a need, and having a sort of an inside feel for what the need was that really drove the ultimate innovation and concepts. And I think that's quite common. If you have a really good understanding of what the problem is that you're trying to solve, so that you can do something useful, then that drives invention, and you have a better chance of doing something useful

Sakamoto: So there you were at kind of an interesting juncture of the microprocessor. It’s about to hit the vertical knee, if you will, to jump up and take over the world. You could see that actually the infrastructure to support that was going to be way too expensive if the legacy technique was going to be the one that was used. And so you came up with a much more spectacular, cost-effective way of doing that. And that's what you decided to pursue as your business that was the core of Megatest. So you and Howard left your jobs and went out to go form this with some, what I guess today we would call, friends and family and angel funding.

Bisset: Well, actually it was all our own money at first. And then when we'd spent our own money, we got some loans from some friends.

Sakamoto: Wow. So that must have been an interesting time. Now full disclosure to anyone who's listening to this. I used to work at Megatest for Steve, and I'd heard this story before. But I think it's one that bears repeating. Most common myth of Silicon Valley is the garage shop company. And I think actually this was close, but not quite the same thing. And I'd just love it if you can retell the story of where and how your company was founded.

Bisset: Actually our garage was a brewery. So it wasn't available. We brewed really excellent, excellent beer in the garage. But things are conceived in bedrooms. And so I built my bedroom out with an electronics workbench, and that's where all prototyping was done. So it was a bedroom shop. Not a garage shop.

Sakamoto: There you go. So where did you actually sleep, then?

Bisset: In the bedroom. On a mattress on the floor, to the green glow of an oscilloscope.

Sakamoto: That's great. And how long were you able to maintain in that facility before you had to grow? What was the thing that happened that allowed Megatest to get past that?

Bisset: Gee, I can't remember exactly what the specific stimulus was to getting a little shop. I think it was on Walsh Avenue. It was somewhere in Santa Clara. It was sort of a 20 x 20 foot cubicle. And we still hadn't released the product then.

By the way, I want to give another bit of credit. Also, part of how we were able to get from start to deliverable product with minimum resources is that it got to the point where it was pretty clear we couldn't do it ourselves. And I did work 100 hours a week for the first two years. It was sleep five hours, grab some junk food, eat at your bench, go home, sleep five hours—seven days a week. And that's what we did. That was our capital. Definitely sweat equity. Especially in the summer, when the place was not air conditioned. There were pools of sweat on the floor. There were days when we worked in our underwear.

But another key thing was another engineer from Intel, Roman Rycerz, wandered in one day and said, “I want to work for you guys.” And we said, “Great. You're hired. We don't have any money to pay you. Would that be all right?” And he said, sure. But he did very well out of that arrangement. And if he hadn't walked in and done that, I don't think we could have quite gotten over the threshold either. So there was some good luck there. And I think just the excitement of what we were doing.

And by the way, another one of these constraints about what qualifies you for a startup, like I say, was financing. So we were pretty clear that we weren't going to get venture funded, so we didn't try. We went to a couple seminars on venture funding, and what do they look for, and nope, they [the things VCs look for] were not on that checklist [the list of what we had to offer]. So we had to come up with our own funds, which knocked off quite a few ideas that we had, because they required too much capital.

And it took us 11 months from when I quit my job at Intel till when we delivered the first product. Howard had quit the year before that. So he had made that commitment earlier. And what was the question?

Sakamoto: I think the question was, when did you make that jump, and why, from the bedroom shop to the outside office.

Bisset: I think when we probably had a third engineer in, and we just needed physical space to do stuff. And it was very cheap. [Correction: We moved into a rented shop space before Roman joined us. I think it was in anticipation of needing more space. We were confident we would need it.].

Sakamoto: I think another follow-on to that is, if we can back up a little bit. So you went out to do this 10 times less—or actually, my view is it was about a 10 times more cost-effective solution than the existing legacy one. So maybe we could talk a little bit to what was the legacy solution like and—in a high-level, technological way—how is that different from what you went out to create with Megatest?

Bisset: I mean the legacy solution—there were two companies that were really in the game there. One was Fairchild Test Systems, that I think at the time was still part of Fairchild Semiconductor, if I'm right. And then there was Teradyne. So Teradyne would be making test systems for small-scale integrated circuits, and was trying to stretch that into the microprocessors—the LSI—large scale integration—that was coming out.

But Fairchild had really kind of grabbed onto that, and they'd adapted a system and expanded it. And they'd made a system that was sort of refrigerator-sized and cost about $300,000 that had some circuitry to condition signals to provide inputs to the pins and look at the output. And they used the bank of memory devices to throw vectors—test vectors—at the chips—ones and zeros to exercise it through its paces and see if it functioned under the conditions.

