Oral History of Len Shustek

Interviewed by:
Gardner Hendrie

Recorded: July 16, 2002
Mountain View, California

CHM Reference number: X2723.2004

Gardner Hendrie: Today is July 16th [2002], and we're here at the Computer Museum History Center's Visible Storage area. We have Leonard Shustek, our Chairman of the Board, for this particular oral history session. Len, maybe a good place to start is what you thought you wanted to be when you were a young child.

Len Shustek: I always knew that my interests were science and engineering of some kind, and if I admitted it to myself then would have plotted a course to be an engineer. But somehow an engineer didn't seem intellectual enough, so I plotted a course to be a physicist. I became a physics major at Polytechnic University, then called the Polytechnic Institute of Brooklyn, and enrolled in a special course they had called "Unified Honors", that allowed you to get a Bachelors degree and a Masters degree in physics in four years. It was an accelerated course. I had come through the public high schools in New York, and been in the accelerated special progress courses there as well. And of course in high school I spent all of my time, as most nerds did, in the science labs. I put together all sorts of electronic equipment, assembled Heathkits -- vacuum tube volt meters and audio generators -- and played with all of that equipment. I really was most comfortable with electrical engineering, but decided that somehow physics was the right, and proper, and hard field to do. Maybe it was a bit of a hairshirt. So I did that. I got my Bachelors and my Masters in 1970 from the Polytechnic University. But in the process of being there and doing some work in physics, I discovered the Computer Center, and that opened up a whole new era to me of toys to play with, in particular the computer, which were a lot more fun than vacuum tube volt meters.

Hendrie: Now were there particular projects in your academic career that led you into the computer room, or was it friends, or were there dames or …

Shustek: My first exposure to computers was even earlier than that. It was in "junior high school" or what's called "middle school" now in California. I must have been in seventh grade, or something like that. IBM had a program for taking kids who showed a proclivity to excel in engineering and science, and having them spend a weekend at the Watson Lab in Manhattan, New York. I think what they were trying to do is convince people to do that, instead of become what was referred to in those days as "JD's" -- juvenile delinquents -- and hanging out on the streets. So I did that, and I spent a wonderful weekend learning how to program an IBM 650. In fact, my first program was in assembly language for the IBM 650, which was a vacuum tube machine with drum as its primary memory. In between junior high school and high school, I guess, I spent a summer programming an IBM 1620, which was a transistorized machine built about 1960-61, at the New York University Bronx campus. So those were my first introductions to computers.

Hendrie: Now were these summer jobs or …

Shustek: They weren't jobs. They were high school programs for bright kids, just to give them something to do, in one case for a weekend, in one case during the summer. But then, as I said, I majored in physics. I'm trying to reconstruct now the background. I did work for a summer; I guess that was the summer before going to college. My first paying job was at New York University, as an operator of a 1620, for New York University Courant Institute in downtown Manhattan. I remember I was being paid a dollar and a quarter an hour, which was the minimum wage at the time, to be the summer operator for the 1620 computer. And I couldn't believe that they would pay me to operate a computer; I felt like I should pay them to operate the computer, not the other way around.

Hendrie: Because it was so much fun.

Shustek: Because it was so much fun, and I thoroughly enjoyed that.

Hendrie: Very good. When in your college career did you start to spend time with computers, as opposed to just sort of focusing back in on your major?

Shustek: Well it's interesting. They were really very separate. I didn't discover and use computers in college because of my physics major. My physics concentration was actually in theoretical physics and quantum thermodynamics. I was doing integral equations and things that were not numerical analysis but really symbolic computation. My Masters thesis was all symbolic computations, as opposed to computer generated stuff. But the University had, at the time, an IBM 7040, and they had a computer center, and they had an active chapter of the ACM -- the Association for Computing Machinery. I hung out there in my spare time and started doing projects there just because they were fun, not having anything to do with my course work. The fellow who really got me started in that was actually, it turned out later, to be a long-time DEC employee. His name was Richard Lary, who may still work at DEC. He worked at DEC out of Colorado for many years. He wrote a BASIC interpreter for the IBM 7040, and took me and a couple of other freshman under his wing, and taught us how to do assembly language programming. I remember very proudly doing the matrix arithmetic routines for the BASIC interpreter on the 7040.

Hendrie: That was your contribution to the …

Shustek: That was my contribution to that project. I just had a great time.

Hendrie: Well, he is still a consultant. He consults in the networking and memory storage areas, for your information.

Shustek: Oh terrific; I haven't seen Richie for many, many years. But he and one other professor at Brooklyn Poly are probably the two people who are most responsible for me abandoning physics and joining the world of computers.

Hendrie: Okay. So what options did you consider, when you graduated, for going out into the real world and doing something?

Shustek: Well, I didn't really have a lot of options, because this was 1970 and the Vietnam War was in full swing. I fully expected that when I graduated that I would be sent to Vietnam and be toting an M-1. So I had not planned -- and I was never good at planning anyway -- my future career. I had no idea whether I really wanted a job or wanted to go to graduate school. I applied, almost at a whim, for a National Science Foundation fellowship to go to graduate school. By that time I knew that I wasn't destined to be a great physicist -- and besides, computers were a lot more fun -- so if I did go to graduate school, it would be in Computer Science, not Physics. I remember very clearly filling out the application for the NSF while standing at the window of the computer center where I was waiting for my [computer job] output to be returned. I got to the place in the application where it said "list the three schools you are going to, or have, applied to go to graduate school." Of course, I had not made any applications, so I turned the fellow next to me and said, "Name the best three schools in computer science that you can think of". And he said "Stanford, Carnegie-Mellon and MIT", and I wrote down "Stanford, Carnegie-Mellon and MIT." And I never applied to any of those schools. About six months later I got a call from Stanford from Professor Gene Golub, who I think is still there, who said "the NSF has just notified us that you've gotten a fellowship to attend Stanford, and we can't seem to find your application. We must have misplaced it. Can you send us a duplicate copy?" Of course, I had never applied to any of those schools, and I had no idea that the NSF had awarded me a fellowship. To make a long story short, I did a quick application, and Gene Golub was one of the people who facilitated that. Stanford was happy to receive me because I was coming with money from the NSF. I moved to the west coast in 1970 to go to graduate school at Stanford.

Hendrie: Very good. Well, before we go into your career at Stanford, could you say something about… you mentioned another mentor at Brooklyn Poly. Tell us …

Shustek: Yes, that was Bud Lawson, Harold W. "Bud" Lawson. There was no Computer Science department at the time. In fact -- this was 1968 -- very few universities had a Computer Science department. He was teaching in the electrical engineering department, but was teaching some computer courses, which I never took. Well, I guess I did take one of his. But the most gratifying association with him was that he decided to start a programming project to make an implementation of the PL/1 language for their new 360/50 IBM computer. He had worked for IBM and was one of the architects of the PL/1 language. IBM had a big and ponderous PL/1 compiler. He had the idea of, instead of making a compiler, of making an interpreter. In fact, an interpreter that would run the compiler, generate interpreted code, and then an "Execution Machine", which was also interpreted, would interpret that code and execute the program. He called this PLAGO, "PL/1 Load and Go compiler". The object was to be extremely fast, and use it for student programs and student courses. I was assigned the job of writing the interpreter for the Execution Machine for interpreting the generated code stream and executing the students' projects. I worked on this for probably about a year, year and a half. It was a project that had maybe ten or twelve students working on it. I found that the most gratifying experience of my academic career up to that point. I really decided at that point that computers were more interesting than physics, and what's more, that I was an engineer at heart and not a scientist, because what I enjoyed doing was building things rather than analyzing things. My physics dissertation was an analytic dissertation, and I found that not as satisfying as the synthesis that you do by being an engineer, starting with a known set of building blocks and constructing something extremely complicated from it. So that PLAGO project was very gratifying, and Bud Lawson was one of my mentors. I've been in contact with him quite a bit over the years. He now lives in Sweden. He married a Swedish woman, has two Swedish children who make fun of his American-Swedish accent, and is variously a consultant, and a Professor at Linkoping University [in Sweden].

Hendrie: Oh, that's very interesting.

Shustek: Wonderful person.

Hendrie: Now this wasn't even in connection with a course, this is just extracurricular activity.

