SCALD Oral History:
#1 of 3
(Tom McWilliams and Curt Widdoes Together)

Interviewed by:
Holly Stump

Recorded: February 12, 2008
Mountain View, California

CHM Reference number: X4429.2008

Holly Stump: So Curt, Tom, tell us the story of the S-1 Project at Lawrence Livermore Laboratory.

Curt Widdoes: That's an interesting story. Tom, do you want to start?

Tom McWilliams: No, go ahead.

[For an overview of the S-1 project, the S-1 uniprocessor and multiprocessor, and the SCALD design system see "The S-1 Project: developing high-performance digital computers", *Energy and Technology Review*, Lawrence Livermore National Laboratory, September, 1979, pp. 1-15. Also, see "The S-1 Project FY 1979 Annual Report", September 30, 1979, Lawrence Livermore National Laboratory, UCID-18619. The S-1 annual reports from 1976, 1977, 1978 and 1979 are all archived at the Computer History Museum.]

Widdoes: So let's see. Really, it got started with Lowell Wood. He was an interviewer for the Fannie and John Hertz Foundation, a brilliant physicist working with Edward Teller at Lawrence Livermore National Laboratory. He recruited both of us for summer jobs at Lawrence Livermore, at different times. I went there in 1973 and Tom went there in 1975. I was doing some other things in Stanford's Computer Science Ph.D. Program in 1975, when Lowell contacted me and said, "I want to build a supercomputer. Tom McWilliams has signed up. Would you like to join?" and I said "Yeah." So that's how it got started.

Stump: And you knew each other previously? How did you meet? Do you remember the day you met?

McWilliams: Lowell had interviewed both of us and he suggested that we meet, so, as I recall, I went over to the Stanford AI Lab [SAIL], and Curt and I met.

Widdoes: I believe that's true. I was working on the design and construction of a TTL, microprocessor-based, multiprocessor system called Minerva. I had my stuff there at the Stanford AI Lab. I had hundreds of SUDS [Stanford University Drawing System] drawings of the Minerva system, so I was an expert at using SUDS at that point. [SUDS was a state-of-the-art graphics editor for doing traditional, flat hardware design, developed by Richard Helliwell]. At SAIL, SUDS ran on the DEC KA10 and III vector-based graphics displays.] When Tom came over, it seemed natural to go off and build a supercomputer for Lowell Wood.

McWilliams: Right.

Widdoes: So we started in the summer of '75 when Tom started in the Stanford Computer Science Ph.D. program.

McWilliams: That's right. Basically, Lowell was an astrophysicist who was very interested in using computers. In those days, the fastest scientific computers were the CDC 7600s at Livermore. I can't remember whether Livermore had the CDC Star-100 yet or not, but it turned out not to be a very good computer. In any case, the CDC 7600 computers were limited in number and very expensive. So what we wanted to do was to use modern semiconductor technology -- a higher level of integration -- to build much more cost-effective computers. Lowell was interested in that. As I recall, the deal was that if Curt and I could design these computers, Lowell would get the money to build them. That started a multi-year adventure together. Lowell spent many years working to keep us funded -- going back and forth to Washington. We made a number of trips with him, visiting Washington to raise money. In fact, from 1975 through 1979, we raised over seven million dollars from the government to build these computers. [The funding level for the S-1 Project after 1979 was much higher.] So Lowell ended up working with us, helping to get it funded, providing lots of support. And we were off building these computer systems.

Widdoes: So in ’75 the mission was to build a supercomputer that would be faster than the CDC 7600. The Cray-1 was on the drawing boards at that point, but it wasn’t available yet. We later matched the Cray’s performance, but the original goal was to beat the 7600 with a cost-effective machine. We decided to build a highly pipelined, ECL-10K based uniprocessor with novel, 2-bit branch prediction and separate data and instruction caches, to replicate it in a 16-processor system, and to connect the processors to memory using a crossbar switch. [The processor caches were private. The S-1 was the first computer to implement cache modes (e.g., read-through, write-allocate, write-through and read-only) using tags in the page-table and in the translation look-aside buffer, the first to use a directory-based cache coherency mechanism, and the first to use load-linked and store-conditional for mutual exclusion.] That was the mission. So we started designing there at SAIL. Did we start with the DEC KA10 or was it a KL10?

McWilliams: I can’t remember.

Widdoes: Anyway we used the SAIL computer. I think it was the KA, which ran the Stanford University Drawing System. SUDS was a wonderful graphics editor. It was the program that DEC used to design the KL10.

McWilliams: That’s right. DEC used the SUDS drawing system. In fact, they hired Richard Helliwell to go back and work at DEC. They used the SUDS system to design the next generation PDP-10, the KL10.

Widdoes: Right. Dick Helliwell was the author of the SUDS software. SUDS was actually part of the initial SCALD system. We just used the existing SUDS drawing system and built other stuff around it.

McWilliams: Right. SUDS was our graphics editor. We used it as the entry vehicle, so we could do our drawings. Then we wrote our own programs to process the SUDS output. SUDS had its own design system. It could produce wire lists, but it was intended to be used for conventional, flat design. You had to draw every chip with every pin in SUDS, just like you would in a conventional hand-drawn schematic. So, basically, it did on the computer what people were doing at that time by hand, on paper. Then, it would automatically produce a wire list. But we were designing this large computer, you know, with 80,000 gates. The first one, the S-1 Mark I, consisted of 5,300 ECL-10K chips. No way were we going to draw all those wires and stuff. So, we basically used the design method that, as a designer, you wanted to use -- hierarchical block diagrams. As a designer, you think of block diagrams and data paths. Well, we said, “Why do we need to draw it all out? We could automate the expansion of the details, right?” Whereas, with the SUDS system you still had to draw it all out, all flat, we said “Let’s automate that and come up with a very concise graphical representation of the design, optimized for ease of entry and ease of understanding. Then, let’s automate all the low-level details.”

Widdoes: Now, importantly, our primary mission was to build a supercomputer. We built SCALD out of necessity, because there were only two of us. We quickly realized that in order to build a large computer we needed advanced design tools, and we needed to use the kinds of techniques that were already being used in programming languages. Programs at that point were structured. Pascal was an important modern programming language at that time. During the fifteen years leading up to 1975, programmers had completely converted to structured programming, yet logic design was still being done just like assembly language programming, with no structure at all. You drew every gate. You drew every wire. You didn't reuse modules. The interfaces were not isolated and not well defined. When you got to more than a few hundred chips, it became a huge job to keep track of all of the low-level details. So, we said, "Look, we're going to design like people write programs. We're going to design in a hierarchical fashion, where modules are defined in terms of a few other modules, the interfaces are well specified, things are reused, and it's hierarchical." So we set off to build just those capabilities that we needed to do structured design of the supercomputer. And it turned out amazingly well. The technique of structured logic design turned out to be revolutionary. Of course, all big designs today are done in a structured way. We were on the cusp of a revolution. Something would have happened with or without SCALD, because the forces were so strong, but I believe that we did the first design of a real computer system using structured design tools. [For an overview and comparison of flat, structured and hierarchical logic design methodologies, including SCALD, see *CAE: A Survey of Standards, Trends, and Tools*, Stephan A. Ohr, John Wiley & Sons, 1990, pp. 85-103.]

