Oral History of Ronald Rider

Interviewed by:
Bob Sproull

Recorded June 21, 2023

CHM Reference number: 2023.0097

Sproull: Our guest this morning is Ron Rider. He's principally known as a key engineer in developing computer printing technology at Xerox PARC in the early seventies. He was the designer and builder of the research character generator (RCG), a rack of high speed digital electronics that created video streams to drive laser scanned page printers. Ron and RCG were the companion to Gary Starkweather's prototyping efforts at PARC that resulted in a printer operated as an Ethernet print server based on a Xerox 7000 copier printing a page a second with 500 dots per inch. Ron moved in 1977 to join Xerox Electronics Division in El Segundo, California, where he held various technical and management positions including Vice President, Advanced Electronics Development. He managed the development of software, electronics and systems used in copiers and digital reprographics products including Docutech, a 120 page per minute digital copier and printing system. Starting in the mid-1990s, he headed the Digital Imaging Technology Center, which was responsible for the design and development of new and innovative digital imaging solutions including iGen3, a 110 page per minute full process color printing system. Today's iGen5 follow-on prints up to 150 pages a minute at 2400 dots per inch resolution, full process color. Ron held various management positions at Xerox including VP Systems Research; VP Systems Architecture; VP Corporate Architecture. Together with John Seely Brown and Frank Squires, he managed PARC for several years. Ron was an iconic figure in the early printer system prototypes at PARC and their evolution into an important Xerox product family.

Sproull: So Ron, let's start with the beginning, your family, where you grew up, any early salient experiences that might have foreshadowed your future?

Rider: Well, I was born in Pasadena and raised in San Marino in Southern California. I had one sister who was totally nontechnical. But for me, I always enjoyed building things, whether it be in the wood shop, working on cars, all of that kind of stuff. When I went to high school, I just sort of did the normal technical-based thing of taking mathematics and all the science courses they offered and, and the last one was physics and that seemed good to me. So I ended up going to graduate school in physics.

Sproull: Well, right, both undergraduate and graduate degrees are in physics.

Rider: Yes. That's correct.

Sproull: So you told me that graduate research in high energy physics also offered an opportunity to build apparatus to support the science: flying spot scanners; computer interfaces; mechanical and digital prototypes, various kinds, probably software too.

Rider: Yes.

Sproull: In fact, talk about how you got into the computer world.

Rider: Well, I was, as you said, I was originally in high energy physics and we were doing an experiment at the Princeton-Penn Accelerator and back in those days, you basically took 35 millimeter pictures of spark chambers and the spark chambers were triggered by what it thought was an event which more

often than not was not. And so you've had to scan a gazillion pictures to find out which ones actually had the event you were trying to trap. And the easiest way to do that or the way we chose to do it was to build the flying spot scanner and, and then write the programs to scan them, find the event, which turned out to be a lot of mathematics because you had to do everything with relativistic equations because the events were happening at high speed. So in doing that, I got involved in computers and I got bitten by the bug and continued doing that. We then built, we decided to, although this doesn't logically follow at all, we decided to move into biomedical kinds of research and look at to see whether we could tell whether people had osteoporosis and look at their bone calcium content in actually the distal radius, this little bone down here. And I had to build a scanner for that and interface it to both a PDP-11 computer and to the IBM 360/50 which we had a shared memory situation that IBM didn't appreciate but worked fine. And, and from there I just kept on in, in computers, and it turned out that when I was a graduate student, George Pake, who started the Palo Alto Research Center,[1] was provost at Washington University where I went and, and was also a professor of physics. So I knew George at that time and in 1970 he went off to start PARC and in 1972 when I graduated, he then arranged for an interview, and I interviewed people like Bill English and Dave Damouth and Butler Lampson and Chuck Thacker, and eventually ended up with a job offer to PARC and came there in summer of '72.

Sproull: Okay. I don't want to let you escape St. Louis and Wash U quite that rapidly.

Rider: Okay.

Sproull: So did you have any contact with CS people at Wash U? Jerry Cox or Wes Clark and the macro modules folks?

Rider: I did know Jerry Cox was actually on my thesis committee and Wes Clark, I knew. They actually had a laboratory over at the medical center which was, oh, five miles from, from the physics department across Forest Park and we were, we had a, a modem set up between the main computer which was on the main campus and the medical school, which was, incredibly unreliable. I mean, packets were lost. Well, they weren't packets, but data was lost continually and, and it was really hard to, actually, it was easier to actually take a disk, get in the car and drive a disk over, than it was to use the communication network.

Sproull: Okay. The other thing I want to explore, so all these flying spot scanners presumably were CRTs to do the spot.

Rider: Yes.

Sproull: And then a photo multiplier or something to detect.

Rider: That's exactly right.

[1] PARC

Sproull: How much transmission the film added.

Rider: Yes, we used an ultraviolet CRT and a photo multiplier to receive the data. We sort of calibrated the film in advance, the photographs by looking at x-rays going through various thickness aluminum plates and calibrating what the absorption was and then used that to calibrate the photos. In this case, it was the bone images and the trabecular structure, the sort of lacy structure inside inside the cortical bone. And by looking at both the density of that and the orientation of the various trabecular structures, we could estimate what the calcium content was. Today they have a cool method of doing it which is quick and easy and really works well and is much better than what I did.

Sproull: Okay. So, so back to PARC, Dave Damouth hired you into the System Science Lab.[2]

Rider: That's correct.

Sproull: And you told me that within two weeks of coming to PARC, Bill English showed you a variant of the SRI[3] mouse that he had worked on with Engelbart at SRI and something interesting ensued. Why don't you let us know?