Shustek: This was just extracurricular activity. There may have been a course of Bud's, I'm trying to remember, that I took at one point, but that wasn't really the point. It was a project that he managed to get students to sign up for.

Hendrie: Volunteer for, yeah.

Shustek: That's right Graduate students and undergraduates are the slave labor force for professors. In that sense I guess we were his slaves, but we enjoyed every minute of it.

Hendrie: Okay. And it wasn't a paying job in any sense, that wasn't a graduate fellowship or anything?

Shustek: It wasn't a paying job Although I guess I did have what was called a "teaching fellowship", under which I did that. So they did pay me a small stipend, although I was living at home. But I think what it really paid was my tuition. I had a Regent's Scholarship from the State of New York to go to college. I come a family background where without scholarship assistance I could not have gone to college. I'm the first generation in my family to go to college. So my entire four years at Poly, and my years at graduate school at Stanford, were all paid for by various combinations of scholarships and fellowships. Without that, I wouldn't have gone at all.

Hendrie: Okay. I apologize for rolling back, but I think it would, since you mentioned it, be interesting to know the background and the interests and vocation of your mother and your father.

Shustek: I come from I guess what you would describe as a lower middle class immigrant family. My mother was born here to parents who were born in Eastern Europe. My father was born in Poland and came to this country when he was five, or something like that. My father didn't get through high school. He left school in the depression to help support his family, worked in the food service industry for many years, owned his own fruit and vegetable store, and luncheonettes, coffee shops, that sort of thing. He was moderately successful and he retired; he had enough money to live on to retire with. My mother got through high school and, I think, one year of night school college. But then again, went into the workforce, became a bookkeeper. Her connection to computers is that she was a comptometer operator for Howard Clothes, and did the bookkeeping using a comptometer.

Hendrie: Very good, very good.

Shustek: An old-fashioned mechanical calculator. But, you know, like many Jewish immigrant families, they put a high priority on education for their children. So there was no question that I would go to college one way or another. They wanted me to be a doctor, and okay, a computer scientist is a good second choice.

Hendrie: Okay. All right. Should we skip ahead to Stanford and talk a little bit about… Now what we're you going to go do at Stanford?

Shustek: I had no idea. I joined Stanford as a graduate student in the PhD program, which had only been running, I think, at Stanford for about four years. I came in 1970.

Hendrie: This is a PhD in what, in …

Shustek: PhD in Computer Science.

Hendrie: In Computer Science.. So you're in the Computer Science Department.

Shustek: I switched absolutely everything: I changed coasts, I changed lifestyles, I changed fields. I had never been west of, probably, Philadelphia at the time, when I drove my Dodge Dart to California. I really had no conception of the geography. I really didn't understand whether San Francisco or Los Angeles were north or south of one another. I was just heading west, and eventually wound up in Stanford. I found the environment very different from Brooklyn Poly. At Brooklyn Poly I graduated with a 3.96 grade point average. I was a big fish in a little pond. Here I get to Stanford with a whole lot of very smart people, and suddenly I'm a little fish in a much bigger pond. I found that first year at Stanford as a graduate student to be very challenging, and hard to keep up. I did, but it was a lot of very hard work. And I was living on almost no money. I had a stipend from the National Science Foundation -- which [also] paid my tuition, and that was the big thing -- of 200 dollars a month. That's what I lived on, including paying for my car, and the books, and the apartment I rented for 90 dollars a month. I lived on my own, for the first time, with not a lot of money. But it was fine.

Hendrie: Okay. So you started this first year. How long was this program going to be? What did you think, when you started out, how long it was going to run?

Shustek: They were very honest when you come Stanford in setting your expectation for how long it takes to get a PhD. Now, there were some "wunderkinds" who did it in two or three years -- I think Bob Tarjan got his PhD in two years -- but the average was five, six, seven years, something like that. And that was okay for me, because I had no idea or plan for what I was going to do after that anyway. So I figured if I found graduate school enjoyable, which by and large I did, I'd stay there forever. My father joked that I was going to be a professional student, and I'd never get a real job, which I came dangerously close to doing.

Hendrie: All right. What did you embark on as a thesis?

Shustek: It took me a long time to get into doing a thesis. I got there in '70, went through the normal wonderful courses there and professors who were terrific, Don Knuth, and John McCarthy, and Ed McCluskey. This was early enough in the computer science days so that people like me, studying for a PhD, were learning from the people who had invented their fields, like John McCarthy. That was absolutely wonderful and fantastic. The first or maybe second summer -- maybe it was the first summer, I guess that was summer of '71 -- I was looking around for something to do during the summer. I spoke to Tom Bredt, who at the time was a new professor there, and who is now a well-known venture capitalist.

Hendrie: I know Tom, yes.

Shustek: His recommendation was, "Well, you seem to be interested in systems kinds of things, and practical things as opposed to theoretical stuff. Why don't you go over to the [Stanford] Linear Accelerator

Center? They've got a bunch of big computers there and a bunch of projects, and you'd probably find something to do there." So I did. I spent the summer at the Stanford Linear Accelerator Center, SLAC. They had just gotten an IBM 360/91, which was the biggest supercomputer that IBM was making at the time. Even more importantly, though, they were about to receive a machine from the Standard Computer Corporation -- of Santa Monica I believe, somewhere down in Southern California -- that Bud Lawson had done some consulting for. It was a very unique architecture. It was an emulation machine that was designed to be able to emulate many other architectures. They promised that machine to Stanford. It was called the MLP-900. They never delivered it. In fact, as far as I know, they never delivered any MLP-900's. But they did deliver to Stanford at SLAC a predecessor of that machine called the IC-7000, which was again a machine that was designed to emulate other instruction sets. I embarked on a research project there basically to use the micro-code of that machine to do performance analysis. To try to understand why computers are fast or slow, to understand the effect of a program being run on the execution of the machine, by modifying the micro-code to collect statistics about the running program, opcode distributions, branch link distributions, information about register usage. It was all wonderfully easy to do with the IC-7000 because you could get in there and, instead of mucking with gates and wires, you muck with the micro-code.

Hendrie: Just insert it in a micro instruction here and there.

Shustek: That's right. Insert a little micro-code branch instruction to the collection routine. The machine would wind up running more slowly, and time would sort of dilate, but it didn't really matter because you would execute correctly, and you collect all the information. The fellow I met, who was a research associate, I guess, at the time there, working on that project was the fellow who was to become my business partner for 16 years in two different companies, and that was Harry Saal. It turns out he was also a fellow New Yorker, although I had never met him in New York. He got a PhD in physics from Columbia University, and followed his physics professor Mel Schwartz out from New York and Columbia, to Stanford, where Mel is a physics professors doing work at SLAC. So Harry also is an ex-physicist, came to SLAC to do physics, and then started doing computer stuff instead. We met there and collaborated on that project, and published a couple of papers about that work in '72 or '73.

Hendrie: Fascinating. All right. Now, how long were you working on this project at SLAC? Have you started writing your thesis, or you're just doing different stuff, and taking courses, and enjoying yourself, and hopefully your monthly stipend is slightly more than 200 dollars a month.

Shustek: Well eventually they got my stipend up to 300 dollars a month, which was a big relief. I could go off my bananas and Coca Cola diet at that point. Yeah, the IC-7000 lasted there for a couple of years. Then they got the 360/91 and I started doing some work on that. What's even more interesting was that in about 1971 Intel started producing little microprocessors, and Harry Saal got the great idea of going to Intel and getting some early microprocessors and development boards from Intel, and starting at SLAC to experiment with the use of microprocessors. We started there, under the direction of Jerry Friedman who was the Director of the Computation Group at SLAC, a little microprocessor laboratory to do that kind of experimentation. I wound up working in that lab on microprocessor-based projects probably for the next three or four years. In the meantime, still attending classes at Stanford on campus and taking my qualifying exams for PhD, and basically taking all the courses that I needed to qualify for the PhD degree, and making no progress at all on getting a thesis topic or writing a thesis. I became what is known at Stanford as a ”Terminal Graduate Student”, which always sounded like a fatal disease, and remained a terminal graduate student until I finally got kicked out of Stanford with PhD in about 1977-78. But the micro processor lab actually became quite successful. We proselytized at SLAC to the other engineers and scientists about the use of microprocessors, rather than the large computers, for doing data analysis and data collection, and automation of the experimental end station controls. I think we had some considerable impact in getting the physicists there to adopt microprocessors.

