# Oral History of Ivan Sutherland: Part 2 of 2

Interviewed by:
Robert Sproull
David C. Brock
Jim Waldo

Recorded March 10, 2017
Mountain View, CA

CHM Reference number: X8081.2017

**Sproull:** Okay. Good morning. This is the 10th of March, 2017, and we're interviewing Ivan Sutherland in a second oral history session. I'm Bob Sproull. David Brock from the Computer Museum is here, and also Jim Waldo, a colleague of Ivan's and mine at Sun, and we're going to sort of trade off questions, but everybody is allowed to chime in as appropriate. So Ivan, last time we left, we left off, as you were headed from Caltech to CMU. So maybe if you can cast your mind back, you can talk a little bit about why you went to CMU and what the new project was going to be.

**Sutherland:** When I was a senior at Carnegie Mellon, Everard Mott Williams, who was the head of the EE department, wanted to find out what the going price for a research assistant was at various universities. He would have his secretary fill out all the application forms to a dozen or so graduate schools if I was willing to apply to them and he would pay the application fees. The only stipulation was I had to tell him what happened. I think he used this to gauge the competition to figure out what he would have to offer to get decent graduate students. So I applied to a lot of places. Cornell invited me to come up for a tour, which I did, but they made a serious mistake. They arranged it in January. I decided that Cornell was not the place to go. Too cold. Too snowy. I got admitted many places. Got admitted to Caltech and finally settled on Caltech, because it was the furthest away from my brand-new mother-in-law, who lived in Pittsburgh, and so my new bride,Marcia, and I took off shortly after we were married to go to Pasadena, California. It was a great idea, the right thing to do, and absolutely fabulous thing. When I got to Caltech, I discovered that computing at Caltech was a little backward. They had not been at the forefront of the computer industry. They had done great work in analog computers. They had some of the biggest analog computers around.

**Sproull:** So I'm going to interrupt. Let me fast forward. We were trying to get back to the point where you left Caltech after the Silicon Structures project.

**Sutherland:** I see.

**Sproull:** And went to CMU to do the robotics project.

**Sutherland:** I see. So we're--

**Sproull:** This part we have covered.

**Sutherland:** I'm a year or two too early.

**Waldo:** This is-- you're a repeat offender.

  

**Sutherland:** What's that?

**Waldo:** You are a repeat offender at a number of institutions and we're looking at the second crime.

**Sutherland:** I see. Okay. Fine.

<laughter>

**Sutherland:** Yeah. Well, let's see. There was this character called Marc Raibert, who came into my office at Caltech and said, "I have some projects in mind. Which of these do you think would be the best?" There were three projects. I forget the first and the third one, but the middle one was to build a computerized pogo stick, and that seemed to me a no-brainer as an interesting thing to do. Raibert subsequently got tenure for figuring out that what we thought was hard was actually quite easy. That you have essentially three servos, each of which is a simple first-order linear servo. One to keep the hopping height right, one to control the position of the foot when you're in flight, in order to make the foot land in the place you want it to be at the next landing, and the third one to control the attitude of the body at the only time you can do that, which is when the foot is on the ground and you have something to push against. All three are perfectly straightforward, simple things. He built his pogo stick and it worked just fine, and he's gone on to newer and better things. You guys ought to see his latest video, which is his two-legged, wheeled--

**Sproull:** Yeah.

**Sutherland:** --two-legged-- he has a biped that has wheels for shoes. I think that's the right way to describe it.

**Sproull:** Well, it's a Segway glued to a body.

**Sutherland:** What's that?

**Sproull:** It's Kamen's, Dean Kamen's Segway, glued to a body.

**Sutherland:** Yes, exactly. Well, but no. But Segway has the two wheels are on a fixed axis. This has two quite separate wheels.

**Sproull:** That's true.

  

**Sutherland:** Literally they have wheels as feet and it's interesting--

**Sproull:** That's true.

**Sutherland:** --it banks when it goes around corners and all--

**Sproull:** Yes.

**Sutherland:** It's really wonderful. So he had determined to go to MIT and I persuaded him to go to Carnegie instead, because my friend Bob Sproull, who is here today, had decided to take a teaching position at Carnegie Mellon. So that's why I went to CMU, and-- yes. Now, I don't remember the sequence of things, but perhaps earlier than that, Charlie Molnar and Sproull and I had gone in to see the Chairman of Computer Science at CMU and said, "We want to rent some of your students to teach a class." Maybe it was afterwards. Was it afterwards?

