DOI:10.1145/1941487.1941501

Jason Fitzpatrick

Interview

An Interview with Steve Furber

Steve Furber, designer of the seminal BBC Microcomputer System and the widely used ARM microprocessor, reflects on his career.

STEPHEN BYRAM FURBER is the ICL Professor of Computer Engineering in the School of Computer Science at the University of Manchester, U.K. Furber is renowned for his work at Acorn Computers Ltd., where he was a principal designer of the BBC Microcomputer System and the ARM microprocessor, both of which have achieved unique historical significance.

The BBC Micro platform was fundamental to computing in British education in the 1980s and directly led to the development of the ARM architecture. The ARM architecture is the most widely used 32-bit RISC architecture and the ARM processor's power efficiency—performing the same amount of work as other 32-bit processors while consuming one-tenth the amount of electricity—has resulted in the widespread dominant use of the ARM processor in mobile devices and embedded systems.

Furber is a Fellow of the Royal Academy of Engineering, of the Royal Society, the IEEE, the British Computer Society and the Institution of Engineering and Technology (IET), and was appointed Commander of the Order of the British Empire (CBE) in 2008.

Jason Fitzpatrick, a computer historian and the curator at the Centre for Computing History at Suffolk, U.K., conducted an extensive interview with Furber on behalf of the museum, which is dedicated to creating a permanent public exhibition telling the story of the information age (see http://www.computinghistory.org.uk/.) A video of the interview is available at http://www.computinghistory.org.uk/det/5438/Steve-Furber-Interview-17-08-2009/; a condensed version of the interview is presented here.

The BBC Micro was just the front end of something that was designed as a dual processor from the outset.

I'd like to talk to you about your involvement with Acorn, and what it's led to today.

I was at the University [in Cambridge]; I read maths as an undergraduate and I went on to do a Ph.D. in aerodynamics. During my Ph.D. I got interested in aspects of flight, and then I heard about the formation of the Cambridge University Processor Group. I thought maybe I should join up with these guys and see if I could build myself a flight simulator or something like that. I was involved in the University Processor Group from its foundation although I wasn't actually a founder. I went along to the very first meetings and started building computers for fun, which was fairly scary in those days because the components had to be ordered from California by mail order using a credit card. I was a student so credit cards were fairly scary then; using them internationally was even scarier. But we got the microprocessors. My first machine was based on the Signetics 2650, which not many people have heard of these days. It had a full kilobyte of static RAM main memory. I assembled the circuit board using Verowire, which is a little wiring pen where you hand-wired the things together; you soldered it, which melted the insulation and made the connections. I understand it gave off carcinogenic vapor, but it hasn't got me yet.

That's how I built these things. I built myself a small rack—I couldn't afford a commercial rack, so I made one and got the 2650 system going. In the Processor Group, enthusiasts exchanged notes with each other. I remember Sophie Wilson coming to my house for one meeting of the Processor Group, looking at my machine and poking away at it—finding faults in the memory and stuff like that. Then while I was still a Ph.D. student in the Engineering Department, Hermann Hauser came knocking on my door and explained that he and Chris Curry were thinking of starting a consultancy company in the microprocessor business. They had been looking to the University Processor Group as the source of technical people who might be able to help them; he asked me if I was interested. I said, "Well, I'm just a hobbyist. I've been doing this for fun. But if you think I can help, I'm willing to give it a go." That's how I joined the embryonic Acorn, before it was Acorn.

Was it based inside Sinclair's building at the time?

Yes, the first things we did were in the Science of Cambridge Building in King's Parade. Chris Curry was set up running Science of Cambridge with Clive. Hermann and Chris did bits of Acorn work in there. In fact the first thing I did for Acorn was actually not for Acorn, it was for Science of Cambridge. I hand-built the prototype MK14; I got a circuit diagram and built one using Verowire, soldering in my front room. The MK14 was basically a copy of the National Semiconductor SC/MP development kit. They had taken what was a masked program ROM from the development kit and copied it into two fusible link PROMS for the MK14, and they managed to copy it wrong. So I debugged this thing in my front room. That was the first piece of work I did for them.