Rider: Yes. So, Bill English showed me a mouse which they had, they had little wheels on the bottom orthogonal to each other. And they had quadrature encoders on each of these wheels and it worked fine if you were going vertically or horizontally, but if you tried to go at 45 degrees, the wheels would drag and turn some. And so it wasn't a very satisfactory way to have a mouse work. And I looked at it and just out of the blue, it occurred to me, why don't you take a track ball, miniaturize it, turn it upside down and then you actually take the sensing points off the side of the ball and it ought to run smoothly in all directions. And I suggested that to Bill and he thought that was a great idea. And I talked to the patent attorney and I never did another thing on the ball mouse except get the U.S. Patent on it. And a machinist down in El Segundo actually, read about it and said, "Oh, I can make that." And so he did. And so this is one of the early ball mice.[4] This isn't the first one, but this is probably the tenth one that was ever made. And it has a big ball here that is the one that encodes everything. And these two little guys up here are just to just to drag along the paper and they don't do anything for sensing. But so that was my first two weeks at PARC and then I started on image generation.

Sproull: Right. Well, and those mice were made in enough numbers to serve all the Altos and Alto follow-ons and so on that PARC made.

Rider: Yes. And then in the time of Carver Mead's book[5], an optical mouse was actually designed and built at Xerox which quickly became the replacement for this one and is the kind of mice that we all use today. And that one had the attribute that you had to run it on a grid of paper that was blue lines one way and magenta lines the other way and--

---

[2] SSL
[3] Stanford Research Institute
[4] In the video of this interview, Rider shows the mouse.
[5] C. Mead, L. Conway, *Introduction to VLSI Systems,* Addison-Wesley, 1980.

Sproull: Right.

Rider: But, whatever.

Sproull: But it got better.

Rider: Yeah, it got better. Now you can run it on almost everything except glass.

Sproull: Right. So, but you mentioned you got into printing quite rapidly after joining SSL. And you also mentioned that you had had a non-satisfactory experience printing your thesis on a chain printer. And so this was somewhat on your mind.

Rider: Yes. So in graduate school, I mean, I spent the last 3 years really living with the computer. And so it was sort of natural to want to use an editor to type it up and print it and that kind of stuff. The printer was the IBM mainframe computer printer and you physically had to load various chains because it was a chain printer. And there was one chain, the TX chain, that had both upper and lower case letters. Probably, it was-- it wasn't Bodoni, it was probably Times Roman. And so I said, well, why don't I do my thesis on this? And nobody had ever done a thesis that hadn't been typed before. And so the university finally said, "Okay, you can be the first one to actually do your thesis this way." But I looked at it a few weeks ago getting ready for this meeting and it was pretty ugly. I mean, it was-- it was not very satisfactory.

Sproull: Well, not only that, but the upper-lower chain was only half as fast as the regular chain.

Rider: Yes. Yes.

<laughter>

Rider: That's right.

Sproull: But, you know, it's interesting you tell this story because, you know, all the Xerox printers that you and Gary and others engendered had huge histories in being the first to offer Xerox original theses[6],

Rider: Yes.

Sproull: And many conversations between students and registrars ensued, I assure you.

Rider: Yes. Well, hopefully it was easier with a decent printer as opposed to a chain printer.

Sproull: Right. Plus it was a few years later.

Rider: Yes.

[6] A Xerox “original” rather than a Xerox “copy.” Copies were not accepted by registrars.

Sproull: So, so let's make contact with Gary. I mean, Gary Starkweather had been fussing around in Webster, New York at the research lab.

Rider: Yeah.

Sproull: And felt a little discouraged about not getting more support for his ideas for laser scanned optics that would drive, that would irradiate or expose a xerographic drum. So he came out to PARC and I gather was well under way at building stuff at the time that you arrived.

Rider: Yes. So when I was hired, you know, three months before I actually showed up to work, there was sort of an understanding that Gary was building this machine which was sort of being used in a fax-like mode. And, and that it needed a character generator and having just finished or finishing my thesis, I was interested in something better and that seemed like an interesting project to me. And, and so I got started on that sort of day one when I arrived at PARC.

Sproull: So, you mentioned it was viewed as sort of a fax-like thing. And in fact, Xerox, I think, had entertained thoughts of xerographic fax machines for some time. They had built up a CRT-based printer called the XGP, the Xerox Graphics Printer.

Rider: Yeah. I mean, XGP was the printer version of a fax machine, yes.

Sproull: Exactly, right. And they had a CRT scanner that went with it.

Rider: Yes.

Sproull: But then the printer alone was actually already in circulation and, well at PARC.

Rider: Yeah.

Sproull: And also in various computer science departments around the country where they had fitted them out with just software-only character generators.

Rider: Yes.

Sproull: To produce a bit stream that printed out theses, among other things, on a roll fed-- a roll of paper, with then a knife to cut it off, just like a fax machine.

Rider: Yes.

Sproull: So, was the fax influence-- Was the fax influence felt at all or was Gary building a computer printer?

Rider: I think Gary was, I mean, can't ask him anymore, but, I think Gary was just exploring the optics wanting to make sure that that worked. And the easiest way to do that was to use the polygon to scan an image with a photo multiplier tube and the same polygon to image onto the drum. That was just a convenient way for him to test the optics is what I think.

Sproull: So the other thing is--

Rider: I mean, I think he had thought about printers, but he knew that he wasn't going to build an image generator to do it, that someone else would do that.