**Sproull:** Must've been after. I don't--

**Sutherland:** I don't remember. But it was-- what was his name?

**Sproull:** I don't remember.

**Sutherland:** Was it Habermann? or… I forget, but we told him we wanted to pay for the privilege of teaching a class at CMU. He damn near fainted. It was really quite a remarkable thing. <laughs>

**Sproull:** Well, this was your doing.

**Sutherland:** What's that?

**Sproull:** This was your doing. Ivan's idea-- correct me if I'm wrong-- was Ivan wanted a little bit of space and access to students at CMU, so he went in and totally preempted the conversation by flipping the sign of the money transaction and basically proposed, "I'll pay you," you know, CMU, "if you'll give me a place to play."

**Sutherland:** Exactly.

  

**Sproull:** Is that right?

**Sutherland:** Yeah, exactly. Exactly. But I wish I remember who it was.

**Sproull:** Well, we'll come back to that.

**Sutherland:** He's dead now, but anyhow, it was a strange, strange deal, but it worked very well, because it gave us the privilege of faculty without the responsibilities. A very desirable position. A position I have now at Portland State, as I-- when we first moved to Portland I went into the Portland State dean and said, "We would like very much to set up a research laboratory in your engineering school," and he said, "Of course, it'll have to pay for itself." So he said, "Okay," and he's been very good to us. He provided a space and we have students. We've graduated two Ph.D.'s and we've got a couple more in the pipe that'll be coming out reasonably soon and having a wonderful time. Being in the university with the privilege and not the responsibility makes a very pleasant arrangement. My wife, Marly, of course, is actually on the faculty and so she actually has some of the responsibility but…

**Sproull:** <laughs>

**Sutherland:** It's okay. So that's why I went to CMU the second time was to do robotics, and Marc Raibert built a machine with only one leg and I built a machine with six legs, and I made a mistake. A six-legged machine should be the size of an insect, because that's when you need six legs. But at the size of a horse, this thing was about a thousand pounds. The inertial forces are so large that you, you should just solve the balance problem and build a thing with fewer legs, and in fact, nature's figured that out. You don't find any large animals with more than four legs. That's not quite true. Kangaroo has five.

<laughter>

**Waldo:** Well, but two aren't used.

**Sutherland:** What's that?

**Waldo:** Only three are used.

**Sproull:** Only three are used most of the time.

<laughter>

  

**Waldo:** So why did you decide six?

**Sutherland:** I'm sorry?

**Waldo:** Why did you decide six?

**Sutherland:** Why did I decide to--

**Waldo:** To have six.

**Sutherland:** To have six?

**Waldo:** Yes.

**Sutherland:** Oh, six avoids the balance problem.

**Waldo:** Okay.

**Sutherland:** You can always keep tripod on the ground. But it turns out it doesn't really avoid the balance problem because at that mass, if you start or stop suddenly, the mass of the thing causes it to tip a bit. So, you know, the main, the main learning that we had was six is the wrong number of legs for a vehicle of, you know, modest size.

**Sproull:** So but Ivan, tell a little bit about actually doing that project, because I got the impression that you enjoyed switching from something that was mostly cerebral--

**Sutherland:** Oh, yeah.

**Sproull:** --in computing to something that was mostly bending metal and casting things in Pittsburgh and so on.

**Sutherland:** Oh, we had a wonderful time. There was-- I did a lot of mechanical design for this thing. Badly.

  

<laughter>

**Sutherland:** But I decided it was Pittsburgh, we ought to use some cast iron, because Pittsburgh is good at cast iron. So I went around, called up various foundries and said, “I’d like to make some castings,” and they said, “Well, how much are they going to weigh?” and I said, “Oh, probably a few ounces.” They, “We don’t do anything under 500 pounds.”

<laughter>

**Sutherland:** You know, this is Pittsburgh. They cast the rollers for rolling mills and stuff like that, right? But I found some small foundry. The guy said, “Well, I’m not going to make any money <laughs> on this, but I’ll teach you how to do it and we’ll make a few castings for you.” So we made a few little castings. I learned how to make a pattern, a wooden pattern, which then gets embedded in sand, moist sand, to leave a hole the shape that you want the iron to be and then you fill it with molten iron and it comes out as a casting. I just recently, going through my storage place, found the fillet material that I had bought in a little store in Pittsburgh. Was a little very dusty old store that sold material to pattern makers, and the question is how do you make the nice, rounded place where one surface meets the other and there’s a nice fillet in there that makes it strong because there’s no sharp corner? And the answer is, you take the, part of the outside of a cow, which is cut. The outside of a cow is relatively flat and then it’s cut with curve inside, so when you push that into the corner the outside of the cow becomes the fillet, and I just recently found I have now four or five meters of leather fillet material for making patterns should I want to make another pattern. <laughs> I can put this fillet material in the inside corners and make nice fillets in the thing. It’s a treasure. It’s really a treasure. <laughs> The store that I bought it in was a remarkable little place, been in business for probably hundred years in the same place and it had, you know, dusty drawers and all kinds of things and it’s a wonderful place. So that was great fun.

