Oral History of Frank Gerald (Jerry) Snyder

Interviewed by:
Dag Spicer

Recorded: November 16, 2015
Mountain View, California

CHM Reference number: X7682.2016

Dag Spicer: Okay. So today we're here with Jerry Snyder. This is November the 16th, 2015. I'm Dag Spicer, senior curator at the Computer History Museum. We're delighted to have Jerry here today. Jerry has worked on some very interesting computer designs as both a logic designer and an architect and manager, a team leader, and we're going to hear his life story and learn a lot about computing history through that story. So before we do that, Jerry, a little while ago you donated a very generous collection of brochures of computer systems to the Computer History Museum, many boxes' worth, and I was wondering if you could describe to us a little bit about what was in that donation, why you thought the items you selected for that donation were important, and any other details, like how you came across these, how you came to collect them and so forth.

Frank Gerald (Jerry) Snyder: I was in the computer development business, commercial computer business for quite some time, and I was usually the leader. And in doing that I would go to computer conferences and meet other people that were in the business also, and collected brochures. I made it a practice to try and get information on what were essentially my competitors at that particular time, and I collected their brochures. I filed them by company name, and over the years I just kept them. Even when I was at Teledyne in the latter part of my career I had them in my office. I was the computer architect guru at that particular time and I would do computer system development. Anyhow, I thought "gee those brochures are just too good to throw away". I was cleaning out my garage, so I thought perhaps the Computer History Museum could use them for computer history of minicomputers of 1950 to 1970.

In my career, particularly when I started, I had to learn a lot about computer design. I took courses at UCLA Extension School. One of the very early courses I took was on Boolean algebra. UCLA offered great night classes for people like me, and there were a lot of us in Southern California. That's where I was introduced to Boolean algebra, which I used in my logic design. I was one of the guys that did equations, and that really was an advantage because designing a computer with equations would allow me to use large-scale computers, IBM computers, to help with the development. A computer could go through the logic equations and make a wire list and loading analysis .Even later a computer could design a whole PCB, Printed Circuit Board, and interconnect the signals right on the board. The computers did that better than people.

I didn't work on a large-scale [computer design] but, boy, did I use the IBM computers as a design aid. In those days the mini-computer that you're designing was memory limited. It couldn't do compiling or assembly of software. You'd have to go to the IBM machine. At first we'd come out with a deck of cards with our program on it, and then later on magnetic tape or paper tape.

Spicer: Just to get back to your brochures for a moment--

Snyder: Sure.

Spicer: Did they inform your work at all? Did they guide your new designs in any way? Did they give you a target to shoot for, for example?

[EDITED 02/17/2016 FGS]

Snyder: Yes, because at that time there was a lot of computer analysis in terms of performance and performance comparisons, and I brought some information on that. Here we go. “Comparative Criteria for Minicomputers.” This is a very old document and it shows how to compare one computer from another. And I’ll leave that with you for your file, your archives.

Spicer: Okay, so this is a reprint of an article from the Journal of Instrumentation Technology by James Butler, and we will add this to the museum’s collection. It’s called “Comparative Criteria for Minicomputers.” Thank you.

Snyder: Throughput was a major thing, calculating the throughput. And there are many equations on how to do that. One of the more reliable and simple ones was: assume that 20 percent of the instructions are multiply instructions, and all the rest are single operations. That means 20 percent were 30 microsecond instructions and the other 70 percent were 10 microsecond instruction execution time which was typical in those days. This yields a throughput of 77 thousand instructions per second.

Spicer: Is this with a dedicated multiplier or done through successive addition?

Snyder: Dedicated hard-wired multiply.

Spicer: OK, well I think that’s a good discussion of the brochures. Shall we begin the interview? Why don’t you tell us a little bit about where you grew up and who your parents were, what they did and what your school was like, that kind of thing?

Snyder: All right, well, I’m an Oregon boy, and I had a really nice childhood because I had an identical twin brother to live with and be with. We gained instant popularity wherever we went.