Stump: When you compare how design was done pre-1975 versus the SCALD approach, what are the practical consequences of the increased hierarchy and level of abstraction and structure and so forth if you're looking at accelerating the design process?

Widdoes: Well it's huge. Today, could you design a new Intel microprocessor chip without using a high-level design language like Verilog? No, you couldn't. It's just too complicated. You get the same advantages for logic design as you get using high-level programming languages for program design -- things take less time, there are less errors, things are more reusable -- all the advantages of structured design. You attempt bigger projects, of course, and you always push your tools to the max, so the risk is about the same. We probably would have built a little tiny computer if we hadn't developed SCALD, because that's all we could have drawn in the old way. So, anyway, there are a lot of interesting stories associated with the S-1 and SCALD. One is that we were guests up at the Stanford AI Lab. Lowell had talked to John McCarthy, who founded and ran the Stanford AI Lab, and he had gotten us permission to work up there surreptitiously. We were allowed to work on the SAIL machine from roughly 5 o'clock in the morning until the real users arrived. So, we were up before 5 o'clock every morning, six days a week, and we would use about three hours to edit all of our hierarchical drawings in the SCALD language. When the real users arrived and we couldn't use the machine anymore, we'd make hardcopy of everything (300 to 400 drawings) and we'd spend the rest of the day marking edits -- design changes and stuff -- and enter those the next morning. This was all done without any underlying SCALD tools; there was just an idea of what the language was like, and we designed in that language. Later on, we wrote the SCALD tools.

Stump: So how many hours of sleep do you think you averaged during those days of the S-1 design?

Widdoes: During the design period, I think I would get probably seven hours a night; you can't survive long-term on too little sleep. It's just that we spent all the time either sleeping or working. We got less sleep than that during the period when we were pushing to get the wire-list out. How much sleep did you get, Tom?

McWilliams: I don't know. It was probably like that. Basically, what it did was push you to go to bed early because you had to get up really early because we weren't allowed to use the computer during the day. We were not a funded project. We weren't paying for the computers. We were using the time that the other projects didn't use, the idle time on the machines in the early morning.

Stump: So it sounds like you had great support from Lowell Wood and from John McCarthy but it was still outside the mainstream -- early mornings and scrounging for funding. Any other obstacles or personal sacrifices?

Widdoes: The S-1 Project was out of the mainstream. I think nobody believed we could do it. They didn't want to take the risk of calling it mainstream because they didn't figure we'd finish. But we got support from a lot of people. Forest Baskett was very supportive. He was Ph.D. thesis advisor for both of us. His attitude was "I'll let them do what they want and see where they go." He didn't actively tell us to work on other things.

Stump: It is interesting that you mentioned that the funding was from the government or the Navy. That seems as though it may have had some far-reaching implications for the SCALD language and how it proliferated, possibly for good or for evil.

McWilliams: Since it wasn't classified, they just put a restriction on it that it couldn't be sent overseas. They classified it under what's called the ITARS, so you couldn't export it without an export license -- that's what ITARS actually means -- you had to go through Commerce Department. This is the case with a lot of technology. The Livermore Lab, of course, was funded by the government. And Lowell had contacts in the Navy. The Navy does a lot of research support. We were able to start small -- we got a small grant of hundreds of thousands of dollars -- and then, over time, as we made progress, we got more money from the Navy. So the Navy became the vehicle with which the government funded our project. But it was a lot of work. We spent many trips going back to Washington, briefing different people on this and that. We found that when you go from one administration to the next they change a lot of the staff. Sometimes it's hard to keep a project funded through those transitions because the new administration doesn't like the old administration. But we managed to get through that. One person who was instrumental was Edward Teller. Lowell Wood worked very closely with Edward Teller, and there were various times when Teller went to bat for us in Washington and argued for our project. So he helped keep us funded, but it was a struggle over the years. How do you keep these people building these computers in a national lab? How do you keep the funding going? It was a lot of work.

Stump: So how did you feel, being primarily technologists, when here you were on the road raising money at this early point in your life?

Widdoes: That was an early education about what you have to do to innovate in technology. And they were fun times, our trips with Lowell Wood. We would go to brief admirals on our work. We'd take a bunch of color viewgraphs and go explain to the admirals how we were going to fit the S-1 Mark I through the cargo door of an Orion P3 aircraft to look for submarines. It was fun. Lowell is such an interesting guy. And there was this creative tension -- we would build good technology, so he would have to raise more money, so we'd have to build bigger, better things, then he'd have to raise more money. It was a question of who would run out of gas first. But Lowell really exhausted himself. He did his physics and, as well as that, raised money for us by traveling back and forth to Washington. I remember one particular meeting with him. We were doing a briefing for an admiral someplace and Lowell was really tired. When he got really tired he would sometimes momentarily fall asleep. We were talking to the admiral, and Lowell was sitting there, and you could tell that he was starting to fall asleep. Of course, we just kept talking, hoping the admiral wouldn't look at him, when we saw Lowell's eyes -- they stayed open but they rolled up and he was asleep. All you could see were the whites of his eyes.

Stump: So what was the dynamic like between the two of you, I mean here you are working on this immense project, trying to do something no one's ever done before, kind of on the side as a sub-objective to your real objective. What personality traits did you have that worked together synergistically?

Widdoes: We worked hard. That was our personality trait.

McWilliams: Yeah. Well, one thing I'd say is that Curt's very organized and methodical, and that was a real asset in getting things done. When we started on the S-1 Mark I design, Curt designed the execution box and I designed the instruction box. We each worked on our own unit. At some point, we stopped working separately and we helped each other. I found that that was critically important because, for me, working alone is much harder. I found that a team design, where two people sit down together, working, figuring out all the problems together, is much more effective. You need somebody to sit there and talk with you and work through each decision; it drives the process. So I found that that was a very effective technique -- when we stopped working separately on our own units and sat down to work through the design together. I've always found that that's very effective for me, and I found that we worked very well together and got things done.