**Sproull:** Well, and then you had to finish the casting and for that you had a machine shop in your Devon Road house.

**Sutherland:** Well, we had some machines in the Devon Road house where we cut metal and drilled holes and stuff like that. The CMU machine shop was the place where we did any precision stuff that needed. They had nice milling machines that we could use and so on, so… But it was wonderful fun and I learned that cast iron machines beautifully, and the reason is it has a lot of carbon in it, so when you cut cast iron, the carbon becomes graphite and lubricates the cutting tool, and so it’s very nice. Machining cast iron is very nice and I had, in fact, yesterday morning, and on my drill press in Portland, I was drilling some holes in piece of cast iron and remembered how nicely it drills and it just very smooth, very smooth.

**Sproull:** <laughs>

  

**Sutherland:** So… One picks up various skills along the way and they last, you know.

**Sproull:** So let's turn to another aspect of this, which is you recruited some students to this project, or they came and darkened your doorstep and you let a few of them cross the threshold. Doing different things with the machine. You remember? Tell us little bit about who they were?

**Sutherland:** Well, the guy who did the most with the machine was Marc Donner. Marc Donner had got a master's degree at Caltech, I think. So he already had a master's degree and he was at Carnegie Mellon as a Ph.D. student and he wrote code for the PDP 11 processor, which was a single-board PDP 11 that was in the walking machine, and he wrote code for it to control the legs, and I think the main feature of his code was that there was a separate piece of code for each leg and they were coordinated by communication between these separate pieces of code on the principle that that might be how a cockroach's mechanism for controlling its legs would work, and so the coordination of the legs was a little loose, rather than just programming the whole thing as a unit, and he did a nice job of that. He's now working as the CTO of Uber in New York City. The students who were the most fun were contemporaries of my son Dean. Dean's good buddy Kevin Nolish was the sense of humor of the group, right? <laughs> He wrote me a note once to the capitalist mongers complaining that we weren't paying him enough, and it was signed you know, the workers of the thing. I forget what it was, and then <laughs> one day, one day Bert was sitting in the cockpit of the robot, which had a stick from an F4. Bert had written to one of his old Navy buddies and said, "You must have a hangar dolly somewhere that we could get the stick out of," and he got a note back saying, "We don't have any junked airplanes here, but here's a brand-new stick for an F4."

<laughter>

**Sutherland:** So it was a nice pistol grip with firing buttons and trim tab controls and so on. So Bert was sitting in the cockpit with a faraway look in his eye <laughs> and Kevin came up to him and said, "Well, Bert," he said, "do you think it's time we discuss the armaments package?"

<laughter>

**Sutherland:** So projects can be wonderful fun. In fact, I have the attitude that if research isn't fun, why are you doing it? Are you exploring the unknown, and why in the world would you bother doing that? You can't do it for rewards, because you don't know what the rewards are. You don't know what you'll find, so you do it because it's an adventure, because it's something that enjoys the spirit, and the camaraderie of a research group is one of the things that makes research worth doing. I think that's important. An important thing to know about research is that if the researchers aren't happy, there probably isn't much research going on. It's hard enough wrestling with nature that you don't want also to have to wrestle with management.

  

**Sproull:** Well, and the machine had a personality too. I think I may have the distinction of driven it-- I broke it the last time.

**Sutherland:** Yes. The last driver of the walking machine was in fact Bob Sproull.

**Sproull:** Yeah. It--

**Sutherland:** It's not the--

**Sproull:** --felt very uncomfortable. <laughs>

**Sutherland:** Well, perhaps so, but I didn't think it was his fault. The castings that broke were, in fact, not nearly strong enough.

**Sproull:** <laughs>

**Sutherland:** So it was clearly the designer's fault. That's me, right? But in these kinds of research projects, you don't assign fault, you don't assign blame. You just learn and go on and try and figure out-- it's like computer programming. It's useless to assign blame to who put the bug in. You just want to find the darn bug and get it out, and you know that the bugs are there. There's bound to be bugs in any complicated thing, so the question is not, "Have you got them all?" the question is, "Are your tools sharp enough to find the next one?"