Hendrie: What had they been using for the real time collection, mini computers?

Shustek: They'd been using a lot of mini computers. They had SIGMA-7's and SIGMA-9's; they had lots of PDP-11's. Eventually they had VAX's. They did a lot of custom hardware circuit design. There was an instrumentation standard for physicists called CAMAC -- I forget what it stands for, it's an acronym -- of large sized boards. They would churn out these boards by the dozens, hardwired with logic to do particular high speed data collection activities. One of the things we tried and succeeded to get them to do was to recognize that they could use both microprocessors and bit sliced processors to do a lot of that in a much more configurable way than the hardwired logic that they had been using before.

Hendrie: Okay. More general purpose collection equipment. Good.

Shustek: Eventually I did connect up with a thesis advisor. Harry Saal was nominally my thesis advisor as a research associate for a short amount of time, but he left SLAC. He joined IBM and spent several years in Israel working at the IBM office at the Technion [in Haifa]. So he sort of disappeared from my life for a while. Forest Baskett, who was a Professor of Computer Science and Electrical Engineering at Stanford, and who also spent some time in SLAC, became my thesis advisor. He also encouraged me to do non-thesis related work. He and I spent a lot of time designing, and I building, a video graphics terminal that we called VGT, which was a very novel 8080 microprocessor-based graphics terminal. At that time at SLAC they did graphics on storage screen terminals, where you could draw vectors and lines and dots, and then have to erase the entire thing in a big flash and re-draw the image. We thought that you could build an inexpensive frame buffer, control it with a micro processor, build some special purpose logic to read out the frame buffer and display it on a standard CRT, and build a more adaptable, more flexible, but still relatively inexpensive graphics display unit. So we did. We built the thing called the VGT. It had a graphics mode where it used all 64K of its memory to display bits on 640 x 480 screen. It also had a very novel text mode where it would record in memory the ASCII characters, and through hardware go through a character generator and display text. With 64K characters you could have a huge typescript of text, much more than could fit on the display. So we built a little scrolling wheel, which was a machined aluminum wheel that looked like the platen of a typewriter. As you moved or spun the platen of this “typewriter”, the typescript on the screen would move instead, in a very smooth fashion. If you moved it a little bit, it would move a partial line. It really was a wonderful experiment in user interface, because people got the connection between the wheel and the typescript as if they were moving a piece of paper. It was a very comfortable way of scrolling. This was pre-mouse days.

Hendrie: Yes. Pre-mouse with wheels.

Shustek: Pre-mouse with wheels. I guess that idea has come back now, a mouse with wheels.

Hendrie: Yes. I think so.

Shustek: We actually tried to sell the design of the Video Graphics Terminal and the scrolling wheel to a couple of other companies. We made the rounds trying to sell the idea.

Hendrie: What, now what year do you think this might have been?

Shustek: This would have been I think, ’75, is my guess. It was just after the 8080 came out. I’m guessing it was ’75. Maybe it was ’76 that we were out making the rounds. We had successfully convinced SLAC not to patent it, which we thought would inhibit our ability to get other people to manufacture it. Our idea was not necessarily to start a company and make money on this, but just to convince people that this was the right way to build graphic display devices rather than storage screens. Actually Tektronix adopted the design. They took it -- they paid me for three days of consulting on the design -- and I believe came out eventually with a graphics terminal as a complement to their storage screen terminals, using parts of our design.

Hendrie: Okay. So that was a relatively early frame buffer based terminal. There are others...

Shustek: There are others and there were others. I think the novel thing about this was that it was targeted for a very low price point. It was targeted to be a terminal, not a frame buffer as part of a much larger system. It was targeted to replace a [Tektronix] 4013 or 4010 storage screen terminal.

Hendrie: Okay. Very good. And so you’re clearly still having fun at SLAC. Now, what do you work on next?

Shustek: Well, eventually I had to get to the point where I was making progress on my PhD thesis.

Hendrie: Yes, did they set a deadline for you at some point?

Shustek: No, they didn't set a deadline. People in universities like Stanford can hang around for years as long as they have some financial means of support. But I was getting to the point, as it was getting to be six years, where I think there should be some next part to my career. Forest would keep leaving these little cartoons on my desk, you know, with a picture of Forest kicking me in the butt, saying "get moving on your thesis." [Laughs] We eventually settled on a thesis that harkened back to the work I had done with Harry Saal on performance measurement on the IC-7000. I did a similar kind of research project for the [IBM] 360. There you couldn't micro program the machine to collect data, but you could build an interpreter. So I built an interpreter for the 360 that did similar kinds of data collection and analysis for running programs in the midst of the complicated operating systems of the 360. It's interesting that I was emulating the instruction set of the 360 on a 360, which is a relatively easy thing to do. There's only one instruction whose properties are ill-defined, and is very difficult to simulate, and that instruction is SVC, Supervisor Call. It's the trap to the protected operating system. The definition of what that instruction does is not just in the hardware, but is in a stack of manuals this wide that only documents part of what SVC does. As an instruction it has very strange properties, like sometimes control is not returned to the instruction which follows the SVC. It goes through a complicated series of control blocks and finds a data exit subroutine address, and the SVC begins executing at that data exit subroutine address. So my interpreter had to know all about the various SVC calls (supervisor calls) and what their properties are, and where they might go, so that I could faithfully emulate the execution of this program when it executes an SVC.

Hendrie: Did you have to do some testing to figure out what it did in some of the undocumented cases?

Shustek: I had to do some testing, but one of the wonderful things about the IBM operating systems at the time was that all of the source code was available on microfiche. So I would spend hours looking through the microfiche of the source of the operating system so that I could understand how it treated various cases.

Hendrie: Ah, you would take cases and then trace them through the source code, see what it did and where it went.

Shustek: As a result, when all that was done, I had something that could execute any problem state program -- not a test case, but any of the physics codes that physicists were running -- and understand the behavior of the program. From that we generated all sorts of statistics. We did things about the instruction set, which opcodes are used and not used, which branches are used and not used, what branch lengths were, what registers were being used. The 360 has a very complicated instruction set. It's the opposite of a RISC. What subset of those instructions actually got used, as opposed to what the manual said you could do? We did a lot of work about paging and working set design, and what the memory reference patterns of the machine were. In fact, I generated many, many mag tapes of various kinds of data, but in particular of address trace data, which I then sent to other researchers around the country to use in their own studies of cache design. They were able to use my tapes of address traces of real running programs to test other designs …

Hendrie: Their algorithms for building … for caching.

Shustek: … their algorithms for cache designs and other multi-level memory hierarchies. And that was gratifying that other people got to use my work.

Hendrie: Oh, that's great. Well after you've sent out those address traces, did you have contact with other people interested in this whole subject of the addressing, the different addresses and the locality of addresses, the things that would be interested in caching studies?

Shustek: I wouldn't say I had much contact; I had a little bit of contact. Alan Smith, who was a PhD student at the time, who went on to become a Professor at Berkley, did a lot of work on those traces, and I helped him some. But my thesis was pretty much self-contained. I did some address trace analysis, but I spent more time trying to analyze the instruction set of the 360, and understand it particularly for high performance implementations. What about that instruction set made it hard or easy to make really fast machines? As you know, IBM struggled for many years trying to build very fast machines. The 360/91 which we had at SLAC was one of their fastest at the time, but still didn't compete with the Cray-1's of the world, and machines like that. My conclusion after doing that analysis was that the instruction set of the 360 made it very difficult, with the technology of the day, to make high performance implementations, and that the best way to make very fast computers was to simplify the instruction set. What's more, most of the complicated part of the instruction set in the 360 never got used by programs. There were bizarre complicated cases of instructions that had to be interpreted by the hardware, but were rarely used by the programmer. If you simply removed those and did that sort of thing by standard software subroutines instead, you would wind up with a simplified instruction set that could be much more efficiently executed for high performance machines.

Hendrie: That could be implemented without all this complication, that could be then implemented with a much faster shorter data path, thus faster cycle times and executions.

