Oral History of Sophie Wilson
2012 Computer History Museum Fellow

Interviewed by:
Douglas Fairbairn

Recorded: January 31, 2012
Cambridge, United Kingdom
Mountain View, California

CHM Reference number: X6409.2012

<crew talk>

Fairbairn: Okay, Sophie, can you hear me?

Wilson: Yes.

Fairbairn: I have no audio.

<crew talk>

Fairbairn: Sophie?

<crew talk>

Fairbairn: Okay, Sophie, are you there?

Wilson: Well, I'm here.

Fairbairn: Oh good. It's working both ways now. Okay, are we ready to go gentlemen?

<crew talk>

Fairbairn: Okay. My name is Doug Fairbairn and it's January 31st, 2012 and I'm in discussion with Sophie Wilson, a primary architect of the ARM microprocessor which over the last 18, 20 years has become probably the most widely used microprocessor throughout the world and so the impact on every day electronics and our every day lives is really quite immeasurable. So, delighted to be here, Sophie, welcome.

Wilson: Hello.

Fairbairn: So, before we get into the sort of general flow of the interview what I'd like to do is ask you, Sophie, to just spend a few minutes talking about what the ARM processor is; what's unique and special about it and what impact, from your point of view, has it had in terms of the micro electronics world? So, if you could just spend a few minutes talking about that and then we'll get into the flow of the interview itself.

Wilson: Okay. So, I'd like to take you back to 1983 <sound effect> we're going back. At the time, Acorn Computers was a reasonably successful producer of small microcomputers, a sort of British company like Apple. We had a machine, the BBC Microcomputer, that had been franchised by the British Broadcasting Corporation; it was, in some ways, quite like an Apple II. It had a 6502 microprocessor in it though we ran it faster, it had higher resolution graphics. It was mainly TTL logic but it had two custom integrated circuits inside it; a video processor and a serial processor. So, in making that machine we'd already stretched ourselves to build custom circuitry. One of the other bits of custom circuitry that was an optional extra for that machine was an interface to a second processor. The BBC Microcomputer was designed as a two processor system from the outset in order to get past the impasse that was in the company at the time where lots of people wanted to build a small cheap machine and lots of other people wanted to build a big expensive workstation style machine and by chopping the thing in two, having what we called an IO processor and a second processor that would do the actual heavy lifting, we could make all of that happen. And the part that was the IO processor was able to run in its own right and we could say it in its own right and that's the BBC franchised. We made lots of second processors for it based on existing microprocessors. So we built 80286, 6809, NS32016. Another company related to us built a 68020 second processor to it. We could see what all these processors did and what they didn't do. So, the first thing they didn't do was they didn't make good use of the memory system. The second thing they didn't do was that they weren't fast, they weren't easy to use. We were used to programming the 6502 in the machine code and we rather hoped that we could get to a power level such that if you wrote in a higher level language you could achieve the same types of results. So you could write 3D graphics games. You could do whatever you wanted to do without having to go all the way down to assembly language and for these processors that were on sale at the time that wasn't true. They were too slow. So between the two things we felt we needed a better processor. We particularly felt we needed a better processor in order to compete with what was just beginning to be a flood of IBM PC compatibles. So, we gave ourselves a project slogan which was "MIPS for the masses" Millions of Instructions Per Second for masses of people. This was very different to what other people were doing at the time. RISC processor research had just been sort of released by IBM, by Berkeley, by Stanford, and they were all after making workstation class machines that were quite high end. We ended up wanting to do the same thing but at the low end, a machine for the masses that would be quite powerful but not super powerful. So, ARM was that machine; a machine that was MIPS for the masses, not super powerful. It kept Acorn alive for another 16 years after we started the project. From about 1987, 1988 onwards we sold very little that wasn't ARM powered. We started selling ARM powered machines in 1986, 1987. Those things that we'd endowed it with, what we'd set ARM up to be with its slogan MIPS for the masses, with its cheap and powerful mindset, they were the things that became valuable; when people wanted to put good amounts of processing into something, that was the really important attribute. We designed a deeply embedded processor, or an embedded processor, without consciously realizing it in our striving for what we thought would be ideal for our marketplace; that's been what's really mattered. As a sort of side effect of making it cheap and simple to use, we also ended up making it power efficient; that wasn't intentional. The MIPS for the masses thing was a slogan that we had but making it power efficient was an accident. In hindsight, it was an obvious accident. We only had 25,000 transistors in the first one. We were worried about power dissipation. We needed to be extremely careful for something that would be mass manufactured and put into cheap machines without heat sinks and that sort of thing. So there were already some aspects of power conservation in the design but we performed way better than that and as the world has gone increasingly mobile that aspect of ARM has mattered as well. But to start off, we designed a really good deeply embedded processor.

Fairbairn: That's a great summary. Thank you very much. So, now I'd like to switch and go back to the very beginning, your beginnings and just, from a human point of view, try to understand how you arrived at Cambridge and doing the kind of work that you did. So, could you take us back and tell me a little bit about where you were born, where you were raised, a little bit about your parents and how they influenced your family life and how that did or did not sort of influence your later life.

Wilson: Okay. So, I was born in Leeds in Yorkshire in England. Very shortly after that my parents moved to a small village called Burn Bridge where we lived for the next whatever it was, 20 odd years. Burn Bridge isn't famous in any particular respect save for the fact that the bridge goes over a burn - or a little river; that little river is called Crimple Beck and we used to go and play in it. And the River Crimple or Crimple Beck was the thing that ICI chose to name a synthetic fabric after, Crimplene. Later on I would go and work for ICI Fibres Research Centre in the nearby town of Harrogate. So that's where I was born and brought up; a very small village. The village itself didn't have a shop of any kind. We were quite isolated in many ways. The nearest school two miles away; well, the nearest main school. My parents were both teachers. If they had one ambition for their children as their children grew up it was "Don't be a teacher." My father's an English teacher, my mother a physics teacher. I don't know what they hoped for their children but actually we all went off to university to do mathematics; neither physics nor English got a look in. My father also does an immense amount of sorts of handiwork, construction. So we grew up in an environment where-- well, when I was extremely young he'd built the family car. He built boats. He built half the furniture in the house. When my mother's Physics Department needed all sorts of instruments or something like that, we built them. You know, we'd have a dining room table and everybody in the house would be sitting around the table and we'd be building Heathkit multimeters all through an evening.

Fairbairn: What a great story.

Wilson: Everybody's sitting around the table…

Fairbairn: How many brothers and sisters sitting around the table?

Wilson: Well, there are three children and two parents. So, everybody would be sitting around the table. My grandmother lived just up the road so when she visited all this stuff had to be taken down and the house made decent which was difficult at times when we had things like a boat mast in the living room going from end to end of it - 26 feet of mast.

Fairbairn: My goodness.

Wilson: So we did grow up in an environment where making things, making things work was a routine thing. You know, the jewelry I'm wearing today, my father made the jewelry. My mum made my blouse. It was that sort of existence.

Fairbairn: Wow. What a fabulous education just sitting around the dining room table, huh?

Wilson: Yes.

