Oral History of David “Dave” Jaggar

Interviewed by:
Doug Fairbairn

Recorded: November 15, 2012
Mountain View, California

CHM Reference number: X6719.2013

Doug Fairbairn: Okay, today is November 15th 2012. We’re at the Computer History Museum in Mountain View, California. And I’m Doug Fairbairn interviewing Dave Jaggar. Is that proper pronunciation?

David “Dave” Jaggar: Perfect, like Mick Jaggar of the Rolling Stones.

Fairbairn: Okay. And we’re going to get an oral history from Dave and learn about his career at ARM and before and after. So, Dave, welcome.

Jaggar: Thank you.

Fairbairn: Glad to have you at the museum. You’ve just flown in from—glad to hear—you’ve just flown in from New Zealand, I understand.

Jaggar: Correct.

Fairbairn: So, I appreciate your coming right down here and spending the time with us.

Jaggar: Pleasure to be here.

Fairbairn: So, we’d like to just start out and you were born and raised in New Zealand. What was that like? What was your family life like? Tell me about the early years of Dave and how that evolved into a computer career.

Jaggar: Well, born and raised and schooled in Christchurch in New Zealand. Born in 1967, youngest of five children. My oldest sister is sixteen years older than me, so quite a spread out family. And my father was forty-eight when he had me. So, he was always kind of an old dad, but I think maybe a wise dad, too that he was—things were a little bit more paced.

Fairbairn: I was the youngest child. There were a lot of good things about being the youngest child.

Jaggar: The parents had learned a few things by then.

Fairbairn: The parents have learned a few things by then.

Jaggar: So, yes it was a great upbringing. We were a very poor family from still a poor part of town in Christchurch. It’s to the east. And recently we’ve had earthquakes in Christchurch, New Zealand. And that’s all—all that stuff’s all flattened now. My old family home is in total devastation. But yes, it was a poor upbringing, but we made do with what we had. And we went to the local schools.

Fairbairn: What was your father—what was your source of income? What was your family—?

Jaggar: We would call it a fitter and turner. It’s somebody that would make something out of metal on a lathe. I don’t know how that translates into American. And my mother was a seamstress. She sewed. And so, the both worked, which was unusual back then, too, to get a decent income, and put us all through. None of my brothers and sisters went to university. In fact, a lot of them didn’t even finish high school and needed to go and get jobs and go on, but I think, again, because I was the youngest, my parents had a little more financial stability by the time I was ready. And, unfortunately, my father died when I was eighteen, just as I started university. So, he never saw any of this. But he told me—and it ended up being the last few lines of my master’s thesis, he said you always try and be both practical and theoretical. That was always the man he was looking for. And I never forgot those words. So, I’ve always tried to be practical and theoretical.

Fairbairn: So you’re growing up. Your other siblings went off to do other non-technical jobs, so to speak. When did you first envision any kind of technical career, and sort of what was your early schooling like?

Jaggar: Well, I remember my mom always telling me—we had an old black and white TV set. And underneath it there was a grill that you could—if you lay on your back on the carpet and look up, you could see the inner workings of this TV set. And my mom used to yell at me because she was sure I was going to electrocute myself because I’d be sitting there looking at it, trying to work out how it all worked. And so, this is my earliest memory of electronics is looking at the workings of this TV set for a long time. And, like a lot of kids, I was batteries and bulbs and buzzers and things like that. But about that time, my second oldest sister, she got a job at a university in a laboratory as a lab assistant. And she ended up marrying a university lecturer. And now, we were so poor that the possibility of me going to university was just not even on the horizon. It was just not even considered. But he sort of opened up the possibility that there was this thing after high school that was a university. And you could do more stuff. So, somehow that became possible for me when it wasn’t for my other brothers and sisters.

Fairbairn: Now, what’s the schooling situation in New Zealand? I mean is it expensive to go to college? Or is that public education?

Jaggar: Back then, it was almost free. But they didn’t pay you, you didn’t have to pay them, either. So, that was possible, but, of course, you had to have some way of supporting yourself while you did that. So, yes about the first thing I remember, though, were the computers. I was allowed to go on a tour of a large

insurance company. And I remember seeing their big IBM system, thinking this will be quite good. I guess I was about ten at the time. I was starting to realize that computers were going to be a big thing. So, what are we talking? 1977. So, they were still pretty small, even in a world stage. And I remember thinking this could be me and my father very much encouraging me away from any other vocation, and pointing me towards this being a computer engineer. So, that was—

Fairbairn: So, you went through high school—

Jaggar: Went through high school, and age sixteen—well, age fifteen, I guess you’d call it tenth grade or so, my mathematics teacher told me that I’d never be an engineer. And I really have to thank him for that because I actually—I have—my degrees are in computer science. And I actually, because of what he said, I didn’t do engineering. And I went on another journey. And it was probably a better journey with hindsight. So, it was just at the time where the whole computer problem was becoming a software problem and not really a hardware problem. So, that was a very curious thing. But right at the start of when I’d just sixteen, right at the start of my fourth year of high school, I first touched a computer. And I still remember the feeling, just thinking wow, this is cool. I can program this thing. And I was programming the thing within days, well.

Fairbairn: While you were still in high school?

Jaggar: Yes. Yes, and that was an Apple II. So, and then learning 6502 assembly, that came pretty quickly. And then it was just how quickly I could sponge it up after that.

Fairbairn: So, when you started university that was the path you were on?

Jaggar: Yes. Shortly after that, that school replaced their Apple computers with BBC computers, which was curious because BBC were made by the English company Acorn. And Apple and Acorn were the founding partners in ARM, along with the major shareholders. So, it was curious that they were my first two computers, as well.

Fairbairn: Right, both 6502 based.

Jaggar: Both 6502 based, very similar computers at twenty thousand feet. And I then got a job working in lots of schools around Christchurch, maintaining their computers and just setting them up and just doing all the things the teachers couldn’t do. And so, that put me through university as well as a supermarket job for a while, stocking shelves in a supermarket. Just like I say, we were poor. I had to get any income I could.

Fairbairn: Did whatever you could. So, is it a four or five year university degree?

Jaggar: Mine was actually six, so three years for the Bachelor’s, and I tried to only—I more than majored, I sort of only did computer science. I had to do a—get special permission to sit all the papers. And I did. So, I just did computer science, a bunch of physics, and bunch of maths. I think there was some statistics in there, too. But—

Fairbairn: You didn’t have to do the humanities and other—?

Jaggar: No. My Japanese is dreadful. High school Japanese gets me into trouble. No, it was just computer science, computer science after that. And then about that time, Acorn announced computers based around that ARM chip. And I started to glean information. And remember, I’m on the other side of the world. This is way before Internet, email. So, any information I could get about this chip, I could—

Fairbairn: So, did you have a BBC computer with the ARM chip in it?

Jaggar: I had a BBC computer with a 6502 in it, which I remember buying with my own money having saved for like years. And yes, but all that, I was starting to outgrow those things. I mean I could take that computer apart and put it back together with my eyes shut, literally. So, it was where next?