<laughter>

Snyder: --and that’s where my name Jerry came from. His name was Tom. We were twins named Tom and Jerry.

<laughter>

Snyder: This was a Holiday Drink. We were born January 3rd and that’s why they named us Tom and Jerry.

Spicer: That’s wonderful.

Snyder: But, anyhow, we had a very nice young life and had leadership roles. Tom, my twin brother, was student body president. I was senior class president, and this was at a little high school, St. Helens, Oregon, during the war. We went to high school when World War II was on. We’d started in ’42 and got out in ’46, and we went directly into the Navy from high school. There’s the start of my career in computers because I was sent to the Navy Electronics School. I was in the Navy for 22 months, and 12 of those months was going to school learning electronics, because I came out of that with a ranking of electronic technician mate, and I would repair ship sonars and radars and transmitters, and so I got into electronics that way.

Spicer: Right.

Snyder: I built a radio, a superheterodyne radio, just to learn more about electronics. Being in the Navy, it was part of my plan to get the GI Bill. I went to the University of Oregon only because I knew from a child that I was going to go to that university. Well, I didn't realize then, but I should've probably gone to Oregon State and majored in engineering, but I didn't. I majored in mathematics, and I took every mathematics course they had to offer and I did very well grade-wise.(I later got my master's degree from USC attending night school .) And with that mathematical background, and with that military experience in electronics I was ready for employment. A team from Northrop came up to the university campus in Eugene and set up a card table under a tree and interviewed me, They thought I was highly suited for a job with their company. Jobs were hard to get in 1952. I graduated, got married, moved to Southern California, and took my job with Northrop. Now I didn't know until I got there that Northrop was one of the pioneers and leaders in computers, and they were working very closely with IBM to develop the Card Program Calculator (IBM CPC).

Spicer: Right.

Snyder: **"**C" used to always stand for "calculator" in those days, not "computer." The CPC was a very good computer, although card-programmed. You executed cards and, of course, they had data on cards, too.

Spicer: Did you use a CPC in your career?

Snyder: I did.

Spicer: Can you tell us a bit about what that was like to use?

Snyder: You'd write your programs on a big sheet with 80 columns and had girls keypunch them, take the cards, that's your program, run them through the card reader, and at the time they're running through the card reader they're being processed. And what was driving it was a huge patch panel, and the patch panel was the key to the operation performing simple instructions from the cards. A fellow named John Postley did that development, and his assistant was named Dick Knowles, I tried to learn a little bit about it, but I just was a user. I wasn't a developer. So we used it for simple functions. It was not capable to process my logic equations or anything like that. That took a bigger computer and that was later. We used it to develop guidance tapes for a special navigation system that went on the Snark missile--it was a stellar-referenced platform, and there were telescopes on the platform to keep stars in sight as the missile went along its trajectory.

Spicer: So this missile used the stars--

Snyder: Used the stars to keep the platform level. And if the stars went out of view of the telescope, then their stepper motors would move it back. This guided the aircraft along its track.

Spicer: Now does that mean you could only launch this at night?

Snyder: Well, at high altitude you can see stars during day or night.

Spicer: Was it a ballistic missile?

Snyder: It was, but we were using a B-25 with a hole cut in the top with our platform mounted through it for test purposes.

Spicer: Oh, I see. Very interesting.

Snyder: And when that would test fly people like me and others--this was at night--would be in the monitoring room, and we would take handwritten geodetic positions of the plane as received from radars down the line. And why I mention that is because that data plus much of the test data that was taken on ocillographs came back to our company. Ladies, girls, would read them, read the data, in sequence as it went down the track, and that information they would keypunch into cards. My job as a performance analysis guy and with my statistical background from was to develop sampling techniques and the measures of performance, you know, like averages, standard deviations and curve fitting. I used IBM computers.

Spicer:. Interesting. And what happened to the Snark missile?

Snyder: Well, it never worked too well, but one time we were testing the real Snark missile, not the B-25 version and it took off and went way off track down into the Caribbean.

Spicer: Oh, my. This is from California?