Widdoes: Certainly the period of those years working with Tom was one of the most productive periods in my life. We started in '75. The first year we spent mostly working on drawings. Sometime late that first year, maybe it was just after the summer of '76, we actually started writing the SCALD software.

Stump: Can you take us through an overview of the different pieces or the flow of the SCALD software?

Widdoes: Sure.

McWilliams: Well, for design entry we used the SUDS system, just as a schematic editor, like you'd use a text editor, but for editing drawings. Then, we used a Compiler, which is one of the areas that I worked on, that took the output of the drawing system and it interpreted it and expanded the hierarchy down to the interconnect. That fed into what we called the Packager, which Curt worked on. I don't know if you want to talk about the Packager?

Widdoes: When you do hierarchical design and you don't express all the details in the drawings, you absolutely have to have automatic physical design. No person can go in and figure out all those physical details; they aren't even necessarily represented in the drawings. So we had to have a complete physical design system to fill in all the physical details (the Packager). It read in the output of the expansion and a representation of the placement of the chips (which eventually was done automatically -- that was written by Jeff Rubin), and it figured out where all the wires went, what types of wires to use and so forth, analyzed the signal waveforms, and wrote out tapes to drive fabrication.

McWilliams: The S-1 Mark I was implemented on 12 boards, each about 2 feet by 18 inches, carrying up to 500 ECL-10K chips per board. It had literally hundreds of cables that went between the various boards -- different places on different boards. They were flexible cables. Initially, we were trying to design the cables by hand, figuring out where all the signals would go in the cables. We were doing it manually because Curt said it wasn't possible to automate that -- it was too complicated. So there we were, sitting there doing it by hand, and Curt looked at how lousy a job we were doing by hand, because who could figure this out? We were sort of randomly assigning signals. He said, "Well, if that's what we're going to do, we can do it automatically at least as well as we're doing it by hand." That was an example of the tight integration between the people writing the design software and the people designing the computer. (They were the same people.) If you had an army of people, you'd just let them go at it, but if you're doing it yourself, you say: "I can write a program instead. This is terrible, trying to do this by hand. You change something and everything else needs to be changed." So there's a whole learning experience we went through: How do you control the placement? What goes where? What do you do manually? What do you automate? And we were just trying to figure out how could we get this thing built. We looked at how difficult it was to do things manually when you've got something so big, so many hundreds of thousands of wires, and cables, and twisted pairs, and different terminations. So we kept adding more capability to the software to automate more and more of this. The Valid software, developed by Valid Logic Systems beginning in 1981, didn't implement as much automation for this particular technology -- for ECL wire-wrap. It was focused on ASICs and PC boards. And most of Valid's customers were doing smaller designs -- they were doing gate-arrays. But we ended up automating an awful lot of stuff. It was really educational to see how that happened when you tried to do it manually. A couple of other programs got written: We wrote a logic simulator. We wrote a placement program. And I wrote a timing verifier. At some point along the way, I decided that it was time to get out of school. Since, I would say, high school, but certainly since undergraduate school, I've always been worried about timing. How do you get the timing right – with all the delays. So, I decided to go ahead and work more on this timing verification problem, the problem I had worked on in earlier years but hadn't made much progress, hadn't really solved the problem. I came up with this technique that is sort of like symbolic simulation. It is similar to traditional simulation, but rather than just values of 0 and 1 in the wires, you also have stable and changing values -- you just know whether a wire is changing or it's stable -- and you simulate with this extended value set. And that program, the SCALD Timing Verifier, allowed us to automatically verify the timing of the design. I believe that was for the second machine, the S-1 Mark IIA. I don't think we had it for the first machine (the S-1 Mark I).

Widdoes: That's right. SCALD I was what we used to design the Mark I. We called the system we used for the Mark IIA "SCALD II". Tom added timing verification in SCALD II. Similarly, Jeff Rubin added simulation in SCALD II.

McWilliams: Right. Definitely a part of that was that I decided it was time to get out of school. I was into my fifth year at Stanford and I needed to pick something that I was going to focus on for my thesis. So, the SCALD Timing Verifier turned into my Ph.D. thesis ["Verification of timing constraints on large digital systems", Thomas M. McWilliams, Stanford University Computer Science Department Ph.D. thesis, May, 1980 – reprinted as Lawrence Livermore National Laboratory UCRL-52995]. I got that done, graduated, and we got the machine [the S-1 Mark IIA] built.

Widdoes: Here is my school story: For SCALD I, we had to implement a physical design system. We thought it would be really easy, but it turned out that there was more to physical design than we had anticipated. It took a lot longer to implement than we expected. After we built the Mark I, we realized that even more was necessary for the Mark IIA, which was going to be a much bigger machine. In fact, there were nearly fatal problems in the way that we packaged the Mark I. It was a wire-wrap machine. You have to use twisted pairs for ECL when the wires get long. Twisted pairs are bulky, and they tended to stack up in a mat on the back of the board. The people who wired the boards would use a fully automatic Gardner-Denver machine most of the time, but sometimes they would use a hand tool. When they used a hand tool, sometimes a little bit of wire would fall off and get stuck in the wire mat. Of course, in such a thick mat, you can't find the little bit of wire. Also, the wires would pull around the sharp edges of the pins and the insulation would get nicked. Sometime you'd get contact there. When we shook the boards, sometimes little bits of wires would fall out and sometimes shorts would occur. So, we realized that a better physical design system was necessary for Mark IIA. We needed much better optimization of all the physical stuff. Since I had implemented the Mark I physical design system, that became my assignment and I cranked through it. It took a long time, and I wanted to get out of school, so I made that my Ph.D. thesis -- the algorithms for the packaging of the S-1 Mark IIA ["Automatic physical design of large wire-wrap digital systems", L. Curtis Widdoes, Jr., Stanford University Computer Science Department Ph.D. thesis, December, 1980 – reprinted as Lawrence Livermore National Laboratory UCRL-53118]. The SCALD II Packager turned out to be much more difficult than I thought when I started.

Stump: Well, there had to be many skeptics, or dinosaurs, at Lawrence Livermore who weren't sure that you two were going to pull it off. What was the reaction when you finished the S-1 and it worked? Did you feel any triumph?

Widdoes: Well, first of all, it wasn't just the two of us. By the time they built the Mark IIA, it was a huge project. How many people?