**Sproull:** Yeah, especially the tools up here.[1]

**Sutherland:** Especially that one, yes, and I'm pleased to say that in Portland now we have a research group which has a good deal of camaraderie in it, and it, you know, it is just a pleasure to be there, and I figure that being involved in that kind of activity is going to add an extra decade to my life because I have a reason to get up in the morning, I have a place to go, I have something I want to do, and I do it with people that are enjoyable to be with. It's almost like Sun Labs was at one time.

<laughter>

---

[1] Bob Sproull points at his own head.

**Sproull:** So Ivan, just to finish off this phase a little bit, I think a lot of people-- this is not my question but a question that a lot of people are curious about, is whether this project was an isolated one or had some relation to robotics research generally as it was going on in the community at the time or, heaven forfend, A.I.?

**Sutherland:** Well, I don't know about A.I. but…

<laughter>

**Sutherland:** But it was current enough. DARPA had just started its robotics project. Perhaps in response to this interest that Marc Raibert and I had evolved. I don't know. But they provided some sponsorship to us to do the write-up and there was a book, we wrote a book about it, which was privately published, and there are copies of it in various places. I'm pretty sure the Computer Museum has one, and if the Computer Museum doesn't, we will rectify that.

**Sproull:** <laughs> Well, Ivan has a small box full of them.

**Sutherland:** I have a small box of these books, right. It's called "A Walking Robot" and it was illustrated by a man named Carlson, with very nice drawings, and there's a few photographs in it.

**Sproull:** So okay. Let's move on a little bit. Still in Pittsburgh. So you mentioned that--

**Sutherland:** I want to say a couple more things about the walking machine.

**Sproull:** Yeah. Okay, good.

**Sutherland:** You know, there's-- it was a hydraulically run machine. There were hydraulic cylinders to run it and hydraulic pumps that powered them, and so there was quite a lot to learn about how hydraulic things work and one of the interesting learnings was about O-rings. O-rings are used to seal, make seals between pipes and fittings and various other places, and I learned how an O-ring works. An O-ring is always smaller than the groove in which it sits. So the volume of the O-ring is smaller than the volume of the groove. But the O-ring works, because it's kind of fluid. It's made of a soft material, and when the pressure of the oil gets on it, that pressure it transmitted throughout the O-ring and the O-ring behaves like a piece of fluid and it goes as far away from the pressure as it can, which causes it to seal the crack into the next place, and it really works very, very nicely and it's very simple, and I made some hydraulic manifolds which were essentially printed circuit boards for oil. There's an aluminum plate and another aluminum plate, and a groove in this aluminum plate, with a little groove on each side for the O-ring, for the seal, and you can buy O-ring material in long lengths and glue it together at the end and put it in the thing and then seal it all in and it doesn't leak. It's quite amazingly good. Except once one sprung a leak and it's, you know, it's 2,000 PSI oil and a leak at 2,000 PSI is pretty spectacular. I mean, it shoots <laughs> clear across the room and makes one hell of a mess, but we had one, so… But by and large the thing worked remarkably well. There was a great deal of learning that I had about the techniques of hydraulics. I don't think we learned anything that wasn't already known but it was interesting to me to catch up in a mechanical field that I knew nothing about. Yeah.

**Sproull:** Okay. So I'd like to move on to Sutherland, Sproull & Associates, which is a little consulting company--

**Sutherland:** Have to speak a little louder, Bob.

**Sproull:** I'd like to move on to Sutherland, Sproull & Associates.

**Sutherland:** Okay.

**Sproull:** Which you and I set up as a consulting vehicle I think in 1980?

**Sutherland:** Yes.

**Sproull:** And this was discovered because I had gone to CMU and would do a little bit of consulting and sometimes you and I had similar opportunities and we discovered it was a lot nicer to do it together than by ones self. The pressure was lower, et cetera. But the main-- but aside from some consulting clients, the main Sutherland, Sproull project was we decided to try to fund a research program in asynchronous systems by signing up some corporate sponsors and agreeing to teach them the results of what we learned. Does that--

**Sutherland:** Yes.

**Sproull:** Does that--

**Sutherland:** This was--

**Sproull:** So tell about how we got that started and--

  

**Sutherland:** Well, this was well into the, well into the period of Sutherland, Sproull & Associates and Sutherland, Sproull & Associates' principal client in the early days was the venture capital firm.