Fairbairn: So, had you traveled at all during any—?

Jaggar: No, no.

Fairbairn: For the university, or—?

Jaggar: We had been to—I went to Australia once before that, before my job at ARM at the end of my degrees. So, no. We traveled a lot around New Zealand because it’s a pretty country. It’s one of those places you really need to go to in your life. It packs a lot of—it’s got dessert. It’s got rainforest. It’s got plains. It’s got deep rivers, big oceans, big mountains, skiing. So, you can spend a lot of time seeing nature in New Zealand without having to travel too far, but no history. I mean the whole place was found a hundred and sixty years ago. So, it’s all pretty—yes.

Fairbairn: So, was there any particular individuals, professors in the university, that—and was the university sort of up with the latest stuff? Or were they—?

Jaggar: That’s two interesting things. It comes back to that New Zealand thing. New Zealand’s a great place for a university professor to have a sabbatical. And we had a lot of people coming out and visiting and giving us talks. But John Mashey came out from MIPS. And we weren’t very hardware microprocessor orientated. But he gave us a talk. And I just sat there god smacked. I knew then where I was going as far as job. I needed to be in designing microprocessors because he was just speaking my language. He was just—I just needed to talk to that guy for the rest of my life and just learn everything I could.

Fairbairn: So, did he just come for one talk? Or did he actually spend some time at the university?

Jaggar: I think he just came for one talk, one talk. I remember being disappointed that no one else was anywhere near as interested in him as I was. So, he came into this big stack of—honestly, he walked in with maybe eight inches of Mylar film, overhead projector slides. And it was just packing them out and sort of talking about random things. And it was all so coherent. And yes, I tell you. He was absolutely a stand out. But we had a lot of lecturers from a lot of different disciplines. And so, and that’s the other thing about New Zealand is it’s quite—although the university was small, because of that smallness, it was quite diverse. So, you could talk to a lecturer one minute about text compression, and that’s become interesting in a minute, why that was very useful to me, or databases another minute, or networking another minute. And so, you’re sitting in the coffee room with maybe only twenty lecturers tops. And you’re talking about the whole field of computer science as it was understood at the time. But then also, the second part of your question’s also critical, too, I think. All the computers in New Zealand were old compared to Silicon Valley, for example. And I think a few things that started to happen was that the computers I were working on were mainframes when it was possible to make a mainframe that would fit on a single VLSI board. And so, when it became possible to stack a whole microprocessor on a board and a system on a chip , that was stuff that I understood that if I hadn’t have been brought up in New Zealand, I think I would have not been exposed to those old sort of computers and understood that. And another big one was we had a—its name escapes me. Data General—

Fairbairn: Nova.

Jaggar: Nova or an Eclipse, I forget which one.

Fairbairn: They had both.

Jaggar: The one that had a writable control store. And once again, this was a computer that could run a couple of instruction sets at once. And, once again, that became very relevant later in my career. And we also had a Prime computer, Prime 750. It had lots of different processing modes, lots of different instruction sets. Once again, programming that thing was directly responsible for the way the ARM architecture ended up the way it did. But if I had have been in Cambridge, or Stanford, or anywhere else,

really, I would have been exposed to much more recent hardware, I think, that didn’t have those facilities. And it became the ARM thumb combination directly came from me using those old computers at university.

Fairbairn: Interesting. So, were there—was there an expert in computer architecture at the university?

Jaggar: No, there really wasn’t. And it was a funny thing because back then, the computer science department and the engineering department really didn’t talk much. And so, there was probably someone in the engineering department, but it was on the other side of the campus. I used their library a lot. But I didn’t go to any engineering lectures. And there was the guy that marked my thesis, I think his name was Bob Doran, I think. He’d been based here. And so, he was probably New Zealand’s expert on microprocessors. And so, but I’d never met him. And once again, as soon as John Mashey walked in the room, I thought there was a whole other world out there that I really started to understand. But I was very fortunate that I hadn’t been trained in engineering because it wouldn’t have been microprocessor engineering in New Zealand.

Fairbairn: So, what was the name of the university? We never—

Jaggar: University of Canterbury.

Fairbairn: And that’s in Christchurch?

Jaggar: That’s in Christchurch, yes right in—just on the outskirts of town, yes.

Fairbairn: And it’s mainly a technical university?

Jaggar: No, it’s very broad.

Fairbairn: Broad.

Jaggar: Fine arts, some law, lot of science. Yes, but arts, as well. Yes, pretty broad university. Not particularly specializing in anything. Maybe ten thousand students, a reasonable size, too, I guess.

Fairbairn: So, you clearly had the bug. You were going to do microprocessors. It was clear you were going to do microprocessors. You understood the whole VLSI world, and what was going on there, and so forth.

Jaggar: And started to see that in parallel to the guys that were designing the ARM, I was thinking I love the 6502. I love the way it works. And I started to look at some of these bigger microprocessors and thinking man, these things are big clunking giants. This just all feels wrong. And so, for my master’s thesis, I started to think of—it started out as a comparison between the ARM, the MIPS, and SPARC.

Fairbairn: Interesting.

Jaggar: Yes, and so I learned a lot about those processors. And what I was intending to do was deeply analyze all three of them and compare them in the same way. So, I started writing a software simulator. And I chose the ARM because it was the hardest. I thought if I could do ARM, the other two will be easier. It was the most complex of the three. So, I wrote that. And that happened fairly quickly. And by writing that, I’d effectively implemented the ARM architecture. So, I understood it at a very deep level. Then I needed a compiler. So, I ported something called the Amsterdam compiler kit, one of the first ARM compilers. And the thesis sort of took a bit of a turn at that point because I became so interested in the ARM architecture. Because I was comparing their compiler to my compiler, I could see what codes the compiler had produced. I could see what mine would produce. And so, I then became sort of a compiler expert for a while. And to try and make my compiler as good as theirs, I actually ended up surpassing it. And the funny thing is, the university had source code for SunOS. And so, I had SunOS running on top of an ARM in 1989. So, yes little known fact. I could boot SunOS on an ARM processor. So, yes all the kernel porter. And then I started doing a detailed analysis of what was wrong with the ARM architecture. And there were a few things wrong at that point because it had only been through one iteration. So, there were some—or, two actually by then. And so, it was analysis of what we could do to make the ARM better. I guess even then it was a kind of a royal we.

Fairbairn: So, you finished up your degree, you wrote your thesis you were describing. What happened? What’s the—?

Jaggar: Sent my thesis to MIPS. Sent my thesis to ARM, well actually to Acorn because ARM didn’t exist as a company then. And sent my resume to Acorn who were still selling computers in New Zealand. And much to my surprise, I was flown up to Auckland, which is a city on another island in New Zealand and interviewed by Acorn, and all good, passed all the interviews, was going to be a technician for Acorn and support all their new ARM based computers. And all of a sudden, there was a hiring freeze at Acorn. And I didn’t have a job. And then about the same time, an email arrived from Steve Furber saying received your thesis. Thanks very much. And I had two or three emails back beforehand. I think I’d been cheeky enough to say are there any jobs going at Acorn. And he had moved to Manchester by then, and said do you want to come and do post-grad work at Manchester. And I needed to get a job by now. And I was just flat broke.

