SCALD Oral History:
#3 of 3
(Tom McWilliams Alone)

Interviewed by:
Holly Stump

Recorded: February 12, 2008
Mountain View, California

CHM Reference number: X4560.2008

Holly Stump: All right. We're speaking with Tom McWilliams, an inventor of the SCALD language, which contributed to all the electronic products that are designed today. And we're wondering about your early years. Where were you born and did you grow up?

Tom McWilliams: I was born in Kansas City, Missouri. By the time I was about one, we moved to Cincinnati, Ohio. My father was transferred there. I grew up in Cincinnati until about my junior year in high school, when we moved to St. Louis, Missouri. Curt and I didn't know each other then. I finished high school in Webster Groves, Missouri.

Stump: So were either of your parents scientists or engineers or entrepreneurs or teachers?

McWilliams: My father was an engineer. He was a mechanical engineer. He was very strong in math and science.

Stump: Did he help you with your homework? Do you think you got that bent from your father?

McWilliams: Well, my father died when I was about eight. That was a big event in my life, obviously. I was never really great in grade school. In fact, I was always building things. I'd be off building tree houses and go carts and making things at home and I didn't really pay attention in school. So I failed first grade and fourth grade. And that was a very humbling experience. At some point, I got embarrassed enough that I decided it was time to work harder in school. It wasn't until about middle school when the more interesting math and science classes came in that I started to excel. I wasn’t very good at spelling and grammar and those types of things. So my education took off in middle school and high school. In fact, that's when I got interested in computers. In middle school I built a relay computer, really an adding machine with binary adders. I made my own relays -- wound the wires, built the transformers -- and hooked it all up. I made a mechanical flip-flop that I designed with homemade relays. And then, in high school, I designed a 16-bit minicomputer built out of MSI logic. AMD at that time came out with MSI-series TTL for about a dollar per chip. I got cosmetic rejects from Intel of 256-bit DRAM chips -- the early Intel DRAM chips. And I built a machine with 230 chips of MSI TTL logic and memory and a bunch of printed circuit boards that I made in the basement using marking pens and etching acid. It turned out that it wasn't very reliable. It had a lot of wires and solder points, you know. It was hard to keep it running, so I never really could do anything useful with it. But it did work. It got in the local newspaper. And so that was my early days -- starting to design and build computers.

Stump: All right, so both electronics and mechanical or architectural… you had both types of interest at that point.

McWilliams: Yes. Actually, in grade school I built model boats as well. I got drawings of ships and boats and I did a lot of work scaling the drawings and gluing balsa wood together. I built a schooner ship that was this long [gestures – about four feet], with three masts. A cousin of mine had a small sailboat called a Y Flier, and he had the drawings. It was an 18-foot, 2 man sailboat. He gave me copies of all the drawings and I spent a lot of time scaling all the numbers. I think it's what helped me in math -- I spent a lot of time trying to scale all those drawings. And I built a very nice model out of balsa wood. So I was into building things. And I always had a love for boats and for building model boats. I loved stories of the sea and ocean-going stuff. That was all a lot of fun.

Stump: So, did you have any heroes or favorite biographies that you read, anyone that you felt an affinity with when you were a child?

McWilliams: Well, I remember I always loved the Hornblower boats and those sailing stories. There was something in me that was always attracted to the ocean. In fact, when I got older, I bought a couple of boats that I kept on the Bay and in Tahoe. But I've always loved the ocean and sailing.

Stump: Where did you go to college, Tom, and why did you choose that school?