Sproull: Well, right. The other thing that struck me about his prototypes is, at least the ones we saw, is he was always trying to fit them into an existing xerographic engine. He wasn't trying to build xerography gadgets, he was trying to expose an existing xerographic engine.

Rider: And that was, that was true for many years to come. And the biggest problem for Gary was, and part of the reason he left Webster was that Xerox is famous for naysayers that, you know, "This can't be done, that won't work. Even if it does work, you can't sell it," and that kind of stuff. So, there were a number of people who conjured up all of these possible problems. Like the laser light because it's coherent will destroy the photo receptor and won't work. I mean, it'll last-- it'll look great for a month or a year or something like that. And none of that turned out to be true. It was all fake news. And it turned out the laser worked very well on a photo receptor, probably better than the, than the regular light source. So Gary had to sort of disprove all of that kind of stuff along the way. And Gary's strength was he was a master optician. And, and so getting the-- just getting the optics right was a huge part of the problem.

Sproull: Right. Your mentioning of the naysayers reminds me that xerography was something of a magic mystery when it first got started.

Rider: Yes.

Sproull: And the early Xerox machines were plagued by all kinds of problems. People, you know, rabbit fur had to be found and fit and tailored for certain processes. There were waxes and so on, commercial waxes that had to be found to--

Rider: Yeah.

Sproull: To polish this and that so that various things didn't happen.

Rider: I mean, the rabbit fur was used to charge the drum.

Sproull: Right.

Rider: And they eventually found a synthetic fiber that would work. But, I don't think it seriously hurt the rabbit population, but the rabbits were probably happy to have the synthetic.

Sproull: Yeah. So, so let's go back to, so, your job was going to be to build a character generator, a source of video that represented text to be printed. And say about how the various ideas and design and challenges, what was the character of that, of that project?

Rider: Okay. So the goal was to make it, you know, at least as good as a printing press, which meant it had to do multiple fonts, multiple sizes, had to be able to do both landscape and portrait orientations. We really didn't talk about graphics much. We could do rectangular graphics, I mean vertical and horizontal lines, but that was sort of done as a kluge of special characters and a special font. But the question always for this image generator was we wanted it to be the quality of a printing press, which meant we did 500 lines per inch. And that meant that we had to average about 25 megabits per second average. But it turned out because of the way the polygon was created, you actually had to output the data at 50 megabits per second. And in later versions of the SLOT machines that Gary built, he had more like 90 percent efficiency on the polygon as opposed to the 50 percent that we had on the original engine.

Sproull: Yeah. And just let me interrupt. So SLOT was an acronym for Scanned Laser Output Terminal.

Rider: Yes.

Sproull: And so a lot of these printing prototypes at the time were SLOTs of various kinds, which was optics fitted on top of some other, some existing xerographic engine.

Rider: Yeah. And in the initial cases, that was always the Xerox 7000. And the reason for that was it was the only Xerox machine that had a photo receptor that was sensitive to red light. And, and at that time, the only laser that was reliable was a red laser, a helium neon. And even argon wasn't very good or was expensive at that time and didn't have the lifetime that helium neon had. And it was sort of greenish, not quite blue. So it didn't really get the photoreceptors very well.

Sproull: I also seem to recall that the Xerox 7000 was built like a tank. It was a very solid machine and there happened to be several hundred in the warehouse waiting to be refurbished or something. They were not hard to come by for experimental purposes.

Rider: Yeah, I think that's true. But the reason for using it was pure and simple. Photo receptor was red sensitive.

Sproull: Yeah. So tell a little bit about how the design of the RCG got done.

Rider: Well, the first thing I did was I went around and talked to various people at PARC, all of whom had ideas about what this should be. But the person who had the most ideas about how this should be and had thought about it the most was Butler Lampson. And Butler-- remember that the paper in the 7000

goes through sideways, so wide edge is the leading edge.[7] And so it meant that everything had to be turned 90 degrees. And later on when you had enough memory and you could create a whole image of the document, that didn't matter because you just created it in memory and, and when it was done, you spit it out in the right order. But back in these days, you couldn't afford enough memory to do that and so you had to figure out how to make the page, how to construct the page line by line, going top to bottom on the page. And, and so you'd start off with all of the letters would, of course, and every line would start off at the same time.[8] But since it was variable pitch fonts, then they each one kicked over to a new letter at a different time. And Butler had this clever sorting algorithm for how to do this in real time, which we originally did on the Nova 800. And I, of course, having spent all my life on a PDP-11 thought that would be a better choice. But lo and behold, when I programmed them both up, it was one instruction faster on the-- on the Nova 800 and that's what we used because the Altos didn't exist at that time. So Butler had that idea and then he also thought that we ought to compress the fonts because it used too much memory and we wanted more fonts. And it turns out that he had this really clever algorithm for how to do it. And Butler always did things on a single sheet of paper. Sometimes you needed a magnifying glass to read it, but he had this one page sheet describing the compression scheme for this. And, and so the first thing to do was to try and figure out what he had actually written and, but then was to program it and compress a bunch of fonts and see what it did. And the answer was if you drew a bounding box around each character, you could then get about 4 to 1 compression at 500 lines per inch.

Sproull: And so the idea would be the fonts would be stored compressed.

Rider: Yes.

Sproull: They'd get decompressed, sort in the scan line order.

Rider: Yes.

Sproull: And put into uncompressed scan line buffers where they were the top-- where the all the lines for that page if you will, would be being assembled.