Then Chris and Hermann got a contract to do some development work on microprocessor controlled fruit machines, which were very new at that time. Up to that date fruit machines had all been controlled by relays and so on; this was an early attempt to do microprocessor stuff. We used two SC/MPs in a rack to control the fruit machine. In fact, the software for that was bootstrapped from the 2650 machine I built in the Processor Group; it was used as a dumb terminal into the SC/MP development kit, and we brought this fruit machine controller up. The main challenge in those days was to make these things robust. Very early on people had discovered if you just sparked electronic cigarette lighters next to the fruit machine, they would often pay out.

Yes, the program counter jumps off somewhere and anything can happen!

Yes. So that was when Sophie Wilson came in. She designed an FM receiver front end that would trigger whenever you flicked one of these cigarette lighters and cause the SC/MPS to reset; it would definitely not pay out. [laughs] That was the requirement—if you interfered with it, it should definitely not pay out. The things were tested by plugging a mains adapter into the wall, plugging the fruit machine into one socket, and an arc welding transformer into the other. Somebody welded metal together while you operated the fruit machine to see if the thing was robust to sparking.

Fantastic! The feeling at that time was very much of the hobbyist. You just enjoyed doing that kind of thing, and the whole industry has pretty much come out of that. Is that fair to say?

Yes, that's right. We are talking about the late 1970s before the IBM PC started, before the Apple II had appeared. There were some very basic box machines. I think the Altair had probably appeared about this time in the States.

In the University Processor Group, the real men built computers with TTL. It was only the wimps like me that used newfangled microprocessors, which were kind of cheating because you got too much functionality in one package. But yes, microprocessors were just entering the public consciousness, so the MK14 from Science of Cambridge was an example of a microprocessor on a printed circuit board with a hexadecimal keypad and seven segment display; you could put assembly code into it and make it run. Sophie saw the MK14 and said something which she said many times—basically, "I could do better than that."— and she went home over

PHOTOGRAPH COURTESY OF STEVE FURBER

Easter holiday and came back with a design she called the Hawk, which was 6502-based. Hermann looked at this and thought he could sell it; that became the Acorn System 1. The name Acorn was introduced originally just as a trading name. The company was called Cambridge Processor Unit Ltd.

If you look at those machines today, the System 1 and the MK14, they are what most people would describe now as unusable. But these things sold in a big way.

The System 1 and the MK 14 sold faster than people could put the kits together. I think the System 1 was mainly sold as a kit, so you got the parts and you had to solder it together. But there was lots of interest. It was really the only way the general public could get their hands on anything that looked like a computer at that time. Real computers cost a million pounds, lived in clean rooms, and were only touched by men in white coats; whereas these things you could buy for £100 or £200 and play with at home.

It was just the want to own and control one of these things. A lot of it was driven by science fiction...

Of course, the real science-fiction aspect is they got used as props in TV shows as well. So the Acorn System 1 was featured as the computer on "Blake's 7." There was quite a lot of competition between Acorn and Sinclair at the time. Clive Sinclair had proudly boasted that you could control a nuclear power station with his ZX81. Well, this was nothing compared with controlling a 21st century interstellar cargo ship [on the "Blake's 7" television program] with an Acorn System 1.

You win, hands down.

That's right! [laughs]

Going forward to the BBC Micro, obviously the BBC came to Acorn with the specification for a machine? How did that change things at Acorn? The Atom was out and it was selling well. Then all of a sudden you were shot into fame in the computing industry.