Fairbairn: Did you gravitate towards any particular thing at the time? I mean, did you find yourself interested in electronics or sailing or, you know, you were building almost everything so…

Wilson: We were building everything. With both parents being teachers, there were long summer holidays where we'd just leave the house and go up to the Lake District or something and spend the whole of the summer outside in tents or later on dad built camper vans. You know, buy a van, strip the guts out of it, fit it out with furniture and bedding and all sorts of things. But at first we would just park it--parked on Honister Pass in the Lake District, in a tent and have to get on with it. There were substantial amounts of freedom; an environment where anything was possible, an environment where there was no real limit to what my parents could do. The sort of people you could always ask to do something and they could always work out a way to do it. You know, books in the house, the full complete volume of the Encyclopedia Britannica, many, many books. We were a very "ready" family. We used to go to the library every week and the three of us would pool our tickets and we'd get out 12 books. We were allowed four books per ticket, three tickets. So we get out 12 books and we'd all read all the books by the next weekend and go back to the library again. And we were given books for presents a lot. There was nothing in electronics available so, you know, these Heathkits that we'd build for my mum's school or basic principles of things we understood but you couldn't do electronics because there wasn't any. When I was about 13 we were given calculators by our parents, one each, but they were mechanical calculators; a little tablet of metal with slidey things that went up and down and you put a stylus in, dragged it up to the right number and the thing would tally and do addition and subtraction and, of course, we all worked out how to do multiplication with it very arduously.

Fairbairn: It sounds like something your father might have made himself.

Wilson: Yes, but it wasn't so easy for them. With parents for teachers you do get used to living very hand-to-mouth; it was not a rich family.

Fairbairn: So, was there interest in you doing something other than teaching, was it financially based or were there other reasons that they found it not a path they wanted you to follow?

Wilson: They thought it wasn't good financially. They thought it wasn't good for us. They both enjoyed teaching but they understood its limitations. So…

Fairbairn: So, they obviously instilled an interest in learning and the importance of education if nothing else.

Wilson: Well, I'd like to think so. I was in a streamed grammar school; so in the UK the school system of the time, there were Grammar Schools and there were High Schools, Secondary Modern schools. Grammar Schools were the old style school; pupils were streamed by ability and we all (the children) went to the same school. And I was only in a sort of middle in ability range. Nothing much happened. I was a very, very unexceptional student. I won a junior essay prize and that was about it. Nobody, I think, had any particular hopes for me but then in the transition from the Ordinary Level, the O Level exams, where again I did modestly, averagely well, to A Levels. It was sort of an expectation that people with a math bent would choose something like math, physics and chemistry so my parents sent me off to the school, particularly with mum being a physicist, expecting that to be the result. But secretly I hated chemistry and was quite fond of the math teacher. We had a math teacher, Mr. Lowe, that's right, who was, again, common to all three children at the school, so we could pass messages to each other through the math teacher, you know, give…

Fairbairn: Do I understand you went away to school at this point and…?

Wilson: No, no, no, no, this is all the local Grammar School.

Fairbairn: Okay.

Wilson: We could walk to school; it was two miles.

Fairbairn: And until what age were you at that school?

Wilson: All the way up to applying for university.

Fairbairn: So, you really had an interest in math but not some of these other scientific subjects?

Wilson: Well yes, and the interest in math wasn't what I felt was really the right sort of thing. I was actually more interested in English and writing and thinking about things and drawing. I liked art classes a lot. But anyway, so my parents sent me off to school expecting me to follow this math/physics/chemistry sort of route - scientific subject base and I sort of revolted at the last moment and picked Maths. Further, Maths and Physics which put me into a class with only six other pupils that were equivalently gifted at my year for math and that was a sort of new beginning. So, two years of that and you apply to university. And, again, partly influenced by the math teacher, I applied to Cambridge University for which I would need to pass entrance exams. So my life changed at that stage. I was born in June, so all the way through school I was very young for my year. End of June, you're part of the youngest age cohort of the year and waiting back a year meant that I went to university sort of more on a par with everybody else, particularly as I'd been out and got a job and done all that sort of thing. So that was overall very good. That…

Fairbairn: So, do I understand you took a year off before going to university?

Wilson: Not exactly a full year off. I spent the first four months studying for the entrance exams.

Fairbairn: I see.

Wilson: Through the entrance exams, passed the entrance exams, go to interview at Cambridge, passed that. So out of those six pupils in the further maths group only two of us managed to get into Oxford or Cambridge. Out of the whole year, I think, only three pupils-- no four pupils got into Oxford or Cambridge; two in history which was the school's headmaster's favorite subject and he was from Oxford so he got them-- he pushed them into Oxford, essentially, and then me and a guy called Peter Taylor who went to Cambridge. Me to do Maths and he went to do Natural Sciences initially and he changed almost immediately to a different course-- a simpler curriculum and then he dropped out. So that left me.

Fairbairn: So you went from, as your self-described, middling student to one of the top students by the time it was time to enter university?

Wilson: Yes.

Fairbairn: So, you entered the university with what goal? What subject matter were you focused on? And tell me about those years.

Wilson: I hadn't a clue. I mean, I was extraordinarily-- I mean, even though I'd had an extra year and the extra year would be extremely useful in developing me and all that sort of thing, I was extraordinarily naïve about why was I going to university. Although both my parents had been to university, and indeed met at university, and my grandmother had been to university that was about it for the family as a whole; it wasn't something talked about very much and the pressure that's on students nowadays didn't exist in those days. I really didn't understand what I was doing or what I was going to study. So I went off to university. I did a year of maths which wasn't too bad. I could still do all of that maths but it was very clear that Cambridge maths was substantially harder than the entrance papers and I'd had to study hard to do the entrance exams. I came home from the first year of maths and I got another holiday job. I went back to university. In the second year of maths I failed. So, then we had one of those really, really bad situations of having to go back to Cambridge to see the Senior Tutor and try and work out what would be my future. So we'll leave you teetering on the brink there. So what had happened with these holiday jobs? So, after I'd done the entrance exams I went to work for ICI Fibre Research and that research establishment it did all sorts of stuff and they put me into a little department that outfitted the production line with all sorts of machines; and that was sort of fine because that was very like what we'd done at home sitting round the table building all these <inaudible>lab instruments and so on. So, that worked for me. So, and the first thing they asked me to do was to build a "wrap detector" - they had this enormous machine that sits on part of the production line, fibres whiz out of the producing machine, go round this and it goes on. If the fibre breaks the machine has to detect it. And I built one of those and got a stern talking to by the Union steward who said "Look, the standard time to build that machine is…" whatever it was. "You were not supposed to build it in half a day." So he had a word with the boss who'd employed me and they decided to put me doing other things. And that was interesting because they wanted other types of machines researching and building. So the next thing I did, at the direction of one of these people in this small department, was to build a machine that counted droplets. So, you want to do very subtle bits of chemistry and the best way to do the concentrations of things that they want is to put them in drop by drop, and they have a little machine that counts the drops, and this was to be an electronic one, infrared beam of light that the drop falls past and it interrupted the infrared light so the detector can count it. So, I designed and built one of those. I wasn't just building somebody else's thing. I made it all work.

Fairbairn: So you knew enough about electronics by this time to put something like this together?

Wilson: Just about. They were using RTL logic blocks at the time.

Fairbairn: What year was this?

Wilson: This would be '74, '75? So that they had big, you know, that sort of size of prepackaged resistor transistor and diode logic packaged in little boxes with leads coming out and you could connect them together to make things; so it was sort of a grown up version of Lego really. And I sort of wasn't very impressed with that. I mean, I could build the drop counter and it was fine. So I said "I'm sure there must be some way better to do this stuff." And I went out and I got the RCA CMOS book that had just been published by RCA. No, I think it was 1972 published by. I went and got a copy and I sat down and read the book cover to cover.