So this was very slow, and it was programmable. So this was equipment that was really driven by the need to be able to fairly easily debug and try out a system [chip] in the engineering lab. Then they sort of tried to be dual-purpose—engineering and manufacturing—which is great if you can do it.

We just saw that the constraints of making something with a program that was very easy to fiddle with and modify in the engineering lab resulted in them pursuing an architecture that was very slow and [low] throughput, and very expensive for production.

So what we did—and this did partly come from an approach that Masatoshi Shima taught me at Intel—this particular aspect of it—is instead of storing up all the ones and zeros, you use a reference device—a chip in situ—and you write a program and let it run the program. And so it's a little algorithmic vector generator that runs through and generates a vast amount of function that you couldn't possibly store in memory, and does it in real time, super fast.

Our innovation was figuring out a very cost-effective way to, instead of putting two things [chips] side by side, to extract the ones and zeros from that, and then condition them to provide the signals to test the chip on the test under very precise conditions, but really using this reference device as a real time data generator.

And then we figured out an architecture where all of the signal-conditioning circuits were built into the tester. You didn't have to replicate them per chip. But we had to do that with custom electronics to make this real-time data generator, which was different for each kind of chip that you're testing.

So there was more applications engineering investment per device, and so it was only really suitable for high volume devices. But massively cheaper cost once you put that thing into production. So we bifurcated between the engineering and the production application and said we're just going to focus on production. Make a tester for production was our model. So it was very focused on high-volume production tests.

Sakamoto: As it turns out, I spent some time as a customer of both of those machines while actually working at Intel at a later time than you did. And I would absolutely back up your assertion that one was easier to get that initial program going, but the other one clearly had advantages for thoroughness of testing, and also expense. So that was actually our migration path at the time, was always bring it up on the one, go to production on the other. And in fact, that's how I later eventually came to work for you.

But I think it was interesting that it did require actually, in my estimation, about a twice-smarter engineer to be able to actually get the Q8000 going than it did to bring up a program on the legacy machines.

Bisset: Oh, absolutely. Once we got our first system in at Intel, I basically moved to Intel. , I mean I was back at Intel. I spent 90% of my time at Intel for months and months and months training those twice-as-smart engineers how to develop the customized modules so that they could use it. So the investment in training the customer to become expert in the application machine was a key part of the success of that program. It's a lot of classes and a lot of hands-on.

Sakamoto: So now, after having done that, what were some of the key challenges in developing that architecture versus the legacy equipment? I should probably restate that. What were some of the challenges you faced in developing this novel architecture?

Bisset: You've pretty much laid it out, is that there's a lot of—in making a custom, real-time data pattern generator for each logic chip—there was a lot of design and invention in this board that we made. So it plugged in and out of our otherwise standardized tester. And that was really the challenge, apart from the other challenges of building a company when we'd never done it before, having to rapidly estimate escalating demand. Hiring people right and left. Trying to figure out how to build things. How to run a

manufacturing operation. I mean learning all this by the seat of our pants, fast, was definitely a challenge. So I was really just addressing the product challenge.

And the thing that actually really made the company, while we conceived it as a solution to the microprocessor and associated peripheral chip testing challenge, we made much more money off testing EPROMS by a long shot. It was kind of full circle. And it was actually Roman Rycerz who kind of really hammered this point one day, saying, here we are struggling for each chip. We've got a major product design effort [to enable the tester to test each new type of chip].

And in some cases, as with Intel, we'd been successful in training our customers to do it. But Intel was actually a special case. They were unusually set up and dedicated and competent to be able to be taught to do this. It wasn't very replicable. It wasn't going to expand elsewhere.

So we got back to finding we could make one module to generate the data sets in an algorithmic way for all different types of EPROMs. And there were a couple of orders of magnitude more EPROM chips per custom design module than there were logic chips.

So we got into the memory-testing business, and we hit that also at the right time when [there were] these UV erasable EPROMS with the little quartz window on the top. And the volume of those just took off. And so we were selling hundreds of test systems at a very high profit margin.

Sakamoto: So it's interesting to note that initially you were recruited and refused a job offer at Intel to go work on memories. You later came in when they said you could work on microprocessors. Subsequent to that, you rejected the idea of testing memories. You decided to go test microprocessors. But that actually ended up not being as fruitful as testing memory. So once again, you go back to the memory business, and that's actually when your company takes off.

Bisset: Yes.

Sakamoto: It's interesting how things work out.

Bisset: Yeah. Yeah. Since you pointed it out. True. You make money from repetition. And the memory business allowed us to do more dollars per iteration. And that was very profitable.