The BBC Micro was a huge phenomenon. Of course, when the BBC came their spec was a Z80 machine running CP/M. The BBC Micro was neither Z80 nor CP/M, although a little bit later you could buy a Z80 second processor to run CP/M; we kind of met the spec in the end. But no, they were sufficiently convinced by what we could do with the 6502 that they moved the spec to the machine that Acorn had already begun to get on the drawing board. The Proton was always designed as a dual processor. The fact the BBC Micro had a second processor connection was actually because the BBC Micro was just the front end of something that was designed as a dual processor from the outset.

I remember that when the BBC was talking to Acorn—I wasn't involved in the commercial discussions, I was just a techie—they were confident this machine would sell, and on the back of their programs we'd sell 12,000 of these machines. That was big numbers to Acorn. Not huge numbers—we'd probably sold several thousand Atoms—but it was really worth going for. Nobody imagined that that estimate would be off by a factor of a hundred—one and a half million were shipped in the end—because nobody really anticipated the wave of interest.

I really realized that this was a phenomenon when Sophie, Chris Turner, and I agreed to do a seminar at the IET in Savoy Place in London. It has a big central amphitheatre that sits about 500 or 600 people. They asked us to do a seminar on the BBC Micro. We went down there thinking this is a big room, I wonder if they'll fill it? Three times the number of people they could get in the room turned up. People booked coaches from Birmingham; they had to be sent home because Health and Safety said they couldn't let them in. Nobody was expecting this, either.

We always had the education market in mind, but this was much bigger than just the education market in terms of interest.

The machine was first sold in January of 1982, so this may have been later in 1982.

And that was the first time you thought this is big?

Well, this is when you first felt the scale of public interest. People were prepared to hire a coach from Birmingham to hear this bunch of techies, who probably didn't know how to speak in public, say something about this hobbyist computer thing. Of course, we always had the education market in mind, but this was much bigger than just the education market in terms of interest. We actually went on tour with this seminar. We gave it twice more at the IET to soak up demand. We did a tour of the U.K. and Ireland. Everywhere we went there was a big turnout. There was real, real interest.

What sort of people were coming to this?

A wide range of people. I think it is the same phenomenon as with the System 1 but on a bigger scale. It was a bunch of people who recognized that computers were about to come within their reach, when they'd been behind closed doors throughout past history. There were lots of companies building machines at the time. We've mentioned Sinclair, but if you go and look at the machines coming out, the 1980s was a real era of diversity. Wonderful quirky machines of all shapes and flavors, all coming out of companies a bit like Acorn: small startups; enthusiasts the public couldn't perhaps fully trust. Unless you were a hobbyist and a real enthusiast yourself, you didn't know who to trust.

Then the BBC put their name on this machine from Acorn. I think that really was the key to the success of the BBC Micro, even though by the standards of the competition it was a slightly expensive machine. It was slightly higher spec and that was partly the BBC's requirements. The BBC imposed—no, imposed is the wrong word—encouraged us to go with a particular spec. The spec was all negotiated and agreed; there was no imposition. But they were tough negotiations. The BBC had a pretty clear idea of what they wanted. The fact that we pushed the technology a bit—you know, we ran the memory a bit faster, we made an early use of semi-custom chips, we had these two ULAs that gave us quite a lot of excitement—all pushed the price of the machine up a bit. It was an expensive machine, but despite that, a much broader market was comfortable with it because it had the BBC name.

The machine itself was low level. If you connected something to the back, like a joy stick, it was easy to communicate with it because you weren't, as you would be with today's PC, so high up in the software stack. The BBC Micro was easy and anybody could read the user guide; if they had a little bit of technical know-how they could build their own electronics to go into the one megahertz bus and program it. It was easy. I'm quite sad we've lost some of that. I think it's much tougher now for teenagers with technical interest to say, "I want to build one of these things, plug it into my computer and make it do stuff."

Yes, I had a BBC Micro and connected up various different bits of equipment to it and had to program this thing down to the metal, talking directly to the chips; you're talking to the very pins that you just wired up something to. Like you say, that's completely gone. You have to go through layers and talk through Windows, etc. How does one get into hardware programming now? We're running so fast that you can't even make circuit boards to connect to it because the frequency just won't have it.