Fairbairn: It's not exactly prose.

Wilson: Well, it's quite good because it's not merely pages that say "CD4046 is a PLL. CD4004 is a Quad Inverter." Not nearly that sort of thing. It's chock full of application notes.

Fairbairn: Okay. So, you got the book and then you…?

Wilson: I got the book and said "Look, this is clearly going to be much better. It works off 15 volts. It's very noise immune. It's very low power, it won't get hot." So, they said "Okay." So I rebuilt the drop counter with CMOS circuitry.

Fairbairn: They had equipment-- you built it out on a printed circuit board or wire wrap or…?

Wilson: I think it was Verowire. So, 160 millimeter boards; you put the chips on it. You wrap the little wire round it and heat up the wire to melt the insulation off. I think it was Verowire. I mean, Acorn would go on to use a lot of Verowire and it's difficult to remember precisely what I used but that sort of class of stuff.

Fairbairn: So this is in between your first and second year at the university?

Wilson: No, this is before I went to university.

Fairbairn: Oh this is before you went to university, okay.

Wilson: This is between school and university.

Fairbairn: Okay.

Wilson: So then the next thing I did was to build another wrap detector out of CMOS circuitry and here we had a very lightweight metal cylinder and, again, opto-detectors and bits of light and dark patterns of spots round the end of the cylinder so you could work out which way round it was going and how fast it was going and count it; and so, the huge wrap detector that I'd started with, we miniaturized, tiny little unit with this ultra light thing (cylinder). You know, one of the guys had had the idea that you could do it all with ultra light stuff. So I built one of those for them and that, I think, was about the end of it but obviously I'd had a huge influence on what they would do and built these two machines. Then I went off to university.

Fairbairn: Right. And you went to university in what year?

Wilson: I think I would go to university in September of '75.

Fairbairn: Okay.

Wilson: Sounds right. If it's '76 I apologize in advance. So, first year of maths, you know, leaving most of this stuff behind. Right at the end of the first year of maths-- well, at the beginning of the first year there's all those freshmen clubs and so on, as is common at university, and I joined all sorts of things particularly Science Fiction Society where people wrote. It wasn't really for reading books but it was for writing them. So we'd write all this prose and publish it in the Cambridge University Science Fiction Society magazine, that sort of thing. But at the end of it there was a Cambridge University Computer Society, which I joined, and later on there'd be a Cambridge University Microprocessor Society, which I also joined. So, first year of maths, then I went home again and I want a job. So, one of the people who used to work at ICI Fibers had set himself up an electronics business, so I worked for him. You know, I'd been recommended by these people in this department; they knew him so off I went. And so, he wanted me to build a completely electronic cow feeder. So, this was to be a machine that would control an actual feed dispenser basically by turning it on for a certain amount of time and it would take instructions from the farmer or from tags that the cow wore to dispense a certain amount of food. So, I was going to build all of this during the summer holiday, which I did. So, I bought a microprocessor, which was the 6502; my very, very first MOS Technology, one in white ceramic with gold legs, for 76 quid (UK pounds), which was a lot of money, and I designed all the electronics it would sit in and, you know, paid due attention to the fact that it would be a farmer doing it and it had ought to be waterproof and buttons that could be pressed and all that sort of thing and wrote the programs for it. It was a very advanced machine for its time because I used General Instruments, Electrical Erasable ROMs, EEROMs, for its memory storage and it had a 32-byte boot loader which I'd written which was programmed into a PROM and then you could load it; so, more of its own software into it that was stored in the electrically alterable ROMs.

Fairbairn: So you absorbed basics of programming somewhere along the way. You never took a class or whatever. This is just something you were doing?

Wilson: This is just something I was doing, yes. When I was working at ICI Fibres I met my first computer. So they had a PDP-8 squirreled away somewhere that was-- well, it was actually a sort of control automation computer; they had some Unibus stuff coming out of it and there would be big ribbon cables, the colored ribbon cable stuff, running down the lab back to this computer; so that was the first computer programming that I'd ever done. I taught it to play to noughts and crosses using DEC's FOCAL language and that was the first programming that I'd done. And then, you know, I was writing this stuff in machine code for the cow feeder and designing my own versions of circuitry for home computers based on the 6502. So, went back to university; failed maths but by then I'd joined the Microprocessor Society and was telling people things about the 6502 processor, about CMOS logic and low power and so on and generally making a nuisance of myself because I know lots of people there who like the TMS9900 and I think-- the Z80 hadn't quite come out yet but there were some 8088 enthusiasts and Signetics 2650 which was Steve Furber's favorite and there were even some people interested in the RCA COSMAC 1802. So, I was a very oddball nuisance with CMOS logic and 6502 processors.

Fairbairn: And you had chosen the 6502 because it was just available because it was CMOS-- what was the-- why did you…?

Wilson: I'd had a talk with another university student and he'd said "Oh there's this new thing coming out. It looks to be cheaper than all the rest." So, that was about the recommendation that went for it.

Fairbairn: Okay.

Wilson: And it was and so I could afford it on whatever little money I had. So, one way or another, people got to hear about me; in particular, Hermann Hauser got to hear about me. And Hermann had a problem, a problem that remains, which is he doesn't remember things very well or attend appointments very well. So he wanted what he called an electronic pocket diary. We'd call it a personal digital assistant nowadays or it's now a phone, smart phone. But he wanted one of those back in, I suppose this would be about 1977, and he knew enough to know that, you know, he was a physics student, he knew that it had to be low power and there was this person in the Cambridge University Microprocessor Society making lots of noises about low power stuff. So…

Fairbairn: And that was you.

Wilson: …he contacted me. That was me. Yes, he contacted me and said "I want to have this electronic pocket book. Can you advise me about it?" And I said "Oh, that's all right. I can build one of those; no trouble. Tell me what you want it to do and I'll write you some designs down." So, he had a discussion with me and I went off and I came back a lot later on, not later on, it was probably two or three weeks later when he wanted to know how the progress was going, and I had a big folder with me because I had all my designs in the folder. So, in the folder were-- I was showing him "Look, this is what I've designed for you. This is how the keyboard bit works. This is how the display bit works," and so on. And he would say "But what's all these other bits of design paper then?" And they were all designs for my home computer stuff and the cow feeder and they were obviously a lot more elaborate than the pocket diary thing. And he was looking at them and said "Well, will this work?" And I said "Of course it will." And he said "Will you build one of those for me?" I think I said something like "Of course it will. Elements of that design are going to prove out your design." And so he said…

Fairbairn: What was he pointing at or what was it that he said…

Wilson: It was essentially the diagrams of the main part of my home computer.

Fairbairn: Okay.

Wilson: So, I failed the course. Came back to beg for my (Senior) Tutor for permission to do something else; and in that discussion, my Tutor said “Well, there aren’t very many one year courses but there is one in computing science that’s a one year course. You could do that if you wanted to.” And I put up some resistance because I didn’t really think of computing as anything that I really wanted to do but he said it’s the only thing. So, I did that. But also I then stayed up the rest of that summer so between the second year of university and the third year working for Hermann. So, I built him my computer and of course it worked and he decided that he’d go off and commercialize it; that became the Acorn System 1.

Fairbairn: So you did that all by yourself over the summer?

Wilson: Yes. And so, you know, just one of those things you do, you go off with-- you make something all work. I had to change the design to suit the parts that he had around but it was still my white ceramic 6502 in the middle.