Rider: Yes. And there were only two scan line buffers, the one that was being output and the one that was being created. So you had no choice but to do stuff in very real time. And that was the problem because it-- this was a time when Schottky logic was the kind of logic used and it had sort of a 20 nanosecond throughput time. You could get a shift register that would barely run at 50 megahertz, which is what I needed. And you had to do the design, in a pipelined fashion, because there wasn't time to do anything. So, so you did a little bit and then you did a little bit more and you did a little bit more and it was about a 12 stage pipeline by the time you got to filling the line buffers at the end. And then the other-- and then when you filled it, you'd swap the two and move on to the next one and all of that. So.

---

[7] The scan-line is thus 11 inches long, oriented along the 11 inch paper direction. In some of the descriptions below, the first scan line is referred to as the “top of the page” but on the Xerox 7000 it was the leftmost (vertical) scan line. A more accurate term would be “leading-edge scan-line.”

[8] Every line of text on a portrait-oriented page (11 inches upright) would start at the same time, namely at a the leading-edge scan-line.

Sproull: So this got to be a fair bit of gear.

Rider: Yeah. This was about a few thousand ICs, 2 to 3000 ICs, about 1000 memory and 2000 logic. And the memory being font memory. And it just had to keep producing and I think it was spec'd that it would do 50,000 characters per page if it had to. <laughs> And people would experiment with that. They would make the font as small as they could and do it till it broke.

Sproull: Yes. Well, so since you mentioned that, to get ahead of ourselves a little bit, that when it became part of a server, you could include special fonts in the thing you sent to the server.

Rider: Yes.

Sproull: It didn't have to be part of a standard font or a library or anything, so people--

Rider: Yeah. We had a library of a dozen or so fonts. But there was also a way that you could create your own mystical font for whatever purpose you want. And a lot of people used it to do interesting graphics kinds of things.

Sproull: Exactly. And in fact, William Newman wrote a program to take arbitrary graphics, bit maps and create the RCG fonts to print them, mostly, making a few errors, of course, because you ran out of font memory or something.

Rider: Yeah. Right.

Sproull: But, but yeah, people-- Given the hammer, given the nail, given the hammer-- Given the hammer, people found a lot of nails.

Rider: Yep. They did things with it that I never envisioned as possible.

Sproull: Yeah. This was part of the PARC culture.

Rider: Yes. And that was great. I mean, that's what made it fun.

Sproull: So let's go back a little bit to the project history. And so what time are we in? And you built two RCGs. You need to explain a little bit why you did that.

Rider: Okay. So I started this in the summer of '72. It took about six months to do the design, draw it up. Back then it was little plastic templates that you drew on sheets of paper. And so it took about six months to do the font compression experiments, figure out how to pipeline it, talk to a bunch of people who helped me understand how to build the piece of logic like this. I had mostly done stuff that didn't need to be pipelined and, and Chuck Thacker was probably my foremost teacher in logic design for this project. And so it took about six months to design. There were, it took about three months to get it fabricated. It was all wire wrapped technology and we had to build special back planes and that kind of stuff. And we

built two because at the same time, a group in Xerox had decided that they were going to build a computer printer, which became the 9700 and, and they needed an image generator for that. And it was only going to be at 300 lines per inch, but it was going to be twice as fast, 120 pages per minute. So that was about the same output bandwidth as we needed for the one we were building. So they in fact sent an engineer up from Pasadena who as I debugged the first generator, any mistakes that I found, of which there were more than a few, required rewire wrapping and he would take care of bringing up the second one sort of in parallel. So when we had one, we had both were ready to go. And it probably took four or five months to do the whole debug and get the software working with it.

Sproull: So, so let's flesh out a little bit more the stuff between the character generator and the polygon, because it isn't just a matter of delivering video. There's the business of talking to the underlying xerographic engine. You've got to worry about timing the video to both the start and end of scan line, but also the bit rate needed for whatever polygon speeds. And there were issues about syncing up to the polygon.

Rider: Yeah, there was-- there were plenty of issues. So as far as the engine is concerned, it's synchronous. You just tell it to start print and it does its thing. You also have to tell it to stop printing, but you tell it to start print and it goes through its cycle. And you basically have only three other signals coming from the engine. You have a page sink which says the page is starting or it's about to start; and you have a start of scan, which is a detector when the laser crosses just before the beginning of the page; and you have an end of scan when the laser gets to the end of the page. Now, you would hope that the-- since the polygon is spinning at a constant, relatively constant speed--that the time between start of scan and end of scan would always be the same. But it isn't. This was in the early days and the best synchronous motors that you could get still did what was called hunting, which meant they'd speed up and then they'd hit a pole in the motor and slow down and speed up. And so the average speed was exactly what you wanted. But scan line by scan line, every scan line was different. It would either be getting a little bit faster or a little bit slower. And so you had to build a servo clock that would put the right number of bits between start of scan and end of scan and adjust the bit clock rate so that worked, because otherwise you-- a straight line would be a wavy line down the page. And, and in fact, there was a test mode where it just used a fixed crystal clock. And when that happened, you just got these wavy lines down the page or wavy text or whatever. And so it was a challenge to design that bit clock and a guy named Tat Lam, who also did the start of scan and end of scan detectors for Gary, actually designed the servo mechanism for doing the bit clock.

Sproull: Yeah. And Tat was also instrumental in designing early Ethernet transceivers for the coax.

Rider: Yes, he did. We bought-- Yes, he was busy back then.

Sproull: He was very busy.

Rider: Start and end of scan, bit clocks, transceivers for Bob Metcalfe's Ethernet.