You're right; the problem isn't simply the stacks of software. The problem is the hardware now...the signal-level hardware is very difficult to work with and requires a lot of skill and knowledge. But of course, you could still build a slow interface. If you want an interface at a megahertz, then connecting to a megahertz is no harder now than it was in 1980. Actually, a megahertz means you can do a million things a second, which is quite a lot for many purposes. So how would people access that? I guess they get some of it through the Lego robotic systems and so on. But in a sense, a bit too much of that is prefabricated. It's a bit too ready-to-go. But you are talking at quite a low level to the microcontroller. There's no reason now why we can't make BBC- like products using microcontroller technology. In fact, you could probably make a BBC Micro in a single chip and sell it very cheaply. I can still run BBC Micro programs on my laptop. In fact I have BBC BASIC for Windows, which is a very faithful emulation. It's produced by Richard Russell, who is one of the people who worked for the BBC when we were negotiating with them in the 1980s. I have an Archimedes emulator that I run on the PC, and I have a BBC emulator that I run on the Archimedes emulator that runs on the PC.

The BBC imposed—no, imposed is the wrong word—encouraged us to go with a particular spec.

Emulating emulation!

And it's still faster than the original BBC Micro because that's how much compute power we've got now. So you can still run the stuff, but you can't interface hardware to it so easily.

There's an interest in actually building something much more basic again around today's microcontroller technology. My guess the market for that is small but not insignificant.

I think the nearest thing you get at the moment is a PIC development kit for about £20; with a PC you can write PIC Assembly code, load it into the PIC, and do all sorts of stuff with it. I do know quite a few people who use PICs this way. That's probably the nearest you get. But with a PIC, in some sense you're going to a lower-level interface than on a BBC Micro.

To go back to the point about the price of the BBC, it was expensive. You had Spectrums, which were under half the price. Was a big part of the negotiations trying to keep the price down?

Yes. You may remember that we launched (I may get the numbers wrong here) the Model A at £239 and the B at £339, but really we couldn't make them at that price. So it went up to £299, £399. We were trying hard to keep the price down. But really, how could you make the machine cheaper? If you look at a Spectrum, then of course everything about it is cheaper, including the keyboard. The BBC had by today's standards a stunningly high-quality keyboard. Today's keyboards are very cheap elastomeric. Every switch on the BBC Micro keyboard had a pair of gold wires that touched when they crossed. When you pushed the key down they sprang together and touched, so you weren't making the contact with the key press; you were removing the obstruction that was preventing the contact. And BBC keyboards were formidably robust. There were some manufacturing problems with them. There was a batch that were manufactured with the key switches about half a millimeter off the PCB; when you hammered on them the force got transmitted through to the solder joints on the back then the PCB tracks broke. But when they were made properly, they would last 10 years of kids thrashing them. The machine took a hammering. You know, mine still works.

They're fantastic machines. So what did the success of the BBC Micro do to Acorn?

Acorn grew rapidly. When the BBC contract was signed, I was still a research fellow at the University; my day job was doing aerodynamics. I guess Acorn at that stage employed maybe 30 people. I joined them full time in October 1981, and by 1983 the company had grown to 400 people, just to manage this stuff. There was no real sales activity involved because the stuff sold faster than you could make it—it just walked off the shelves. So it was really building the technical team up; and we had a strong manufacturing team. If you're going to make millions, you want to know something about procurement. Acorn didn't actually manufacture; it was all sub-contracted. We had to build skills in manufacturing. The BBC contract also required lots of exotic technology beyond the basic machine. The Prestel telesoftware receiver was a bit of unknown technology we had to make; the second processors including the Z80 running CPM. There were a lot of things to go around it, so we grew technical teams.