Fairbairn: Tapped out.

Jaggar: Yes. And probably over university by then anyway, probably time to move on. And about the same time, Apple approached Acorn and spun out ARM. So, the time was just perfect. I must have—I finished my thesis around February. And I was awarded first class honors, which you don’t normally get honors on your master’s. But they gave me first class honors on the master’s because they liked the research so much. And so, I went on from there. And a little bit of touring with ARM because they were getting set up. And they didn’t know how much money they had. And this was a real start up. Do they have a place for me? And they really wanted me to come out and port my instruction set simulator. So, they gave me about a three or four month contract became they’d never met me, telephone interviews. And I was lucky enough to be there about two months before they said we’ll put you on the full time payroll.

Fairbairn: So, you moved to Cambridge—?

Jaggar: Moved to Cambridge. So, second time out of the country was all the way around the world. I bought a great big wooden packing crate and just stuck everything in it and put a big stamp on it, and posted it over.

Fairbairn: How was that? What was the—?

Jaggar: Well, it was huge. I mean looking back, I guess it took a lot of courage. But at the time, it was just no one told me I couldn’t, so off I went.

Fairbairn: What did your family think?

Jaggar: You have to understand, here I’m going to—let me take you back, I’m from a very poor part of the world. And here I am, I’m about to be standing at the foot of the gods at Cambridge University, goodness.

Fairbairn: That’s why I was wondering what did your parents and family think of this?

Jaggar: Obviously, they didn’t see that. But my mum was—she really didn’t understand the whole thing. She—when I came home and told her that I had first class honors, I remember she was—she really didn’t understand. I was over the moon. But because she had no concept of university, it was are you going to get a job now, I think was pretty much what she was thinking. And—

Fairbairn: How much money do you make?

Jaggar: Exactly, how much money do you make? And what did I make? I think I started on fifteen thousand pounds, senior programmer. I remember the job title was senior programmer. I thought oh god, I’m senior already.

Fairbairn: So, who was it that hired you? Was it Robin—?

Jaggar: It was Lee Smith.

Fairbairn: Pardon?

Jaggar: Lee Smith, he ran all of—he was Mr. Software at ARM. And he had to get approval from Robin. That happened. And so, I arrived over with my bicycle and a suitcase. And—

Fairbairn: Found a flat or apartment nearby, and—

Jaggar: Begged, borrowed, and stole for a couple of weeks until I work out what a bank account was. And everything’s different when you change country. So, get over the jet lag. And I think I arrived ten days before my start date. And I just worked through those ten days. So, I’m pretty sure ARM never paid me for them. Maybe I should send them a bill. The—so, no just got set up. And then shortly after that, Robin Saxby, the CEO, said he was looking for an apartment. And I said well I am, too. He said you find me one with an extra room in it, and when I’m up in Cambridge—he lived two houses to the south, when I’m up in Cambridge, I’ll stay in your apartment and give you a little bit of money every week. And I said that’s great. I can get a better place and have the place mostly to myself anyway. So, that was fantastic. And we actually had another guy, too, a polygon pusher we would call him - a VSLI layout guy, He came and he lived with us for a while. And so, funny enough, Robin and I worked out we were born on the same day twenty years apart. And so, this has always been a joke between him and I because my father was long since dead. And Robin, honestly, sort of took over a lot of fatherhood role for me. And it was an interesting time at ARM because he was the head. And then my boss was underneath him. So, it was that relationship you get with a grandfather and grandson, and the parents are in the middle sometimes. Sometimes it was a slightly difficult relationship for my boss because I was talking to his boss.

Fairbairn: Because you had this relationship—

Jaggar: Over dinner .

Fairbairn: Over beer, or whatever.

Jaggar: Yes, exactly. So, yes, Robin was absolutely abusing this. I didn’t know this at the time, but he was finding out what was going out on the shop floor just by asking me questions. And then my girlfriend from New Zealand, she joined me—Carmel joined me in—about six months later when everything was sort of stable. So, she came out. She actually ended up working for Digital Equipment Corporation just as—just secretarial. But that was funny. Digital was up in New Market. And so, she came out. And strength to strength really, my boss—so I finished my software simulator. That was all fairly quick. And they then wanted me to do an implementation of the next one. So, I started working on that with a guy called Al Thomas who was working on it.

Fairbairn: Right, implementation of what?

Jaggar: The next—so it was going to be the ARM seven. So, just starting on the ARM seven with that one. So, I was going to implement that in software so we could check everything before he did it. And about six months into that project, Al died. And this was huge tragedy for ARM. We—I was employee number seventeen. So, there were thirteen, I think, that came from—I think Robin might—thirteen came from Acorn. Then there was Robin, our secretary, and two engineers. And so, we were still tiny. And yes, Al died. And he took all the ARM knowledge with him because the original two architects of the ARM processors, Sophie Wilson had stayed at Acorn, and Steve Furber had gone to be the professor of computing at Manchester. And so, really no one at ARM understood the ARM any better than this guy from New Zealand who had only been there twelve months. Everyone else was kind of busy on other projects.

Fairbairn: Who had no formal education in computer architecture.

Jaggar: Exactly, who was a computer scientist who’d written a couple of compilers and programmed a 6502 once. But I knew that architecture pretty well. So, and I think with hindsight, that was probably a really good thing for ARM because then, I think I was just smart enough not to know that I didn’t—to know that I didn’t know much. And so, I had to ask a lot of questions. And that became—my career then took—was talking to a lot of customers. And whereas the processor had grown out of the workstation, a lot like MIPS and SPARC, Robin was saying very much, this isn’t a workstation CPU. We’re going to be embedded. We’re going to be first into the embedded market. And he had this partnership business model where anyone that wanted to buy this core, could and manufacture silicon. And despite a lot of pressure through the years, he never buckled once about ARM making silicon. Many, many times did various people consider it, including me. And he never said—he said no, we’ll never compete with our partners. That’s their job. We design, they build. And it was absolutely right with hindsight.

Fairbairn: So, that was actually one of the things we had talked to Robin about was this tension between Acorn, which was still a partner that wanted it as a work station computer, and Robin’s goal for the company that was to be an embedded product, which was a novel concept at the time, I think. What—you were the guy, it sounds like, in the middle. You were the guy implementing the architecture, making trade

offs. And so, tell me how much you felt that conflict and what—did Robin give you a clear message about what you were doing, and that sort of thing?