Sproull: As PARC was struggling to be digital, Tat needed to be analog and we needed him desperately. <laughs>

Rider: Yes. Yes. That's exactly true.

Sproull: So let's talk a little bit about getting-- going from the RCG and Gary's slot prototype to a print server.

Rider: Okay. So basically what we did was we, we initially got the RCG running on a Nova 800 and tested, well, mostly tested. I'll tell you about another bug we found later. But it was creating multiple fonts and doing portrait and landscape, and worked as designed. And we had a couple of simple programs that would allow people to take a text file and print, you know, directly on the Nova 800. There was no network at that time to send anything to the printer. We-- Bob Metcalfe at that time was busily working on the Ethernet and Chuck Thacker and Butler Lampson and Ed McCreight were working on the Alto. And so pretty quickly the idea came that we could take the SLOT machine, the RCG, the Ethernet and the Alto and put them all together in a print server that we called EARS, which was for Ethernet Alto RCG and SLOT. And so we conceptually had that idea and I think one of the first Altos that were created, I actually got down in the lab to interface to the Research Character Generator, and Bob and Dave Boggs did the Ethernet card that went in that Alto. And then Ed McCreight built a RAM card for extra microcode, which was necessary to drive the RCG. Basically, the microcode did this sorting algorithm that Butler Lampson had designed, and pushed the characters in order into a FIFO, which then went into the RCG. So we started putting all of that together. And I wrote the software for doing all of the spooling and running the RCG, and Bob Metcalfe wrote the software for doing the Ethernet transmission. I think it was EEFTP or something like that was one of the early protocols. And we then eventually got that working, and you could spool hundreds of pages on the disc, and it would print them and that sort of stuff. So it was-- I don't think it was the first print server, but it was certainly the first Ethernet-based print server.

Sproull: But all this took a while to develop, because the Ethernet-- even the first card for the Alto was substantially[9] after the first Alto.

Rider: Yes.

Sproull: And then, of course, the network wasn't much good until there were lots of Ethernet cards and lots of Altos. So I think you told me that it wasn't until really late-- well, mid-'74 that the EARS as print server came to be a common utility.

Rider: Yeah. It was used as a print server a little bit in the old building, but really it came into common use when we moved to Coyote Hill Road building, the new building, 3333 Coyote Hill Road. And that wasn't until '74.

Sproull: Yeah. Well, I think you said '76 that you moved to building 35.

[9] Several months.

Rider: Yeah, you're right. Yeah, yeah. Yeah.

Sproull: Yeah. So it took a while.

Rider: Yeah. And I would have guessed earlier, but someone else had a document that definitely showed that's when we moved, so must be true.

Sproull: Okay. At some point, I want to turn and talk more about the 9700[10], but before we do that, another piece of what you did at PARC was recognize that the interface between character generator and slot was something that could be routinized a bit more. And you came up with the idea of a thing called a ROS adapter.

Rider: Yes.

Sproull: ... Or Raster Output Scanner. So talk a bit about that.

Rider: The idea was that you wanted to be able to create an image generator and attach it to a laser imaging system. And oh by the way, you'd want to be able to connect it to multiple imaging systems. And you wanted sort of a standard interface. You didn't want to have to deal with all of the start and end-of-scan things, which eventually even disappeared. You only needed start-of-scan, because the motors got more stable. So the idea was to build an interface that anyone could build an image generator to, anyone could build an imaging system to, and you could plug them together. And that ended up having more variations to it than I had imagined, because you had to do how many lines per inch? How many bits per inch? How fast did the polygon have to go? How big was the paper? And all of that was part of the engine. And all you really then delivered to the engine was scan lines. And so the imager created scan lines, sent them relatively asynchronously over to the ROS adapter, and the ROS adapter took care of the synchronization with the engine. So in fact, you and I worked on that for a while. We debugged the first one-- I think we did it on a engine we called Pegasus, which was a 9200-based engine. And so the ROS adapters ended up having two versions, an ECL[11] version, which we needed for that engine, and later there was a TTL version, which was the more common one that was used for Orbit, which again, you were involved in, and future things. When I went off to El Segundo to do product design. We stuck with the ECL version for a number of years because we were doing higher-speed engines and 600 lines per inch.

Sproull: Right. Well, so the whole idea of standardizing the interface was extremely useful in being able to experiment with different Xerographic engines. For example, hooking up a color 6500 turned out to be pretty easy. Or a 3100, if you wanted to do-- don't forget, at the time, the Xerox 7000 xerography was not grayscale. It was just literally black and white. It didn't do half-toning very well at all, but other machines, like the 3100, did. And so if you wanted to experiment with half-toning algorithms..

[10] The Xerox 9700 computer printer, introduced in 1977. Described in more detail below.
[11] Emitter-coupled logic, faster than Schottky/TTL.

Rider: Yes, that's true.

Sproull: ... You took an image generator that you had, and you hooked it up with the ROS adapter to a different machine. It worked very easily.

Sproull: So you mentioned that you had built two RCGs, two Research Character Generators. One of which went off to Pasadena, which was Xerox electro-optical systems, and they had been playing with scanners of various kinds. Dale Green, who worked on color machines down there, I worked with a great deal. But as you mentioned, there was a move underway, a project underway, thinking about computer printing as a product. And this is what ultimately resulted in the Xerox 9700, which was a two-page-a-second printer based on the two-page-a-second copier. That's the 9200[12]. So there was no real new major xerographic engine, but a SLOT gadget had to be built, and the image generator and so on. And of course, a big component turned out to be interfacing to an IBM machine, because at the time that's where all the source of computer printing came from.