Fairbairn: So, did you build the box and everything that it went into or did he take a circuit board and have the system built or how did that all come together?

Wilson: It was the Verowire prototype again on the Eurocard,160 millimeter card, with a seven-segment display that he’d taken from Clive Sinclair’s Science of Cambridge and I changed all the MOS technology IO systems using their peripheral IO chips into the National Semiconductor 8154 IO chips that Hermann could get from Science of Cambridge as well. I wrote the monitor for it which was 512 bytes of machine code, which doesn’t sound very much, but remember this is written entirely by hand with no ability to debug it, blown into PROMs and put into the machine. I think they were overly impressed when it worked the first time, frankly. It was only 512 bytes. So that was Acorn’s first computer, essentially.

Fairbairn: So, did he form the company Acorn or how did…

Wilson: So, Hermann had set up a consultancy, Cambridge Processor Unit, which is an Austrian’s idea of a joke, and he had this idea that you’d do, like the man back in Harrogate that I’d worked for the previous summer, that you do electronic consultancy on this new thing. So, he got a contract with a company that made fruit machines, the big gambling machines, and they were going away from big heavy relays and spinning reels into fully electronic things. So, then CPU Limited won a contract from Ace Coin Equipment of Wales to build a completely electronic version of that. So, Hermann had got this contract going and Steve Furber and Chris Turner, who would later be-- Steve was co-designer of the ARM and Chris Turner was Chief Engineer at Acorn in later life, they built a system based on Steve’s design for his home computer and got that all working with a 2650 in it when Ace contacted them to say “We've heard that there are people going around with Piezo electric lighters and they go up to our fruit machines and they press the lighter to generate electrostatic sparks and under no circumstances must your circuitry fail and pay out any money." So they were all a bit worried by this and so they got me to build them some circuitry to stop that. So, I found that somebody in the building, in fact it was Chris Curry, had a sample single integrated FM receiver. So I converted that into something that would be a wide band receiver so it would detect electrostatic activity and when it reached a certain threshold it would just tell the microcontroller to shut itself down, hold its reset button down; and that seemed to work very successfully in our trials. Chris Turner then helped them take the machine down to Wales and went through their tests. What they did was to take it out onto their factory floor, plug it into the mains, start it up and start checking that it functioned correctly and was programmed correctly and then they plugged an arc welding machine into the socket next to it and started striking electric arcs. And apparently the machine was perfectly happy to shut itself down and say “No, no, you're trying to fool me.” So we passed that one. But then for the actual production machine we changed everything over to use my 6502 rather than Steve’s 2650 processor; it was cheaper and reliable and faster.

Fairbairn: Now, Steve was working for Hermann Hauser or was he also at Cambridge?

Wilson: Steve was doing a Ph.D. for Rolls Royce in the Engineering Department.

Fairbairn: At Cambridge?

Wilson: He was sort of moonlighting for Hermann.

Fairbairn: So, getting back to the university, you completed this year in computing; is that correct?

Wilson: So, it would be wrong to say I completed it so much as with Acorn, with the CPU Limited starting up; I spent most of my time at CPU Limited building stuff for Hermann and a fraction of my time at the university completing the course. Sometimes this worked to my advantage like when they’d carelessly said “Using any computer do the following problem.” Then I could use an Acorn computer. Fortunately, I had a supervisor who was minded to accept that. I did manage to break Martin Richard’s BCPL compiler. So sometimes they’d have questions that were “You must use X to do Y.” You know, you’d use the university IBM370/165 running FORTRAN 77-- can’t be FORTRAN 77, it must have been FORTRAN 66, to compute the angle a tone arm must take on a record player for minimum audio interference. I maximized the tracking ability of the tone arm with the angle. So, I did that in FORTRAN on the university computer because you had to. And there was another question that was using BCPL, Martin Richard’s BCPL compiler “Do this problem” and I wrote the thing very elegantly and recursively and broke the compiler and they could never fix the compiler to accept my program nor could they trivially modify my program to make the compiler accept it. But I mean, other people could write the program in different ways and the compiler would process their stuff but it never did accept mine. I never did work out what they did with that for a mark but then when it said “Use any computer” then off I could go and use machines I was building for Acorn.

Fairbairn: So, did you sort of gradually move over to working for Acorn full time or did you complete the university work or…

Wilson: I had a job offer before I graduated. Hermann attended the graduation ceremony just as a matter of form. My parents had come down to talk to Hermann, I think in the April before the exams, and they’d come to some agreement about my future. I pretty much didn’t care very much went on then. Hermann took them out for cream tea in Grantchester on the river side and they talked money and stuff. I got paid 1200 pounds a year.

Fairbairn: So is this to say your parents had some role in negotiating your position at Acorn?

Wilson: Yes.

Fairbairn: Is that standard procedure?

Wilson: No, I think it was a bit unusual.

Fairbairn: But that’s the way it ended up. So you went to work for…

Wilson: I was still pretty unworldly. <inaudible>.

Fairbairn: So you went to work full time for Acorn?

Wilson: Yes, I went to work full time for Acorn, 1200 pounds a year, and if I built a computer out of Acorn parts I could keep it.

Fairbairn: And so, this was 19-- what? What year was this that you went to work for Acorn full time?

Wilson: I think it must be summer of '79 by now, straight out of university.

Fairbairn: Okay, so by this time you’d designed the original Acorn computer on your own over the summer.

Wilson: Yes.

Fairbairn: What was the state of the world then? What was the next job or…?

Wilson: Oh it was an incredible task of bootstrapping things up. We had nothing. So, you know, I was writing code entirely by hand on pieces of paper. You know, you'd write out mnemonics then you'd hand assemble the mnemonics down to the binary and then laboriously, bit by bit, you would program the binary into a PROM blower. So the first thing I did after that was to spend a lot of time writing an automatic assembler. It was about four kilobytes of code which of course would have to be written by hand but it was the last thing that had to be written by hand importantly. And then I went home over Christmas and wrote the first BASIC interpreter.

Fairbairn: So, what were people using the first Acorn machine for if they didn't have BASIC or other-- is it purely a hobbyist's machine?

Wilson: Well, it was being used to quote (i.e. to respond to a request with a quote) for these control things. So the machine that Hermann had decided would be sold and the kit of parts that they were producing to help them develop things for companies they were consulting to were sort of distinct. So, we're saying the Acorn System I, which went out with my little monitor in it and, you know, I did everything for that machine apart from design the cassette interface, which Steve did. So, I wrote the software, wrote the manual, designed the printed circuit boards, designed the circuits and later on would help put all the components into the box and ship them and then answer the telephone calls from people who couldn't get them to work, handle posting them and returns and all-- everything.

Fairbairn: You were a one person computer company.

Wilson: Well, but everybody did the same. Yeah, Hermann; his fiancée, Pamela; Steve, Chris; we'd all be doing bits of the same stuff.

Fairbairn: I see.

Wilson: It was only later on that people actually got proper roles, were concentrated on one thing.

Fairbairn: How were people programming this initial Acorn 1 computer?

Wilson: So, the monitor that I wrote allowed you to program it in Hex; so they were programming it pretty much as I did and when you'd written a program you'd save it on cassette tape because anything else was horrible.

Fairbairn: Okay, so then…

Wilson: Because we'd have to type it in again.

Fairbairn: So then you had written a BASIC interpreter and did that run on the Acorn 1 or by this time was there Acorn-- what was the progression of machines then?