Snyder: From Hawthorne, California. That's where it took off from. It crashed down there, and they had a hard time finding it. I think it did crash into the Caribbean Sea. And then one of the funny guys would say, "Now that's Snark-infested waters--"

Snyder: And so that's what happened to Snark. It just wasn't quite good enough to go into production.

Spicer: Did it use inertial navigation?

Snyder: Oh, yeah. It was inertial navigation keeping that platform level, but it had to keep that platform moving, and we did that by tracking stars and making sure that the stars were in the field of view, and if they were out they would move it and that would move the platform back, because that platform is supposed to be perfectly level with your track.

Spicer: Right.

Snyder: So that was a fun program. However, back to Northrop, though, they did some pioneering work in computers, and they decided to build a special computer using technology of the day, which was a magnetic drum and vacuum tubes. They called it the GFC, Guidance Function Computer, to develop those paper tapes that would later be converted to Mylar tapes that would position these telescopes.

Spicer: Right.

Snyder There were some other early pioneers at Northrop. There were the guys trained at UCLA. UCLA was very advanced in computer development with their SWAC computer. Some of the people that I worked with got trained on that, the SWAC, and many of the engineers that actually were my bosses came from UCLA, and they had experience with the SWAC. That was their introduction to computers. They brought that technology to us. There was one fellow that was my mentor.

Spicer: Like a model?

Snyder: Kind of model. But, anyhow, his name was Jerry Mendelson. He passed away recently.. Not too many of my contemporaries my age are still alive. Jerry taught me a lot about computers. Later he left Northrop and went to NCR, which happened to be down the street, to develop a computer for them. That didn't work out. Many new computer developments and many of the computers that I have provided you with brochures, none of them really were totally successful. Some of them were a little bit successful like DEC and IntraData.

Many of the computers in those days didn't work because they were too limited in memory, too limited in speed and flat out weren't reliable. I went to ALWAC from Northrop. AlWAC got started by a group from Northrop. They took a big magnetic drum with them drum and they developed an ALWAC III-E serial machine with drums and telephone relays.

Spicer: ALWAC was a spin-off of Northrop?

Snyder: The guys that went there started a new business. Northrop donated the drum. The company was called Logistics and we were situated on the second floor of a furniture store in Redondo Beach, CA.

Spicer: Can you tell us what the ALWAC was because there was an original one and then a couple of other ones?

Snyder: They started the design of the ALWAC III. What does ALWAC come from? There was a Swedish industrialist named Axel L. Wintergreen, A-L-W, Automatic Calculator. He put the money into this little company called Logistics at the beginning, but Logistics changed to ALWAC when he came in with the money. So they developed the ALWAC III-E, which was a serial drum machine. They sold and delivered 12 of them. Litton bought two of them, a place at which I worked later. But the point I want to make is that the MTBF, Mean Time Between Failure, of those computers of those days was like 20 minutes.

Spicer: Amazing.

Snyder: Computers failed because the hardware technology did not support them.

Snyder: I have a funny comment to make. I hope it's funny. In those days there were many companies of major size that were spending money and hiring people to come in and get them into the computer business; General Electric, for instance, and of course RCA; you know, that list of firms that I provided you on that list. It's just interesting. Honeywell, everybody. All these big names. So, anyhow, the story then is that General Mills, the maker of Wheaties, had and were developing a "cereal" computer. They really did try to build a serial drum machine.

<laughter>

What was the ALWAC used for? You mentioned two of them went to Litton.

Snyder: Actually, you know, when people like Litton, for instance, bought and tried to make those work it was just more experimental, driven by the curiosity of having a computer. And I can't remember some of the applications for the ALWAC IIIE. I really can't. I helped to develop the MAC-16 at Lockheed and there were some interesting applications for that little computer. That was not a serial computer. That was a parallel computer made with transistors and diodes. But I'll mention it now that one of the applications was Disneyland, which was nearby. Disney bought a MAC-16 to control Abraham Lincoln, their robot that they had in this show, and Abraham Lincoln would stand up, sit down and speak under control of the MAC-16.