**Sproull:** Right.

**Sutherland:** And we did, we did due diligence on deals for the venture capital people, which was what basically supported the firm. We incorporated. A good thing that we did was to incorporate, and I, for some reason, I knew that we should do that, and so it was set up as a corporation. It was a California corporation. Very easy to set up a corporation, and it was well worth doing because it separates the individual owners from the liability of the corporation, After a while I had been, started getting interested in the asynchronous world. How could we make things self-timed so that each thing would happen when it was ready to happen, not before and communication delays would automatically be accounted for? So while I was lying on the beach in Australia… I took two, two holiday trips, three holiday trips to Australia, of two months' duration each, and in one of those trips, I forget which one, I was lying on the beach and I thought, "Well, if we hooked a bunch of Muller C-elements in series we could make a FIFO, a first-in, first-out device," and I puzzled about it and figured out what the latches should be and so on, and I thought, "This can't possibly work," because there was no asymmetry front to back. The input end and the output end were perfectly interchangeable. So how could it know which way was first-in and which way was first-out? And I couldn't believe this thing could be made to work. It was, it just, it was ridiculous that you should think that a symmetric thing could be able to provide FIFO action, and so when I got back to Pittsburgh I got one of the assistants that we had to actually build one out of some little chip things that we had and a protoboard. That was David Douglas. David Douglas is now working at Sun Micro at Oracle, in the computer business, and he built it and, by God, it worked. Was really strange, and that became the basis of the paper that was published called "Micropipelines," which the Turing Award people called me up and said, "Would you accept a Turing Award?" This is like asking, "Is the Pope a Catholic?"

<laughter>

**Sutherland:** Not a question to which you likely say "no." They said, "On the other hand, you're going to have to give a speech and you're going to have to write a paper," that we're going to publish in the ACM communications, with no reviewers. Hmm.

<laughter>

**Sutherland:** Now, that's the best part of the Turing Award, right? You get a free, a free pass to a published paper, right? Without the idiot review process.

<laughter>

  

**Sutherland:** I mean, this is a prize worth more than the money, right? Well, it was worth more than the money then, but maybe not now.

**Sproull:** <laughs>

**Sutherland:** So I wrote this paper called "Micropipelines," which talked about this very strange FIFO. Talked about transition signaling, the idea being that if you want to send information, one way to send it is by just changing the state of a wire, and the transition can carry the information rather than the level. I talked a lot about transition signaling and in retrospect I wish I hadn't done that, because it set the asynchronous community off on a bad track. What I believe now is a bad track. I would advise anybody who got into this to avoid building a micropipeline. It's got too much symmetry. It's got front-to-back symmetry, it's got Vdd to ground symmetry. It's got all kinds of symmetries, which makes it very hard to get right, because any two things that you do can cancel or-- it's just very hard, and since then we've learned quite a bit about what we should've done, but it's taken a long time to learn why and what to do in a field that's new.

**Sproull:** So we'll come back to a lot of that, but you also gave a talk as part of the Turing Award.

**Sutherland:** Yes.

**Sproull:** And I recall you had a favorite way of engaging the audience in talks about micropipelines.

**Sutherland:** Yes. We did what I subsequently learned was called a KLA, a kinetic learning activity.

**Sproull:** <laughs>

**Sutherland:** Okay. This is the proper pedagogical term for this thing. It's basically a demo using victims, using people, I'm sorry, <laughs> as the demonstrators, okay? And my favorite of that was to get each person to emulate a Muller C-element. I had the great pleasure of giving this demo in, at the University of Illinois, in a lecture I gave where Muller himself was present, and Muller at that time was, you know, at retirement age. His wife had recently died. I think he was not a very happy person at the time. But I picked the row to demonstrate this in which Muller sat, of sourse. And so I asked him, "I know that you invented the Muller C-element, but have you ever before *been* a Muller C-element?"

**Sproull:** <laughs>

  

**Sutherland:** And he admitted he had never before been a Muller C-element. He did tell me, by the way, why it's called a C-element. I thought this was kind of an interesting piece of history. He said that they always drew them and the inputs were always called A and B and the output was always called C and so it became called the Muller C-element, and that's where the name came from. At least that's what he told me, now. Your mileage may vary, but that's the story I got from Muller.

<laughter>

**Sutherland:** So the idea was in the micropipeline, each Muller C-element follows the rule, "If predecessor and successor differ, copy predecessor. If predecessor and successor are the same, hold whatever state you have," and each element is a state-holding element, so I have everybody demonstrate that they can hold a state and we use hand-up and hands-down to demonstrate the state, and I warned some of the participants that they may have to support their hand because it may be up for quite a long time and it's okay. It's not okay to let it go, but you can-- it's okay to hold it up by whatever means you have, and it's quite an effective demonstration. You see the flow of things going from one end to the other, flow of fullness going in the forward direction, the flow of emptiness going in the reverse direction, which is quite interesting. You've all seen this on the freeway. You've seen a crowded freeway. You see an empty space coming toward you from the front and then when you get to move, you move, and that moves the empty space to the back, and this is the symmetry of FIFOs, which is what happens at each stage is you trade a data item for an empty space. It's a trade, and indeed, you can, in fact, put data in that empty space instead of an emptiness and so you can have data going both directions at the same time. The symmetry of these things is really very interesting, and it appears over and over and over again.