Wilson: So, because of the consultancy side, they were designing card frames to take things. So, the Acorn System 1 would plug into the card frame. There was a backplane and you could plug in other devices. So, we were bringing up all sorts of other things. In no particular order we had a memory card, eight kilobytes of memory on it. We had a video display. We had a teletext video display. We had a programmable 80-character video display; three different video displays I think we built. And later on we'd have an in-circuit emulator, a big programmable IO port board, disk drive interface, HPIB, GPIB, bus controllers; everything that you'd expect from some of the consultancy to our company that is trying to make control units. And then anything they wanted to do would be assembled out of the kit of parts that they designed and then crushed down onto one printed circuit board or whatever the customer wanted. Meanwhile, as Acorn, CPU Limited would sell all the things to hobbyists who wanted to build the same sorts of machines. So you got System 1. System 2 was a card frame. System 3 was a card frame with a disk drive, in the end with the floppy disk controller or <inaudible>disc drive. And then System 4 was two racks of card frames. System 5 was as System 4, everything at which was running twice as fast because it would move up to the next 6502; that sort of thing. There were lots and lots of parts.

Fairbairn: And so at some point BBC came along and was looking for a computer? Tell me that part of the story.

Wilson: So, we had all these system range bits and pieces and we were selling them. Chris Curry left Science of Cambridge and joined Acorn; he wanted something that was better for the home hobbyist. He thought that the home hobbyist wouldn't really want to do all this stuff in the rack and anyway it was very expensive. So he had a guy called Nick Toop sort of build, out of a kit of parts, a small home computer that we then put all the system range software onto it. And it wasn't a very good computer because it had things like an American standard display controller, a Motorola 6847, which could only develop a US standard TV output and when you plugged that into a UK TV it may or may not work. It was also built with all its chips upside-down. So if you pounded away on the keyboard the chips would fall out of the sockets and fall onto the bottom of the case and the machine would stop working. And there were numerous bad aspects of it but it sold in quite large volume and, you know, it had a lot of the flexibility of the system software, a lot of the flexibility of the system range of being kept as it was put in there.

Fairbairn: And was it…

Wilson: And hobbyists we're buying Atoms, you know, 12,000 or so got sold and that was enough to whet Chris and Hermann's appetite for something more home computer market orientated.

Fairbairn: And was this first one, did it have your BASIC interpreter?

Wilson: Yes. After I wrote the first Acorn BASIC it just got used everywhere and I mean everywhere. It ran the company. The company, they employed a guy called Graham Tebby who went off and wrote, in Acorn BASIC, a full accounting system; so it was running the company. And Atom had a version of Acorn BASIC in it with a floating point extension that the original hadn't had. So they wanted it to do more so we had these big arguments. Andy Hopper, who is now the head of the Cambridge University computing service, wanted us to build machines that were more like workstations, you know, 68000 processors or that sort of thing. Chris and Hermann wanted to build something that was more like an Atom. So, I came up with this concept, as I said earlier, of building a machine that natively had two processors but that you could cut it in half and sell just the IO processor half of it. So the IO processor half of the machine we were designing - all of Acorns machines at the time all had code names that were about Atoms. So, you know, the Atom and then this was the Proton which was short in our minds but professional or properly done Atom. So the Proton design was being discussed and discussed and discussed and discussed and refined and so on and then Chris Curry got wind of the BBC wanting to have a home computer of their own; they'd been working with a company called NewBrain to design something that they could write a TV series around for educating the populace in microcomputers. The BBC had got very aware of this sort of thing because they'd made several science shows about the microchip and how it would change the world and they wanted to get involved, particularly the bit of the BBC called the Continuing Education Department, which is trying to put out programming for people who've left formal education because they wanted to do something and, you know, at the time, all computers were different. Nothing would run on anything else. If you wrote something for an Acorn machine it would not run on an Oric or an Apple or a PET or whatever. So they realized they had to have a custom machine. So Chris Curry got wind of the fact that they weren't getting on very well with NewBrain for their prototype and said "Well, why don't you come and see the prototype we're making." Glossing over the fact that we hadn't made a prototype and we were still in the arguing stage of the Proton. And the BBC…

Fairbairn: What year was this? What time were they…?

Wilson: This was 1981, I think; 1980? [SW: No, 1981!]

END OF TAPE 1 <audio ends abruptly>

START OF TAPE 2

Wilson: --all computers were different. Nothing would run on anything else. If you wrote something for an Acorn machine, it would not run on an Oracle or an Apple or a PET or whatever. So they realized they had to have a custom machine. So Chris Coey got wind of the fact that they weren't getting on very well with NewBrain, for their prototype, and said, "Well, why don't you come and see the prototype we're making?" Glossing over the fact that we hadn't made a prototype and we were still in the arguing stage of the Proton. We perceived--

Fairbairn: What year was this? What time were they--?

Wilson: This was 1981, I think-- 1980 or 1981 (yes, 1981!). So the BBC sort of reluctantly agreed. They were coming under a little bit of pressure in the press for the late delivery of this project. It wasn't very much, but I think they were a little bit sensitive. So Chris and Hermann now had to convince their technical staff, me and Steve in particular, to build one of these things including discussing it to death. So Hermann came up with a strategy. So on Sunday evening, he rang both of us separately, and told each of us that the other one agreed that we could indeed build the Proton to show it to the BBC on Friday. And of course each of us said, "Well, that's crazy. We can't possibly build the machine that we've been discussing for ages by Friday. There's so much work to do." But Hermann said, "Well"-- he said to me-- "but Steve's agreed that we can do it." "Oh, well, if Steve things we can do it, perhaps we can." So he told both of us porky pies. But nevertheless, we came into Acorn on Monday morning and started drawing out the schematic of the BBC machine. You've probably never seen one, but it was a very complicated machine, 100 ICs in the BBC machine. So it took us two days to draw out the schematic, and in parallel with that, we were trying to obtain all the parts. One of the hallmarks of the BBC machine was going to be a very high-performance memory system that would allow the processor and the graphics display systems to collaborate, share. Up to this stage, graphics on microprocessors had been fairly poor, because the frame store, every time the processor accessed it, the graphics display controller wouldn't be able to use it. And so you'd get little flashes across the screen. So we decided we hated that, and it had to stop now. So we'd build a memory system that was much more powerful and could service both the processor and the display controller at the same time. So this would be a 4-megahertz memory system, four million accesses per cycle. Four million-- yes-- four million access cycles per second, I mean to say. And to do that, we needed some very special DRAMs. Hitachi were the only people who made a DRAM that went that fast, at the time. So we got hand-carried by the rep the only four 4816s in the country. 6502 was no real problem, and all the other parts we gradually obtained. And then we had the nightmare of building the thing, hundred-chip machine. That was wire-wrapped. Hermann managed to spring an expert out of the Cambridge Computer Lab, Ramanuj Banerjee, who it was claimed was the fastest gun in the West, fastest person with the wire wrap gun. And we set to, on Wednesday, when we had assembled all the parts, wire-wrapping the thing. We completed the wire wrap by the end of the Thursday night, and it didn't work. It was attached to one of our system range computers with an in-circuit emulator, so a big umbilical cable coming out of the system range computer into the processor socket of the Proton prototype, and we were troubleshooting it and correcting errors in the wire wrapping, and the thing still didn't work. And there was no logical reason why it wouldn't work. At that stage, I went home. It was something like two in the morning on Friday morning, when the BBC would be coming at ten in the morning. And I said, "Look, if anybody's going to have anything working, it will need a program to run on it, and I'm the person who's going to write it. So if I'm to write anything, I have to go home now and get some sleep." So I went home. I came back at eight a.m., and they were all sleeping on the floor and things. They got it to work, and Hermann will probably relate the story to you himself but, in brief, Hermann determined that if there was no logical reason why it didn't work, then it must be the fault of the in-circuit emulator, and that what they should do is take the in-circuit emulator out, put a native processor in it, and surely it would work then, because there was no logical reason it shouldn't. And annoyingly, he was completely right. It did work. So I <inaudible>returned to find a piece of hardware working, but obviously it had no program. So in the next couple of hours, I needed to adapt the Acorn operating system to run on this new hardware, bootstrap BBC Basic, get a basic interpreter running on it, which I did. The BBC arrived, and while Hermann and Chris were talking nicely to them, delaying them on the stairs, that sort of thing, I was speedily programming with BBC Basic the CRT controller on the machine, a new CRT controller we'd not used before, sort of poking its registers directly with Basic. And by the time Hermann and Chris let the people into the lab to see the prototype, there it was-- that high-resolution screen generated by the CRT controller on the BBC computer displaying, I think, at first just a little random line across the screen. But it was clearly doing it, and it was repeatedly drawing it. So the machine was clearly working. The BBC seemed to be rather impressed. We found out later that NewBrain hadn't got that far in the whole year, that the BBC had been talking to.