These little computers were fixed-application devices, dedicated to performing specific functions repeatedly. The other application that was popular for the MAC-16 was that the Olympics of 1976, I think. For the first time they had a computer to select the winner. You know when a swimmer comes in and he touches the pad?

Spicer: Right, right.

Snyder: It was the MAC-16 that would measure that, who was first and all of that.

Spicer: That's wonderful. Well, we're almost there, but before we get to discussing the MAC-16 in detail, let's talk about your time at Ramo Wooldridge and the RW-400 computer. First of all, can you tell us who the customer was for that machine?

Snyder: I can now but I couldn't then. It was a very secret program. They built this huge system controlled by more than one RW-400 computers, which was the first CPU for which I did the complete logic design, algorithms and instruction set.

Spicer: Wow.

Snyder: But this system was to process photographs as taken by the U-2 aircraft that flew over Russia and sent the information back as photos. The RW-400 was used as a big controller for processing the reconnaissance film.

Spicer: Okay, so it controlled the developer machines, something like that?

Snyder: Yes, and pieced the images together. That was the application for which the contract was awarded by the US Air Force. And that's critical because when Gary Powers was shot down and the covert spying program became publically known, they canceled the program, thereby canceling movement of our system into production.

And what is also very important in terms of technology advancement, Lowell Amdahl, the brother of Gene, was my boss and the leader, and he developed in about 1960 a new computer architecture that he called the 'polymorphic' system. It was the first parallel processing system. There were four computers down this side and memory banks across the top, and then buffer units, which would buffer I/O units down here. And you could interconnect these processors with any memory bank, with any processor, and all four processors could be working at the same time, sharing memory or not sharing memory, and it was called polymorphic.

Spicer: What was the application for that computer system, the same?

Snyder: The application was the control and processing of film as retrieved by the U-2.

Spicer: Interesting.

Snyder: That was a good, fun program. I have a funny story about that.

Spicer: Sure.

Snyder: It was also one of the processors, I mean computing systems that took large relay racks, lined them up and filled the whole room, an air-conditioned room with a raised floor. These relay racks would hold memories and computers and buffers and things like that. Well, we would try and demonstrate it to the Air Force that it was complete and done but it kept stopping. It kept having a hang-up. So what we did was in front of the control panel we put another relay rack bay empty with a door on it and we put a guy in there. His name was Jim Nyberg, an excellent circuit designer. And we put him inside this box, and he could peek out and see the control panel and the lights flashing. When he saw the lights stop and hang up he'd open up the door, sneak out, and push the button and then hide back in his cabinet. <laughs>

And the Air Force didn't know that the computer had a hang-up. The light kept blinking.

Spicer: How was the RW-400 different from the RW-300?

Snyder: Oh, they're totally different divisions of the company. They were adjacent. John Bacon was the name of the leader in that development. The RW-300 was almost an analog computer. But it had some analog function, and it was just a controller, and it was just a different application. I think they were later bought by General Electric, and they used it for something. But, it was not a full digital computer.

So no relation. It was done by another division of Ramo Wolridge same building, but another division.

Spicer: What was the device technology in the RW-400? Were you using transistors yet?

Snyder: Yes.

Spicer: Were they point contact still, or junction transistors?

Snyder: Junction transistors. General Instruments made them. Yeah, I didn't do circuit design. We had good guys doing circuit design. And we even designed our own flip flops and everything. We designed everything from scratch. We didn't take anybody else's product. As I recall, we made our own magnetic memories.

Spicer: Let's move onto the next interesting project you worked on. The RCA 4100. I'm not familiar with that line of computers.

Snyder: Oh, that is a military computer that they only made a couple of and that's it. I didn't design that one, I did some documentation on it, but it had some unique designs. Related to the question you just asked, it used some chips that allowed us to make 16 registers, 16 by 16, because there was a 16 by one chip. We could use 16 of them, and have 16 words of 16 bits. And we assigned a CPU function to each one, so that they would hold accumulators, the program counter and memory address for 16 different programs. When you had an interrupt, instead of having to store your A and B registers and everything for backup, you just switched to the other program. You could switch from program to program, very easily. That was made possible because we finally had some chips available that provided more than a few gates. The start of large scale integration.