**Sproull:** So as I say, we'll come back to some of this, but while we're on the Turing Award thing, there've been some subsequent aspects to the Turing Award in that there's an annual meeting of winners that I think you've gone to several times.

**Sutherland:** There is--

**Sproull:** Is that interesting in any way?

**Sutherland:** There is an annual meeting in Heidelberg of a thing run by what's called the Heidelberg Laureates Foundation, and a wealthy German industrialist gave a large amount of money to the foundation to run an annual meeting. Apparently there is an annual meeting of Nobel laureates and this is patterned after that. It encompasses the Turing awardees and the Fields medalists and a couple of other mathematical awards, of people who get together to meet and there's some 30 or so old guys like me who show up and then there's 200-odd young people who are chosen by application to somehow interact with these, with the older generation and presumably something rubs off from the older folk to the younger folk or maybe the other way around, I don't know, but that's the idea, and I've found them a very pleasurable thing to do. I've been to every one so far. There's been four, and I plan to go again in 2017.

**Sproull:** So okay. Let's go back to SSA. So you talked about the ATV connections. We've talked a little bit about the asynch project. So somewhere in there Bert came aboard.

**Sutherland:** Yes. Bert, Bert decided to leave the Xerox Corporation, where he was working at PARC, and Bob and I both knew him well and said, "Bert, will you join us?"

**Sproull:** <laughs>

**Sutherland:** And he said, "Yes," and he grumbled quite a bit at how much money he had to put in because the company was bigger than when we had started it. He stopped grumbling when we finally sold it to Sun and he got a major return back from selling the consulting company. But it was good to have Bert in. Now, yes, it was a strange company. It was an incorporated company, but it ran like a partnership, and the partners all knew each other very well and so it was an odd partnership. Most partnerships, at the end of each year, have a big fight over who should get the bonus, and in Sutherland, Sproull & Associates we had a similar big fight at the end of the year but it was over who should have to take the bonus.

<laughter>

**Sutherland:** And it was really quite a remarkable place because the three founders knew each other and cared for each other enough that we were interested in the welfare of everybody and it worked out very well, so for 10 years in the partnership and for some many years afterwards, I got to work with Bob Sproull, my favorite student of all time, and my older brother, the two men from whom I've learned the most, and one of the joys of my life is that I had that experience of working with people, closely with people that I care for deeply.

**Sproull:** So and if I recall correctly, it was your idea that the three of us should all be vice presidents. Perhaps another nod to symmetry.

**Sutherland:** I have no idea whose idea it was. <laughs>

**Sproull:** I think it was yours.

  

**Sutherland:** Well, it may be, but it turns out that Bob's wife was a student of organization theory. She studied organization theory as part of her professional activity, and seemed to be reasonable that she should be the president. She was the only one <laughs> who knew about organizations. So she was the president, and Bob and Bert and I were all vice presidents. She's one of the best bosses I ever had.

<laughter>

**Sproull:** So why don't you talk a little bit… You mentioned that at the end we sold SSA to Sun, and we were part of a founding nucleus of Sun Microsystems Laboratories, and that involved me in Boston because Lee wanted to, needed, to go to work at a university where she had an offer, and you and Bert out here in the Bay area. So why don't you talk a little bit about that. So why we did that and how that progressed.

**Sutherland:** Well, a consulting firm is consumptive of intellectual capital. You work with various people and you have intellectual capital, which you bring to the table, which is why they pay well and… But there's not adequate mechanism in a small consulting firm to generate new intellectual capital and I think after a decade we figured out that it would be good to be generating intellectual capital and so we went and talked to the Digital Equipment Corporation. We talked to a number of other places and talked to Sun. I think Eddie Frank was key in bringing us to Sun's attention, and Sun elected to buy us. Now, there's a bunch of interesting stories that resulted from that. So how does a big company buy a little company? Well, we were, in fact, a little company. We were incorporated in the state of California, so it was not just a straightforward thing, it was a corporate merger, and there's a complicated legal process that the big company goes through to make sure that the liabilities of the little company don't surprise it later on and so on, so three-way reverse merger, blah, blah, blah. I don't know what all the legal forms were, but we're sitting with the lawyers and the lawyers said, "Well, now you'll all become new employees of Sun." "Wait a minute. Wait a minute. If Macy's buys Gimbels and you have a 25-year veteran at Gimbels, he doesn't suddenly become a one-month person at Macy's. A corporate merger from the point of view of the employees is not change of employment, it's continuity of employment." "Oh," said the attorney. "That's quite right." It is quite right. Now, the problem was that Sutherland, Sproull, & Associates was two years older than Sun.