Shustek: Exactly. And I've been told that the people who started the next generation computer architecture, the RISC generation, all read my thesis. I'm gratified by that. Lots of theses get written and never read; they are a sort of write-only medium. To the extent that mine was used by people, and influenced in some small way computer architecture, is very gratifying.

Hendrie: That's wonderful. Was IBM showing any interest in what you did, or were they very interested in it not being published?

Shustek: Well, interestingly enough, IBM independently had their own project, which had come to similar conclusions to do RISC architectures. That was being run by George Radin at IBM Research at Yorktown Heights in New York. Because of my interest in this area, one of the places I applied to to get a job after I got my PhD was IBM Yorktown Heights to work with George Radin in that group. They turned me down.

Hendrie: Now, was that the same group that John Cocke worked …

Shustek: Yes, exactly, it was the group that John Cocke was in, or part of, doing the 801 RISC-like machine.

Hendrie: They just …

Shustek: I guess they thought I wasn't smart enough. I don't know why they turned me down.

Hendrie: [Laughs] That's fascinating. Okay. What else can you tell me about the wind-down of your academic career at Stanford? What else went on?

Shustek: Well, one of the interesting things that I did while there, because of the involvement in starting the microprocessor lab at Stanford, was to get involved in the hobbyist activities surrounding microprocessors. In January of 1975 the cover issue of Popular Electronics touted the Altair, which was an affordable kit computer that anybody could buy and build, based on the 8080 microprocessor. One of the physicists at SLAC, Dave Gustavson, who's a really good friend of mine, decided for his physics group to order two of these machines, to experiment with using them, instead of the very expensive minicomputers, to do process control for the physics experiments. So we wound up very early on – "we" was really SLAC but I wound up working with Dave on them -- with two Altair's that we put together and did all sorts of software and hardware for. The machinist for Dave's group, Group F at SLAC, happened to be a friend of an engineer by the name of Gordon French, who decided that a lot of his friends were interested in his computers, and he would start a club for people interested in computers like the Altair. He held the first meeting in 1975 in his garage, in Gordon French's garage. A group of people, including John Grant, the machinist friend of Dave Gustavson. There were probably a dozen people in the room at the time. And that, as it turns out, was the founding meeting of what became the Homebrew Computer Club. Another SLAC'er and I, Frank Rothacker and I, arranged for the Homebrew Computer Club to meet in one of the meeting rooms at SLAC; originally the Orange Room, but then later the auditorium. For the next four years, the Homebrew Computer Club met at SLAC. Lee Felsenstein was the moderator. I think that history is known; it's been written about. It was one of the important early places where hobbyists shared information about small computers and how to build them yourself. It was one of the places where Bill Gates got angry. He wasn't there, but he got angry at the pirated software that was being distributed freely at the Homebrew Computer Club. People would stand and throw copies of paper tapes out into the audience. Bill Gates' Altair BASIC. Bill wrote an open letter that got published in the Homebrew Computer Club newsletter, decrying the activity of pirates -- software pirates -- and saying, "how do you ever expect high quality software to be written for microprocessors if the people writing them can't be compensated for their efforts?".

Hendrie: Very interesting.

Shustek: It was a fun time, and the Homebrew Computer Club lived on for many years after that.

Hendrie: Were you at that first meeting in the garage?

Shustek: I was at the first meeting in the garage.

Hendrie: Very good.

Shustek: We had a reunion of the Homebrew Computer Club, I don't know, about three years ago, something like that, which I guess might have been the 20th reunion of the founding meeting.

Hendrie: Now with your building… you built Heathkits and things like that in high school -- did you ever actually build one of these for your own personal use?

Shustek: Never for my own personal use, because the only thing I built for my own personal use was an ADM-3 Dumb Terminal Kit from Lear Seigler. But I didn't have any computers at home. I worked at SLAC and Stanford for 18 hours a day, and when I came home I didn't want to see a computer and didn't have any time to use one anyway.

Hendrie: Right, you had to sleep.

Shustek: So I had my fill of computers in my work life.

Hendrie: Okay, and obviously built all sorts of stuff.

Shustek: And built all sorts of stuff, at SLAC.

Hendrie: Yeah. So you had plenty of time on somebody else's nickel.

Shustek: That's right.

Hendrie: Yes, very good. Well, when you finished your thesis, you needed to do something else.

Shustek: That's right. The time had come..

Hendrie: What were the thoughts that went through your mind?

Shustek: Again, I had no plan for really what I wanted to do, or what my career was. I applied to several companies for industrial positions, and I applied to several universities for teaching positions. Eventually I decided that I spent all of this time getting my PhD -- and mostly what a PhD is useful for is that it's the union card you need to teach in a university -- so I decided I'd give that a go. I got an appointment at Carnegie Mellon University in Pittsburgh as a new wet-behind-the-ears Assistant Professor in the Computer Science Department.

Hendrie: Okay.

Shustek: I discovered that a PhD was a wonderful background for my field, which was computer science, but no background at all for what my job was, which was teaching. This was the first time I ever taught. I'd been a TA, but never taught a course. So I was now thrown in to an environment where I was expected to teach, and I didn't understand anything about how to teach, or what the theory of teaching was, or the theory of grading was. Do you grade based on some absolute standard? Or do you grade based on comparison of the rest of the students in the class? On comparing previous students in the same class? It was all a mystery to me. I put a lot of effort in that first year into trying to do a good job at teaching.

Hendrie: Now who hired you? Who decided, yes, you can be in my department?

Shustek: Let's see, who was there at the time. I remember both Joe Traub and Bill Wolf as department chairmen at various times. But I think who really hired me and had the most influence in my being there was Sam Fuller, who had also been a student of Forest Baskett's, and had gone to Carnegie Mellon University, and was about to leave to go, I believe, to DEC.

Hendrie: Ah, yes.

Shustek: They were looking for somebody to take his place. Part of what he was doing was managing some research projects that had to do with performance analysis of instruction sets. He had some contracts with the Army to select their standard instruction set for military computing. He was doing a lot of analysis similar to the kind I had done at Stanford about the performance of computers and the associated instruction sets. So I was a natural, I guess, to fill his position and to continue on those contracts, which I did. I slipped into his chair while it was still warm.

Hendrie: Okay, very good. And what did you have to teach, what were you assigned to teach, or did you get to -- you don't get to choose when you have no seniority do you?

Shustek: No, you don't really get to choose, you get to teach what needs to be taught, and …

Right. And what nobody else with higher seniority has said "that's what I want to teach".

Shustek: That's right. So I did the normal complement of advising graduate students, but I [also] taught programming courses.

Hendrie: Undergraduate programming courses?

Shustek: Yes, I guess it was the senior level undergraduate programming courses, using a textbook [being written by another faculty member].

Hendrie: And what language did they teach that in?

Shustek: It was Algol, actually, which was a little bit strange at the time.

Hendrie: This is what year?

Shustek: This was 1977. Algol was still in use, in some places.

Hendrie: Now, I'm interested in what companies you thought you might want to go to. What your vision was of places that might be interesting and/or fun, and/or remunerative, in some proportion, to go work for?

Shustek: I don't know all of the companies I spoke to. I remember three, one of which was very different from the other two. I applied to, as I mentioned, IBM, to work at the Yorktown Heights Research group under George Radin on the 801 project, I don't know why they rejected me. They told me because the project was winding down at the time, but I don't know whether that's true or not. They did make me an offer to work in the queuing theory department there under Kobayashi, because I had done some queuing theory analysis in my thesis of some parts of computer architecture, but I wasn't very good at that and wasn't particularly interested in that, so that's not a job that I would have expected.

Hendrie: It's a little theoretical.

Shustek: A little theoretical for me. although I was a little flattered that they wanted me there. I had applied to Intel to work in their advanced architecture group up in Oregon. I can't remember whether they made a job offer to me or not. I applied to, of all places, General Motors in Detroit because they had a research project on the use of graphics for design. So my work on the VGT, the Video Graphics Terminal, fit into some interests they had in doing graphics work for design automation. They actually made a job offer to me. But eventually I decided that I wanted to try academia. I did not apply to Stanford because I had been given some advice, which was probably good, that you shouldn't teach at the place where you got your PhD because they will always view you as a graduate student and not as a professor.