Spicer: Can you say what application that was for? Do you know?

Snyder: I don't recall. It was some kind of controller, for a Navy radar program. I don't remember. That was back in New Jersey anyhow. We were just doing the computer, the rest of the system was back there.

Spicer: Right, so did you have to move back East too?

Snyder: No I never had to move from Southern California. This company, RCA, in Van Nuys, California, had an excellent capability for computer design, but we were in competition with the people back in Camden, New Jersey, the real computer designers. The ones that built the commercial systems. They treated us like country cousins, they wouldn't give us a good budget, they wouldn't put sprinklers in the lawns, because they didn't do that in the East, they said "You don't need sprinklers out there." Anyhow, it just really didn't go because it was neglected by RCA, and they closed that division down later. George Brightweiser was our president. He was a pioneer in computer hardware development from Litton.

Spicer: Was there any way for you to meet and speak with designers from other companies?

Snyder: Oh, yes.

Spicer: Can you tell us a bit about that, and how you might have shared…?

Snyder: Well, there would be meetings, and of course I've mentioned that we'd go to computer conferences and they'd have one major one every year, and other computer conferences. I've given you a list of the papers that I gave in some of those. But, more locally, we had the ACM. That was an organization that we belonged to, and people would get together and get to know one another. There was one that was more related to software, and it was called DCA, Digital Computers of America? I can't remember what DCA stood for, other than Drinkers Club of America. Because they met in bars, and--

Spicer: Did you hear about SHARE?

Snyder: I've heard of it, yes.

Spicer: Did you ever go to their meetings?

Snyder: No. I was a hardware guy. But there were also a few publications. A lot of people would read a newspaper called *Electronics Design and a* magazine: Computer Design. They would cover people and they would write stories about each company's new developments. You could keep up with it that way.

Spicer: I'm quite interested in understanding how you came to be so comfortable with logic design and electrical engineering with your background in mathematics? How did you learn to do logic design?

Snyder: Well, that was just kind of secondary. Really what I was, was an expert in algorithms. I designed instruction algorithms, you know, microprogrammed, micro steps for very complex programs. I don't know if I was the first, but I provided that computer we talked about, GFC, with complex instructions, such as a sine, cosine, tangent, square root.

And so, I wrote the algorithms for very complex functions. And one of the reasons that I could do that, was because of the book, *Approximations for Digital Computers*, by Cecil Hastings. He later became a friend of mine, and worked with me at Teledyne. He was a genius, and he was a teacher at UCLA, and probably one of the best programmers there ever. But he wrote this book on approximations of digital functions. Mathematically, there was a theory, or law that said "Every function can be a geometric progression, such as A+BX+CXX + DXXX" all the way to as many terms as you needed. And then, a mathematician named Chebyshev published, that you could convert these geometric progressions--equations down to about three terms--to be accurate. He changed the coefficients of your progression, your series, so that within three terms, generally, you could get a good approximation of the function. And therefore, when I implemented those instructions, like trigonometric functions, I used Chebyshev's equations.

Spicer: Polynomials?

Snyder: Polynomials. And, that was a breakthrough for me, and I relied on that book from Cecil.

Where did I go to next? A short stop at RCA.

I didn't do really hardware design there, because they didn't have anything for me to work on. But I spent a year and-a-half just writing papers on variable instruction computers, what is the best length in bits for an instruction and other architecture considerations. I read papers, because RCA had a library of technical papers, and you'd get points for writing articles that they could put into their library. So, I have a list of my papers, but I did as well as studies of computer architectures. How many bits do you really need to make a good instruction? You know, six for the opcode, three for the memory access options, like indexing and indirect and that sort of thing, and then some bits left over for your address. You had to use tricks to get enough bits, to be an address for large size memories. So, where are you going to get those extra bits? If you want to keep it small and low cost and everything. We worked hard on making programs short. A guy would design a program, which took a hundred instructions, and that's large for those days. If you can get that down to 90, that was a real plus because of memory storage problems.