Jaggar: Hugely, hugely conflicted. So, half the board is saying you must build this chip that we need, that we will sell maybe only a hundred thousand of, tops. So, even though I didn’t know much about business, I could do that math and go this company can’t survive on that alone. So, you think to yourself, how can I satisfy—and I’ve got Apple, which wants something similar. It wasn’t embedded, but they were interested in ARM for the power consumption. But it was still basically a system CPU and not anything deeply embedded, stand alone CPU on the Apple Newton product. And so, they were pulling us in a direction, as well. And maybe they had some better volume, so that helped a little bit. But we could play those guys off against each other a little bit. And Larry Tesler was on our board. Larry was—he’s a bright cookie. Let’s be blunt. And he, I think, could put the interest of ARM before the interest of Apple or Acorn. And so, he was a huge ally to Robin and therefore, a huge ally to me. And so, but as time went on, we delivered CPUs, but we came up with this concept of an embedded core, putting a cache around it, and calling that our CPU. And then going one beyond that and putting peripherals around it and making something embedded. So, we started heading down that track. The problem for ARM was we needed to start doing things that were absolutely useless for Acorn. One of the biggest ones was a bit of debug. When the chip was embedded inside, it’s our credit card. How do you debug it? And that was, again, where my expertise really started coming in. Because of the software background, I was used to proper source level software toolkit. And how do I get that working on top of a chip that was embedded. Acorn really didn’t need us working on that because they plugged the CPU in and programmed on the computer. The other thing was that we were starting to hear, and this was—took a long while for this to get into my thick head is that ARM had a code density problem. In fact, all risk chips had a code density problem. ARM’s was bad, but not dreadful. And I’m still not sure to this day that the guys at MIPS or SPARC or even PowerPC know that the reason we stole their lunch is because we could replace our 8-bit chip and use less memory. So, we could take an 8-bit or 16-bit, and once we had the thumb instruction set, we had better code density in every one. Suddenly, they could take their existing product, free up memory. And you know what products they always—you’re always running out of memory.

Fairbairn: Right, absolutely.

Jaggar: So, if you could free up memory, suddenly, for no cost, they could put more features in. And so, it was a huge win. And it was three or four companies told us that. Nokia [ph?] were the loudest. When we first went and saw Nokia, they said your code density’s too big. It was pretty much that blunt. Your code size is too big. We compile. And it just—it’s just too big. These 32-bit things are everywhere, aren’t good enough.

Fairbairn: So, that became a major direction? I mean what are the steps that you took in order to combat that? Can you describe that?

Jaggar: Yes, well once again, I mentioned earlier that one of the lecturers at university was very into code compression, Tim Bell. He’d worked at Calgary with a couple of Cleary and Witten, the two big experts in text compression at the time. And he taught me a lot about compression. And so, when I first started thinking about it, I thought I’m just going to compress this somehow. And I’m going to have some sort of decompressor that’s going to turn it back into the ARM instruction set. So, I don’t have to play with the whole ARM processor. That keeps Acorn and Apple happy because they’ve still got their beloved ARM instruction set. And I will have this new compressed thing. Trouble is, as time went on, I started to learn more about hardware, and doing text compression in hardware is just hard. So, the direction became let’s give this thing almost writable control store from the—I can’t remember if it was the Eclipse or the—one of those DG machines and the Prime, that executed two instruction sets, not at the same time, but you could unload the one instruction set, load in a new one, and off you go again by rewriting the whole control store. And so, those two things together, I think put me in a unique place to go let’s make a chip that can have a 32-bit instruction set, the original ARM instruction set, and the 16-bit instruction set. When you want to go fast, use up a little bit of memory, maybe on chip wide memory to use the ARM instruction set , but ninety-nine percent of the time, execute the thumb instruction set and save an enormous amount of code, probably thirty percent of your memory. So, yes. And off 16-bit memory, off chip memory, the thumb instruction set was faster than ARM because it got an instruction every cycle.

Fairbairn: So, you actually had two instruction sets?

Jaggar: We had two instruction sets. And we could switch between them on the fly, whenever the processor took an interrupt; it jumped back into ARM mode because we needed that to be running fast. But any other time, you could branch and switch instruction set on the fly. And that was careful design, too because we—I didn’t want a pipeline on the front. By then, I sort of understood processors pretty well. And I didn’t want a pipeline down the front of the machine that had 16-bit and 32-bit instructions in it at the same time. So, by doing it in a branch, the pipeline was being flushed anyway. So, I only had one kind of instruction down the pipe. And it turned out this was a fairly simple bit of hardware on the front. The thumb instruction set is a subset of the ARM instruction set. So, you can decode it back to ARM quickly as it flows down the pipe.

Fairbairn: So, at the compile time, you just switch say which one you want to compile to?

Jaggar: Exactly. Different modules can be compiled and different sub routines can be compiled in different ways. The compiler just put the right branch instruction in to switch between them. So, you could say to a customer you got to start profiling your code. Which bits do you need to go fast? If you don’t know on day one, compile it all on ARM, and then slowing it down a little bit, or it might even be speeding up if you’re running at a 16-bit memory, but start moving stuff over to thumb. And, in the end, ARM was running—an ARM processor was running thumb instructions probably ninety-nine percent of its time.

They would jump back into ARM for high speed interrupts, but that’s about it because it was fast enough. It was much faster than anything else.

Fairbairn: So, you’re beyond getting this initial software worked on, your first job was to—basically the chief architect for the ARM seven, is that—?

Jaggar: Yes, exactly.

Fairbairn: So, that was the third generation of the ARM processor by then?

Jaggar: It was, yes, yes. Five and six—sorry, four and five were forgotten because 486 and 68050, and 586, and they were all out there. Once again, Larry Tesler, I think, suggested let’s skip a couple of generations and just to get the jump on everyone. If everyone else is up to version six, we will be, too. And it’s funny ARM’s campus back in Cambridge now, there’s no—the buildings are numbered. So, there’s ARM one, two, and three. There is no ARM four and five. The next building in the sequence is ARM six. I only found that out recently myself. So, you got—but to come back to your question about pressure. We were under enormous pressure from Acorn not to do this because, of course, putting this instruction set on that was no use to them was a complete diversion away from—and the same with—

Fairbairn: Pulled in a lot of resources to—

Jaggar: Yes, and we didn’t have many resources. I mean I was it. And once we put embedded debug in, as well, we put a faster multiplier in. And no, this was all useless for them. And they fought hard, but Apple stuck up. And, as I recall, we brought Steve Furber in from Manchester University as sort of an arbitrator. And Steve reviewed the paper, reviewed my thoughts and the answers and basically came down. And I’m pretty sure the summary was it’s ugly, but it’s a good idea. So, that was kind of it. Off we went down that track. And we built a software simulator. I did a software simulator straight away. We hand coded some stuff to it while the compiler was being done. I think it went—I remember the instruction set went through thirteen iterations. And I was thinking geez, we need another iteration. We can’t be having version 13.

Fairbairn: You can’t be shipping 13.

Jaggar: Yes, and there was another iteration after it when the compiler work was done. But yes, hand coded and we started to see the numbers that we wanted to see. We had this 30 percent in mind and we were hitting it. We took it back to Nokia and they were very blunt. In Finland, you’re maybe 200 kilometers off the Arctic Circle in a place called Oulu. And they were very blunt with us and they said if you can make this we’ll buy it. So we felt like was the most…

Fairbairn: So you didn’t have to get a 2x or anything like that, just down 30 percent.