Sakamoto: So it's interesting to note that—at least my interpretation is—a lot of the lessons you learned were almost more business and management lessons than they were technological after forming the company.

Bisset: Yeah, definitely. I mean, we're engineers. We had the technology down. That [the technology] was [the] easy [part].

Sakamoto: So when you look at that, can you attach some numbers to how many units of maybe a particular logic design would get made at the time in that market, versus how many units you'd do to a typical memory design? Just roughly?

Bisset: Not really. That was a long time ago. It was thousands, and then tens of thousands, and then hundreds of thousands, and then millions. I mean, this was 1977, '78, '79, '80. I mean, it was going exponential.

Sakamoto: I was thinking more of the number of test units that you needed to create.

Bisset: Number of testers. I think between 1977 and 1980 we probably sold 200 to Intel, and a few dozen to other people. And then when we got into memory testing after 1980 for Samsung—you go visit a Samsung facility in Korea and you feel like you've come from a small village to the big city. The vastness of the test floor, and the rows and columns of wafer probers with our Q8000 orange boxes on them was an awesome sight. And so that was actually quite a lot bigger than anything even we did with Intel. So those were the financial underpinnings of the growth of the company.

But again, we're making product that was very simple by comparison. Very small. After we went through the iterations of fixing the various reliability disasters that came up and gave us various near-death experiences—one in particular—but we ended up with an exceptionally reliable product, and the customers came to depend upon it and like it, and we shipped them and serviced them and trained them and they kept buying them—as long as we're in a growth cycle.

So one of the really big challenges of being in the capital equipment business, as everybody in the capital equipment business knows, is that if you are selling product where the demand is generated by increased unit volume for the product, you sell product when the unit volume's going up. And when the unit volume goes flat, and your customers are whining that the revenues are flat, and things are really difficult, their purchases of production equipment from their vendors have gone to zero.

So you're really at the end of a whip in being a capital equipment supplier to the semiconductor industry. So there are a lot of stories associated with that. But survival and figuring out how to run a business that inevitably is cyclical like that—always was, always will be—still is, as far as I know—that was definitely one of the big challenges.

Sakamoto: Yeah, having stayed in it for a while longer than you did, Steve, I can absolutely guarantee you, it got worse. But the observation is absolutely dead on.

Bisset: You're clearly a glutton for punishment, right?

Sakamoto: Lack of adaptive behavior patterns—yes.

Bisset: I got out of it after 20 years.

Sakamoto: So during that time, I think your estimation was that you had an impact on Intel, but the industry in general. What do you think the impact was of that particular invention and what the industry was able to do at the time?

Bisset: We were a bit player, but we were, I think, a significant bit player, because the promise of large-scale integration is that you can do an awful lot of function in a small space. And smartphones and all that—that's where that all came from.

And that one of the many, many, many things in our little bit part was we made a significant contribution to enabling things to be built cheaply in extremely large numbers with high reliability. And you have to have all those things. But volume cost and reliability, and so we were doing very ….

We weren't compromising test quality with what we were doing. We were just focused on high-volume production test. And it was that particular focus that was sort of the strategic differentiator between us and many of our larger competitors, which allowed us to have a way to get in and give customers a reason to take a chance on doing business with a smaller company.

Sakamoto: So one of the interesting paradoxes, I think, in automatic test equipment in particular is that you're using essentially the products that you're going to help test to build the next generation of testers. And there's a chicken and the egg problem that seems to be kind of a conundrum. How did you view the interplay of that—not just in your first generation product, but going forward?

Bisset: Well, it's the classic challenge. You've got last year's tools to test next year's products, because there's a complex product development cycle and you never have—as you said—you never have the latest in technology available to test that technology when it comes out, because you had to spend two years and commit to a component design a couple years before that.

So it's just an engineering challenge. That's sort of the way it is. But what it did mean is that you've got a complex system. It's got an analog circuit design, complex logic design software, mechanical interfaces to the testing. And certainly at the time—I don't know if it's gotten any quicker—but even though we did our first product in a year, for these more complex products it's a two to three-year product development cycle. And it's a lot longer if you keep changing your mind in the middle of it.

So the real challenge there is figuring out what specifications to commit to for product, and sticking to it until you get it done. And the best way to do that, of course, is to engage with your key customers. Ask them what they think they want, and try to understand their business, and figure out if we think they're right about what they think they want. Because if you just take the checklist of what customers want, and you end up with extremely complicated products that do everything they need, but they don't buy any of, because they cost too much and they're late. They're late to the market. So you have to think ahead of that.

We had different phases in the company. We came out with first with the Q8000 product, which was … I mean, the fine detail of that was design. I sat down with production technicians in the Intel cafeteria, and we looked at every aspect. What are the little things that make this usable and workable? Where should the buttons go? What are the connections? What do you do with this stuff? Why? What's it like out there?