Spicer: So, you went to Litton next--

Snyder: I did go to Litton, yes.

Snyder: Oh, okay, I was in both Divisions: Guidance and Control, and Data Systems, and both did computer development. When I was in Guidance and Controls, I was head of the logic design section, and we would design military computers definitively for navigation purposes. One of the computers we developed for the Navy was called the Phoenix Computer. It was for the F-111 aircraft which and was a fire control and gun control application. At the beginning, I asked myself, “What instructions should we put into this computer?" Well, thinking of software compatibility and making it easy for the software people, I decided we were going to implement this computer with an IBM 1401 instruction set. And that's what we designed into the computer, trying to emulate the 1401, so that we could use their software, their assemblers and maybe compilers if they ever had them. The Navy owned the 1401 instruction set and approved our use.

Spicer: So, Phoenix could run 1401 software?

Snyder: Yes. I'll tell you what killed that program. While we were developing this computer, we were also developing a core memory. And Dick Searing, one of the pioneers of developing magnetic core memories, failed to meet the requirements and the specifications and we were late in delivery. Finally they cancelled the project because of the failure to come up with a reliable magnetic core memory.

Spicer: Really? The years we’re discussing are 1963 to ’67… they should have had reliable core memory by then.

Snyder: Yep, they should have. They kept shrinking the size of the core, down to where it was so difficult, they couldn't even get the three wires through it. Under 16 mils.

Spicer: I have a general question: Throughout your career up to this point, what role did packaging, power and cooling constraints play in your work?

Snyder: One of our real goals was to reduce power requirements to reduce heat. We started using integrated circuits more and they ran hot. So we had to develop unique heat sinks. Sometimes, we'd take some of these large scale integrated ICs and put them on metal or a thin cover on top of it, to get rid of the heat. It was a tough problem. And the computer still did ran hot.

Spicer: And the Phoenix, I imagine, being intended for a jet aircraft, must have been quite a small, black box, with a bunch of connectors on the end or something--

Snyder: Right, and fans.

Spicer: And fans inside?

Snyder: Yes.

Spicer: And all the usual environmental tests that it has to pass, you know, salt spray and being dropped and frozen, and boiled and all that good stuff.

Snyder: Right, mechanical designers would come up with clever schemes and do that.

Spicer: That's interesting. Did that feed back into the way you designed things at all, or thought of things, or as a--

Snyder: Only in that we wanted to reduce-- hold the number of components required to a minimum and we didn't worry about the power, we let somebody else get rid of the heat. And, of course, we had to develop schemes for addressing large memories.

Spicer: Right. Okay, well your next stop from 1967 to '74, was at Lockheed. And that was the era of the MAC-16…

Snyder: Yes, and I mentioned the MAC-16 was the first computer that they built there. I didn't do the logic design on that. Lowell Amdahl's company had a hand in it.

Spicer: Lowell Amdahl?

Snyder: Yes. He had a company called Compata, that provided consulting, and one of the things he decided was to eliminate as many components in his computer design as possible. He says, "We're not going to have a B register." You know, all computers have A and B registers. You can't do a multiply without a B register. Anyhow, he convinced us that we have to use just enough parts to get everything on a single board. And so, that's what we did. And the MAC-16 only had an accumulator, no B register.

Spicer: I know Gordon Bell, the famous computer designer of many computers once said "The most reliable parts are the ones you leave out."

Snyder: <laughs> Yeah, I believe it. Yeah.

Spicer: What did “MAC” stand for?

Snyder: Good question. Multiple Application Computer.

Spicer: Oh, okay. You came up with that?

Snyder: Yes.

Spicer: Tell us about the application then, because it has many.

Snyder: The applications of the MAC-16 was driving the robot, Lincoln, at Disneyland and then, measuring the times of swimmers that would come and touch the panel in the 1976 Olympics. And that's about it.