Hendrie: Right. You can come back after you've established your credential somewhere else.

Shustek: Right. You can come back later, which I subsequently did. But I went off to Carnegie Mellon to teach, and spent a lot of time learning how to teach. And at the end of my first year the Department Chairman said "You did a great job of teaching, and you got all these wonderful reviews [of your] teaching, but what did you publish?" And my answer was: very little. So I quickly realized that although they expect you to teach, what they grade you on is your research.

Hendrie: Yes, publish or perish.

Shustek: Publish or perish, and I hadn't done both.

Hendrie: So did you continue there?

Shustek: Well, I would have continued there. I was on a tenure track, and as a result of that first year review would have corrected my errors, and put more attention to…

Hendrie: You would have decided on something to publish about.

Shustek: That's right. I was still doing research in computer systems of various kinds, and would have put more attention into publishing. But what happened is that Harry Saal, whom I had worked with at SLAC, and who had gone off to IBM and lived in Israel for a while, returned to California. He was still working for IBM in their Santa Teresa lab, and was similarly affected by the new personal computers that were now beginning to appear. Not just the Altairs, which were really hobbiest computers that had to be assembled, but the ones that were ready-to-go when you bought them from a store; "plug and play". The Apple-2, the TRS-80, the Commodore Pet. He had this idea, having been exposed, as we both were, to the Internet and the advantages of networking computers, that these little computers would be a whole lot more useful if you could connect them together in a network so that they could share resources and share information. He had this idea to start a company, and convinced me to… Actually he had several ideas to start a company, but [for] the second idea, which was the one we eventually decided to start, he convinced me to take a leave of absence from Carnegie Mellon and come out for a year to help him start this company to do networking of personal computers. And then I could go back and resume my academic career. So I agreed to do that, and went to Bill Wolf, and Allen Newell who was still alive at the time, at Carnegie Mellon. I got their blessing to leave for a year, and left, and never went back.

Hendrie: Okay.

Shustek: The interesting company that Harry had tried to start before Nestar Systems (the one that did networking of personal computers) was actually a quirky company to do interactive gaming. He wanted to start a company called GameNet. The idea there would be to use personal computers as intelligent terminals connected -- over the phone line using modems -- to a centralized computer, for people to play games with each other who weren't in the same place. So you could play chess. You could play shoot-'em-up games. You could play all sorts of interactive games in a big network, using your personal computer attached to a central computer [that was] relaying information to other people's personal computers. It's actually a clever idea. I think it's been reincarnated over the years in different forms many different times.

Hendrie: Yes, certainly. People do do that today.

Shustek: They do do that today.

Hendrie: I don't know of a company that is based on that.

Shustek: Well, we didn't form a company either. We made a business plan. We shopped around for people we thought could provide financing -- we were looking for 750,000 dollars, or something like that -- and found nobody willing to fund us, for many good reasons. One is that none of us -- there were three of actually: myself, Harry Saal, and an administrator from Stanford that we both knew named Nick Fortis -- none of us had any experience in business. We were all academics, and had no idea what it was like to start or run a business. And the idea was probably not well formed either. But it was our first attempt at starting a company and getting funding for it. Then we decided to do Nestar, this networking company for personal computers, and tried to do it basically without any funding because it was clear that nobody would give us money.

Hendrie: Now who did you go to, where did you look for money?

Shustek: We looked to friends and relatives, and we looked to venture capitalists to let us in the door. I don't remember the list of people we spoke to. It was a long list. Nobody was interested, I think because we were not people with a track record.

Hendrie: Now had you moved to California to do this, or was this Harry doing this there, and you were still at CMU, and you'd move if this got going?

Shustek: That's exactly right. Harry was there, I was at CMU. I was doing my CMU work. In the meantime, on my kitchen table, I was designing and building modems for personal computers so that we could get GameNet going, because they weren't available inexpensive modems. I was designing modems that could be inexpensively manufactured that would plug easily in to these personal computers for GameNet. And that was a kind of natural transition to doing networking instead of modems -- local area networking for the same personal computers -- when we finally decided to do the networking company instead. When it looked like that was going to take off, I applied for the leave of absence, and came back to California to start Nestar.

Hendrie: Okay. So you did some of this exploration without even doing the leave of absence or anything?

Shustek: Right. In fact, Harry and I burned up the Internet air waves, so to speak, sending email back and forth to each other: he from IBM Santa Teresa, and me from Carnegie Mellon. In fact, I think I still have the typescripts of our correspondence over that summer, trying to talk about starting these companies.

Hendrie: Oh, that would be probably pretty funny. Okay. So you decided to go do it. Now you had to have some funding?

Shustek: We had to have some funding. I had no money. My net assets were, if you generously include an allowance for my Dodge Dart, about 8,000 dollars at the time. Harry had some more money, so he put in the money to start Nestar. There were two other founders: Nick Fortis, who was the administrator from Stanford, and a fellow that Harry had known while he was teaching for a year at State University of New York in Buffalo; his name was Jim Hinds. He was a good programmer/software engineer. The four of us started Nestar Systems. We rented a little office building in Palo Alto, and started doing both hardware and software and design of our own networks for personal computers.

Hendrie: From scratch.

Shustek: From scratch.

Hendrie: Now were you patterning them on anything? Had Ethernet become something at this period? Had the alliance between 3-Com, Intel, and DEC to try to make Ethernet happen…?

Shustek: Well, it was originally DEC, Intel, and Xerox, the three companies that did the ethernet standard. That was the three-way alliance. Certainly that had not happened [yet]. I can't remember whether Ethernet existed or not. It probably did exist in some form. I think probably by then -- this was 1978 -- Xerox PARC had the three megabit per second version of the Ethernet. But the implementation of that was of a scale that was not appropriate for small computers. It took a big board of electronics to implement Ethernet. We were looking for something that could be fit on an Apple-2 board, or on a little plug-in board on the side of the Commodore Pet, so it had to be something really, really cheap. So we invented, in fact, through the history of Nestar, several different networks. One of which Harry got a patent on, a very unique kind of addressing for a local area network so that individual stations did not have to be assigned an address. It would be auto-configuring based on the position of the station in a linear string. Each station would be given a unique address. It had to do with cables that had interior twists in the cable; a permutation of the wires. That was Harry's idea. He got a patent on that, the only patent that Nestar ever got. And we implemented that and used it for our first version of our network.

Hendrie: How fast did it go?

Shustek: I think it was 200 K bits per second.

Hendrie: Okay. So it was relatively slow.

Shustek: No, I was about to say it was relatively fast compared to modems of the day!

Hendrie: Yeah, compared to modems, it was very fast.

Shustek: Compared to Ethernet it was slow, but it was very inexpensive. The implementation for the Commodore Pet actually had no hardware except for connectors and wire twists, because the Commodore Pet had an IEEE-488 port sticking out the back of it, and you could program that to do all sorts of level changes on the individual wires. So we did the implementation of the network protocols in software at the data link level. So for the Commodore Pet, the board was a ten dollar board.

Hendrie: And how many wires in this?

Shustek: Let's see. It had eight wires for the addressing, in a group of three, and a group of five. Two relatively prime numbers, which gives you a clue about how the addressing might work. And then eight wires for the data, so it was a 16-wire flat cable that carried the data.

Hendrie: Okay.

Shustek: Eventually we moved actually to a CSMA type system, an Ethernet like "carrier sense multiple access with collision detect" kind of system, again using a parallel cable -- a flat cable -- rather than a serial cable that Ethernet was using. So we could do a lot of the protocols on a byte by byte basis in software rather than building special purpose hardware, and that kept the cost down.

Hendrie: Very good.

Shustek: Of course, that was the communications level. Then we had to build all the other stuff. So we adopted the Xerox Network Systems protocols, eventually, XNS. It was that or TCP/IP, and we made the wrong bet.

Hendrie: But it took a very long time for that…

Shustek: Well, Xerox was actively promoting the XNS protocols. They were publishing papers out of Xerox PARC explaining all the protocols, and trying to get people to adopt them. We thought, “they’re smart people, this is good stuff – we’ll use it”, which we did. We built servers out of machines like Apple-2’s We built file servers, we built print servers, we built file transfer servers that used modems to connect remote networks to each other, and did automatic gateways for electronic mail. I wrote an email program, so we had an email system running. We had lots of great early software. And no customers.