And that level of deep customer engagement was a definite factor in the success of the product. I mean the people who had to use the product liked it, for the most part.

And then we get to 1980. It was founded in 1977. By 1980, we were doing $10 million a year in revenue. We were making better than 40% pre-tax profit. We had caught the wave. We'd hit the explosion of microprocessor and memory expansion. We had a differentiated product. We could sell it very cheaply at a very, very nice profit margin. We had hired tons of people.

And then venture capitalists heard about us, because they give you money when you don't need it. They really begged us to take their money, and we kind of got sucked in. They described all the advantages. 1980 to '85 was one of those silly upside periods in semiconductors where you could raise money. Well, if you already had a product, and you were making money, so that you really didn't need money, it was really easy to raise money.

So there were a couple hard lessons there. There were mistakes that we made. I'll give you one mistake that was sort of operational. This is before computerized MRP systems, or ERP systems. It was all manual. We'd do our quarterly sales forecast. We had a couple of different product lines. Every quarter, the sales forecast for this product would be up and that one would be down; and then the next quarter it would switch. We really didn’t have a mechanism for backing out the thousands of different part types on order, so it would kind of ratchet up the inventory. There was a day when my CFO came to me and said,

“Steve, we've got a lot of inventory, and it's kind of used up the cash.” I said, “And when are we going to use up the cash? Oh, in about two weeks? Oh…”

So I called up our investment banker, who was Robertson Stephens [one of the “Four Horsemen” of technology investment banks in Silicon Valley] at the time, and they said, “Good. Let's do a mezzanine financing road show.” We went to Baltimore, Chicago, Edinburgh. And in two weeks we had $17 million. No need to solve the inventory problem.

We eventually did [solve the inventory control problem], and we went through and became an early user of computerized ERP, which was an intense, year-long transition that the company went through. We were also pioneers in outsourcing. We were one of the major early customers of Solectron [contract electronics manufacturer, now part of Flextronics Corp.]. We figured out how to do these things. It took a lot of work and a lot of discipline. Definitely, easy money with inexperienced management can lead you into some traps. That was one of them.

The other one [trap] was that we conceived that it was time for the tester-per-pin concept to come to fruition. Going back away from the original concept to much more programmability, but exploiting the fact that chips had come way down in costs per function, and that we could make something that was faster and better and more flexible and easier to program by rearranging the architecture of the tests. And this turned out to be the way everything ended up being done. It was technically correct.

But we thought we were so clever that we should keep it a secret. So what we didn't do was engage with our customers who would have been more than happy to give us all sorts of information about the ideas that we were coming up with—some of which were good and two of which were really dumb.

One is we hired a bunch of computer scientists—and Unix, boy this was hot. This was hot in the colleges. So we designed a test system that—the test engineers would program the programs to test the chips in Unix, interacting directly with the operating system. And we didn't ask them [customers] if they thought that was a good idea. It didn't occur to us. We knew this [that we should ask the customers], but we'd hired a bunch of people, and we forgot it. That what we think is right, if it's not what the customer thinks is right, it doesn't matter what we think is right. And if we had asked, they would've told us.

But we went way down that path. Then another company came in and did a far worse job of implementing a system. But it [the competitor’s system] looked very nice and much easier to use on the outside, and they sold lots more than we did. They had unhappy customers, but they made money on it—much more than we did.

The other one [large mistake] was just this mechanical thing. We had a very well-functioning test head. But it was giant, and for emotional reasons it put people off. We didn't ask.

Sakamoto: For people who are going to listen to this, or read it later, the test head is actually the place where the rubber meets the road, and it brings all the connections to and from the actual chip, which tends to be a tiny fraction or a few thousandths of the size of the actual automatic test equipment itself. All that has to be funneled down to a very tiny area. This is the part that does that. I think at the time the Megatest piece that moved around and did this and connected to the robotics was maybe on the order of four or eight times bigger than the next biggest one.

Bisset: It was. We had all sorts of reasons why it worked better as a result. I think it [actually] did. But our competitors came out with, “Look, they have large test head!” And then they'd say, “Hey, we have a small test head!” It was messaging, and it felt better to the customer. If we'd have understood that that was going to be important, we would have worked harder at figuring out a way of making a small one that worked well.

But we didn't. We decided and we did our own thought experiments. That led us down a couple of serious mistakes that cost us a lot. We could have had the dominant new technology there. We kind of had the basic pieces right. But then we had to go back and redo all that at great expense later on.