**Sproull:** <laughs>

**Sutherland:** And so my date of employment should have been before Sun was founded. So naturally I demanded badge number, minus one. Seemed appropriate, right?

**Sproull:** <laughs>

**Sutherland:** For… <laughs> Well, some things you can do. Some things you can't.

  

<laughter>

**Sutherland:** So the compromise was that our date of employment with Sun would be the date of the founding of Sun. With a result that as Sun employees, there was no one at Sun who had been there longer than we had, and when Sun was 25 years old, Scott McNealy handed out 25-year badges and very carefully said, “And by acquisition, 25-year badge to these people.” It was, that was rather gracefully done. But we did. We started the Sun Laboratories and Bert I think had a major hand in formulating the policies that made Sun Labs an interesting place to be. One of the ones I think, and you should get his correction on this, but his plan was each year to not quite spend his budget. You have to spend almost all of it to make a convincing case that you’re doing the work, but you want not to go over budget. You want to be just barely under budget so that you return a little to the corporation and he and the management of Sun got to trust each other this way and it worked well.

**Sproull:** So what role did you play during the early founding? And so Wayne Rosing was the first head of the lab, and he had an advisory committee of various people that met and so on. I’m sure you were given a title of Fellow. You were viewed as one of the wise heads in figuring out what the lab should do. Was there any interaction in that way?

**Sutherland:** We had quite an interesting advisory committee. I don’t remember who they all were, but seemed to me that an advisory committee should be people that you were to seek advice from, and I come to know a great many people and many of them fit in that category and so we got some of those people to work with us. I think Bert knew the people from Xerox who were also key, key thinkers in the industry, and some of them were involved in the advisory committee and it worked very well. I wish on the spur of the moment I could remember names.

**Sproull:** Yeah. We can--

**Sutherland:** Those may come to us collectively when we have a break here.

**Sproull:** So and you, from a research purpose, continued. Well, and with me and others, your asynchronous systems work, and building ever more chips and so on, but you had to build the new research group.

**Sutherland:** Well, I was interested in the asynchronous business and I had become interested in integrated circuit design from interacting with Carver Mead at Caltech, and so I was interested in how we could do that, how we could build self-timed chips that would make use of the notion that things could happen when they could happen in order to gain speed and to not waste energy when it was not necessary to do anything, and so the group that I headed was called the Asynchronous Research Group. But its name was subsequently changed to the VLSI Research Group, I think appropriately, because there's much more to VLSI than just asynchrony. We got Charlie Molnar to come and join us. Charlie Molnar was a certified genius. He, the man, was amazing. He had been at Washington University working on self-timed systems when they discovered the glitch phenomenon. It turns out that a flip-flop can be flipped or it can be flopped, and of necessity, in between there is a place where it can be metastable and hang, neither flipped nor flopped, for an indefinite period of time, and so he wrote a paper about this, he and Chaney, Tom Chaney, wrote a paper about this, and Charlie told me when they sent it in for review the reviewers said, "This is really important stuff. If it were true I'd know it. Rejected."

**Sproull:** <laughs>

**Sutherland:** Well, of course, it is true and the reviewer just didn't know it. It's not a part of the digital view that a flip-flop has a third state that's metastable, but it is, of course, true, and since that time, since that paper was published, it's become widely recognized in the industry that every synchronizer that you have may hang neither flipped nor flopped for a period that you'd better be careful of, generally longer than a clock period, maybe longer than two clock periods in modern technologies, and so you need to do calculations about the error rate that you might have, and I think that's by now become well understood and widely practiced, not always correctly, but widely practiced.

**Sproull:** Well, and in fact, so one of the roles of Sun Labs was to help out Sun product engineering as needed. There were early opportunities to help in SPARC processor design and some CAD tools and so on, and one of the guys in your group, Ian Jones, became the company expert about synchronizers.

**Sutherland:** He came and still is a company expert on synchronizing.

**Sproull:** Still is, and there are thousands of synchronizers in every Sun--

**Sutherland:** And each synchronizer has--

**Sproull:** --SPARC processor.

**Sutherland:** --associated with it a possible error rate, and the product of all those error rates had better be small enough that it doesn't cause you grief, and so there's some careful statistical calculations to be made there.