McWilliams: I'd say over 20 -- 20 to 30 people working on the Mark IIA. Just building one machine… See, one Mark IIA had 72 boards, which were 2' by 2'. The machine would literally fill up most of this conference room. I mean, it was an enormous machine, physically, and to build even one machine we had a whole group of technicians. In fact, it was all done in trailers at Livermore. At Livermore it was hard to get real buildings, but you could lease trailers easily. So, we had this whole park of trailers, and we were all working in trailers. In fact, they even built a data center in trailers. It looked pretty good. It had a false floor. It was quite a project. We had a lot of people -- a bunch of people out of Stanford, plus a bunch of full-time staff at that point. It was all externally funded through the Navy. Most of the Livermore Lab, the Computation Department and so forth, was separate. We were really a separate group in the Physics Department -- what's called the Special Studies Group ["O Group"] in the Physics Department -- with external funding. [Lowell Wood's O Group at Lawrence Livermore National Laboratory, including the S-1 Project, is the subject of the book *Star Warriors*, William Broad, Touchstone, 1986.]

Widdoes: Now, if you look at the design of the Mark IIA, it was done with basically four people right?

McWilliams: That's right, four designers in two-and-a-half years. In fact it took the same number of man-hours per chip for the Mark IIA as for the Mark I. For the Mark I, Curt and I did the logic design.

Widdoes: We did it in two man-years.

McWilliams: Then Mike Farmwald and Jeff Rubin joined in. Jeff helped us debug the Mark I. Mike Farmwald wrote microcode for it

Widdoes: Right.

McWilliams: Then for the Mark IIA, Jeff and Mike and another guy, Bill Bryson, and I worked on logic design. Curt worked on the SCALD tools for the Mark IIA.

Widdoes: It might be helpful if I gave a timeline, so people who see this could understand better how things fit together. Tom and I got together in 1975 and Lowell started up the project. For the first two years we worked on the design of the S-1 Mark I and the SCALD I software. That was basically done by the middle of 1977. SCALD I involved the macro expansion and the SCALD I physical design system that allowed us to build the Mark I wire-wrap boards. No timing verification. No simulation. SUDS was the graphics editor. The Mark I was a 5300 ECL-10K chip machine on twelve 500-chip wire-wrap boards. It took two elapsed years to design that. During that time, the drawing was done at the Stanford AI Lab and the back-end processing was done at SLAC, using the SCALD software. We'd get done entering the changes to our drawings, write out a magnetic tape and drive over to SLAC. Then, during the day, we'd process everything at SLAC and we'd get all the errors figured out. Early the next morning, we'd go back to SAIL and fix the drawings. In the fall of 1977, SCALD I was basically finished and the Mark I was out to fabrication. The next year (mid '78), we published the SCALD papers in the Proceedings of the 15th Annual Design Automation Conference. Those were important papers because, before that time, essentially all design automation papers were about flat design. We did two seminal papers about structured logic design using the SCALD system and we published them at that conference. We subsequently received the 1984 IEEE McDowell Award for introducing structured computer-aided logic design to the industry. That was the middle of '78. By the middle of '78 the Mark I was running and the Mark IIA design was underway. We continued from '78 for the next couple of years enhancing the SCALD system for the Mark IIA. . [For a more complete timeline of SCALD development, see "Automatic physical design of large wire-wrap digital systems", L. Curtis Widdoes, Jr., December, 1980, Stanford University Computer Science Department Ph.D. thesis, pp. 18-24 – reprinted as Lawrence Livermore National Laboratory UCRL-53118.]

Stump: What part of the flow for SCALD I, was the most difficult and what would you have done differently in retrospect? What was the most challenging technology in SCALD I?

Widdoes: Well, I think we should have done simulation of the Mark I. A key guy that worked with us later, Jeff Rubin… By the way, Jeff Rubin is an amazing guy. He writes software like Glenn Gould played the piano… they're both just unbelievable. Anyway, Jeff Rubin's statement was that whatever you don't simulate doesn't work. Even if you're designing in a hierarchical way, you still put logic mistakes into your design. The structure helps, but it doesn't do everything. So, it took a while to get the Mark I running. We had various logic errors that we had to find, and we had to rewire things. So, if I were to change something in the SCALD I flow, we should have simulated the Mark I. Probably, it would have shortened the whole development cycle.

Stump: It is interesting that your subsequent companies that you founded after Valid were in the area of verification. Some scar on a synapse there.

Widdoes: No matter what language you design in, there will always be logic problems that you need to worry about. We advanced the language end of things a lot with SCALD I, but you still need other tools to help you do the verification.

Stump: Were you aware of other people developing similar technology at that time?

McWilliams: Not at that time. What was important about the S-1 Mark I… It was such an unbelievable thing. I remember that we gave talks… Curt and I gave talks at DEC and other places, particularly DEC. We gave talks about the SCALD design system. People couldn't believe it. It was such a radical difference when you look at what the two of us did in designing that machine versus commercial design teams with large groups of people working for long periods, doing it manually, drawing everything out and so forth. The productivity difference was so striking that it was really revolutionary. In fact, one of the things that happened when Valid came along was that, because of the example of the S-1, Valid ended up with Digital, IBM and Mitsubishi Electronics as its first customers. In the beginning, Digital was the biggest customer Valid had. We hosted a whole room full of DEC people who came to Valid to talk about features they wanted in the Valid system. They couldn't wait to get it. So basically the S-1 was such a notable example of what could be done using automation versus what people were doing manually that it was like the best marketing you could have. People really wanted the tools when Valid finally came out with a commercial version.

Widdoes: You should understand that we introduced structure to logic design. What we did was the first practical use of structured design for designing a real computer -- not a toy research project in a university, but a large computer that really worked. We used structured design, and that was the cusp of a revolution. Essentially all logic design now is structured. Our language was graphical. It was very early in the structured design revolution. When we started the S-1, we couldn't imagine, and nobody that worked with us could imagine, <u>not</u> designing the computer graphically. Computers were always designed graphically, so we developed a graphical system. Later on, in the mid '80s, structured design came to mean designing with text – for example, Verilog is structured design using text. So, the revolution started with graphical structured design and morphed to text. I think major innovations sometimes come in steps because our imagination only reaches so far. But in those days... You have to go back to '75 and realize what the world was like. People really would mark the nets and type them in to a sorter to get the net list so that they could give them to a guy who would manually lay out the PC boards. There was no ASIC design.

McWilliams: In fact, the workstation hadn't come along yet. The Sun workstation didn't exist yet. We were doing this before Sun, before Apollo. None of the graphical workstations were there yet. So, Valid built a graphical workstation, which was very similar to the Sun workstation that was being built at Stanford at that time, before Sun started as a company. There wasn't anything available off-the-shelf for Valid to use. We wanted to run Unix. If Sun had started earlier, we probably would have used Sun and not built our own hardware. But the first Valid products were based on a proprietary workstation. For the S-1 Project, we built our own system. We had the Mark I, and we hooked up a graphics display and made it all work so we could do drawings for the Mark IIA. We had to build our own system. It had a fancy graphics display system [a III vector display, same as used at SAIL] that we bought and hooked up to the Mark I. The Mark I had 16 megabytes of memory, which was huge in those days.