Fairbairn: What an incredible story. How long was it before you and Steve figured out that you'd been hoodwinked by Hermann in terms of--

Wilson: I think we figured that out on Monday. Not long at all. But kind of too late. So we got the contract for the BBC Microcomputer. A hundred ICs was too many. We had to condense the thing into some custom circuitry, so the video processor part of it, and the __________serial processor part of it we had to make custom circuits for.

Fairbairn: And you used gate arrays for that?

Wilson: We used uncommitted logic arrays. And that was our baptism by fire into the world of designing your own chips.

Fairbairn: So you clearly have a detailed memory of the event. Do you know the specific date the BBC came to your lab and you finished this?

Wilson: Yeah, sometimes in February.

Fairbairn: Of what year?

Wilson: Of 1981. Unless it's 1980. I can remember the time of year better than I can remember the year. Long time ago. [SW: Yes, it was 1981]

Fairbairn: It was cold when you went home to bed.

Wilson: It would be on the Internet somewhere. Some geek will have written it all down.

Fairbairn: So that launched a whole new world in terms of the volume and things that Acorn was about, right?

Wilson: Yes. Steve and I thought the BBC were crazy and that we'd sell 50 thousand of the Proton design, now rechristened the BBC Microcomputer, which we also thought was a little pedestrian, as names went. In the end, we sold one and a quarter million of them.

Fairbairn: Forty? Four-zero?

Wilson: One and a quarter-- 1.25.

Fairbairn: One and a quarter million.

Wilson: One and a quarter million. U.K. is quite small.

Fairbairn: So tell me about the time from that. You launched that machine. Sort of take us quickly to the point where you then had to talk about doing the ARM processor.

Wilson: So that was 1981. We converted a lot of the system range capabilities to make the BBC machine in the first place, and we built a lot of peripherals specifically for the BBC machine. And then we started implementing language processors. So the BBC machine had a socket underneath of it for the second processor, and its two-processor model. I designed the Tube, interconnection between the two things, and wrote all the software protocols that ran the Tube. The Tube was basically a dual-direction DMA controller plus four FIFO channels of communication so that you could virtualize your operating system. So a second processor operating system, all they had to do was send the right commands across the Tube, and the I/O processor machine would do all the right stuff for them. So we got all of that working and we started building second processors. The only one that sold a lot was a 4-megahertz 6502, 2-megahertz 6502 in the base machine. And with these two things linked together, then everything just ran. You put your 4-megahertz 6502 on, and programs were copied across to it, and executed twice as rapidly. And without-- well, more than twice as rapidly if you were doing lots of I/O stuff at the same time, because the other processor would still be working away. And that was quite successful. Some people even wrote games for the dual processor, so there was a very successful game called Elite, which was David Braben and Ian Bell's space trading game, with three-dimensional wire-frame graphics, and that had a version for the second processor as well. So we'd built lots of second processors, and we came up against this thing that they were no better than the 4-megahertz 6502. You could get an 8-megahertz 68000, I think it was in the days, just before 68010 came out; an 8-megahertz 80286, which we built into a machine, and Hermann had this plan to sell an Acorn business machine based on the 80286 running CPM86, multitasking version of CPM. And we'd gotten with National Semiconductor on the chip originally called 16032, and then renamed 32016, and built the same processor in that. And they were all very disappointing. These high-end processors with their clock rates twice the 6502s with their bus widths twice the 6502s were actually, in many circumstances, slower. This was because they didn't use the memory system effectively. On the 6502 at 4 megahertz, each cycle is a memory access. So it accesses 4 megabytes a second. On all these other ones, a memory access is four cycles of the master clock on a 68000. So an 8-megahertz 68000 would only do two bus accesses per microsecond, and they would be two bytes wide. So they only had four megabytes bandwidth, just the same as the 6502. But in building the 6502-based systems, we knew how to make very high-performance memory systems. And if we made the memory system wider, a 16- or 32-bit wide memory system, we could easily design a memory system with 16 to 32 megabyte-per-second memory bandwidth. But we couldn't find a microprocessor that could eat that. We became convinced-- which we think is still true for cacheless microprocessors-- that memory system bandwidth is direct predictor of system performance. You fetch in your instructions, you're doing all your I/O; without a cache, the memory system bandwidth is the system performance. So we knew we could build a faster memory system, but we couldn't find a microprocessor. One of the avenues we followed to get a faster microprocessor was to go back to the developers of our favorite 6502. So Western Design Center was discussing a successor to the 6502, the 65SC816. So Steve and myself and a guy called Andrew McKernan, who ran Acorn's Advanced Research department at the time, flew out to Phoenix to visit Western Design Center and talk to them about their processor. And that wasn't too unusual for us. We had flown out to Israel to talk to National Semiconductor about the NS32016 development, and we saw in Israel kind of what we expected-- massive building on a science park with loads of people working. When we came into Western Design Center, these people who had build our favorite processor, what we found was essentially some bungalows next to each other with ordinary drawing boards in them, and William D. Mensch and some college kids designing processors by sticking bits of sticky tape, Rubylith tape, on transparent things to-- this was less sophisticated than we were.

Fairbairn: What year was this, or when was this you visited Western Design Center?