Sakamoto: I think it's actually very interesting that that was a big technological and economic jump from the legacy product line to go to this, from what had been tens of thousands of dollars per unit machines to millions of dollars per unit. And aside from the availability of easy money and some other encouragement, were there any other driving forces for kicking that off and taking that big jump, instead of doing something evolutionary? I might add that one of the products that happened between the Q8000, the founding product, that was actually sort of a rational “twice a Q8000” kind of product called the Q2.

Bisset: Yeah, and it [the Q2] was spectacularly successful. It was an evolutionary product, and it was also evolved as a direct result of our interactions with our customers and their needs and their problems. And so they bought a ton of them—I think ultimately more than Q8000. It was a product that had more programmability and more flexibility and cost two or three times as much. The unit volumes were extremely high, and that was really the financial underpinning of the company.

But we had grown. We were $10 million, $20 million, up to $30 million. I mean, we're starting to be somebody. And so we went after a product that could take the whole industry by storm. Hit the big time.

And a number of people said going from a small low-cost product to a large high-cost product—it's much different selling something for a million dollars than selling something for $50,000. You guys are going to find that it's a whole different business and you can't do this and it turned out that wasn't right. Customers were the same. We had, I think, really good sales people, and really good applications engineers. They figured out how to sell this stuff. It was really a question of what the customer problem was. The manufacturing transitions, we were able to execute that.

I'll go back to the problem was there were a couple of product-definition flaws, where if we had asked the customers before we were locked in, we would have developed a far more successful product. All these other so-called scale hurdles were ones that we managed fine. We just had a product-definition problem based on a bit of hubris that you get from being young and successful and having people throw money at you and tell you how great you are. We kind of all got into it.

This was exacerbated by doing very rapid hiring of very smart people from very good computer science schools who had not been part of the customer-driven, detail-driven, product-development process. They were very smart and were trained to believe so. An awful lot of those people learned from that a great deal, too, and subsequently became very practical and effective. We all kind of paid a learning price there together.

In 1985 we were just introducing this very expensive product with a couple flaws. Then we had one of those [economic] cycles. The Q2 and Q8000 were our cash cows. So here's something. In the fourth quarter of 1984 we booked $10 million worth of our legacy product. In the second quarter of 1985, we booked half a million dollars. So we had a 20 to 1 bookings drop. And, trying to do this very expensive product introduction at the same time—the arithmetic doesn't work.

So that was where we went grow, grow, grow, grow. And then we went into a dark tunnel for quite some period of time as a company. Which was also a learning experience.

Sakamoto: So that was kind of an interesting time, because here we have a market that actually, even at the time, most technological financiers would not have actually understood or heard of. And yet you were able to attract and retain some fairly well-known venture capital firms and keep them engaged through this whole time.

Bisset: Well, up until the crash. And then, let's see, I think it was either 1985 or 1986. We'd done a series of financings. We'd raised $42 million in venture capital. It was the easiest thing to do in the world. I mean, we'd done it in a series of financings where we never even needed money, went out and gotten some more, because we were profitable. I would explain the strategy of what we were doing and what its unique distinctions were and what was the driving force behind why the customers who buy our product versus the competitors, and how I understood that, and they would go like this, and I don't think they were listening. They just saw that we were making money and growing. There were notable exceptions to that.

But at a time when wise heads could have said, you're going down the wrong path here. Things aren't going to keep working out this way. We weren't getting as much of that as we could have. Maybe I was young and wasn't listening. Hard to know exactly what the history was there.

But yeah, in 1985, 1986, we had to take the company from 450 people down to about 200 just to survive. It was really heartbreaking. Those were my friends. I laid most of them off myself. I kind of thought it was the end of world.

We haven't touched on this, but the company was not only a venture, it was a social experiment and a management experiment. We did a lot of things because we wanted to, and we thought that would work, and we were full of our own ideas, and we didn't have to do things the old way.

Some of those ideas were very good, and some weren't. We weren't too good on accountability, but we were very good at communications and teamwork, and innovation, and hard work, and loyalty, and honesty, and all of those things. We had a mixed bag of management experiments. But there were definitely some interesting aspects of that time. We had to downsize.

We didn't even have a bank line. Couldn't get any money to cover our losses, so we had to stop making losses. I could go on about how we actually survived during that period of time, but from 1986 through 1990—I think in '86 and '87 there were eight quarters in a row where the one sale that we made that made the difference between having a profitable quarter or a losing quarter was booked, sold, and shipped in the last week of the quarter, eight quarters in a row. In an industry where you're supposed to have three to six-month lead times. And this is competitors saying, “Did you hear that Megatest is in Chapter 11?” I mean, all this sort of thing. So the first item on every sales call was our financial statements, and saying, no, we're actually here, and we're profitable.