McWilliams: Well, as I said, I went to Webster Groves High School. The counselor there gave me the names of a couple of schools, one of which was Carnegie Mellon University in Pittsburgh. So I applied there. I did very well on the science and the math SATs, but my language skills -- writing, reading, grammar and that stuff -- were never very high. When the Carnegie Mellon people interviewed me, I showed them all this stuff I had done and they admitted me. But I think that, because my verbal SATs were low, I was lucky to get in. I had done a number of projects that were probably helpful. It was so difficult designing and building a computer the hard way -- actually designing and building all the boards with all the chips -- that I took a class at University of Missouri on computer programming. I learned FORTRAN. And while I was there, one of the graduate students wrote a logic simulator in FORTRAN. I thought that was fascinating. I also took a class at Washington University where we programmed the LINC computer -- a DEC PDP-8 designed for laboratory work. It had an interactive graphics display. We programmed that machine in assembly language. So I learned a lot about computer architecture and that sort of stuff. I ended up getting a computer account over at Washington University on their IBM-360, which we programmed with punch cards. I wrote a logic simulator there in FORTRAN. It was horribly inefficient. My simulator was constantly searching for things. As the circuit got bigger, the speed of the simulator went like the square or the cube of the circuit size -- everything got slower and slower. So it was not practical; you couldn't simulate anything large. I had a number of jobs as I went through school; we never had much money because my father had died young and my mother supported four kids. One of my jobs was selling ice cream. I had one of those trucks that drive around the neighborhood, selling ice cream. I can remember that as I was driving around thinking about my logic simulator, I discovered pointers. And I thought, “Ah-hah, I'll just remember and put a pointer to where I need it rather than searching all the time.” So I was excited that I could revolutionize my logic simulator and make it run much more efficiently. When I got to Carnegie Mellon, I met a professor there, David Casasent, in the Electric Engineering Department, who turned out to be very influential in my career. He introduced me to another professor, Gordon Bell, who started the Computer Museum. At that time, Gordon was a professor in the Computer Science Department at Carnegie Mellon. I explained to him what I wanted to do with my simulator. So he gave me the print set for the DEC PDP-8 -- all the engineering drawings showing how to build a PDP-8. And he said, “Go off and simulate this.” And he gave me a computer account (it was actually his account) on the PDP-10 system in the Computer Science Department. At the time, the PDP-10 was one of the best computers in the world. So that year, as a freshman, I spent a lot of my spare time writing a new logic simulator using pointers. I wrote it in BLISS, a language they'd created at Carnegie Mellon that supported a lot of low-level manipulation. It worked so well that I could simulate the PDP-8 computer efficiently. Gordon was quite impressed and he offered me a summer job. So that summer, after my freshman year, 1971 or maybe 1972, I went with Gordon to DEC in Maynard and worked in the Old Mill -- an old wool mill from around the time of the Civil War, Digital's Headquarters. I worked there using DEC's PDP-10. I worked with a group including Alan Kotok and the PDP-10 designers. We went out to dinner together frequently. That summer, I wrote a new generation of my logic simulator. (Actually I took the old version and enhanced it.) And it became the Sage simulator -- the first logic simulator that Digital Equipment used. Over a number of years, DEC ended up growing a whole group to enhance the simulator. They used it as their in-house simulator for simulating their new machines. So Gordon Bell turned out to be quite important in my career. DEC ended up giving me a consulting contract and for the remainder of my time at Carnegie Mellon. I consulted to DEC, working on the logic simulator. That was how I put myself through school. In addition, David Casasent got a contract from Gordon Bell to design a highly pipelined PDP-8 with cache memory. At that time, cache memories were new in computers. So he and I designed a PDP-8 with cache memory and we simulated it with my logic simulator, which I'd enhanced at Carnegie Mellon. That was interesting. Then there was a debate, "Should we build the computer, or should I write a new version of the simulator?" At that time, I had some ideas about how to enhance the logic simulator. We ended up deciding that I should do another generation of simulator, which I did. And then for a Master's project, I designed a PDP-11 out of Intel bit-slice chips. Intel had just come out with the first bit-slice chip -- a two-bit slice of the data path. I designed the machine on a wire-wrap card. It was about 100 chips of logic on a wire-wrap card. And I actually built it and ran it. It was a working PDP-11. That was my Master's project at Carnegie Mellon -- designing a PDP-11, simulating it with my simulator then building it and running it. Like Curt, I qualified for an interview with the Hertz Foundation for a graduate fellowship. My second-round interviewer was Lowell Wood. Lowell is a most amazing guy, a brilliant astrophysicist. At the end of the interview, he offered me a summer job. He said, "Well, would you like to come out and work at Livermore?" So I ended up taking that job. After I got out there, [in the summer of 1975,] I ended up deciding to go to Stanford for my Ph.D. program in Computer Science. Then Lowell said, "Well, why don't you go out and meet Curt Widdoes?" Basically, Lowell introduced the two of us, and Curt and I started talking. That grew into the S-1 Project. We decided to work together and build a large computer using ECL logic and wire-wrap boards. The first generation machine that Curt and I designed consisted of 5300 chips of ECL-10K logic (about 80,000 gates). And while we were designing, we came up with SCALD. Curt and I were doing all these drawings with the SUDS Design System at Stanford. (SUDS stands for the Stanford University Drawing System, written by Dick Helliwell.) We drew graphical, hierarchical block diagrams on the SUDS system to design this large, 36-bit highly pipelined, cache-based supercomputer [the S-1 Mark I]. At the time, people designed electronics by drawing flat schematics. They would draw every pin of every chip. And they would draw lines to show the wires. Then they would make a wire list. To make a wire list, they would highlight each pin in each drawing and type in the name of the signal connected to the pin. It was a completely manual process. You drew everything manually. You made the wire list by hand, and so forth. Well, there were only two of us, Curt and myself. We were designing a machine with 80,000 gates, 5,300 chips of logic. How is this going to happen? Since we were both aspiring computer scientists, we decided to write a set of software programs (the SCALD Design System) that would automatically compute all the low-level details. So we would draw graphical, hierarchical block diagrams and out of the computer would come magnetic tapes with all the detailed instructions to drive Gardner-Denver wire-wrap machines. The first machine we designed consisted of 12 boards, each about 18 inches by 2 feet, containing up to 500 chips per board. All the low-level design details were automated. So that was how Curt and I connected, built the SCALD Design System together, and built the S-1 Mark I. That machine was phenomenal. We had the same thesis advisor, Forest Baskett. I heard that he thought we were a little bit crazy. You know, two graduate students, Curt and Tom, off building these big computers. You know, how are they going to get this thing built, right? But we had our own funding -- we were funded by fellowships from the Fannie and John Hertz Foundation. So we had freedom. I certainly have to thank the Hertz Foundation for giving us that freedom; we never would have gotten our project funded by the University.