Stump: So take us back to the 15^{th} DAC in 1978. Who was there on the team? Tell us about the papers. Was it anything like the design automation conferences we have today?

Widdoes: Well, it was tiny. It was in Las Vegas. There were 650 people, I think. I was just reviewing the materials and I think I read that it was 50% bigger than the year before. So, the year before there were slightly more than 400 people at the conference. Everybody fit in one room, so there were no split sessions; there was just one continuous sequence of papers. And there was no industrial participation -- no equipment, and no companies at all. Tom and I presented the papers on SCALD, which explained the philosophy, how the programs worked, the language, and everything like that. [McWilliams, T. M., and Widdoes, L.C., "SCALD: Structured Computer-Aided Logic Design," *Proc. of the 15^{th} Design Automation Conference*, Las Vegas, Nevada, June,1978, pp. 271-277; and McWilliams, T. M., and Widdoes, L.C., "The SCALD Physical Design Subsystem," *Proc. of the 15^{th} Design Automation Conference*, Las Vegas, Nevada, June,1978, pp. 278-284.] Those papers resulted later in our winning the 1984 IEEE McDowell Award. It was a very different conference. Today the DAC is highly commercial. This one was all just about the papers.

Stump: So how did you decide to make that transition from technology that you were building yourself to get your own job done to the idea of starting Valid? What motivated you? Who encouraged you?

Widdoes: That's a good story. It was 1980. A guy named Jerry Anderson read about what we'd done with SCALD for the S-1 Project. He was founder and CEO of a company called Two Pi that was making machines to run IBM 360 code, and he was selling them cheaper than IBM.

McWilliams: He'd worked for several companies before. He wanted to start a new company as the CEO. He was looking for some technologists. He had read the papers about the work that we had done and he approached us. He called the Lab one day and wanted to talk to Curt. The secretary said, "Well Curt's not available, he won't answer." Jerry said, "I'll stay on the line. I'll wait. I'll hold." So Jerry was on hold, I forget how long, 20 or 30 minutes, until Curt finally agreed to take the call and talk to Jerry. Curt was still busy coding, so he wasn't going to talk to somebody on the phone. So, that started quite a dialogue that went for a number of months. Jerry wanted to start a company, we were building these computers, and were we willing to leave the Lab or not, and so on.

Widdoes: Very importantly, Jerry had the stamina and the drive to get us up out of what we were doing to try something different. We talked about how to structure things. We wanted the S-1 Project to continue, so Tom and Jeff stayed back and continued on the S-1 Project while they worked 20% of the time as consultants to Valid and became part of the Valid founding team. I went over to Valid full time and ran the technology development. Jerry was CEO. We raised money. Don Valentine of Sequoia Capital led the first-round investment. It was an amazing success. We took the SCALD technology and we adapted it for commercial use. The company went public in 1983, when it was only two-and-a-half years old.

McWilliams: We hired a new group of people. They used the tools that we developed as a prototype, in other words, as a model for how to do it. They actually ended up rewriting the Valid code from scratch, which is not atypical when you bring in new people; they want to write their own code. They also oriented it towards a commercial product and made it support commercial technologies like gate arrays and PC boards. And we ended up with a new graphics editor written from scratch (SUDS was written in PDP-10 assembly code). Valid built workstation hardware, graphics boards, [networking, Unix operating system, CAE software,] the whole thing. It was an amazing effort.

Widdoes: Importantly, the original SCALD was written in Pascal. You might say, "Why use a teaching language like Pascal?" In 1975, when we started, Pascal was the only portable language that was available (other than FORTRAN, but FORTRAN wasn't a suitable language). We wanted our code to run on various machines, including the PDP-10 and the IBM-370/168, so we wrote everything in Pascal. We continued writing in Pascal through about 1980. So, what we put into the public domain and what Valid took from the public domain was coded in Pascal. Pascal turned out not to be the best language going forward, so Valid recoded everything in C. Also, the needs of commercial users were different from the needs of the S-1 Project. We had a great development team at Valid. Lou Scheffer, a brilliant guy, wrote the graphics editor [GED] for the Valid system. Glen Miranker was in charge of the simulation effort and the timing verifier. Mike Price was in charge of compilation and the physical design stuff. But we took all the SCALD Pascal code and basically used it as an existence proof. We took off from there and wrote new code that was more suited to use by mainstream ASIC and PCB designers.

Stump: What other differences or learning experiences did you have, now that you were building code for commercial use as opposed to your own proprietary, really focused use?

Widdoes: At Valid?

Stump: At Valid, yes, when you built the team and re-coded. Any other insights to share?

Widdoes: It turns that there were a lot of details that we didn't have to worry about when we were in the S-1 environment but that were necessary for the commercial user. For example, details related to getting a wire list out. Initially, that was a problem for Valid. The style of wire list that we output for the S-1 Project wasn't appropriate for PC boards and ASICs, so we had to cover that area, and it took longer than expected. Libraries, libraries, libraries. The user wasn't willing to put in the effort to build the low-level libraries of gates, adders, memories and so forth, so we had to spend a huge amount of effort creating libraries. Simulation. For system-level simulation of complex chips, you had to have hardware modeling. Well, for the S-1, we didn't care about hardware modeling because we didn't use complex VLSI chips. So, we had to cover all of the needs of the actual customer, you know, the guy who buys stuff and writes the checks. It turns out that Valid was a wonderful success. We ended up being one of the three principal public CAE companies (Daisy, Mentor, and Valid). And we did well primarily because we started early and we worked hard. If we'd started a year later, we probably wouldn't have been successful. There were 20 or 30 other CAE companies that started late. We took all their market because we started early. Thanks to Jerry who got us started early.

Stump: I want to go back to Jerry. I want to go back to the courtship period when you were deciding whether or not to make the jump from Lawrence Livermore Lab. I want to explore what motivated you at that time to take that risk.