Rider: Yes.

Sproull: And in fact, I just have to mention, I remember that Xerox management at the time actually was pretty heavily populated by IBM refugees. And had the attitude that if IBM was in a business, that might be a pretty good business for Xerox to be in, too.

Rider: Yes.

Sproull: Because IBM had kind of figured it out and managed to get the-- done what today you'd call the going-to-market steps. Making sure the customers really wanted it, and worrying about all the adaptations that their business practices might need if you wanted to insert computer printing. So I think that-- I'm sure that must've been part of the impetus to develop the 9700. But the other, of course, is they had this magnificent new 9200 technology, two pages a second. Magnificent machine, but as you pointed out when you were talking about the 7000, it was a blue-sensitive belt. And so it wasn't a trivial process to build SLOTs and get the project going. So although the 9700 wasn't a direct descendent of EARS, it was certainly coincident and shared character generation technology and other experience that came from what you and Gary had figured out in the research lab.

Rider: Yeah. So Gary, in fact, consulted with them on the SLOT part of the machine. And this was actually done by a group in Dallas under Jack Lewis, but the optical part was done in Pasadena by Electro Optical Systems, EOS. And there were a number of problems. I mean, the blue-sensitive belt, photoreceptor, was the biggest problem, because that needed a blue laser. And the only blue laser that existed at the time was helium-cadmium. And they had an incredibly short half-life. They worked, but they-- so there was a development project to stabilize the helium-cadmium laser, which was big and ran hot and all sorts of things. They also decided that they were going to use two beams of light so they could run the polygon more slowly. And they could actually get a more stable polygon.

[12] The Xerox copier product used as the basis for the Xerox 9700 computer printer.

Sproull: So these two beams would be illuminating two adjacent scan lines?

Rider: Two adjacent scan lines.

Sproull: On the same scan? Okay.

Rider: And so they were, in fact, worried that you were going to get interference effects. But it turns out that when they split the beam, it split into two orthogonal polarizations. And if you ran them at 90 degrees to each other, they wouldn't interfere with each other. So the two beams were finally done that way, and it meant the character generator had to play out two scan lines at the same time. And this came out in '77. So a few years after we had built the slot for EARS-- and the motors-- the polygon motors were better. And so they didn't have to use a servo bit clock. They used a high-frequency crystal, and then started it with a start-of-scan pulse with the phase that was closest to the start-of-scan pulse and then went from there. And that-- and they were 300 lines per inch, so it ended up being about the same actual image bandwidth as EARS had. So there were lots of issues with the SLOT machine, just having to do with the photoreceptor. The image generator was a simplified version of the RCG. Since it was 300 lines per inch instead of 500 lines per inch, the compression algorithm wouldn't have done very well. You might have gotten two-to-one instead of four-to-one on the images, but also memory was two or four times cheaper by then. So they just bit the bullet, didn't compress the fonts, used it in raw form in the font memory. And this made the character generator considerably simpler, and probably took three or four stages out of the pipeline. But they used the same microcode algorithm. As a matter of fact, all of the commands for that character generator are exactly the same as the RCG. And they went back to a PDP11. And they discovered a clever but-- well, no, it's not questionable. It was a very clever way of making it run faster by going in and patching a code after it had compiled. Because there was no way to compile what they wanted to do, but the machine would happily do it if you just went in and magically tweaked one of the instructions. So they did that, and it was efficient and it started off on a PDP11-34, and eventually went to the J11 processor, which was the PDP11 in a chip a few years later. It was a very successful product. It easily made a billion dollars for Xerox corporation.

Sproull: So that was introduced in June '77..

Rider: Yes, that's correct.

Sproull: Which was just about the time that you moved from PARC down to El Segundo.

Rider: That's correct.

Sproull: So I'm wondering, did you get any glimpse before moving of what product development of this sort at Xerox was like? I mean, obviously you were on the fringes of it, you and Gary. And yet you were about to step out of research into product land.

Rider: I got convinced that this would be good, for me to do this, and I did it. Product development wasn't at all like PARC. I mean it has interesting attributes. I mean, you can't do anything as quickly as you could

do yourself, but when you have 100 people working for you, you can do bigger, more complex things at least as fast. And so it's an interesting tradeoff. I mean, it was always more fun doing it yourself, but it was satisfying, being able to do large and complex things that there was just no way that any single person or five people could do. It took a lot more people to do. So the first thing we did was we built three prototypes, which were called Xenia, Tor, and Verde, which were 20 page-per-minute, 50 page-per-minute and 120 page-per-minute engines. Maybe it was 110 page-per-minute engine. They were all 600 lines per inch, and they were all-- the middle one was basically a computer printer. The slow one and the fast one were really what today we'd call multi-function devices. They scanned, they would print as well, and then output always a compressed image that was decompressed in real time going to the engine. And back in those days, when you were at 600 lines per inch, and you were at 120 pages per minutes, the only technology you could use was ECL. It was the only thing that was fast enough, so it was-- I mean, there was a decompressor that was 200 chips of-- and it was pipelined as well. So we did those, and we recommended that they take either the fast one or the slow one. And so Xerox made the middle one a product. And that was the 5700, and it wasn't very successful.

Sproull: And this was a computer printer product?