Stump: It sounds as though you have a real theme in your life of just following your own technical interests and building tools that then turn out to be more interesting than the mainstream things you were supposed to be working on.

McWilliams: Well, I have a passion for computers that started in junior high. I want to say it's because I'm lazy in a sense -- I hated to do things by hand. I was fascinated by the idea of a computer that you could program once and it would do the tedious things over and over again. Back in junior high school I developed a passion for computers -- building computers, programming computers. And I've just followed my passion. I also found that when I get into things, they always turn out to be much harder than I thought. If you had any idea how hard it was going to be, you never would have started, right? When Curt and I started, we thought we'd write the SCALD software in a month. In fact, it took us about a year. It was about 35,000 lines of code when we were done, just for the software to compile the drawings and get the wire list out. We were working around the clock. We were only allowed to use the PDP-10 computer at the Stanford AI Lab in the early morning. They wouldn't let us on after about eight o'clock. So we'd arrive at the AI Lab at five in the morning and they would kick us off at eight o'clock because we were a non-funded project. They weren't going to let us use any prime-time computer cycles. It was an enormous effort. We had deadlines because we had raised Navy funding for the project. And we had visitors coming out. We needed to get the computer built for a demo. I have to tell one story… I was going to move to Livermore for a year -- I ended up buying a house in Livermore. We were madly working around the clock to get the wire list done. Late on moving day, I went off with my wife to move. When we got to the house, I stayed for dinner, ended up sleeping there, and didn't come back until about five or six o'clock the next morning. That morning, Curt said, "We're hopelessly behind schedule! We lost the whole night!" All because I went off and moved. He was quite unhappy because we lost the whole night. What was I doing, sleeping at home? In fact, Curt developed a technique where he could sleep for just about three minutes at a time and then he could get up and work a little longer. One morning, he had been working all night long and his wife came walking in to our work area at the Stanford Linear Accelerator Center [SLAC]. Curt didn't recognize her. He was so tired that it took a while before he recognized his wife [laughs]. She came walking in bringing breakfast for us. Anyway, it was quite an enormous effort. We had to get the computer built. So we just made this enormous effort. That is part of what gets things done. If you don't make that kind of effort, it just takes too long. We were working around the clock, just working like crazy. We had portable teletypes that we used from home to type in the code. We were using SLAC's big IBM mainframes to run the SCALD code because we needed a lot of memory. [We were using the SLAC machines for free, courtesy of Forest Baskett.] When we finally finished the wire list at SLAC and they saw the bill -- it was for about ten or twelve hours of time on the big mainframes – they told us to please leave. We got our first wire list and they told us not to come back. Then we went off and designed a second-generation of the S-1, the Mark IIA, that had 25,000 ECL chips on 72 boards, each two feet by two feet. For the Mark IIA, there were four designers and another generation of SCALD. But this time we were able to use the S-1 Mark I, the first machine we built. We made it run SUDS and all the other SCALD tools. The S-1 Mark I had 16 megabytes of memory, which was enormous at that time. (A PDP-10 has only one megabyte.) So we used the first machine we built, the biggest computer we could get our hands on, to design the second machine.