We also realized the price point was quite good for schools and professional users, but it was too high for hobbyist and most home users, so we developed the Electron, which was a cost-reduced BBC Micro - not an entirely happy story. There wasn't much wrong with the machine, but for Christmas 1983 when there would have been a huge market we couldn't get the electronics reliable enough. By 1984 when we cracked it and could make lots of them, the market had gone. We ended up with a quarter of a million of them in the warehouse that were eventually sold below cost.

But yes, the company went from a small, experimental start-up to big, established…well, is 400 big—medium-sized maybe?

What was the atmosphere like with the people that worked there? Or was the excitement lost in the numbers of people?

No, there was a core group who did the really ambitious technical stuff. We did begin to get the idea that we could take on anything. We knew there was competition. We were a bit over- focused on Sinclair as the competition. Standard problem: seeing the parochial competition and missing the real competition from much further away. Technically we felt very much on top of what we were trying to do, and we kept taking on bigger and bigger challenges. We felt we'd developed the Midas touch when it came to advanced technology; that's a lot of the background to the development of the ARM microprocessor. This was a very short period of time when you look back. The first sale of the BBC Micro was January 1982. The ARM design started in October 1983 and the first ARM chips were in our hands in April 1985. It was only four years from beginning to sell BBC Micros to having ARM chips in our hands.

Going on to the ARM chip, how did that come about? What started you guys working on a new type of processor?

The advanced R&D group, which I was in with Sophie and several other folk, was responsible for looking furthest out in terms of what the company was going to do for product. Hermann was very hands-on—he was always very involved in this. We were already thinking the BBC Micro has been a big success; we need to build on this. We could put second processors on, which tided us over for a bit. But really we needed to be thinking about the next big machine. It was clear that we were going to step up from 8 bits. 16-bit processors were already around and going into some competing products.

We looked at the 16-bit processors that were around and we didn't like them. We built test bench prototype machines with these. We didn't like them for two reasons. Firstly, they were all going to 16 bits by adopting minicomputer-style instruction sets, which meant they had some very complex instructions. The anecdote I always remember is that the National Semiconductor 32016 had a memory-to-memory divide instruction that took 360 clock cycles to complete; it was running at 6 megahertz, so 360 clock cycles was 60 microseconds; it was not interruptible while the instruction was executing. Single density floppies, if you handled them chiefly with interrupts, give an interrupt every 64 microseconds, double density every 32. Hence you couldn't handle double density floppy disks. The complex instruction sets gave them very poor real-time response.

The second problem we had was that none of them kept up with memory. Commodity memory at the time was, as it still is today, DRAM—it was rather smaller DRAM chips—and this DRAM had a certain bandwidth. If you ran the DRAM at full spec, you get a certain bandwidth. We deduced from a number of experiments that compute power goes with memory bandwidth. But these microprocessors wouldn't even use the bandwidth that was there; they couldn't keep up with the memory. This struck us as the wrong answer.

So we were feeling unsatisfied with the microprocessors we could go out and buy when I'm pretty sure it was Hermann who dropped a couple of papers on our desks, which were early RISC papers from Berkeley and Stanford, where a grad class had designed a microprocessor that was competitive with the best industry could offer. We looked at this, and thought how would we really like a microprocessor to look? Sophie began tinkering with instruction set architecture, inspired considerably by the Berkeley and Stanford RISC work, but also by what she understood of the 6502, and also what was needed to write a good BASIC interpreter. Sophie had written several BASIC interpreters by then for the Atom, for the BBC Micro, for the 32016 second processor, and so on. She sketched out an instruction set.

Then in October 1983 Sophie and I went to visit the Western Design Center in Phoenix, Arizona. They were designing a slightly extended 6502, the 24-bit address 6502 that became the 65C816. We went in expecting to find big, shiny American office buildings with lots of glass windows and fancy coffee machines. What we found was a bungalow in the suburbs of Phoenix. They hired school kids during the summer vacation to do some of the basic cell design. Yes, they'd got some big equipment, but they were basically doing this on Apple IIs. My strong memory is walking out of there saying to each other, "Well, if they can design a microprocessor, so can we."