Rider: It was a computer printer product, and hooked onto an IBM mainframe, again, but it also wasn't-- unlike the 9700, which was a heavy-duty, robust engine--the 5700 was built on a xerographic engine that was not so robust, and just couldn't deal with the volumes that a computer printer needed to deal with. We then went on and did a product called Docutech, and Docutech was 600 lines per inch. And again, did both scanning-- did copying and printing as well. And the printing was done, in this case, all in software. And then when the image was created, it would be compressed, so it looked like a scanned image at that point. And the trick was having a compression algorithm that did okay with half-tones, and did okay with edges, and maintained the fidelity on both half-toned images and on text.

Sproull: So just out of curiosity, you were able to-- if you're going to scan and print, and you wanted to scan a picture, then you got a grayscale image that you then had to half-tone?

Rider: We half-toned it.

Sproull: And so I know Xerox had been working on half-tone algorithms with xerographic output for years and years, but that meant you had to actually choose some half-toning algorithms that were going to be good enough for both the current digital technology and the current xerographic technology.

Rider: Yes, which mostly-- I mean, this was still a black-and-white machine. So really there weren't a lot of choices of half-toning other than you do it at 90 degrees or you do it at 45 degrees. And how many dots per inch do you half-tone? And we experimented with all possible dot combinations that worked with 600 lines per inch, and then had to design a compression algorithm that would compress the half-toned data.

Sproull: Meaning after half-toning?

Rider: After half-toning.

Sproull: Right.

Rider: And we did that, and of course that means it's a more complicated algorithm, which means a deeper pipeline to get it to run at speed and that kind of stuff, but that all worked.

Sproull: And in all these cases, you're building page images of the whole page, compressed, and spooling those?

Rider: Yes.

Sproull: So there's some discs involved in all of these Docutech engines.

Rider: There were three discs that ran in parallel, because that was the only way you could get enough bandwidth off the discs in those days. We tried to get a manufacturer to stream multiple heads off one disc, and never happened.

Sproull: Right. Right. So you wound-- I don't know whether this was Docutech or later on, but eventually these kinds of reprographic systems involved custom VLSI, custom-- was it custom ECL, custom CMOS?

Rider: Docutech had one and a half micron CMOS. And we did an architecture on that which had about ten custom chips. It had a microprocessor which was conceptually an Alto. I mean, and it had microcode in it, and the microcode ran Mesa, which was the language that was being used at PARC and moving towards Cedar, but it never did Cedar. It had a cache memory chip that I think was a four-way associative cache, and it used an algorithm that either Chuck Thacker or Butler Lampson had initially proposed, called a snoopy cache algorithm, which allows multiple processor systems to know when another processor is accessing a piece of data that's in your cache and provided instead of the memory, which I think is still an algorithm that's used in a lot of systems now, all in one chip.

Sproull: Well, it's just cache coherence among multiple caches.

Rider: Yes. And so I think we were the first ones to actually put that in VLSI. And then we had a bunch of other chips that had to do with compression and decompression and things like that. But it was a-- it ended up having about ten processors in it, and coherent caches.

Sproull: And I think you mentioned that the architecture was such that you could scale it around performance and speed, essentially. Speed and resolution needs.

Rider: It was. I mean, and so it had a little one that only had one processor, and it had a big one that had, like, 20 processors. Xerox tends to like big machines, so they of course built the biggest one. And I don't think it ever got used for a smaller machine. And Docutech's went into the field in '92, and I think it stayed in the field for almost 20 years. For the life of me, I don't know how they continued to build the 1.5-micron CMOS. By that time, I had moved back up to Palo Alto and such.

Sproull: Okay, but before you move, I have to ask you one more question. So was this project probably the biggest project outside PARC that used Mesa?

Rider: Yes.

Sproull: And I'd be very interested in your reflections on what it was like to teach a bunch of programmers Mesa, and what their reaction was, because this was a highly-typed, strongly-typed language at a time that that was not particularly popular.

Rider: They actually liked Mesa, because it made-- because Mesa had a lot of hooks in it that allowed you to do multi-processing conveniently. And this was, I think, the first multi-processing-- well, it was the first multi-processing product that required a coherent system. And Mesa was very helpful with that. Eventually, a good deal of the code-- I guess not in Docutech, but later--then moved to C++.

Sproull: So talk a little bit-- so Docutech must've also gone through a series of-- well, confronted an outside world different from an IBM mainframe. So there was networking involved, there were different data forms, different page description languages and so forth involved. Talk a little bit about how those-- how were the products defined, and how did you deal with evolving needs?

Rider: Okay. So I came back to PARC in '77. And this-- excuse me, '87. Yes. '87.

Sproull: Off by ten.

Rider: And this became a product in '92. And I stayed involved with a lot of the decisions on upgrades even after I was back at PARC. But so technology kept marching forward. They had to keep making 1-1/2-micron chips, but for example, you could no longer get the disks. So you had to change out the disks and the disk interfaces changed from proprietary disk interfaces to more standardized disk interfaces. And so whenever they did that, they had to upgrade the microcode that controlled the disk and change the board that interfaced to the disk somewhat. And so there was a constant upgrading process. The initial protocols for Ethernet were the old XNS[13] protocols from Xerox, and they finally moved over to the standard protocols. And so they had to make those changes. They probably started with a subset of Interpress, and then moved over to PostScript as time progressed. That wasn't as difficult, because the way Docutech worked, the Interpress or PostScript was all done in software and then compressed, so it was-- you had to rewrite the software, but you didn't have to rebuild the hardware. And so that was good. But it's interesting thinking about products in this timeframe, because it used to be that you built a product, and basically it stayed the same for its entire product life. And this was just getting into the point where things evolved during the product life, and you had to keep up with the evolution, because you needed disk drives that-- well, one, they were faster and better, but also you couldn't even buy the old one, so you couldn't even do spares. So we're now to a point where the hardware is more or less constant, or at least the major changes are software changes, and I think that's a better place to be.