But that was back to the roots. We got customer funding of a complete redo of our memory and logic product lines. We worked with IBM on the logic tester. We went back and worked with Intel on the memory tester.

So we kind of went back to what worked, but with more layers of experience and systems layered on top of that, and pulled it out. By 1990 we were back to profitability. We'd grown. We were doing $50 million and had two completely brand new and very solidly based technologies that were set to grow for the future. That was a turnaround. It was very, very hard.

Sakamoto: Yeah, I remember that's where I learned the phrase, "If the last minute wasn't good enough, they'd call it something else." In particular, the eight quarters of "this is the order that we must get."

Steve, one of things I think is interesting as a Megatest alum myself is sometimes I'll go in and I'll tell stories about what it was like to work there, and the different philosophies that we had—or you had—managing the company, versus everywhere else. And the reaction that I get lots of times—particularly from younger people—is, you made that up. I was wondering if there's any stories in particular that you're

willing to share on the record about the company that you thought were just exceptional differences that you had from maybe what people would expect.

Bisset: Somewhere I have it typed up in Courier from an actual typewriter. We worked on the Megatest Philosophy. I mean we sort of had something. And by the way, certainly for me, after a couple of years, Howard and I co-founded the company in 1977. We were two years into it. Howard came to me and he said, our customers are confused that we don't have any defined organization. We need a CEO. I went, yeah? OK. And so we actually tossed a coin, and that's how I became CEO of Megatest, and not just founder.

But it was definitely a teamwork thing. I mean, we were on an adventure. It was a family. Certainly in my 15 years as CEO, one of things that I had to learn after that was that the line between where “me” stopped and the company began was very blurry for me. This happens with company founders. It's something you go through. But it was something. It was just we're all in together.

Let's see. Particular things. I don't know. Things that stick in my mind are some methodologies. For example, teamwork was key. So at a time when the standard method of hiring people was that a boss gets some resumes—somebody's a manager of a group, and kind of looks at them and interviews a couple of people and hires somebody who may or may not be any good, or may or may not get along as a team with the other people. So you'll remember this. We had this when we hired somebody—and I'll say especially if it's in a management position—they would interview with their prospective boss, anybody that they'd have a working relationship with, all their peers, and anybody who was going to report to them.

Then all the people who'd interviewed that person—with the person gone—and this became our standard documented Megatest hiring practice—which was that we'd all get together in a room. And the culture was as equals. I mean, everybody understood that it didn't matter what level you were on the org chart, your input was really valued. And just share observations and thoughts and questions about this person. And it was very accurate. It was like a jury. Very accurate. I mean I think the same mentality—didn't think of it at the time, but very accurate method of identifying talent and qualities. Weeding out BS.

Really interesting when you get somebody in a management position to find out if they're productive or political. The political ones will tend to treat the interview differently depending on where you are in the organization. And then when we get together and talk about it, we would see that. Whereas if you're the boss, you can't see that, because politicians are very good at looking upwards the way you want them to look, which looks exactly the same as the people you really do want to hire. Otherwise, the politicians wouldn't be any good at self-selecting.

And so that was a practice, and also got buy-in. And so that was one example of an institutionalized practice that caused just a lot of employee ownership—apart from all the stock options—ownership in the

company and I think that was a factor in us getting through the hard times, because there was a lot of mutual loyalty there.

Sakamoto: Steve we were just talking during our brief break here about some stories about Megatest back at the time. And I guess as we're kind of coming towards the back end of our interview here, I think I'd like to ask you a little bit about what do you think was the influence and the role of software in what had largely been a hardware development? I think you mentioned before that there were some issues with software driving some of the factors in the development of the MegaOne product. But I think just in general, how did you see that evolve over the time that you were running Megatest?

Bisset: I think software engineering and hardware engineering—when I kind of stand back and look at them—in their fundamentals they're not much different. I mean the issues and the management issues that come up, I think it's a little easier for people conceptually to be kind of grounded in customer issues if they're doing hardware. But if you're involved in pure software businesses, you wouldn't agree with that at all. You've got a function, you've got to perform, and it's got to be useful. And the definition issues and the management issues and so forth are really the same.

So our issues that we had with software were more to do with internal culture and what we were trying to accomplish, and how related it was to understanding the customer needs, I think, rather than any particular difference between hardware and software.

Historically, because we'd started out as a hardware company, we had grown with our profitable product lines, which were mostly driven by hardware and firmware and very direct and light on software. Then what happened with software was that we hired a bunch of software people from very good schools who hadn't gone through the training of being successful with the customer. And they got hired at a time when we were getting tons of venture money. So in their heads they kind of didn't really seriously make and believe the connection between understanding what the customer needs and is doing with it versus the value of what they're doing. So I think that was just the coincidence of the timing that had to correspond with when we hired the software people. And we were in easy money phase.