Jaggar: No. Yes, we needed to get 30 percent. That took us about—we were about 15 percent too big. And, of course, the trouble with memory is they come in pairs of two. So if you’re going to jump out of 256 kilobytes, the next thing is 512 kilobytes and you’ve just blown your budget. So we were about 15 percent too big. If we could come in 30 percent overall smaller it would be 15 percent smaller than their old chips and suddenly they had free memory. So they were really happy with that. Yes, no, we didn’t have to do 2x. The sum instructions said because it’s simpler you need more instructions but because they’re half the size, that doesn’t add up to be. That’s a net saving overall. But yes that was really the starter. We want and saw I think it’s still confidential. I don't think these companies have ever come out that they’re using ARM. There’s an some engine manufacturing in Japan and one of the first automotive applications. And they were thrilled with that code density as well, so that made a big difference to them.

Fairbairn: So was Nokia, you know, one of the first sort of big design wins? And they were sort of a bellwether customer that kind of…

Jaggar: Yes. And they were also the one that brought TI on board as a licensee. And TI they were the big brother for ARM for a little while. They were a real champion. They had some big volume. They were the one that could really secure someone with the volumes that Nokia needed. They needed a big supplier. They needed a guaranteed supply. And in the end, I think, even TI they were dual sourced out to another big company. I don’t remember who the other one was.

Fairbairn: So these were some of the features you had in the ARM 7, what were the major driving forces in the 7? Was it reduced code density and what about performance and other more normal…

Jaggar: We went down to three volt. So ARM 6 still had transistor stacks that were not full complementary. So we went down to full complementary throughout the whole chip which put the silicon budget up, again, which the original designers of ARM weren’t too happy about because we were using silicon that wasn’t completely necessary for the applications but we needed to be down at three volts. So it pushed up the silicon budget quite a bit more. We put in a bit of debug. We put in a four times faster multiplier because we started to pick up some DSP applications. And then, of course, the thumb instruction set and that all wrapped up into the ARM 7 TDMI, T for thumb, D for debug, M for multiplier and the I was some I-stuff the breakpoint registers that you could took off when it went into production. And that was ARM’s real hit for a chip. That thing was a great bit of processor. It’s many billions of that sold now.

Fairbairn: Is the chip that got you heavily into DSP applications because of the multiplier.

Jaggar: Well, no, we started doing—we couldn’t quite run MP3D code on that chip at 20 megahertz, I think. As I remember when we pushed it to 33 megahertz with a few more tweaks then we could start running MP3 on it. But we also had to be careful, here’s another political thing with ARM is we couldn’t be too DSP because TI were our partner. So as soon as we wanted to do DSP, TI would go no, no, we don’t want any of that. We do DSP. You do microprocessors. And they were a major partner.

Fairbairn: They didn’t have a board position did they?

Jaggar: They didn’t have a board position, no. But they were pretty influential.

Fairbairn: They were a driver.

Jaggar: Yes, they were a big driver. But in parallel to all of this we were doing the ARM 8 series which was really just to keep Acorn happy. I think the original charter of ARM had to deliver two generations. So we delivered the 16 and 17 were pretty much the same. I think we counted that as one generation. And then there’s 18 was the next generation and that was very tailored towards Acorn’s situation. And it was a difficult design because of that. They needed requirements in the backwards compatibility that no one else needed. Namely, they had code that wrote to itself. They had a self-modifying code. And so we couldn’t have an instruction case that wasn’t kept coherent by the data side. So we ended up with every other CPU was a Harvard architecture. ARM 8 wasn’t. It was a full Von Neumann shared ID cache. And it’s really hard to get performance when you’ve had your data bandwidth cut in half.

Fairbairn: So you described these various early influences in terms of sort of what stood in your long path, what about as you were doing this, were there sort of major aha’s, revelations, aha’s, whatever, from outside? I mean were there any other major outside influences in terms of your…

Jaggar: Well, Nick’s big one was Digital. So just as we were about half-way through the design of our ARM 8 Digital came along and it was the team that had just finished doing Alpha. These are pretty bright guys. And they had an ARM just about implemented but we had never written down exactly what the architecture was. So they weren’t completely sure what they had implemented because there were so many details that were just undocumented. So they were in Austin, Texas. And I moved to Austin, Texas for eight weeks and I wrote the architecture manual. So that’s the first time the architecture had actually been properly defined. And it was a mess up until that point, to be honest, because what it was was whatever was executed on the current chip. And so I started basically taking stuff out and saying no, we’re not going to support that in the future. So if you do this wacky instruction we’re not going to do that. And gave them an architecture manual at the end of that eight weeks. But at the same time, once again…

Fairbairn: Yes, the Alpha was a RISC processor.

Jaggar: Yes, a very fast one. It was at 200 megahertz when everyone else was at about 50 or 60. And so it was a real wow chip in about 1991.

Fairbairn: And it was largely ARM compatible?

Jaggar: No, no. They did the Alpha. But then this design team they did two or three more generations. So then they said, well, let’s so something new. And so they decided to implement an ARM and so they called it StrongARM. So the StrongARM team led by Rich Wittick down in Digital, down in Austin. He had implemented this thing and it was a beautiful clean design and I looked at it and I thought why are we fluffing around with Acorn and this ARM 8 thing? I knew it was just we need to be designing like this. It’s just so simple. It was a very clean architecture. I distinctly remember flying back to Cambridge and writing the ARM 9 business plan on the plane. I called it ARM 8E at the time, E for embedded for completely political reasons because it was just going to be a version of the ARM 8 and therefore, under the radar as far as the board were concerned.

Fairbairn: Right.

Jaggar: But it rapidly became ARM 9. I think everyone saw through the guise that suddenly it was a clean five stage pipeline full Harvard architecture. And we were going to migrate people—you know, ARM was still very young as far as the amount of code that we had to migrate. We didn’t have a great deal of hand written assembly to migrate. So we thought a customer was going to be a full Harvard we didn’t need the ARM 8. And ARM 8 basically subsequently died because of that as an ARM processor. We delivered some to Acorn. But the StrongARM processor was so much faster in clock rate that they actually threw away their own backwards compatibility needs and ported it to the Harvard architecture. And so that was really the death nail of ARM 8.

Fairbairn: So what was DEC’s goal in doing that? Who were they going to sell that too?

Jaggar: I don’t know. It was that stage of computing that there was a lot of money around and these were good guys. I think Digital had a big fab in Hudson, Mass. that needed filling. I think it was a 0.35 Micron fab or something like and it was pretty empty producing Alphas. And so they sort of said well let’s get into something high volume to fill that fab up.

Fairbairn: Did they buy an ARM license and everything?

Jaggar: Yes. Well, they invented the architecture license for them. We’ll give you the architecture and you can build it. Between working at Digital, those guys had all worked at the Somerset facility that was the combined IBM-Motorola-Apple facility in Austin. So that’s why they’d all end up in Austin. But I think

the politics of Somerset had sort of driven them all bananas. So they just wanted to go and do something new, a new processor. So that was where StrongARM came from. And it was a relationship that for me I was used to a bit of climate in England. And so it sort of changed my career because I fell in love with Austin, great place to live. And so yes, I was sick of not seeing the sun.