Widdoes: Right. So I took the full jump. What motivated me was that I saw an opportunity to do well in the market. Right then was a time when things were aligned for CAE so that you could create a successful CAE company. There were the types of CPU chips that you needed to build a decent workstation. (We used the Motorola 68000 because it had a large address space. Sun used that chip as well.) There were displays that could actually display a big enough drawing. (You had to have at least 512 X 512 pixels to do logic design. Displays somewhat larger than that were just becoming available.) There were the memories. (Memory technology was just on the verge of providing memory chips large enough for a workstation to run this new software.) And there was the market -- the whole world was moving to ASIC design. So things were converging. And we had a head start since we had the SCALD methodology and software. So, I said to myself, "We can build a successful company that accomplishes something significant." But if that convergence hadn't been there, I wouldn't have gone for it.

Stump: It's interesting to think about what was at risk at that point in your lives. You had houses. You had marriages. What kind of personal sacrifices did you make in order to go off and start Valid?

Widdoes: Let me correct you. I was not married. My first marriage was over at that point.

McWilliams: Actually, if you get venture funding, it reduces the risk for the individuals. If you fund it yourself, that is, if you take all your savings, you get a second mortgage on your house, and you try to fund it yourself, you're at tremendous risk. Valid got venture funding, so everybody got paid a salary. This is one of the reasons why there's so much innovation in the United States; it is because of venture funding. If you can get funded, it reduces the risk. If you start a restaurant or a business that is all your own capital and it goes broke, you lose all your own capital. But the VCs have reduced the risk for individuals by providing the capital and paying salaries to people who work in those companies. So that is one of the reasons why there are so many startups here in Silicon Valley and elsewhere -- the VC money has basically reduced the individual's risk.

Widdoes: That said, it wasn't clear at all that we were going to be successful. It was a new market. It wasn't clear that we could actually build the product and make it work. We had to build our own workstation and all the software on top. So, yes, it looked to me like there was a lot of risk. But I was at a point in my life when I could take a lot of risk, so I decided to do that.

Stump: So who encouraged you, besides Jerry, of course? Who encouraged you, and who in your life, personal or professional, was more for risk mitigation? Do you have any memories of that?

Widdoes: I talked with Tom, Jeff and Jerry. We talked it out. Other than that, I didn't listen to anybody. I remember the meeting when we told Lowell that we were going to do it. He wasn't very happy. At least Tom and Jeff were staying back to continue. If they had left the S-1 project, it would have collapsed -- they were in the middle of the Mark IIA design.

McWilliams: The Mark IIA wasn't finished yet. We hadn't built the machine. I felt a personal obligation to Lowell and to the government, due to the fact that we'd raised a lot of money from the government, to see the machine completed and built. So, Jeff and I stayed on until the time when the machine was built and running, then we joined Valid on a full time basis. We basically worked out a deal where the S-1 Mark IIA could be completed and the company could get started anyway.

Widdoes: Importantly, Jeff wrote the simulator in SCALD. He was part of the SCALD team as well as the hardware design team. Of course, the simulator was used only for the Mark IIA. He also implemented SUDS for SCALD II. We had built a 36-bit machine, the Mark I, which was a basically a superset of the PDP-10…

McWilliams: Basically, he wrote an emulator so that he could run the SUDS binary code. He emulated it. The emulator translated each PDP-10 instruction into S-1 Mark I.

Stump: I want to take you back to the day or the evening when you really decided to start Valid Logic or the day you closed the financing. Was it an evening of fine wine at the Lion and Compass or was it a microbrew?

McWilliams: Well, actually we met and negotiated a deal at Mountain Mike's Pizza Parlor in Livermore. The problem was that Jeff and I weren't ready to leave the S-1 Project, so we negotiated a deal where Curt would go to Valid full time and Jeff and I would go part time. That way, Jeff and I could finish the Mark IIA. We wrote the deal on a napkin. That was when we made the deal. Jerry had signed a contract to stay at Two Pi for a year (he had sold it to Phillips), and the pizza meeting occurred right after his contract was finished, so he was free. He went off and got the company incorporated in January of 1981.

Widdoes: And I had just finished my Ph.D. at Stanford. I took until 1980 to do that; Tom actually finished his Ph.D. first. I had been rewriting the Packager. I finished up at the end of 1980, joined up with Jerry in January of 1981, and went out to raise money.

Stump: So you've shared a little bit about the steps of Valid importing and using the SCALD code as a prototype and rewriting it and so forth. What was the team like in the early days?

Widdoes: The Valid team?

Stump: The team that you formed at Valid, yes.

Widdoes: Well, we attracted some really exceptional people. That's something that I have been fortunate enough to do in each of my companies and that is the most important reason why my companies have succeeded. The first engineer who joined was Glen Miranker, also a Hertz fellow. He was previously at IBM Yorktown. Then Lou Scheffer joined from Stanford and HP. If there is an individual in the early technology team at Valid who was most responsible for Valid's success, it is probably Lou, because GED [Valid's graphics editor] is the primary reason why people bought Valid's product in the first couple of years. They mainly wanted to enter their drawings. Fancy SCALD-like software was cool, and they used that later on, but in the first couple of years we sold many tens of millions of dollars of product because people wanted to enter logic diagrams. They'd never been able to do that before. So, one thing you learn is that, despite all the fancy technology you develop when you're trying to get out in front, people are really mainly interested in the first little bit working properly. Anyway, Lou built a wonderful graphics editor. It was so far ahead of Daisy and Mentor that there was no comparison. And we sold systems based on the graphics editor. Mike Price was recruited into Valid by Lou, who knew him from HP. Mike, Lou, and Glen led the Valid development team and developed a complete set of CAE tools that replaced and improved on SCALD.

Stump: You mentioned you had some customer champions that carried over from your efforts at Lawrence Livermore. Were some of your early customers at Valid people you knew from DEC?

McWilliams: That's true. In fact, I had worked summers at DEC when I was at Carnegie Mellon. So, when we did the S-1 Project we had contacts at DEC. Curt and I would talk with people in various companies. In particular, we went to DEC and gave presentations about all of our work. We created a lot of interest and a lot of desire on the part of the designers to have that kind of technology. I remember when Valid was newly incorporated, a whole room full of DEC engineers came to visit Valid. They wanted specific features. For example, they wanted properties on properties in GED. They were SUDS users. So they were out there at Valid telling us the features that they wanted. It was the best kind of customer input you can get. We ended up putting various features into the commercial product that were specifically requested by DEC. They helped guide some of those features and they worked with us through all the trials and tribulations of getting beta software working. In the first product we shipped, you couldn't get a wire list out. You could enter the design, but the design flow wasn't complete. It took six months of working through issues after we shipped our first beta units before we could get the printers working well, the wire lists coming out, and lots of other things. Fortunately, it turns out that in the design process, the first thing you do is enter drawings -- you just draw things and print out hardcopy. But DEC was instrumental for Valid. They were our biggest customer and they were a reference site. A lot of people said, "Well, if DEC can design computers with it, it must work." So, they were instrumental in helping us get more customers. They really provided valuable input to the company and were a tremendous help.