And a lot of them had been trained at their school by their illustrious and eminent professors to believe that the future is Unix. We had people saying, well, there was all this belief about the structure of software, and the elegance of it, and it was beautiful. And it was all good, but a disconnect from “let's figure out what the product actually has to do.”

So if you build a complex software structure which actually executes very slowly, there's a mentality that, well, computers are getting faster, so you stay with your elegant structure and just buy a faster computer, without actually really analytically considering how fast is it? How fast does it need to be? What are we going to have in the way of computer power? Is this solving the customer problem?

To me, the problem that we had was an accident, the timing of when we started bringing in software people, and what was going on with the funding of the company at the time. But the lesson is: engineering—including software engineering—is about being useful. And being useful means you have to know a problem that needs to be solved, and you have to put a lot of work into figuring out what the problem is. And I don't really see that as any different between hardware and software.

Sakamoto: Yeah. It's interesting. I guess the song remains the same. If you're not marching towards the right spot, you're going to end up in the wrong place.

Bisset: Yeah. And for us, that turned out to be hardware/software distinction. But that was a timing issue. What was the climate in the company when we hired the software people? They didn't go through the struggle. We could hire people, and we could go off and do things for a long time, and spend a lot of money before the fact that it wasn't really addressing the customer problem became painful. The feedback loop—when you have lots of money to throw around—is way too long.

Sakamoto: It's interesting. And you mentioned before the pain that came from having gone through that particular cycle, which I would sort of say kind of answers the question about what was the worst moment. But what was the best time that you can remember at Megatest? Or if you've got a couple of them that just kind of stand out. There must have been some really moments in the sun where you said, yeah, we've really done it.

Bisset: I don't think we ever stopped and said we'd really done it. Things were moving too fast to do that. But certainly the late '70s, when the company was three, four, five, six years old, it was a heady time. We were growing rapidly. We were making tons of money. We were 100% employee-owned, 100% self-funded. We had issues all the time with customers, but we were fixing them. People were buying stuff and we were shipping it. And we were engaged with our customers and our suppliers. And the culture was definitely a work hard, play hard culture.

Another thing. We actually had quite a lot of siblings or spouses hired in the company. We developed a set of rules for that. Instead of saying no, we won't do it, we'd say things like, if you bring in somebody that you have a personal connection with, you are held to a higher level of responsibility for your recommendation, because of the suspicion that this could be a favoritism thing. It kind of added to the family atmosphere. But there were are also some very painful situations where you had people related to somebody else in the company, and they weren't working out, and you had to terminate them, and it caused massive angst.

So there aren't simple solutions to anything. If you want some upside, you have to manage the downside. But those were heady times. People would work long hours. People would party together. There wasn't a big distinction between work and family.

Most of us were young at the time. A few of us were starting to have kids, but most of us didn't. People had the time to work and play hard, and there was an enormous sense of belonging. And you can see that statistically, some of the numbers that come out is that because we didn't manage to be founded and then go public in, say, five years, and because we'd gone through this heady growth period where in say 1984 we were almost ready to go public, but then it crashed. And then we went through a five-year dark tunnel to rebuild.

And in the midst of that—well no, it was actually, into the '90s, we had a—I have to remember the year. We had a problem. We'd given stock options to everybody. The idea of ownership was a big part. We gave stock options to secretaries, janitors, what have you. Different numbers. And over more than 15 years of doing this, there was turnover, and we had accumulated a large number of shareholders.

So there was a rule that said if you have more than 500 shareholders you have to start reporting as if you're a public company. You don't have to do a public offering, but you have to do your 10-Ks and 10-Qs and be audited, and do all that sort of stuff. And that's a real pain in the neck. That's extremely expensive and disruptive. And our attorney said, you guys have 600 and something shareholders. We're going to have to deal with this, or we're going to be made to deal with it.

So we came up with a scheme. We figured out that if we could take everybody who had a hundred shares or less and buy out all those shares, that there were enough people there that it would get us down below the 500 count. So we consulted legal, and we made an offer that said we will buy up to 100 of anybody's shares—providing that if you have less than 100 shares you have to sell all of them. You can't keep one share.

At the time we were still struggling month-to-month. The fair market value of our stock was a penny. Our investors had all completely written off the investment—all of the venture capitalists. And we offered $1.50 a share for the buyback. I don't know what's not to like. We offered it to 600 and some people. We got three takers.

So I mean, because these are people that we'd laid-off. And they just went, no. Megatest is going to make it. It has to. And they wouldn't sell their stock.