Fairbairn: So you went back and then you started back on the ARM 9?

Jaggar: I went back and started working on the ARM 9. And then we started working on a DSP because by then we thought we were big enough to do a DSP, but we couldn’t do a full DSP still because of TI and a few others. So we started on something called Piccolo which was—my names are always slightly puny, very corny but puny. Thumb was the useful bit on the end of your arm, so that’s where the thumb thing came from. Piccolo was pitched at the high end for audio. So we didn’t want to take on full DSP based band of cell phones. We just wanted to do high end audio. So Piccolo was high end audio, high pitched audio. So we went down the Piccolo architecture for a while and it was a disaster mainly because the software wasn’t in place and I left ARM Cambridge and started the Austin Design Center. And so the Piccolo went nowhere. We sold a few licenses for it but I don’t think it ever got programmed up to do anything real in the real world, which is a shame because it was a cool architecture. It needed a software owner, though, and ARM was very much a hardware company then. It had not enough software on the expertise which is something I had started to change. We started to hire more software people doing applications, engineers and sample code and things like that.

Fairbairn: So what go the whole Austin thing going? How did that happen?

Jaggar: We employed—ARM employed everybody that could spell microprocessor in England let alone design one. We just could not get any more employees. But we liked Cambridge and England because the salaries were much lower. We could just be more cost effective. But we just got to the point that if we were going to do the next chip and we had been watching 2 guys design ARM 2, 3 guys designed ARM 3, 4 or 5 guys on the ARM 6, ARM 16. But the time we were up to ARM 8 we had a design team of about 20, ARM 9 about the same thing, about 20 on StrongARM. So we knew we probably needed, you know, rough guess 30, 40, 50 engineers for the ARM 10. And so we just weren’t able to hire those people in Cambridge. And so I put my hand up and said I’ve got to go and get some sunshine because it’s just raining and cold here. I’ll start the Austin Design Center. What we’ll do is we’ll partner with Digital and we’ll design StrongARM 2 and ARM 10 will be the same chip. So we’ll keep the same pipeline. They’ll take into some high performance applications. We’ll do the embedded stuff and license that. So we’ll halve our costs. So the board looked at that and thought well this seems like a very reasonable thing to do. So I took my credit card in the middle of 1996 and went to Austin and started buying second hand PC’s from the Dell factory outlet. Set up a design center. About two months after I got there Digital sued Intel over patents. Intel took one look at Digital and bought them outright just to settle the whole thing. So Intel bought Digital, that part of it, that fab.

Fairbairn: Yes, that Alpha team or whatever.

Jaggar: The Alpha team and everything. I thought that’s okay. What we’ll do is we’ll work with Intel, partner with them. We can do that. They’re good designers. But the Digital guys didn’t want to work for the big company anymore. They had that Somerset experience with Motorola and IBM. And so I worked hard to hire those guys but no they wanted to go and do a startup and so they went and did a startup called Alchemy. And so suddenly I had committed customers, no design time, no design. I had a building. That was a start. And then a couple of months after that I think Synopsys bought the design tool company we were using Compass and put them out of business effectively. So I had no design tools, no engineers, and a customer. So that was a scramble. Those years in Austin were a complete scramble to put a design team together, a completely new design flow from scratch and deliver a chip. And we got there about three months late which was about as late as they were anyway, so we were pretty much on time. We delivered the ARM 10 family.

Fairbairn: So the ARM 10 was really a souped-up StrongARM.

Jaggar: Well, the StrongARM design, those designers that were finishing off some other StrongARM embedded designs and never actually started on StrongARM 2. So ARM 10 was actually all me. I was starting waiting for those Digital guys to finish what they were doing instead. I had got into Austin a wee bit earlier as it happened and they never did. They sort of never really put much influence into it. And I was keen to do a lot more floating point because we were starting to see floating point to become important. So I took a lot of the design that was Piccolo and made it into the victor floating point unit. So I started all of that floating point, stuff.

Fairbairn: So you managed to hire people from everywhere…

Jaggar: Organically, we’re approaching people left right and center and starting to get a bit of a presence and exciting company. People want to do something different. So we hired some really bright cookies. So anyway we put a design team together probably 50 to 60 people in the end over those two or three years.

Fairbairn: So what year was it you moved to Austin?

Jaggar: Ninety-six I moved out there. Ninety-seven is when we really started hiring. Yes, we had a few sales people but no engineering—I think the first engineer joined about the middle of ’97.

Fairbairn: And when was the ARM 10 design complete?

Jaggar: End of ’99. So two years. It was supposed to be two years and three months, but two years and six months in the end. And that led on to ARM 11 too. ARM 10 didn’t sell hugely. We made a mistake not chasing enough clock rate. Back then Intel were pushing clock rate with AMD and clock rate, clock rate, clock rate. And we were trying to wave our arms and say look clock rate doesn’t matter down here guys. We’re running, we’re booting operating systems twice as fast as ARM 9 because we’ve got a cleverer architecture that’s not burning as much electricity. Yes, that fell on deaf ears a little bit and we need to push clock rate a bit harder, so that was what ARM 11 was about, basically, the same pipeline as ARM 10 but pushing the clock rate harder.

Fairbairn: So from my conversations with Sophie and Steve, it seemed like the low power element of the ARM architecture was sort of a necessary accident in terms of well they couldn’t design anything very complex and they were worried about heat so we’ll just keep it as low as possible whatever that is.

Jaggar: Yes.

Fairbairn: And it ended up being very low.

Jaggar: Right.

Fairbairn: So did that remain—did the low power aspect remain a top priority as you started through the ARM 7 and so forth?

Jaggar: If you had been present the halls of ARMs, we didn’t have halls we were so small, the rooms of ARM, we didn’t know why we were low power. We did not know why we were. We didn’t really know what we were doing. We were low power but we didn’t really know how we had got there. And many years later I was talking to an engineer out of IBM that had come to work, Eric Shawn had come to work at ARM and he just said your transistors are just so much smaller. He had transistors in some of the big IPAP EC chips that were about the same size as our entire header because we were just so much smaller. And it turned out to be with hindsight it was just everything was smaller. And the wires were smaller, the geographical distances were smaller, everything was smaller. It was completely by accident, though, Steven is absolutely right. But we were sort of worried about this because there were a few emails backwards and forwards, does anybody know why we we’re low power? No, we don’t really know why we’re low power. And you try to explain it and we’d say we’re simple, but we’d never really seen the high power chip up to know what they were doing wrong if that made any sense.

Fairbairn: So you didn’t analyze the differences in what led to your…

Jaggar: No, no.

Fairbairn: So it was the overall simplicity of the chip and you didn’t have drive big busses or something.

Jaggar: Exactly.