Stump: So, Tom, you were at Lawrence Livermore National Laboratory. Starting what year?

McWilliams: Let's see. I guess it was 1975 because it overlapped with Stanford. I graduated from Stanford in 1980. I continued at Livermore until about 1983.

Stump: So what are some of the post-SCALD highlights at Lawrence Livermore Lab?

McWilliams: While we were there, we generated two versions of SCALD -- SCALD I and SCALD II. Since SCALD was funded by the Navy, it was ultimately released into the public domain. Over 150 companies and people around the country requested a copy. If you wanted a copy of the tools, all you needed to do was to send in a magnetic tape and a letter. The people at the Lab would then make a copy and send it to you. So Daisy Systems, Mentor Graphics, Valid Logic Systems and others got copies of the tools, including all the source code. Valid Logic was founded in 1981 to commercialize the SCALD tools and ended up being a phenomenal success. At the Lab, we built two generations of the S-1 machines, the Mark I and the Mark IIA. The first Mark IIA was partially running when I left and several copies were finished after I left, but they were never production machines. They were prototypes, machines built to demonstrate a concept. It was interesting that the design tools ended up being much more valuable than the machines. I think the S-1 experience provides some useful insight into the process of innovation. Curt and I were hardware designers and we wrote design tools for our own use. We did not write the tools for other people to use. That wasn't the goal. The goal was to get a computer built. We needed a way of getting it done. We didn't have an army of people to do it manually. So we automated all the low-level details. Since we were hardware designers, we knew what we wanted. What happens if you get somebody else to build a tool for you, assuming that they're not a user of the tool, is that they don't know what is really needed. So our tools evolved to be exactly what we needed to get the job done -- to get the computer built. We didn't put anything in the tools that we didn't need. The effort was very focused and very intense. We were the users, so we could change the tools to be exactly what we needed. I think that's the best way to innovate: An end user who is an expert in a field goes off and writes a tool to automate what he is doing. Since he is a user, he has insight about what is really needed -- the market requirements. One of the great things about the S-1 Project was the amount of innovation. It turns out that most of the project consisted of software people writing software for the machine. And that drove innovation in software. The project was staffed by a large number of faculty and students from the Stanford Computer Science Department. The MIPS compilers came out of the U-Code compilers built for S-1. The MIPS compilers became the SGI compilers. The innovations in hardware architecture that we developed are frequently referenced in patents and computing literature. A lot of innovative work was done there. So it was a really good research project. A number of Ph.D.'s came out of it, and a lot of people got trained.

Stump: It's interesting what you say about innovation and being a user and having your own market research just driven internally. You're going to come up with something that's discontinuous, a completely new idea. Do you think things changed in that regard in EDA from the early days?