Spicer: Right, but it wasn't designed just for those purposes was it?

Snyder: No, it was mostly a mini-computer used for all kinds of special applications, however it was programmed.

Spicer: So, it was a commercial product?

Snyder: Oh, yes.

Spicer: Okay, not a military one.

In that sense, you were one of the many entrants to the minicomputer market at that time.

Snyder: Yes, but that wasn't our main product, and because just when a computer like the MAC-16 was developed, it's already obsolete because there are new parts available to make a better one. And so, we just set up to build a new one, called the SUE. Now we didn't come up with that name, some advertising company did it for us, and they thought that was clever because of a "Computer named Sue, a boy named Sue…

So, I had named it the System User Engineered, but it was really just plain SUE, like a girl's name. Anyhow, we built that computer to be directly competitive with the PDP-16 (sic). And it was very much like the PDP-16. It was so much like the PDP-16 that I got sued.

Spicer: 16? I didn't know there even was a PDP-16.

Snyder: Almost everyone that had a 16-bit parallel computer had a -16 in their name.

Spicer: Wasn't the PDP-11 the 16-bit computer?

Snyder: Oh, okay, you're right. My mistake.

Spicer: I think so, that was their blockbuster product.

Snyder: Yeah, that's right, I'm sorry.

Spicer: So DEC thought you were a little too close to the PDP-11 architecture?

Snyder: Oh, we were. We had a common bus in the back but ours was better than theirs, which was called Unibus.

Spicer: So, it being called SUE was actually a good name because you got sued by--

Snyder: <laughs> Good question, yeah. But, no. we fought that. I made a deposition. I said "You're off base, you can't patent that," and those kind of things. You know, in an instruction, there's three bits that pick your address options, such as indexing and indirect; They put a patent on that, and they also put a patent on microprogramming, And things that everybody was using.

Spicer: That they didn't invent.

Snyder: Yeah. But the lawyers that were fighting this didn't know anything about technical matters. So this was not settled on a technical basis. It was settled on a business basis. And that was-- they agreed to not sue us, Lockheed Electronics, and me, the named, if we stopped selling our computer in the state of Massachusetts. Now that's very significant, because a Massachusetts company, Bolt, Beranek and Newman who were the leaders in the design of the Internet, they called it something else like--

Spicer: ARPANET?

Snyder: Yeah, yeah. They designed that, and then they came up with the pack technology of transferring information.

Spicer: Packet switching.

Snyder: Packet switching, yeah. They developed that, but more than that, they selected my computer, the SUE, for the computer application that would drive the bus and the reason they did that, is because my bus, common backplane bus, was designed so that you could plug your cards into just about any slot, because it had a common backplane. But with the SUE, you could put two computers on the same bus, again parallel processing. Then they could work simultaneously on that bus, and communicate. And that's why Bolt, Beranek and Newman thought that this would be a better design for them. But, it turned out that we had to give up on that.

Another SUE application we had was an insurance company, and we built a version of it that was contained into a full-size steel office desk, in one drawer. The other drawer was just for data and storage. And on top, a CRT and paper tape readers and punch. The insurance company developed software to run in their insurance office.

Spicer: Is there a technical debt between SUE and the MAC-16? Are they related technically? Or are they completely different?

Snyder: Oh, completely different.

Spicer: Because of the word size, or something else?

Snyder: Yes, just a whole new design, The MAC was rack mounted in a relay rack and SUE was a, you know, normal minicomputer much smaller, with its own cabinet that held a single board CPU.

You remind me of a system that I developed for Teledyne, which was also a dual processor. So, we did have an application for a dual TDY-52B processor system, for a fellow in Germany. His name was Doctor Kubeck. He put the computer system into his missile control application. And it was dual for redundancy, he had two processors. Now that processor, was one of the very first that you could hold in your hand. And, the forerunner of computer on a chip, because that was a computer on a 2" by 2" substrate.

Spicer: Ceramic hybrid, right?