Fairbairn: Everything just sort of…

Jaggar: Exactly. The code density helped because you’re not driving the memory so much. All of those things help. And it all sort of compounds up to—And if you can get a 2x or a 3x or a 5x and that’s kind of the stuff we were seeing 2 or 3x better. And it was very hard for the big guys to come down. Once you’re huge it’s hard to shift weight. It was easy for us to go up.

Fairbairn: When did that become a real design criteria? At first it was an accident and not necessary for the original application. When did it become a driving force?

Jaggar: I’m not sure it ever did. I’m really not sure it ever did. I think it was always by evolution rather than revolution.

Fairbairn: It just happened given the other things that were going on.

Jaggar: Yes, just what we designed was small. We were always small. You know, I remember the salaries that I was signing off for ARM 10 and trying to send these bills back to Cambridge going these are big numbers for ARM back then. We were spending a lot of money on these chips and we needed it to be small and simple. So if you look at the Victor floating point you notice it’s a very small, simple chip but it’s fast. And it reuses its silicon well and similar sort of things. We were always small and light weight. Always focused on it without even knowing I think. I’m not saying we’re doing it completely by accident. I think it was just sort of ingrained in us, I think. Yes.

Fairbairn: So yes, I had always thought that was a major—it seemed to be a major reason for the success and I figured this must be a—at some point it must become a major focus of your design effort.

Jaggar: Not really.

Fairbairn: It was the fallout of everything else you were doing.

Jaggar: And then some of the other things, later on, of course, we were worried about leakage because then we were starting to use fab processes that were leaking all over the process. So then we started to—and this is after I left ARM, but now they worry a lot about how to manage leakage.

Fairbairn: Yes, leakage is becoming a huge problem with the small geometries and everything.

Jaggar: Stopping the clock just doesn’t help. We were always—well, we weren’t always static, but for a long time ARM was also static so you could stop the clock. That also helped. And things not sitting there executing. It was amazing how many chips couldn’t have their clocks turned off and not lose time.

Fairbairn: So you did the ARM 10 and that became a much bigger success than the nine, was that the…

Jaggar: Yes. Nine and 10 sort of came out about the same time. Another contribution I made to the team personally was ARM 9 was having a lot of—we started booting operating systems on our models at that point, for ARM 10 because ARM 9 we got silicon back and it didn’t boot operating systems. And there were just these bugs, you know, esoteric bugs that were catching us. And the digital guys had told me that booted on the hit list, that booted a cut down Unix, I guess it was Ultrix. But this amazed me that they were actually able to get that far. And, of course, the problem is have you made any mistakes along the way. I actually went back and rewrote my original software emulator from my thesis and instrumented it with—it produced a big file of everything the processor was doing, what was transferred two and from memory and what was transferred to the instructions that flowed into the machine. And we played that back into the net list to make sure the net list performed exactly the same. And we caught maybe 15 bugs with that. And when the ARM 10 silicon came back it ran everything the first time. And about the same time I think maybe only two or three months earlier ARM 9 had working silicon that was booting the two or three operating systems we were interested in at the time. So that was another problem that we had sort of ARM 9 and ARM 10 right next to each other. And ARM 10 didn’t have the clock rate that everyone wanted. It was twice as fast booting operating systems as 9 but everyone wanted more clock rate. So ARM 10 sold a few licenses but then they very quickly did an ARM 11 which was a rehash and lengthened the pipe a little bit and got the clock rate back up. Which was good because by then, I guess, ARM 10 being like that was a little inevitable. It was a new design team, new software tools. You know, everything was just too new for it to be perfect. But they didn’t want to deal with the A series chips at the Austin Design Center. That’s a pretty good design team there now. Yes, once they cut their teeth on that first one, they all got to know each other a little better.

Fairbairn: So how much longer did you continue?

Jaggar: Well, officially after that…

Fairbairn: You did the 11.

Jaggar: I started down on the 11 architecture and basically said yes gave it my blessing but I was completely burned out at the end of the Austin thing hiring all of those people. I really had gone over there to raise my children, my eldest was born in 1996. So I was going back to Austin to travel less. I remember in 1996 when she was born I crossed the Atlantic 26 times. And I just thought that was crazy. And Heathrow Airport was two hours from Cambridge.

Fairbairn: So your family didn’t move?

Jaggar: I was going back and forwards setting all of this stuff up. And my wife had a newborn baby and I remember working at the sums that I spent a complete working week sitting in the taxi cab between Heathrow Airport and Cambridge in that year. A 42-hour week I think it was sitting in that taxi cab. And so this was crazy. I was just completely burned out then too.

Fairbairn: Exhausted, yes.

Jaggar: We started the design center so I could go back into engineering. I’d been doing a lot of marketing and selling the ARM 7 and 8 and 9. So yes that was what Austin was about. And, of course, a year after I got there, Digital suing Intel and being bought out meant that I was back in the thick of it again and it wasn’t going to be an easy design. We didn’t have a bunch of learned Digital employees to do most of the work for us.

Fairbairn: So you had to start from scratch.

Jaggar: We had to start from scratch and it was a crazy time. Absolutely crazy. Fantastic time. But a crazy time.

Fairbairn: But your family did not move?

Jaggar: They came over at the start of ’97. My second daughter was born in Texas, raised a little in Europe and resided eventually back in New Zealand so she has three passports.

Fairbairn: So being born here she got to be a U.S. citizen.

Jaggar: Exactly. Yes. A handy thing to have. A real handy thing to have. So I’ve got one British daughter, one American daughter and one Kiwi son, a New Zealander. Yes, they’re all over the place. So I went back to New Zealand then and I was going to design a little design center there. So I said let’s start working on…

Fairbairn: So you didn’t ever go back to Cambridge, really.

Jaggar: I never went back to Cambridge.

Fairbairn: No. You just sort of bailed out of Austin.

Jaggar: Yes, time to raise the children back home was really what I was thinking about. And like I said, right at the start, my dad had retired when I was young and I think that must have influenced me because I just wanted to go back and have more time with the kids. ARM was public by then so the stock price was healthy. We had sold up three-quarters of our stock, my wife and I, and put a healthy amount of money in the bank. So we didn’t need to worry too much anymore about what the future held. And so I went back there to start a new design team that was cut off purposely where it could work on something really far out. And maybe bring people down for a sabbatical every so often and work on whatever I was working on for six months, twelve months, something like that and give good people a rest. So that was the plan. About two months after I got there the stock crash happened at the start of 2000. And it didn’t look that smart anymore to put a new design center down there. We sort of battened the hatches a little bit like everyone did. And so I just worked on down there. I started pushing code density harder, pushing performance harder. And about that time ARM was being sued by a couple of companies for patent infringement which in my opinion were bogus. And as the judge finally said were bogus. But I was deeply involved in that as the inventor of a lot of ARM’s patents. So that took up a lot of time. And then at the end of 2002 ARM missed a quarter. The first time ever it missed a sales target. And so they took a bunch of voluntary redundancies and I took a voluntary redundancy. And I thought I’m just going to raise the kids. I really liked the CPU I was working on. I gave that back to ARM and started fresh. I had some fresh ideas. And you’ll be seeing a new CPU soon.