Hendrie: Basically, it was hard to sell?

Shustek: It was hard to sell for a couple of reasons. One is: remember, this is before the IBM PC. Businesses didn’t treat any of these machines as serious computers. We couldn’t get a business to buy Apple-2’s or Commodore Pets or TRS-80’s, because they were toys, they were hobbyist machines. They bought “real” computers. We had some modest success selling to schools, who used them for kids for training and games and whatnot. But schools didn’t have a lot of money. They might buy one or two of these personal computers, but they weren’t about to buy -- at least not too many schools -- weren’t about to buy 15 of them with a network and file server to connect them all together. So we had not a lot of success.

Hendrie: So, you built all sorts of stuff, you got it out there, you tried to sell it, and …

Shustek: Part of our problem was that we made many mistakes. Part of it was we bet on the wrong technology several different times. We bet on the wrong technology for software standards: XNS instead of TCP/IP or even Novell standards. We bet on the wrong technology for hardware networking a couple of different times. We did two versions of our own, then we decided to adopt a standard. It was a company in Texas that was promoting their internally-generated network as a standard. They were, what’s even more exciting, building integrated circuit controllers so that a one-chip controller could implement all of the low level protocols. So we said “Great, we’ll adopt that”. That turned out not to be Ethernet either. That was DataPoint’s ARCNET, which was a wonderful system. It had many attributes to it that recommended it over Ethernet, but obviously was the wrong choice because Ethernet became a standard and ARCNET did not.

Hendrie: Right.

Shustek: At about the same time, Bob Metcalfe was starting 3Com. In fact his original office for 3Com was his apartment on Sandhill Road where we visited him. We tried to hire his chief engineer, Ron Clark, to be the hardware engineer for Nestar, but lost out to Bob and 3Com. Networking was sort of in the air. Everybody was struggling to find the right standards and the right protocols to use.

Hendrie: That's interesting.

Shustek: So we switched to ARCNET because we thought that would be the next standard, and it wasn't. And then IBM decided that they were going to provide the standard for networking. They, by this time, had the IBM PC, so the PC was legitimized as a device that businesses could buy. We figured if they could standardize the PC, then by God they were going to be the one that were going to standardize networking. So we adopted their network, which was the IBM Token Ring.

Hendrie: So you hopped from..?

Shustek: We hopped from ARCNET to Token Ring.

Hendrie: But previously you'd had your own proprietary eight-bit..

Shustek: We went from two versions of our own proprietary network to two different networks that promised to be the future standards for networking: one was ARCNET and one was Token Ring. And of course, as you know, neither of those became the standard for networking.

Hendrie: Now at some point you must have raised some more money somewhere along here?

Shustek: We were raising some money. I have to remember how that went. We'd gotten an investment from The Rank Corporation, the British company who owned the Xerox subsidiary in Britain. So Rank Xerox was the major investor in Nestar. That was facilitated by a fellow I knew at Stanford named Skip Stritter, who had worked for Motorola, for Colin Crook.

Hendrie: Yes, I know him.

Shustek: Colin Crook went back to England and became involved with the Rank organization. And the connection between Skip and Colin got us funding from the Rank Organization, from Rank Xerox.

Hendrie: Yes, and then he went came back and went to Data General.

Shustek: Colin Crook did. He also worked for Citibank -- Citicorp I guess -- as their networking guru.

Hendrie: Yes okay. Very interesting.

Shustek: Skip Stritter, of course, had been a fellow graduate student of mine at Stanford under Forest Baskett, then went off to Motorola and designed the 68000 at Motorola. Then when he left Texas and Motorola, we hired him at Nestar to be our architect of advanced products. So he worked for me at Nestar for many years and got us the funding from Colin Crook and Rank whom he had met at Motorola.

Hendrie: Oh very good It's all very incestuous. That's fascinating, I didn't know that's what Skip did after the 68000.

Shustek: It's the people networking story. Skip left Nestar, and I don't know if it was immediately thereafter but soon thereafter joined MIPS and was one of the founding members of MIPS, doing reduced instruction set computing.

Hendrie: Right, exactly.

Shustek: We ran out of money eventually at Nestar. We got some money in from a Texas corporation, and eventually we sold the company. We had built it up to about 125 or 130 people and we sold it to DSC, Digital Switch Corporation, of Texas. They wanted to be in the networking business. Harry and I decided that we didn't want to be part of the Texas environment and decided to leave, but negotiated our leaving. We thought that the Texas investors were doing things which were borderline unscrupulous, and decided that we wanted to get out. The terms by which we got out was that we would forfeit all of our shares in Nestar in exchange for the rights to an internally generated test tool that we had done in the engineering department to help us diagnose problems with networks. We called it The Sniffer. It was done originally for ARCNET because that was the network we were using at the time. It was basically a promiscuous packet receiver that would watch the communications between computers, and have software that would display at a relatively high level what was going on. You could use that to diagnose problems. People before that really had no clue how to diagnose local area network problems. We had done a little bit of experimenting with building the Token Ring version of that, because we were in the process of switching to Token Ring, when DSC bought the company. Harry and I decided to leave, and we got out of them, in exchange for all of our shares and an agreement not to sue them, the rights to The Sniffer. And Harry and I started all over again.

Hendrie: And that was your plan? When you did this agreement, that this was the plan?

Shustek: Right, that was our plan. We decided that an easier sell was to build test equipment for networks rather than build networks themselves. We had realized how hard it is to get these damn things to work. It wasn't easy. IBM and Ethernet companies were going headlong into building these very complicated systems with very little idea of how to diagnose problems. We thought we could capture this niche market. At the time we thought if we could sell 100 Sniffers, that would be great, and in the meantime we could figure out what we really wanted to do. Did we want to start another company? Or did we want to go back to academia? Or industry?

Hendrie: Yes, what do you really want to do when you grow up.

Shustek: That's right, what we really wanted to do when we grow up. I basically wrote The Sniffer. There were only two of us; it was Harry and me. We had a few consultants along the way eventually, but first it was just Harry and me. I was the engineering department and he was everything else. We put our own money into it -- what little we had. We got some free offices in the back of Frederico Fagin's company Cygnet, which was being shut down at the time. We built the first Sniffer, which was a Token Ring Sniffer, based on a Compaq portable luggable computer. I think its biggest advantage was that it was supremely easy to use. It was a device. You plug it into the wall with a power plug, you plug it into the network with the network connector, and you press go. It generated these wonderful screens of high level information about what files you were trying to open on which computers, and [helped] diagnose the problems which, by this time, were almost always high level software problems. They weren't "I can't send a packet from A to B"; we'd gotten that right. The problem was at a much higher level in the networking hierarchy and The Sniffer was designed to diagnose those kinds of problems.

Hendrie: As opposed to problems in the physical layer?

Shustek: That's right. And about the only disagreement Harry and I ever had was what the appropriate price was to sell this thing at. The cost of goods was, maybe, $2,000 or $3,000. It was mostly the cost of computer, the Compaq computer.

Hendrie: Right, with a little bit of interface?

Shustek: With a little bit of interface, a special-purpose logic board that we plugged in. For some networks later on, not even a special purpose board [was needed]; the standard boards could do promiscuous reception. So the cost of goods was fairly low. I figured you double the cost of goods -- if it cost us $2,000 we sell it for $4,000. Harry said, "No, the price is $20,000." I looked at him and said, "What? You're crazy. Nobody will spend $20,000 on this." And he gave my first lesson in marketing, which is value pricing. You price it based on what the value is to the customer, not what it costs you to make. His concession [to me] was that, okay, he would try it out -- special one-time-only for a three month introductory period -- $15,000. And we sold all we could make at $15,000. The price went [back] up to $20,000, and we were cranking them out the back door, selling mostly to developers. This was the

Token Ring version, remember, and IBM was proselytizing like crazy trying to get people to design hardware and software systems using Token Ring, and having a devil of a time getting things to work. We were selling to developers who couldn't get their systems to work, who wanted to buy a tool that would help them. Harry's son [about 15 at the time] was our "vice-president of manufacturing". Manufacturing was simple, because was we got the boards manufactured elsewhere. We'd plug them in the Compaq Portable machines, and stick labels on the front of the Compaq saying "Network General Sniffer". We were off and running. The company name was funny too. We decided we would call ourselves "General Network". We went to Sacramento and found that we couldn't get clearance for that name, so on the fly we made an instant decision to, okay, try the opposite. If "General Network" wouldn't work try "Network General". And we became Network General Corporation.