McWilliams: Well, I think they did. What happens once you become a commercial operation is that you end up with a large staff of programmers implementing the tools and a marketing organization that talks to customers. Somehow, it's no longer the end users who are designing the tools. After the EDA companies started, some of the end users continued to write their own tools. But the tools became so large and sophisticated that eventually the companies couldn't afford to have end users write them. It's harder to innovate when you've got a large group of programmers who aren't users, and a marketing organization trying to sort out the user requirements. It turns out that Valid completely rewrote the SCALD tools that we built. Valid threw out almost all the code and rewrote it. We didn't write the tools for general use. We wrote the tools specifically for building a machine using ECL technology and wire-wrapped boards. In certain ways, our system was more automated than what Valid implemented. For example, we automatically assigned the cables and interconnect wires. We automated a tremendous amount of stuff, but much of it was technology-specific.

Stump: So, Valid started up in 1981. You commercialized some of the SCALD technology and it seems as though, from the early days, you were interested in both EDA tools and building computers. After Valid Logic, did you return to your love of building computers?

McWilliams: Yes. Actually, the next company I started after Valid Logic was Key Computer. I'd heard a talk by John Cocke at IBM. Cocke was building super-scalar supercomputers. I thought, “Well, that's a great way of doing things.” I was still in love with designing computers. So we started a company to build a super-scalar supercomputer using ECL gate-arrays. It seemed to me, as a computer architect, that Cocke’s idea, executing multiple instructions in parallel, was a great way of doing things. This was not widely done in computers at that time. So I decided to start a second company to build a super-scalar supercomputer using the new high-density ECL gate-arrays. We got the company funded and we used the SCALD design tools from Valid to help us design the machine. The company was acquired by Amdahl about 18 months after we got it funded. It turns out that we were going to need a lot more money than we had raised from the venture community. We were looking for corporate partners and we were talking to a number of different companies. One of them was Amdahl. Amdahl decided they didn't want to invest, but they wanted to buy the company. So that was what happened with Key Computer.

Stump: How did you go about gathering the team for Key Computer? Who was working with you at that point?

McWilliams: Well, Jeff Rubin. He started working with Curt and me back at Stanford, and he worked with us through the S-1 Project as a key contributor in both design tools and hardware design. And he was a co-founder of Valid Logic. So Jeff and I started Key Computer together. And then we started recruiting some of the people that we'd known, some of the old S-1 people and others, and pulling a team together.

Stump: All right, so after Key Computer, what was your next adventure?

McWilliams: Well, by the way, at Key Computer I learned something about how difficult it is to have a successful outcome where everybody makes money. We had a successful outcome because the company was acquired, but you never know. You can't predict what is going to happen. We could have failed to raise money. The company could have gone bankrupt. So, I looked around and said, "Well, gee, this is kind of out of my control." It was one of those interesting things in life -- no matter how hard you work or how smart you are, the outcome for a lot of these companies is beyond your control. I ended up spending some time at Amdahl; I was in no hurry to do another startup. I spent a few years at Amdahl, a few years at Silicon Graphics, and then five years at Sun Labs, where I had my own research group. That was a wonderful place to be. A research group is one of the nicest places to work. It's a shame they're disappearing. Companies have pressure on corporate profits so they are really squeezing their research labs. So I took some time off from start-ups. Then, in 2001, I decided it was time to start another company. Sun was not doing well, as you know. The internet bubble had burst and Sun's stock was coming down. My intuition told me it was time to go start another company. I have to say, I spent a lot of time contemplating this. Basically, I rebuilt my confidence. There's this thing inside us called intuition, which we all know about. But you need to listen to it. And my intuition told me it was time to start another company. So I did that. I left Sun. It turns out that I got seed funding right after 9/11, right after the terrorist-attacks on New York City. It was a terrible time to raise money, but I managed to raise $1.5 dollars, which funded the company for a year. Our business plan evolved and we ultimately raised over $12.5 million. Initially, we were going to provide low-power servers using some of the advanced technology that Broadcom had developed -- low-power processor chips. But it turned out that those chips didn't really do what we expected them to do. When we actually started running code on them, their performance wasn't what we thought it would be. But in the process of doing that, we decided that if you're building clusters, the interconnect is the key -- it's not the servers. And so we changed the company's plan to build a very low latency, high-bandwidth interconnect. And we re-named the company PathScale. PathScale ended up developing a revolutionary technology that lets you communicate user-mode to user-mode between two servers in a cluster in 1.29 microseconds. And it lets you send up to 10 million messages a second. So you could send a lot of small messages through Infiniband switches. It was really a revolutionary technology for interconnect. We built that product and it was successful. The company ended up being acquired by QLogic. We had a lot of market interest. In particular, there was a large university in Germany that had a government grant to build a 30-million-Euro cluster. They specified that it had to have the PathScale interconnect, which meant that companies like IBM, in order to bid on that big procurement, had to incorporate our technology into their product line. In fact, that is one of the reasons why QLogic wanted to buy it -- because companies like IBM needed it. QLogic wanted to OEM the technology and build it into their product line. So PathScale ended up being acquired and it's still there. It's still selling and shipping that interconnect. I ended up taking a year off. And that was a good thing; I decided it was a lot more fun to work than to stay at home. By the end of the year, I decided to start another company, Schooner Information Technology, which got funded last year. And it's using the PathScale interconnect. We're building cluster applications for internet data centers and financial service centers. It's a very exciting company, still early stage. We haven't announced the product. It's still in Stealth Mode. I can't talk much about it, but that's what I'm up to now.