Wilson: This would be sometime in early 1983. So we came away, well, fairly convinced that, although the 65SC816 was an advance on the 6502, it wasn't what we wanted. We also came away fairly convinced that if they could build a processor, then we sure as hell could. So we started thinking about building computers, processors ourselves. Professor Hopper, Andy Hopper, at that stage saw the very first publications of RISC chips. So he put on my desk <inaudible>Berkeley RISC and Stanford MIPS's first press releases and papers, and about a month later the IBM801 papers were published. So I had them all on my desk, and I looked at them all, and thought, "Hmm, we could do this." So then we spent-- by we, I mean me, Steve and Hermann-- Hermann was an integral part of this, not in terms of providing any of the technology, but in providing somebody for us to talk about. So I started designing an instruction set. It wasn't the first time I'd designed an instruction set. I did one during university years as well, but this one would be more real. Steve started researching how you made these things work, what a microprocessor pipeline is. We had figured out that 6502's sort of semi-pipelined structure was noticeably more efficient than what Motorola did the 6800 and 6809. We already appreciated that things overlapping well really worked for us, and we had a philosophy for ARM that we actually pipelined the memory BUS as well, which was later on removed because people found it too confusing to design with. But it was very good performance gain-- so I mean, Hermann, Chris and I, we-- Hermann, Steve and I would work separately for part of the time and then we'd walk down to lunch from the Acorn building to the nearest pub, talking all the time, and come back and revise things and keep doing it. And we got more and more convinced that we could indeed make this thing. Hermann set up an official project in Acorn to do it, and we had me, Steve, and Hermann gradually acquired four VLSI design engineers, originally led by Robert Heaton, and later on by Jamie Urquhart. And we did it.

Fairbairn: So was this a controversial thing within Acorn, and did-- I mean, did a lot of people think you were crazy, or what was the--?

Wilson: We thought we were crazy. We thought we wouldn't be able to do it. But we kept finding that there was no actual stopping place. It was just a matter of doing the work. We gradually converted Acorn's Advanced Research and Development (AR&D) department into a whole set of people doing this. After we determined that we were going to do this, we decided to build not just one-- we didn't just build a microprocessor. We built an I/O controller, a video controller, and memory controller as well. So we built four chips, of which the ARM microprocessor itself was about half a year ahead of everything else, because it started first. So AR&D, Mike Muller now at ARM, Tudor Brown now at ARM, did the I/O controller, and the video controller; Alistair Thomas, who sadly committed suicide, built the memory controller with Steve. And Steve and I built the microprocessor, and we did all those chips. A complete system.

Fairbairn: So was there sort of a major insight or breakthrough or whatever in doing the processor, or was it just sort of you read the papers, you sort of absorbed things and sort of went through the process? Were there any sort of major aha moments, or points that--

Wilson: There was nothing aha about it. It was just, "You can do this stuff. You just get on and do it." As I say, it was more the reverse. We expected to find a roadblock. We expected to find why National Semiconductor employed all those people in the building in Israel and why they found it so difficult. And we just never did. We also-- we thought it was going to be much harder than it really turned out to be, so we put a lot of effort up front into verification and modeling. So Steve wrote in BBC Basic the first behavioral model of an ARM. We ran test programs on that. So I led a large group of people who wrote test programs to check all the behavior of the instructions. We then also started writing instructions and interpreters so we could write more and more software. The BBC Basic behavioral model of the ARM was quite slow, only a few cycles per second on that. But with pure instruction simulators, we could have things that were running at hundreds of thousands of ARM instructions per second on a 6502 second processor. And we could write a very large amount of software, port BBC Basic to the ARM and everything else, second processor, operating system. And this gave us increasing amounts of confidence. Some of this stuff was working better than anything else we'd ever seen, even though we were interpreting ARM machine code. ARM machine code itself was so high-performance that the result of interpreted ARM machine code was often better than compiled code on the same platform.

Fairbairn: That must have been encouraging.

Wilson: Oh yes, that was just quite encouraging. We sent ARM off to VLSI Technology to be fabricated. It came back on the 26th of April, 1985. We plugged it into the ready and waiting second processor board. The Tube operating system booted up. It ran BBC Basic. We said, "PRINT PIT," and cracked open the bottles of champagne, because everything worked. We didn't realize that things were hard, though at that time I think the NS32016 was into Rev J, and they still hadn't got it right.

Fairbairn: So you basically had a team of less than 10 people-- yourself, Steve, four VLSI designers, and maybe some other people helping as well?

Wilson: Well, we tried to do a lot of things in parallel, so it ended up-- Hugo Tyson, Jez Wills, Jon Thackray, David Seal, myself were all writing verification software, validating software and interpreters. We set up a team of people, led by Lee Smith and Harry Meekings, to write compilers for it. So there were more people, but yes, essentially making the processor itself was a low number of people, and it was a simple processor. There were only 25 thousand transistors in it. Why shouldn't it work the first time? In fact, all the chips that we were building worked the first time. All but one of them went into volume production in their Rev A's.

Fairbairn: Well, that's a tremendous achievement. So this then, very quickly, went into a new BBC computer, or what was the--?

Wilson: We sold ARMs as a second processor development system. There were lots of people eager to get their hands on that amount of compute, particularly people in universities, researchers, and that rapidly produced a lot of software for it. So we had Prolog and LISP. We hadn't got a C compiler out of Acorn, but in short order we got a C compiler and a BCPL compiler from university people. Then when all the chips were qualified, we produced a whole Acorn machine. It didn't have a very good operating system at that stage. The operating system was little more than a scaled up version of the BBC operating system. And we sold that machine, and then we revised the operating system into the operating system RISC OS, which is what it's called today, and that became much more usable. Even on the first machine, we had some world firsts. We had anti-alias typefaces. In RISC OS we had anti-alias typefaces designed from hinted glyphs all computed at one time, and that was five or six years before anybody else would do that. We had much more computer power than they did - and the motivation.

Fairbairn: So how did the ARM transition to becoming an embedded processor, and were you involved in that process, or how did that come about?

Wilson: So lots of things then got very difficult. We were quite successful, and we began to get people wanting the processor themselves. So a little bit of Acorn started producing a PC-based ARM development system, which we called Springboard, so that was a plug-in board for the PC with all our software on it. And that got a lot of people interested. But at the same time, they wanted to sort of keep their distance from Acorn. I mean, Acorn controlled the whole thing, and so we began to get feedback that people who wanted the processor wanted it free of Acorn as well. By about 1988, we had plans for the processor that Acorn couldn't afford, and we began to look around desperately for some way to fix that. It took us three attempts at putting together a business model that would be useful, and we probably would have failed it each time, apart from that on the third time, Apple were one of the people who wanted to use it. So Apple became very supportive of this notion that ARM would be taken outside of Acorn and given its own existence and its own resources, jointly controlled by Apple, VLSI Technology, who'd got very interested, because they were making a business out of selling this stuff, and Acorn. So the three founders-- the three owning companies and the founders of that ARM startup would control the destiny of the processor, and each of these people would be sort of customers for it. That was quite interesting, and so in 1990 ARM spun out of Acorn as a separate company on this basis. And it had some very tough years trying to satisfy people before it could get big enough to spend enough money to make it really possible to satisfy people. And it also had to invent a proper useful business model. So we were enormously lucky in finding Robin Saxby to come and run ARM, and he brought with him what became the ARM business model, that ARM would license the processor, license the design tools, and franchise it generally, rather than building silicon for people, which was sort of how Acorn had envisioned it before. So without that, it wouldn't have survived. But with that, people that were interested in building it could do so at lower risk. So Robin essentially created the ARM ecosystem that exists today out of that business model, out of licensing access to the architecture and the designs, and essentially selling a design. So the next step beyond fabless chip companies is design-less. You only give away the design. You don't even get involved in actually making an SOC at all.

Fairbairn: So how much longer did you stay with ARM after it's been spun out and this?