Widdoes: When we started the company, it was actually called "SCALD Corporation". That was the official name of the company we incorporated. I remember one day Jerry came in and he said "Curt we have a problem." This was right before we raised the first venture money. "The venture guys want to invest, but they are demanding that we change the name of the company." "Oh, why Jerry?" "Because they don't want anybody to get SCALDed." They were afraid that, if we were successful, we'd be trying to take the company public, and everybody would joke about the fact that they might get SCALDed. So, that was a serious problem. We changed the name to Valid Logic Systems.

McWilliams: That turned out to be a good name -- Valid Logic Systems.

Widdoes: Oh yes. Much better than SCALD. SCALD sounds like the name of a Scandinavian drink.

Stump: So I'm just fascinated by the fact that the original work was in the public domain, and I'm wondering how you think the SCALD work affected the rest of what turned into the CAE industry… spawning some competition, or…

McWilliams: Well, Daisy, Mentor, Valid and a number of others did pick up SCALD. Livermore sent out over 150 tapes containing the software to people and companies that requested it. It is interesting that the competition actually helped Valid. CAE was a new industry. Three companies (Daisy, Mentor, and Valid) went public, and others followed. Customers took the products more seriously because there were multiple suppliers. If you were starting a design project, you'd have multiple companies coming in and giving presentations. I think all the competition made it clear to the customers that they needed to buy something like this. After that, it was just a question of whom they bought it from. So, the competition greatly accelerated the growth of the market. The market grew bigger, faster and got big enough that multiple companies could be financially successful.

Widdoes: The SCALD software that we put into the public domain really set the ground rules for all the CAE companies. The various companies all did their own implementations, but SCALD was important in saying to the industry that you must provide a certain basic set of capabilities. If you didn't provide the basic capabilities, your competitor would. The basic set of capabilities obviously included a graphics editor -- that was obvious. But you also needed to provide hierarchical design; you couldn't get away with just old-style flat design. So, all the successful CAE implementations included hierarchical design. They probably would not have done that if SCALD had not set the ground rules.

McWilliams: Right. There were earlier CAD companies that did conventional design, but they were more focused on PC board layout.

Widdoes: So, SCALD says you must provide the graphics editor, hierarchical design, simulation, timing verification -- all that stuff is part of the minimum offering. It was a forcing function. Some of the companies didn't actually succeed in providing certain pieces, but they were all aiming at that. And so there were multiple offerings of roughly the same set of capabilities and customers felt safe in adopting the new methodology. But you had this ground rule: This is what modern design is. You must provide this set of capabilities in your commercial offering. Different companies had different levels of matching the specification. SCALD set a specification and people went off and did their own versions of that specification.

Stump: Now, Curt, isn't it true that even today you've got design capture, you've got the idea of hierarchy and design reuse, abstraction… There have been a number of changes in EDA, but they've been along the lines of that original flow and consistent with the concept of raising design to a higher level of abstraction?

Widdoes: Yes. The game after SCALD was how do you get to an even higher level of abstraction. Today, the languages that are used are at even higher levels of abstraction, much higher than SCALD was. With SCALD, you explicitly represented every element -- there was a representation of every gate. We were basically drawing in a fancy way. In today's systems, the gates are inferred from a text description and design is done at a higher level of abstraction.

McWilliams: Well, synthesis came along later. Synopsys came along with synthesis so that you could basically just write equations. In SCALD, you could always find a drawing that would show which gate to use. Synopsys started automatically calculating: "Well, I'll put it in this gate and that gate and optimize it." That's more automatic. Synthesis automatically generated and optimized the logic based on constraints. So, that was another level of advancement. When you went to synthesis, engineers no longer controlled what gate or what cell got used. Once you start doing that, it opens the door to even more abstraction because the engineer has less need for control.

Widdoes: So we did structured design with graphics. Later on, design was done using text, but still describing every gate in the text. Then, synthesis was introduced and people stopped describing every gate. But things moved in little steps. At the time we did the SCALD work, you couldn't imagine not drawing. Later on, text took over, but you couldn't imagine synthesizing it; synthesis was too hard. Later, Synopsys introduced synthesis. So things progressed and engineers proceeded to design at higher and higher levels of abstraction. I think there may be a point at which our algorithms have problems getting any higher. We may be at that point now.

Stump: I want to go back to the days of Daisy, Mentor and Valid, the market leaders in early EDA. Did they have different personalities as companies, or different quirks in how they approached technology and how they approached the market?

Widdoes: Yes. I should say that those were the cowboy days of EDA. There was a lot of intense and personal competition. Everybody was struggling to get their products out on the market first. What we tried to do at Valid was to ship stuff that actually worked. I think we probably had the best record there. But I don't want to say too much along those lines.

Stump: So when you think back, what were the effects of SCALD on the industry, EDA and electronics overall?

Widdoes: Well, SCALD was a revolutionary methodology for designing electronics; it was the first practical implementation of structured logic design. But in a sense it was inevitable. If we hadn't developed SCALD, structured logic design would have come along a little later, but it would have come along eventually. SCALD got things moving, maybe a little earlier than they would have gotten moving otherwise, but SCALD is not really responsible for the existence of structured logic design today. It's clear how to do it (programming languages did it), and it's necessary. We were just early in our implementation. So what effects did SCALD have? SCALD got people working on structured logic design early. It got the CAE industry moving in that direction. It helped launch the CAE industry.

McWilliams: I'll say one thing: One of the reasons people drew out every gate and all the low-level details is because they were doing the physical design manually; the designers were making all the physical decisions. They needed the pin numbers and they needed to show the wires and which connectors the wires connected to. In order to make it possible for the designer to avoid having to draw everything out just in order to have to have a place to put the physical information, we had to automate the low-level physical details. Automating the physical design is what made it possible to not draw all the gates. In the back-end packager, we had mechanisms for automating most of the physical design and also ways of manually specifying some of it. For example, in our placement language you could say that a whole unit or a whole design would go on this board or in this section of a board. So, we had a way in which we could specify some amount of physical design information and the rest would be automated. Similarly, synthesis made it so that now you don't even have to draw as much as we did. The more you automate the low-level details, the more you can move to a higher-level representation. Today, in a full-custom design shop using full-custom, manually optimized, transistor-level design, they still use schematics. A lot of them still use GED. If the designer really wants to optimize the transistor layout, he needs to show every transistor and he needs to carefully tune the physical design. It turns out that people can't understand the physical design very well from a text-based language. Talk with the designers of the Alpha chip that DEC developed. Well, they tuned that thing. The way they got it to that speed was that they actually laid it out using the schematic -- the way it flowed across the chip. So, there are still people doing full-custom design where they need to have complete control, and they still use graphics. But if you can automate the low-level details, then you can get rid of graphics and you can move to a higher-level representation.