Stump: What were the key takeaways from your experience at PathScale, because there was some serendipity.

McWilliams: Well, one of the interesting things is that after we changed our direction, I went and talked to our investors, including Charles River. Charles River said that they had looked back at their most successful companies and most of them actually changed course along the way. My conclusion was that it's like a football game. You're the quarterback and somebody gets loose in the field. There's suddenly an opening in the field. There's a receiver who got free and you see it and you toss the ball to him and he makes a touchdown and wins the game. In business it's the same. You need to keep your eyes open and see what the opportunities are. If you see a much better opportunity than what you currently have, you should go for it. Basically, you've got to stay flexible and listen to the customers and build what the market really needs. And that's part of how you build a successful company. It's hard to know what to build before you get a team together. So, first you get enough money and people together to have a critical mass, then you can go out and talk with the customers and start working with the technology, trying to do things. It was the same with the S-1 Project -- we wanted to build a computer, but the most interesting thing turned out to be the design tools we had to build along the way. PathScale was the same -- we wanted to build clusters, but the most interesting thing was how you connect them, not which processors you use. So, you get in there and try to solve a problem, and when you get a bunch of smart people together working on the problem a better idea often comes out of it. It's not always what you set out to do that turns out to be the best idea.

Stump: So do you measure any part of your success based on how many people have benefited from the technologies that your companies have created, the fan-out?

McWilliams: Yes. I think that's a good measure of success. In fact, I've been amazed by how many people I have come across over the years who used the Valid products. And SCALD had a tremendous influence on the industry. Daisy, Mentor, and Valid all built hierarchical systems. Interestingly, the Daisy founders tried to recruit me, Curt and Jeff as co-founders before Valid started. They had heard about our research and read the papers that we'd written. Lots of people were interested in that research and many of the CAE companies got copies of the SCALD software. So, I think that the work we did on SCALD had a profound impact on the industry. And the interesting thing is that it wasn't what we set off set off to do. We wanted to build computers. I think that's often the way that innovation happens.

Stump: Well, Tom, if you could change or redo some decision or event in your career, is there any one thing that stands out?

McWilliams: Not really. I've become a believer in a saying that comes out of Yoga: Everything happens for the best. I think that if you look back, even the things you wish hadn't happened – there are lessons there. And a lot of times the biggest lessons are the things that didn't work right. That's where a lot of insights come from. You try to do something and it doesn't work right. But then it turns out that it did work right in terms of helping you innovate.

Stump: Is there's anything I've forgotten to ask you?

McWilliams: No, I don't think so.

Stump: Thanks, Tom.

McWilliams: Thanks.

END OF INTERVIEW