Wilson: So there was a lot of pressure at that time. It was a very difficult decision for me to make. Steve had left Acorn to go to be a professor in Manchester University. I did a lot of things apart from designing ARM microprocessors, since I wrote a lot of software for Acorn and designed a lot of end-user computers for Acorn as well as designing the processors that went into them. So I stayed with Acorn, because I felt that joining a small chip company wouldn't let me do the things-- obviously Acorn contracted my services back as a consultant to ARM. ARM needed my services, particularly in the very early time until they could actually get it running themselves. So I stayed at Acorn, and I spent all of the '90s doing a whole bunch of things. I wrote a multimedia software system, “Acorn Replay”, essentially the equivalent of Apple's QuickTime. I wrote all of that from scratch. We did a video phone. We still built new computers, and so on. Eventually the Internet sort of got invented along that period, and people got very attracted to the idea that they'd have client computers running over the Internet. And Oracle in particular wanted a company to produce a client company that would be very cheap. You couldn't do that with Windows machines at the time. They were thousand-pound machines, and Oracle wanted one at a tenth of that. You could do that out of Acorn's kit of parts, and we built such a thing. We got involved in a cable TV system, built an entire video-on-demand cable TV system for the local Cambridge cable company-- all sorts of things. And we build some fascinating computers, and we build some fascinating systems on chip. We got involved with Digital Equipment Corp (DEC)to build, firstly, the StrongARM processor, and then a successor to the StrongARM processor for doing multimedia, with the Digital Semiconductor people. We built a chip set called the SA1500 and SA1501. SA1500 is a StrongARM augmented by a media processor. SA1501 is a single-chip I/O system-- memory controller, video controller, I/O chip generally. With those two chips and the memory array, you had a complete set-top box, which was software-programmable. And it could do things like decode two MPEG-1 streams, one picture in picture, all decoded in software. So it was quite a powerful system. When Intel bought DEC-- fell into buying Digital Semiconductor, they didn't know what to do with that project, and effectively killed it. So we did lots of things.

Fairbairn: So you stayed at Acorn for some time, and then tell me about how you transitioned to your current position.

Wilson: Well, if you remember, Acorn owned 40 percent of the ARM shares. This was fine for a long time. Indeed, it was quite good for Acorn, because about the only way a speculator could speculate on ARM's success would be to invest in Acorn shares. It was the only publicly quoted company that you could do that. So for quite a long time, that went along. And then ARM floated. So you had a means of valuing ARM and Acorn's share of ARM at the same time, and that was okay for a while. But then ARM's share price went up a lot. As ARM got more successful, its share price went up and up and up. And there came a time, somewhere around about '96, that the value of Acorn's holding in ARM was worth about Acorn's valuation on the stock market. And then a bit later there came a time when it was worth that, including all the taxes. And then all the proceeds as well. So we knew we were doomed. Morgan Stanley Dean Witter put together a takeover bid to buy Acorn's ARM shares with the difference in value of the ARM shares. So it made an offer to Acorn Shareholders that they would exchange their Acorn shares, advantageously to them, for ARM shares, and it left a couple of million left over for Morgan Stanley. The market valued Acorn itself negatively to its holding of ARM shares. So what did we do about that? I'd been developing another microprocessor. <laughs> So at the time it was code named ALARM – A Long ARM. So in some sense, ARM is a sort of 32 world, 32-bit processor. It has 27 registers. That's near enough 32. Everything about it is 32-ness. What happens when you set out to design a processor where everything about the processor is 64-- 64-bit busses, 64-bit instructions. So that's what I was spending my time thinking about. And obviously we worked with Digital Semiconductor on the media processor that was inside the SA1500, and I thought that had some good ideas and some terrible ideas. And it just wasn't powerful enough anyway. So I was building this processor ALARM, and other people in Acorn found out that I was doing it, and they thought, "Hmm, what could we do with that? Could we start a company with ALARM as its seed, and essentially produce another ARM but at a different capability point?" ALARM was substantially more capable. And they hawked this idea around Silicon Valley venture capitalists, and basically the answer was no, they wouldn't fund a company like ARM with new ideas. They didn't see that there was much return in it for them. After about the twenty-ninth rejection, we found a venture capitalist who was sort of helpful and said, "Look, what you need to be is to set yourself up with a much shorter time period business model," because of course if you're licensing IP like ARM, the time to get a payback is long. "So you've got to take some of those steps out, otherwise venture capitalists just won't like you forever. You've got to be the people who make the system on chip that exploits your IP. You should become a fabless semiconductor company. And I know you've had these ideas about media processors and digital TV and so on, but I don't think that will fly. I think you ought to get into communications. I think you ought to build something on this newfangled standard, ADSL"-- well, it was G.Lite at the time, pre-ADSL-- "and do that." So we adjusted our business plan and we got funding, and ALARM got renamed FirePath, and we found a silicon implementation team out of INMOS/ST, and we found a modem team out of Alcatel, and put together a three-site startup called Element 14. That exploited the FirePath processor to build-- well, we started off saying G.Lite, but FirePath was much more capable than that. When the ADSL 1 standard came out, we said, "Right. It can do that." In fact, it could do six ADSL-1 modems per FirePath, and we put together multi-core chips. We were a multi-core chip company from Day One. And pre-silicon Broadcom bought us. We'd made big enough waves by then. So one funding round, we got bought by Broadcom. Broadcom is now the volume leader by a long way in central office DSL modems with FirePath processors. And I work for a chip company, designing chips. <laughs> What I said I wouldn't do.

Fairbairn: You have two wildly successful microprocessor designs.

Wilson: Yeah. So I can do it twice. Yes, they were both wildly successful.

Fairbairn: We're going to have to wrap things up. So you're currently at Broadcom and continuing to work on similar things? Do you have another microprocessor in your--?

Wilson: Well, FirePath is set up in a different basis to most processors. It's much more flexible and FirePath has been changing more as a result. The old world of microprocessors is sort of a world of binary compatibility. The program I wrote in 1983 as the very first demonstration of what an ARM can do still runs on a Cortex A15 today. It's that building compatible. The binary runs. FirePath isn't built like that. FirePath is only assembly language compatible, and the binary encodings and so on changes out of all recognition each time round. So FirePath is morphing faster than anything I've ever seen, and changing as a processor radically over time.

Fairbairn: So of all these-- and just to wrap things up-- of just tremendous accomplishments throughout your life, what do you look back on as the thing you're most proud of or the thing that you really sort of feel the best about? Is there one moment or thing or time, or maybe it's your career. What comes to mind?

Wilson: Probably all of it. I mean, I've always had the pleasure of seeing things that I design get out there and be used in volume. I mean, back in 2008 when I was first told that they'd shipped 10 billion ARM cores, I was impressed. Now they've shipped over 32 billion and are shipping 2.2 billion a quarter. . Who cares about that anymore? But you still have the satisfaction of seeing your stuff be in things. Things are made to work. I suppose that goes right back to my parents. You make something that works.

Fairbairn: Well, I think that's a great point to end on, making things that work, and in billions and billions of quantities. That's a hell of an accomplishment. So congratulations.

Wilson: It's a bit worrying. I mean, ARM, they're still selling 44 percent of things, ARM7s, that I worked on. So out of those 32 billion, something like 20 billion of them are ARM7s, and to have 20 billion copies of a little bit of your brain running around the world is scary at times.

Fairbairn: All right. Well, thank you very much for your time, and it's been a tremendously interesting and educational, and I look forward to meeting you in person in April. Thank you.

Wilson: I did email Carina. She's not replied yet, about arrangements for coming out in April. So. I expect it'll just take its course.

Fairbairn: All right. We'll talk about it then. All right, thank you Sophie. Drive safely.

Wilson: Thanks, Doug.

END OF INTERVIEW