So we did the public thing. And so we were already public for a couple of years before we did our IPO in 1993, which made the IPO very smooth from an accounting point of view, because no restating of the books or anything like that. But three takers, offering $1.50 for something where the fair market value was a penny a share. Actually, that moment was the very middle of a tough time, but it kind of warmed my heart. I thought, well, this is something.

Sakamoto: These people have confidence—why shouldn't we?

Bisset: Yeah. Yeah.

Sakamoto: That is an interesting story, because out of everyone, they're the ones who should have the least confidence, and they're saying no. Stay with it.

Bisset: Yeah. Exactly. And you know, those people, the ones who timed it perfectly later on, some of those people who wouldn't sell for $1.50 did sell for $100 per share. So they were right. Crowd wisdom.

Sakamoto: It is amazing. Why, I remember even at the time—if I can inject my own story again—that you and I were talking in the hallway, and some people had just left, and it had been when we'd come through the very darkest time, and people were abandoning ship. And you looked at me and said, “I could understand why someone would have left a couple years ago. I don't understand why they're leaving today.” Because they left just before the payoff.

Bisset: Yeah. But then a lot of them came back. People get tired. They need a break. And then they get revived.

Sakamoto: Yeah. It was great. It was an open place for that.

I think another question that I want to get down towards the end here is when you look at all this—you look at test, you think of Megatest, and you think of all these other different companies that were in play at the time, and the impact that they must have had on the semiconductor industry—what do you think was the impact overall that test—and in particular Megatest—had on the development of the microchip, and where we are today? And how do you view that looking back?

Bisset: No, that's way too strategic and global of a question. I mean, you could say, well, if we hadn't have done it, somebody else would, because there was a problem to be solved. But we did it. We accelerated that process. So I mean I think we just helped. Like I said, we were a bit player playing an important part in the growth of that industry.

For me, I started out just for the adventure of the invention and doing the startup. Didn't really think about it much. I wasn't leaving because I was frustrated with a big company. Didn't do it because I wanted to get rich. It was the adventure. And I think we drew in a lot of people like that. They were focused on what we could do, not what we could get. These are people who were asking not what Megatest could do for them.

But for me a couple years into it, it became about the people—not the technology—and about what we were doing as an organization, and how we did it. And that really became my passion.

And frankly, part of the one downside of it is that, in terms of the greater scheme, is the world a better place? I mean, chip testing is way deep up in the industrial food chain. I never could explain it to my mother. It's from an external point of view, an iPhone, yeah, that's pretty cool. But test equipment for a production line—why is that cool?

Yet we managed to inspire a lot of very talented people to come in and get the bit that this was important, and it was worth doing really well. A lot of those spun off and became very successful in other areas too. I think we have a great track record of having hired people who then went on to do other startups and who were very successful.

So to me, more the legacy, was that we were kind of the leading edge of bit of a management revolution. A lot of the things we did back then are now considered very ordinary in Silicon Valley. Company cafeteria. Not wearing suits. Focus on results. Efforts to do consensus decision-making where you can do it. Working on getting everybody on the same page. The idea that you can be nice to people and finish first. These were radical ideas when we were playing with them, and a lot of people told us we were crazy. You can't run a company that way.

And then there was that book “In Search of Excellence.” We read in that, and we said, hey that's us—or part of it. And so to me, looking back on it was the experiment with the team, with the management, with the organization, and how to get useful things done through that that was much more of a lasting impact than the technology itself. Which was important. But like I say, it's a cog in the big machine of the semiconductor industry. And each cog has to work well. So we did our bit.

Sakamoto: Well I think that is actually just a great kind of cap note on the history of Megatest and the part that you had in starting it and seeing it through. So I want to thank you for that. I do want to add my own personal note, though. I think you need to be a little more generous to the impact the company did have. I think, as I recall, it was the first DRAM tester at Samsung. And although they may have found something else that would have suited I'm sure certainly their purposes for their early couple of lines, the fact is that Megatest did enable their early economic success. It was the machine that probably enabled Micron to ever actually test their own first chips.

Bisset: Oh yeah. I'll cop to those. And just the 8080. Being able to have the first high-volume LSI microprocessor chip take off. It really would have been very difficult—it couldn't have actually built all those chips. And Intel, they did their calculations. How many buildings? How much real estate they would have had to fill with testers if we hadn't done that. And somebody would have done it two or three years later. But that would have made a difference. So yeah, on behalf of Megatest, I'll take that.

Sakamoto: There you go. Well, Steve, I want to thank you very much for spending the time with us today, and recording these memories on our oral history program. And so once again, thank you very much, and we hope to be able to talk with you about this again.

Bisset: All right. Thank you.

END OF INTERVIEW