We went back and from the tinkering that Sophie had been doing with instruction set design, which Hermann had entirely supported and approved of, we put the project on an official footing.

The other infrastructure aspect of this is that Andy Hopper from the Cambridge Computer Lab, who was a director at Acorn, had persuaded Hermann that if he was serious about staying in the computer business, he needed to get serious about chip design. Andy advised Hermann to get chip design tools from VSLI Technology, and Apollo workstations. They recruited IC designers, a group led by Robert Heaton. I can't remember precisely the order they came in, but Jamie Urquhart came early, Dave Howard, Harry Oldham—all names still associated with ARM. So we'd got these tools and the IC designers, but no chips to design. Sophie and I were thinking we should have a go at designing our own microprocessor.

We looked at this RISC stuff, and thought this is kind of obvious, this is a good idea. So we'll set off using these ideas and try to put something together. But it's clear that big industry has got far more resources; they're going to pick up on these ideas too, we're just going to get squashed underfoot by big industry as it catches up. But if we set about doing this, we'll learn something, we'll understand something about what it takes to build a good microprocessor; and then we'll be better at recognizing a good one when we see it. We didn't expect this to go through. To us, building microprocessors was a black art. The big companies had hundreds of people, and it took them 10 revs of the chip before it started to work sensibly. It just looked like a black hole, and Acorn couldn't afford that size of black hole.

But we got on with it. It turned out there is no magic. Microprocessors are just a lump of logic, like everything else we'd designed, and there are no formidable hurdles. The RISC idea was a good one; it made things much simpler. Eighteen months later, after about 10 or 12 man-years of work, we had a working ARM in our hands, which probably surprised us as much as it surprised everybody else.

In July 1985, we'd had the processor on our bench running for a couple of months; we decided it was time to say something to the public. I rang a journalist and said, "We've been working on this microprocessor design and we've got it working." He said, "I don't believe you. If you'd been doing this, I'd have known.", and put the phone down. [laughs] We'd actually done this in considerable secrecy; the secrecy was so good that we couldn't even persuade people when we got the working silicon in our hands. In terms of timescale, this was all happening at exactly the time when Acorn was going bust and being rescued by Olivetti. I believe Olivetti wasn't told about the ARM when they bought Acorn. When they bought it, we thought, maybe it's time to own up: they didn't know what to do with it.

My strong memory is walking out of there saying to each other, "Well, if they can design a microprocessor, so can we."

So we're talking about a chip now that is in something like 92% of mobile devices today?

Yes. Around the end of 2007, the ten-thousand-millionth ARM had been shipped, so there are more ARMs than people on the planet. I believe production is currently running at about 10 million a day. It is projected to rise to about one per person on the planet per year within two or three years.

They're mind-blowing numbers. Looking at all this and seeing how it's changed us as people -to have this computing power in our pockets has completely changed the way we are and the way we live our lives. And you played an absolute key part in that. So what does it feel like, to know that you played a big part in it?

It's kind of magic, isn't it? I mean it's largely serendipity. I spent some of my last two years at Acorn trying to work out how to build a business plan for a company that could take ARM out. Acorn's desktop PC business was not big enough to support proper processor development; we needed a bigger market, so I tried to work out how to spin out a company. I could never get the numbers to work. You have to sell millions before the royalties start paying the bills. We couldn't imagine selling millions of these things, let alone billions, which is where we are now. But a lot has happened to make that happen—it hasn't gone there on its own. When the company was spun out, Robin Saxby was brought in, and he and the team evolved this business model, which has been instrumental in its success. Had Apple not come knocking at the door wanting the ARM for the Newton, and Robin Saxby not been brought in to head it up...You know, there are lots of ifs.

If these things hadn't happened, we wouldn't be where we are today. But where are we today? I've been trying to work this out. I suspect there's more ARM computing power on the planet than everything else ever made put together. The numbers are just astronomical.