Hendrie: While you were in Sacramento? Yes this is real time: "let's try this". That's fascinating. What year is that?

Shustek: That was in 1986. in the summer of 1986 I believe I did most of the engineering for that original Sniffer, into the fall of 1986. We probably shipped our first machine in November 1986, something like that. Our first customer was Bridge Communications.

Hendrie: And it grew and grew and grew.

Shustek: It took off from there. We started hiring people, we were--

Hendrie: So you had no venture capital in this either?

Shustek: No, no venture capital. We had just come out of eight years of working on a company which we had sold, but was basically a failed company. We didn't even try to get venture capital. We knew at that point nobody would fund us, so we put $10,000 apiece in it, and took no salaries, and had no employees, so we really didn't need any money except to buy inventory. And the first few $20,000 Sniffers paid for enough inventory at $2,000 a pop for the next 10. So it was a nice cash-flow-positive kind of business from the beginning. Then we branched out. Token Ring was the first Sniffer. Then I think in the late spring or early summer of 1987 I did an Ethernet version, because clearly Ethernet was becoming the established standard. We just cranked them out after that. We did more and more protocol interpreters. The big value added, in addition to the simplicity of use, was that we committed ourselves to writing high level interpreters for all of the crazy protocols that were being proposed and implemented at the time: IBM SNA, Banyan VINES, Xerox XNS Networking Standards, Corvus protocols, Novell Netware. You name, it we did it. We had 125 protocol interpreters, all built into the machine. A sort of semi-intelligent algorithm tried to figure out what it saw, so you didn't even have to tell it what you were running on the wire. You just plugged it in and it would figure it out. Which was great, because lots of people had multiple programs running on the wire and didn't even know it. It was really fun to watch what people didn't know about how their networks were run. We would take it into an IT manager and plug it onto his network -- if he would let us -- and that would almost always guarantee a sale, because he would see stuff going on in his network that he had no idea was happening.

Hendrie: Yes, it's fundamentally invisible.

Shustek: Security violations, passwords that were not encrypted, that were in cleartext, that might have crept into your network.

Hendrie: Well, that's very good. Alright, you were talking about Network General and its success. You've just kept pumping them out the door, adding protocols. Did you charge extra for the protocols, or did you just get more and more value for your $20,000?

Shustek: I think you just got more and more value for your $20,000. Eventually the price began to came down, but we actually kept a high price for quite a while because there was no competition. There were a few other companies building similar devices. HP was one of our competitors, probably our best competitor in the sense that they're a very reputable company and very honorable competition. They don't misrepresent what they're selling, and we could easily do better than they did. But there were lots of other little competitors as well. We got about 50% of the market share, I think HP got about 20% of the market share, and then lots of little people after that, for these kinds of instruments. Our success was pretty dramatic. We started the company in 1986. In February of 1989, when we had 29 employees, we went public.

Hendrie: Now what were the revenues; do you remember a little bit about what happened…

Shustek: I would have to do some research, I don't really remember the revenue. I don't even remember how much money we raised in our public offering. By today's standards I'm sure it was a small amount of money. But we were lucky, because by that point we had a reasonable revenue stream, we were a profitable company, and we dominated the market for our particular product. So it was a good profile and we were able to go public and get a good valuation for the company. Harry and I and the employees that had stock options were able to retain a fair bit of the company. We did our IPO in February, and then in August the market conditions were good, we did a secondary offering as well.

Hendrie: So now you had plenty of cash to go and expand.

Shustek: I should point out that just before we went public we did accept some venture capital money, mostly as a way to position us to go public. We got $2 million from TA Associates probably six months before we went public, or maybe a little more than that. We didn't really need the money, but we thought since we were doing well, we could get the money for a good valuation at the time. It would be good to bank it. And also, they would help us go public, which would have been something that with our experience and connections it would have been much more difficult to do otherwise.

Hendrie: Who did you work with at TA?

Shustek: Mike Child was on our board from TA, for a long time, until eventually they sold their shares. He actually added a lot of value as a board member. We had a good experience with venture capitalists at that point in our careers.

Hendrie: Did you have other people on your board?

Shustek: We had Greg Gallo on our board, who was our lawyer from Grey Cary Ware and Friedenrich, which was then just Ware and Friedenrich. We had [Larry Hootnick from Maxstore and Howard Frank] on our board for a while. I can't remember all of our board members, but we did have outside board members.

Hendrie: Yes, so over time you built a reasonably strong board with outside board members. Trace some more of the path of Network General. What happened next? You presumably were still basically a one-product company when you went public?

Shustek: That's right. We were certainly a one-product company, with variations of networks and protocols, when we went public. We broadened some in the years that followed. We had Distributed Sniffer systems. We became known as "The Sniffer Company". In fact we debated whether we should change our name to "Sniffer". But the company was Network General and Sniffer was the product. Everybody knew it as "The Sniffer" and we fought pretty hard to keep the word Sniffer from becoming generic. We had a trademark on it. We ran full page ads in the Wall Street Journal saying, "Sniffer (Trademark)" with a little arrow that said, "This little TM is brought to you by Network General Corporation." We were trying to protect our trademark so that it wouldn't become Kleenex or Scotch Tape. I think we did do that reasonably well for a while.

Our product line gradually became somewhat broadened. We had Distributed Sniffer systems, because networks became complicated with lots of segments interconnected. We had these little black boxes built on PCs that you would throw in the wiring closet and attach to a segment or multiple segments. Then someplace else you would have the user's console -- the administrator's console -- that would poll all these various remote distributed Sniffers and see what they were doing, and use them as proxies to do experiments and that sort of thing. A distributed diagnostic tool system. Those were our products. We were into the tool business -- the diagnostic business. We did not want to get back into the networking business; there were too many of those companies. One of the things we liked -- I particularly liked -- was to be in an area where there wasn't much competition. Where it really was easy to dominate a small market. We were a big player in a small market. Maybe it goes back to my Brooklyn Poly days when I was a big fish in a small pond, as opposed to my Stanford days when I was a small fish in a big pond. [Laughs]

One of the things we did very early on -- and this entirely is to the credit of Harry Saal -- at Network General was to establish a philanthropic program. When we were four employees in December, I think, of 1986 -- that was me, Harry Saal, Harry's wife Carol who went under her maiden name as vice-president of marketing, Carol D'Esopo, and George Comstock, who we had hired as our first VP of sales. We were four employees and Harry said, "Okay, it's time for us to start our corporate philanthropy program." And George and I looked at him and said, "You're crazy. We have no revenue, we have no profit…"

Hendrie: Yes, we've got no money. <laugh>

Shustek: Right. What are we going to do? And he said, "This is the time to start." He proposed that we establish a precedent of donating, at the end of every calendar year, $1,000 per employee to charities that we would select. "We", the employees. So at the end of 1986 we donated $4,000 to four different charities; $1,000 each. We kept that going until Harry and I left the company, when we had 1,000 employees and were giving away a million dollars a year entirely to charities that our employees were recommending. We gave precedence to charities that either they benefited from or that they were involved in, that they volunteered for. It was wonderful. Employees really appreciated being employees of a company that would do that, number one. But number two, being involved in the process of helping to decide where the company's philanthropic budget would go. I'm very pleased to have done that.

Hendrie: That's very interesting.

Shustek: It's entirely to Harry's credit that we did that.

Hendrie: Did he have a philanthropic background?

Shustek: I'm not sure where that came from. He certainly taught me about philanthropy, and whatever success I've had in giving away money since then has been due to his acting as a role model in that regard. He came from a family of slightly better means than mine, as his father was an osteopathic surgeon in New York, in Brooklyn. I don't know whether his parents were particularly philanthropically inclined, but certainly he and Carol have been, and have taught others how to do it.

Hendrie: That's wonderful. Do you know of any other companies that have gone and done that sort of thing?