Fairbairn: For ARM?

Jaggar: I hope so. I hope they buy it.

Fairbairn: This is something you’ve done on your own?

Jaggar: Yes, new architecture coming up, completely new like nothing you’ve ever seen before.

Fairbairn: This is something you’ve done on your own.

Jaggar: Yes. New architecture coming up, completely new like nothing you’ve ever seen before. Yes. So that’s what I’ve been working on the last little while.

Fairbairn: Interesting. When will the world see this?

Jaggar: Well, we’re just starting now. We just started talking. I was back in Cambridge to demonstrate it to them two months ago. So it’s still very early days.

Fairbairn: And you’re also talking to other companies about it?

Jaggar: Well, ARM is my only company. I’ve only ever worked for one company, the only company I’ve ever worked for. *<inaudible>* because now they’ll lower the price. I don’t want to sell it to anyone else. And they’re the right people anyway. They’re the people with the channel to market. They’re the people that can license this.

Fairbairn: Were you working with other people down there? Or is this completely your own…

Jaggar: A little bit. But mostly me, just some stuff I’ve seen and some great it’s a little bit like RISC all over again. I can give you a few clues. A lot of people have been working on data flow machines recently, the trip [ph?] stuff out of the University of Cambridge, the University of Texas especially. And there’s been some good data flow machines coming out and it solves a few problems. But it’s a bit bigger than I would have done.

Fairbairn: So has this been sort of a side thing? Or have you put a lot of, you know, significant part of your time into it?

Jaggar: Both. I knock off work every day at three so I can come and pick the kids up from school. It’s not an…

Fairbairn: It’s an everyday project.

Jaggar: It’s an everyday project.

Fairbairn: This isn’t something that you…

Jaggar: No, no, this isn’t—no. No, I’ve been chipping away at it. So it should be good. Yes, it solves a lot of big problems.

Fairbairn: Fascinating.

Jaggar: Yes. It’ll be good. Maybe it will be the next chapter of the Computer Museum.

Fairbairn: So what else do you do when you’re playing with your kids? Do you have any other hobbies or interests?

Jaggar: Yes, I mountain bike a lot. I’m bored with home automation. My home automation is all broken. Someone needs to fix home automation and do it properly for once. I’m actually working for a company called Schneider a little bit to integrate a lot of stuff that I’ve got in my house back into their products. It’s crazy that you can’t—well at least I can but to turn on your air conditioning from your phone when you arrive home so that your house is warm and things like that need to be all easy. New phones have got new field communications and these new NFC chips. You know, you hold your phone near a disk on the wall and that becomes the remote control for the air conditioning. Hold it near the one for the TV and it’s the remote control for the TV. And put it in the car it’s automatically sinks to your car. Things like that are becoming. And it’s all very phone centric which is good because…

Fairbairn: You can get a business card with that.

Jaggar: A business card, yes, absolutely. All of your business stuff on it. Yes. And, of course, electronic wallets are based on the same technology. And as ARM have just announced their 64-bit architecture, V8, which is kind of cute. So the way the cell phones are going up, you know, you’ve got this powerful computing device with you and the world is changing so fast. You know, the augmented reality stuff that’s coming up is just fantastic in my opinion. And suddenly this phone will become—well, if they’re not a necessary part of your life now, they’re certainly going to become everyone will have to have sooner or later, just to control your house, to control your car, yes.

Fairbairn: So how old are your kids now?

Jaggar: Sixteen, fourteen, and twelve.

Fairbairn: Okay. So you’re about to give them advice as far as the path they ought to pursue, what’s your view having been through the career you’ve had and perspective you gained?

Jaggar: I had to show my kids my primary school or elementary school reports recently because my marks weren’t very good. There wasn’t a teacher until later on at high school that really inspired me. And so I said to them this is not a sprint race that you’re in even at age 16. She’s doing papers, my oldest is doing papers from a year old than she should be. So she’s doing very well. But being careful not to burn themselves out with education, to enjoy themselves and to find out—they’re trying to find their vocation, I believe, up until about 17, 18, 19. And then they can start to specialize at university and then really, that’s

when they need to start accelerating at full speed. But I tell them just—it’s an old Robin Saxby line and I thoroughly believe it, “Work hard, but have fun.” It was ARM’s motto for a long time and I think it’s a good one. We worked hard but gee, we had some fun. And I’m a little bit like that with my kids too. So they’re achieving. That’s what’s good. My middle daughter is profoundly dyslexic so that’s been a challenge, but we’ve sort of cracked that now. So the oldest one is very much her father’s daughter. She picks things up quickly. She’s very good at science and math and physics. So it’ll be interesting to see where she ends up. She jokingly says you’ll just get me a job at ARM, won’t you dad? The youngest one he’s only twelve, plenty of years before he…

Fairbairn: Who knows?

Jaggar: Yes, who knows? Yes. He smiles a lot so that’s all right. He’s a good kid. A good kid. So that’s all good.

Fairbairn: That’s great. Anything else that I missed thing or anything you want to…

Jaggar: I don’t think so. No. What didn’t we touch on? I guess the other thing—the other secret for ARM is we listened to customers. We were out there listening. We weren’t a big behemoth like Motorola and Intel. We were little and hungry. We also made some mistakes. We did a game console for—it was part of the Time Warner organization, 3DO it was called.

Fairbairn: Yes, I remember that.

Jaggar: Yes, and Panasonic licensed it and it went nowhere. They were going to make money off game hardware and not so much the software was their business model. And the hardware was just too expensive. Again, Newton was we thought they were going to throw bazillions, they didn’t. But luckily Nokia came along at the right time. As did the cell phone. I mean the whole cell phone market came along at the right time.

Fairbairn: Yes, you’re in every cell phone, right. I mean…

Jaggar: I think so. There are some Intel midfields now. If you dig in the bottom of China I think somebody is shipping some midfields, but why I don’t really know to be honest. I mean I don’t know. I really don’t. There just seems to be a whole lot of hassle for no gain. I mean it’s interesting that the ARM Intel thing to me is fascinating how ARM has slowly gnawed its way up the trouser leg, I suppose of Intel.

Fairbairn: Yes. You ship far more processors than do anybody, orders of magnitude, right.

Jaggar: It would be nice if we could be making their margin, wouldn’t it. But, you know, it’s a very broad base now that as you said because we’re shipping so many very broad base that ARM fires from. And ARM is big enough now that it’s hard to buy. It’s a big chunk of change for anyone to buy them. That was always a threat. We were always being threatened with just being bought out. And Robin was very smart. He’d say to people as soon as you buy us you destroy us because you’d turn us into a silicon foundry and that wasn’t the business model with the licensing that worked. Yes. It will be interesting to see in another ten years where ARM and Intel are. Very interesting indeed.

Fairbairn: All right. Well, thank you very much. I appreciate your time, Dave. We’ll get on with the tour now.

Jaggar: Thanks.

Fairbairn: Thanks.

END OF INTERVIEW