[13] Xerox Network Systems

Sproull: Right. Certainly all the networking and page description formats were entirely software things.

Rider: Yes.

Sproull: And I would think that one of the trickier parts of all the software was dealing with the real-timeness of the machine that-- the ultimate engine you're controlling.

Rider: Yes.

Sproull: And getting stuff off the disk fast enough et cetera.

Rider: And a lot of that happened in microcode as well as a Mesa overlay that directed it.

Sproull: So we're reaching the end here. So talk a little bit about your satisfaction of working in product development. You told me that you had wanted to go off in technology into products, and I think you also got help from a number of people, certainly were not-- it took you a while, I think you said, to figure things out. But there were people to help along the way, including-- I think you mentioned when I asked you why you went, you said you thought it would be good for you. And I think there were people who counseled you in that direction.

Rider: Yeah. I mean, a mentor I had at that time was Harold Hall, who was up at PARC, and he thought that it would be a good management experience, and it was. And it served me well in the future. As I said, management has its pros and cons. I mean, it allows you to do more things and more interesting things and bigger things, but it also means you don't get to do as much yourself, and I'm a hands-on tinkerer kind of person. So I always stayed pretty involved, to the dismay of many of my employees. But it certainly served me well as I continued on in Xerox.

Sproull: And I think you also mentioned that the PARC people continued to be helpful on technical things, if necessary.

Rider: Oh, absolutely. I mean, at least once a month I was up at PARC, and I'd spend time with Chuck and Butler, and they'd always have good thoughts and good ideas.

Sproull: So after leaving El Segundo and then back at PARC and so on, you then took on a variety of more sort of corporate functions. The system architecture, corporate architecture, talk a little bit about that. What were you trying to achieve? What could you achieve? What was available in those roles?

Rider: Yeah. So the idea, of course, was to have a corporate architecture so that we didn't use a different processor in every product, and a different language in every product, and different interfaces and everything. And we worked on that for a few years and made some progress. But it's slow, and the complaints were real. I mean, if you do something using the latest, greatest product out, you can almost always do something that's innovative and new and different and that kind of stuff. But the bottom line is it

takes you longer to get there. And it was-- I mean, when I finished, Xerox was better at that, but they weren't as architecture-centric as I would have liked.

Sproull: Yeah. I think you also mentioned that a lot of the pushback was about performance. And it was very hard to get anybody to appreciate-- I think this was true of PARC people, too. Anybody to appreciate the incredible forcing function that Moore's law was providing for the whole business.

Rider: Yeah. And it took time for me to really understand that as well. I mean, you basically were involved in the next character generator beyond EARS, and the technology had totally changed in, what? Three years?

Sproull: Well, it was principally memory was the issue there.

Rider: Oh, memory was the big issue.

Sproull: Yeah.

Rider: I was going to say it slowly creeped into software, but it didn't slowly creep. It was an avalanche. I mean, every two years you could do so much more in software and so much less in hardware.

Sproull: Well, so-- and to exaggerate slightly, we were able, with one card[14] and an Alto, to do what you did with ten in the RCG.

Rider: Yeah. That's about right. Yeah.

Sproull: And that was about five years later, or four.

Rider: Yeah, yeah. And I mean, if you kept-- I mean, some of the things I worked on in El Segundo, we kept pushing the envelope, so it still-- there was still a lot of hardware in it, but even there, more and more was moving into software. And that's just the way it is. I mean, Moore's law is-- I mean, what was it? It was 1990 and we were already talking about how Moore's law was going to end and it couldn't last more than a couple of years, and it's still going.

Sproull: Okay. Well, so in preparing for this interview, I did a bit of thinking about, obviously, listening to you, your career. And it struck me that you had spanned quite large evolutionary changes in several areas that became, as you were just talking about, became qualitative, not just quantitative. You went from sort of a bit-starved world where memory was so expensive you tried very hard not to use it, to VLSI becoming a standard building block, custom chips, for example, were something that you did for performance and not just cost. And you saw software overtaking hardware in many, many ways, again because things got faster and so on. And then there was-- you didn't mention much about this, but there was the whole networking evolution from sort of a start with-- a chaotic start, frankly, in product land with

[14] This was the Orbit design, which actually added 4 cards to the Alto.

many, many proprietary protocols, and it must've been a huge problem in deciding what to support and what to-- how much tolerance you had for the chaos. And we eventually came to a much more stable world, but it took a long-- it took quite a while. So several aspects of the technology changed quite dramatically in that period.

Rider: Well, I mean, one thing to note is in the 50 years we're talking about, Moore's law has been more than a million-fold. I mean, actually I think it's more like 30 million now. That means something that used to take you roughly a year to compute now takes you roughly a second.

Sproull: Well, and moreover, I'm reminded actually of a famous quote due to Bill Wulf, which is a factor of two in performance difference usually means you should be looking out for a qualitative difference as well. There may be an architectural difference or an algorithm difference, something you need to pay attention to that's not just the fact that it runs twice as fast. And you were confronted with that again and again.

Rider: Yep. Yep. We're talking about two to the 25th now. Yeah.

Sproull: Well, I guess maybe let's just close by agreeing it was a great 50 years for both of us to be in the computer business.

Rider: It was a fun time.

Sproull: It was a fun time. Thank you, Ron.

Rider: Thank you very much.

END OF THE INTERVIEW