Shustek: There are certainly other companies that give away money. There are lots of big examples. I don't know if there are any examples of companies that started that small to give away money. Certainly HP gives away lots of money.

Hendrie: Or that model, which is based on number of employees and has this employee input into it -- that sounds unique to me.

Shustek: I've not heard of that being done before.

Hendrie: Very good. So, when did you start thinking about maybe you didn't want to work for this ever-larger company?

Shustek: I think the company was gradually getting too large for me. I realized by then that I'm a guy who likes small start-ups and hands-on activity. I was VP of engineering, and I didn't get to write as much code and design as many circuits as I really liked to. And we were growing by acquisitions. We were acquiring other small companies to grow the company, in some cases, frankly, to take out competition. We were buying companies that were doing similar things, that had value to add, but were also synergistic.

We were over 1,000 employees at the time, and I wasn't having as much fun. I'd been doing it for seven years or something like that. I realized that my cycle time is about seven years: seven years at Stanford, seven years at Nestar, seven years at Network General. So I pulled back to a part-time position. I started by working four days a week and one day a week at home, then started working four days a week only, and then three days a week, then two days a week. We eventually merged the company with two other relatively large companies. One was McAfee Associates, and the other was PGP. It was almost a three-way merger. Network General, McAfee Associates and PGP (Pretty Good Privacy) merged, and the result of that was Network Associates, "Network" from Network General and "Associates" from McAfee Associates. The fellow who was put in charge of that merged company was Bill Larson, who had been the CEO of McAfee. For me at least, that was the end of my association, because I didn't like his style. I didn't like his values, I didn't like the way he was running the company, and I left. I left by becoming well and fully retired, as opposed to partially retired, semi-retired.

Hendrie: Now when was this?

Shustek: This was about 1994 or 1995. Probably the fall of 1994. That's right; that's when I took that my big scuba diving trip to New Guinea.

Hendrie: Oh. Allright, well, yes, you could go do something like that.

Shustek: Well yes. I could something now; I didn't feel guilty. Throughout the time at Nestar and Network General I was a workaholic. I was single, and I worked six or six and a half days a week, eighty hours a week. I didn't take any vacations, and sort of had this pent-up relaxation. So I took a few months off and learned how to scuba dive and did scuba diving trips.

Hendrie: So when you left in 1994, it was 1,000 people or.. Well, because of the acquisition--

Shustek: It had been about 1,000 and merged to something bigger at that time. So I was out. I decided that one of the things I thought would be fun to do is to reprise my original career as a beginning professor, and to go back to teaching. I was now fully committed to being a Californian by this point. I went to Stanford, talked to Ed McClusky and said, "Any opportunities for teaching?" And he said, "Yes, of course." So I taught computer architecture as a consulting professor at Stanford, which was a fun experience.

Hendrie: Very good, that would be very interesting.

Shustek: In fact, I taught the same course, by number, that Ed McClusky had taught me 20 years earlier, EE282, which is the second-level graduate course in computer architecture. Of course, because of the intervening 20 years, there was [only] about 5% overlap between what Ed McClusky taught me and what I taught.

Hendrie: Things had moved a little bit.

Shustek: Yes, things had changed. My course was mostly the design of microprocessors, which of course I knew very little about. So I had to take a crash course in it myself. I was always two or three weeks ahead of the students in designing microprocessors. It was hard. Teaching that four unit course was basically a fulltime job for me, or almost a fulltime job for me.

Hendrie: Yes the first year at least, yes.

Shustek: Yes, the first year. And that was the only year I taught it; I never taught it again. I have made an academic career of never teaching the same course twice, which is a lot of work and is the wrong thing to do.

Hendrie: People with real academic careers I don't think do that. <laugh>

Shustek: I don't think so.

Hendrie: That's very hard to do, and to find the time to publish maybe.

Shustek: Teaching at Stanford was one of my part-time activities. Some friends of mine and I also started a little "angel" financing group. We put together our own money and started a little fund of about $10 or $12 million dollars called VenCraft to do start-up financing -- seed-round financing of small companies. The jury is still out, but it looks like we did a very poor job. We learned how difficult it is to be a venture capitalist. Betting on good ideas and good people -- or people you think are good -- is not necessarily good enough. You still have to play the odds, and with a $10 million fund you couldn't make enough investments to broaden the base and play the odds. Probably it was not financially a wise investment, although we certainly learned a lot. Teaching at Stanford is what got me into the computer history business.

Hendrie: Yes, now that's obviously the next thread I'd like to cover.

Shustek: What happened was that when Ed McClusky taught me computer architecture -- and the rest of my fellow graduate students -- he taught it primarily out of the Bell and Newell book ["Computer Structures"], which contains reprints of original papers about the design of computers: design of the Atlas, the design of the Stretch computer and all that stuff. We learned how to build computers by studying the design of historical computers. Fast forward to 1995, and by and large that's not the way computers are taught anymore. You learn how to design computers by looking at the latest and greatest microprocessors. You see how they're designed, and pay only the slightest amount of lip service to what went on in the last forty years in computer design. So what I decided to do, as a modest step toward some historical content, is to spend the first five minutes of every class in what I called my "show and tell" period. I would bring in a computer artifact -- I had some of my own -- a core memory module, or vacuum tube module, or a tape read only storage device -- and talk about the technology, and the computers that they appeared in, and the design of those ancient computers. Much to my surprise, I discovered that students liked it, they were interested in it. They wanted me to keep doing it at every class session, which I often did.

When that course was over, or during that time, I got to thinking: Why is there no historical museum of computers in Silicon Valley, where I can tell my students to go see the stuff which they were clearly interested in? I decided, okay that's a great retirement project: I would start my own computer museum. So when the class ended, I did not sign up to do another class the next semester. I decided to look around at other examples of computer museums, and low and behold couldn't find any. I went to Smithsonian and met David Allison, and discovered they had a very aging exhibit of computer technology and some collection that David had not the foggiest notion of what was in. I went to the Deutches Museum [and the Science Museum in London]. Eventually I wound-up in Boston, at The Computer Museum. Harry Saal had wrangled me an introduction to Gwen Bell, whom he had met. I discovered that they were really the only credible museum of computer history, and that they had wound up with two separate missions. One was computer history, which was the original one that Gwen and Gordon Bell had started. The second was what they had been led into doing by the exigencies of funding, which is kids' education about how computers work. That is a wonderful thing, and every city should have a science museum that educates kids about how computers work, but that wasn't a history museum. Gwen and I, and later Gordon [Bell], started talking about how it be great if we could reinvigorate the original mission of the Computer Museum, and that maybe the right place to do it 20 years later was not in Boston but in Silicon Valley. Why don't we start up an outpost on the west coast, to be the computer history mission of The Computer Museum in Boston. It had already one west coast employee, Carol Welsh, whose job was primarily to do organizing for the Computer Bowl, which is a bi-coastal [fundraising] event. She was based out here working out of her home, and I sort of joined with her. Gordon and Gwen at this point were bi-coastal; they were spending part of their time in Boston and part of their time in California. And we started noodling about how to start a computer museum in California.

Hendrie: And eventually made a start with a separate division of…

Shustek: It was a subsidiary of The Computer Museum. How did this happen? I had known John Gee, who ran, for NASA Ames, a high-tech incubator being paid for by NASA, called the Ames Technology Commercialization Center. He had office space that NASA was renting, and we needed offices. So, I went to John and said, "How would you like to incubate a company that is <u>guaranteed</u> never to make a profit, because it's a non-profit [corporation]?" And he sort of laughed, and said, "Sure, I have some spare offices." He gave Carol and me two offices in Santa Clara. That was the beginning of our association with NASA. We eventually moved into offices that the ATCC had at Moffett Field. Then we began to make acquaintance with the Moffett people, and prevailed upon them to give us warehouses to store the collection. Once we had that, we started loading up the 28 wheelers in Boston and pulling the stuff out of the back rooms where Gwen had hidden away all of the historical artifacts, and moved them out to these aging warehouses at Moffett Field, where we now sit.

Hendrie: Yes, very good. Alright, that’s excellent.

Shustek: So that is my story.

Hendrie: Good, well thank you very much.

END OF INTERVIEW