Stump: And that's what made the 80,000-gate computer possible back in the days of S-1?

McWilliams: Yes. Of course, 80,000 gates is nothing today. In fact, Intel just announced a 2-billion-transistor microprocessor. Today, 80,000 gates is just one little gate-array. The whole S-1 Mark I computer would fit easily on one gate-array. The S-1 was simple compared to today's advanced microprocessors. The design technologies have evolved so much. Designers use more automatic placement and synthesis, more automation. But parts of the advanced chips are still designed manually, and custom designs still use schematics in sections where the designer needs manual control over the physical details.

Stump: Tell me more about Lowell Wood, who enabled this whole project.

Widdoes: Frankly my career was sparked by Lowell; he was a key mentor for me during the S-1 Project. Lowell is a remarkable guy. He probably has more drive than any other person I've known. He's a brilliant physicist. He was a real inspiration during the time we worked with him. He built a large team of very talented young people. He was a Hertz Foundation interviewer, so he would cherry-pick Hertz Fellows and get them to work at the Lab. As a result, he built a group of top students from top colleges researching interesting things and working really hard. There are a lot of funny stories associated with Lowell. For example: I was working with Tom on the S-1 Project. I had a house in Sunnyvale. I was married at the time. I noticed that my water bills were really high -- hundreds of dollars per month. And that was when water was cheaper, so this represented tens of thousands of gallons of water used per month. Also, I noticed that water was pooling up around the edges of my concrete-slab house. I didn't have any time to worry about it because we were working so hard. So, I didn't worry and I didn't worry. This went on for about a year and…

McWilliams: So Curt told us about this, and Lowell and I went over to his house. It seems that his house had water pipes under the concrete slab and they were leaking. Lowell had a station wagon, so we went down to the rental place and rented a big air compressor and a jackhammer setup.

Widdoes: When I looked out my door, there was a huge compressor and jackhammer assembly coming up my driveway, towed by Lowell's station wagon.

McWilliams: So Lowell and I went into his house and we got started.

Widdoes: They jackhammered up my kitchen floor and my utility room floor looking for the leak. Eventually we did find the leak.

McWilliams: Well, actually, no. We didn't find it. At some point Curt decided to call a plumber.

Widdoes: But this just illustrates that Lowell <u>acts</u> and gets things done. When it was time to jackhammer up my kitchen floor, he got the jackhammer and he came and he did it.

Stump: Do you think this was a thinly disguised ploy to have you be at work more?

Widdoes: Well, the real point is that Lowell helped. Lowell was always willing to help out in whatever way was required. He would do stuff with his own time in order to free up the same amount of time on our side. He would do whatever was required to keep us maximally productive. In this case, he figured that jackhammering up my kitchen floor was going to help me, so he got it done.

Stump: Are there any other moments that you remember that really crystallize in your mind or that are icons of the whole SCALD project? People? Events?

Widdoes: Well, Tom, were you there when Lowell borrowed a bulldozer and worked on damming up the trailer site so it wouldn't flood?

McWilliams: I don't remember that.

Widdoes: I can't tell it well enough. I wasn't actually there that night.

Stump: So, either of you: If you could have done the SCALD project differently in any way or if you had but known then what you know now… Were there any takeaways you could crystallize for us?

McWilliams: I think that, if we had known what we were getting into -- how much money it was going to cost and how long it was going to take -- we never would have done it. So, some of these things you get into, you need to be a little bit naïve, otherwise you wouldn't even attempt them. The S-1 Project was so much beyond what any of us expected -- the amount of time and money and how many people we got involved. If you don't expect it to be so difficult, then you just get started. You just go step-by-step, one day at a time, one thing just leads to another, you keep making progress, and it grows. I found that starting companies is the same; if you knew up-front how much money it was going to cost and how long it was going take, you would never start.

Widdoes: That's right. The funny thing is, although the S-1 Project was much more difficult than I expected, I re-learned that same lesson at Valid, I re-learned it again at Logic Modeling, and I re-learned it again at 0-In Design Automation. I keep re-learning the same lesson.

McWilliams: Right. I agree. Every company that I've started ended up taking more time, effort and money than I expected, but in the end it has always worked out; it has always been worth it.

Widdoes: Another takeaway, not only from my SCALD experience but from my later companies as well, is that if you want to build a successful company, you need to attract high quality people to work with, especially in the early stages. If you work with people like we worked with, for example, Lowell Wood, Jeff Rubin, and Mike Farmwald, you can do great things. Similarly, at Valid, if you can attract people like Lou Scheffer, then you can accomplish great things. Great people hire and motivate other great people. That's one of the most important lessons I've learned. Your success in building a company is really critically dependent on the quality of the people that you attract to work with you. Certainly, my success in the companies that I've started has been primarily the result of having been able to attract really great people to work with me.

Stump: So what advice would you give young scientists or engineers who are inspired by your journey?

McWilliams: I would say to follow your passion. There's some inner place of knowing in us where we have a passion. If you're not excited and passionate about the thing you're working on, it's probably the wrong thing. I've found that you have to have passion and follow it, and see where it leads.

Widdoes: For me, work with great people.

McWilliams: Right. I agree with that 100%.

Widdoes: If you're not going to work with great people, then don't do the project at all.

McWilliams: That's definitely true. Every place I've been it has been the team that built it. You can get things started, but if you don't have a good team, then you won't be able to execute long-term. Also, if you get a good group of people together, then, even if you're not doing exactly the right thing, they can figure out how to change it. You need to stay open-minded and not lock in. Sometimes people don't want to admit that they need to change things. There may be bigger opportunities if you make some changes. You may need to evolve what you're doing, listen to customers, see what's really required, see what the market really needs. It's hard to know all that when you get started. You have to get a critical mass of good people together, get out there, play the game, interact with customers, figure out the technology. You absolutely have to have the best people.

Stump: All right, have I forgotten to ask you anything?

Widdoes: No. Thank you very much. It has been quite an exciting ride over the thirty-three years that we've been working in EDA. And without the advances in EDA that we've talked about, there would be no fancy computers today. There would be no Computer Museum, because there wouldn't be anything worth storing in it. Certainly, people wouldn't be designing multi-billion-transistor chips today if they didn't have advanced EDA tools.

END OF INTERVIEW