Snyder: Right, and those are chips instead of components, and they're put on a substrate with other components. Teledyne had a hybrid company that did that. And you can see the first one that we made, I'll show you the chip. The first one we made used the 4004 from--

Spicer: Intel?

Snyder: Intel.

Spicer: 4004?

Snyder: There were four of them, 4004s operating in parallel to provide a 16-bit computer. The second one we made used the 8008. And it had a little more power. That was a TDY-52B.

Spicer: So this was the early '70s probably right?

Snyder: Yes.

Snyder: So that was one of the very first computers you could hold in your hand.

Spicer: Oh, and they're 1200 dollars each, for this little thing, but it's got a lot of power.

Snyder: Yeah, however they were very hard to make work.

Spicer: That was a lot of money back then too.

Snyder: Yes, but the problem was, those chips that we got from Intel, had to be matched, so that they would be in-phase, timing wise, and it was very difficult to get them all matched so that they worked together.

Spicer: So, they had different propagation delays probably?

Snyder: Yes, different things like that. So, I wasn't involved in trying to do that, because it was our hybrid division who were building them and trying to check them out.

Spicer: Wonderful. Do you know any other systems that this went into?

Snyder: Let's see, we did System DK (Dual Komputer for the Germans) as mentioned before.

Spicer: We're looking at the TBY-52 brochure.

Spicer: Well, that brings us to the end of Teledyne, which was were you ended your career, I think, in '92?

Snyder: Yes, I've retired and I've been retired over 20 years and I've watched digital technology explode in the 21st century.

Spicer: Is there anything you'd like to leave us with? Some parting thoughts, or great ideas that you'd like us to record?

Snyder: Let's see, ideas. I do want to thank you for the opportunity to give you this story, because I think it's an important story because it tells the story of a lot of work in computer development that took place before a "computer on a chip" was possible. Many mini-computers were developed, but few were successful. Even the PDP-11, which was probably one of the best, was put out of business by large-scale integration.

Spicer: Yeah, well they actually put the PDP-11 on a chip, they called it the LSI-11.

Snyder: Did they? Oh that's good.

Spicer: So, they were smart enough to do that.

Snyder: Oh, that's true, I remember it now, but everybody went out of business. Most all 125 companies that I gave you information on stopped making computers. There was one other point that I wanted to make about that, and it's just interesting to me, but of that 125 companies that were in the computer business, about 50 of them were in California, including Silicon Valley. But there were about 25 companies in the Los Angeles area alone, in the early days, developing computers. Some of them still exist because they were large companies, but they're not in the computer business, like Bendix and Varian. They all made processors, but they were not commercially successful.

Spicer: It was a real gold rush, I think for the minicomputer era.

Snyder: Yes, everybody jumped into it.

Spicer: Yeah, the barriers to entry must have been very low, and TTL or earlier forms of logic were around.

Snyder: Yes, we were trying to get too much out of technology. There was another time when I was with ALWAC that, people thought that two million dollars would be plenty to build a computer, and put it to market. Well, not really. When we were at ALWAC, Axel L. Wennergren gave us a lot of money to literally compete with IBM. I mean, his goal was to compete with IBM. And, we started building this high-speed new, large-scale computer, in racks, that-- it was Alwac-800 I think we called it. And he put in a couple of million, and we spent that, and it still didn't work.

We said, "We need more money." And finally, he just said, "No more money." And that put us out. But, the real problem was technology-wise, there was a device called a tranfluxer. It was a dual core, two cores together, only it was all connected. It was a switch and it would toggle when pulsed. And that was the toggle that's needed to do the logic design. It never worked out. It was again, a timing problem, because by the time that you sent a signal down a long wire in those racks, it's out of sync with the rest of the things. You need compactness for those kinds of things, and the technology didn't support that. So anyhow, the ALWAC-800 was a failure because it used the wrong technology.

Spicer: Well, that's fascinating, thank you very much for speaking with us today.

Snyder: Oh, you're welcome.

Spicer: And thank you again for all those great brochures that chronicle the history of computing so interestingly. It's a great donation.

END OF INTERVIEW