**Sproull:** Well, and there's also careful design required.

**Sutherland:** Correct.

  

**Sproull:** Okay.  So any other comments?  So we were at Sun Labs quite a long time.

**Sutherland:** Yes.  That's the, that's--

**Sproull:** 1990.

**Sutherland:** Yes.

**Sproull:** Was when we joined Sun.

**Sutherland:** That's the longest continuous period of employment I ever had.

**Sproull:** So what went wrong?  I mean, what--

**Sutherland:** Nothing went wrong.

<laughter>

**Sutherland:** Sun was sold to Oracle, and at that time I elected not to continue with Oracle or was asked not to continue with Oracle.  I can't tell quite how that was done but-- and my wife, my new wife, decided that she wanted to live in Portland, and so that was okay with me.  Portland's a fine place to be.  It's not quite as warm as the Bay Area, but it has water.

**Sproull:** <laughs>

**Sutherland:** Hey, hey, hey.

<laughter>

**Sutherland:** You could flush your toilet without worrying about it. <laughs> You can water your lawn if it needs it, but it generally doesn't, <laughs> and I like to say in Portland it only rains once a year.  Starts in September and ends in May.

<laughter>

  

**Sutherland:** But the result is that the summers are quite spectacularly beautiful.

**Sproull:** So can you say approximately when it is you think of yourself as having moved to Portland? Because it was more like a wave function change than…

**Sutherland:** Well, I don't know. I moved to Portland. <laughs> I obtained a Portland address. We put an Oregon license plate on the car and all of those kinds of things. We were renovating a house. We renovated a house for a decade and the house is now almost complete.

<laughter>

**Sutherland:** So it's livable, and it's a very nice place.

**Sproull:** So let's talk a little bit more about Portland State, and did you get recruited there or you--

**Sutherland:** No, no.

**Sproull:** You mentioned you walked in and just said, volunteered, to start a new center.

**Sutherland:** Yeah, yeah, yeah, right. Well, I like this notion of being associated with a university, especially if you get the finances right so that they aren't paying you. Right. So we wandered in to the acting dean, who was a man named Dick Knight K-N-I-G-H-T, and Knight had been an executive at Tektronix before going to Portland State and he did various jobs at Portland State. He was the acting dean when I was there. After he found a replacement for the real dean, he went off and became the acting Vice President of Development and after he found somebody to replace him there, he became the acting Dean of Arts and Sciences, with responsibility for the History Department and a bunch of other things, and then he found a replacement there and he's now retired and spends most of his time with this two granddaughters. He's a fine man, and he, he set up in the engineering school a thing that we call the Asynchronous Research Center. It reports to the dean. It's physically located in nice, two nice rooms that the dean provided for us. Halfway between Computer Science and Electrical Engineering, and I have a nominal appointment in the Computer and Electrical Engineering Department and Marly has an appointment in Computer Science Department so we're kind of a bridge between the two organizations, and I, I find the difference in the organizations quite striking. That the folk who know about electrons know less about lists and about data structures and the folk who know a lot about data structures and lists and things know a little less about electrons and for reasons which I don't really understand, communication between those two groups is rare, in any university. Maybe anywhere, I don't know. They just have a different mindset about how the world works.

  

**Sproull:** So, talk a little bit about what you consider some of the highlights of the Portland State research operation.

**Sutherland:** Well, I think the biggest single highlight was in 2015. Marly published a paper in the Asynchronous Conference. And I forget what the paper is titled. But it's Roncken et al. There's half a dozen authors. And basically, it describes self-timed systems in a very simple way. It says there are links, which are communication channels that cover distance. And links provide storage. They store data. And they transmit data. That's their job. They don't compute. And the important part of a link is that a link can be either empty or full. In a synchronous or a clocked system, whether a register is empty or full is known only to the designer. The register itself is no different when it's empty than when it's full. So, you have to know on any one clock pulse whether the data in this register is useful. In a self-timed world, you don't have that luxury. So, each link has with it some additional storage that says whether it's empty or full. And at the input end of a link, you can say, "Here's some data. Accept this data and become full." Now, because a link has physical length, it may take time before it announces, at its output end, that it has data and that it's full. That's the latency of the link. And at the output end, you can accept the data, make use of it in whatever way you want, and then tell the link that I no longer need these data, so you may become empty. We call that draining the link. You tell the link to drain. But you do that at the output end whereupon the link becomes empty. But it takes time before the link will announce, at its input end, that it's now empty and can accept new data. And those delays caused by physical distance are automatically encompassed in the link. Links are connected together with joints. And if you think of this as a graph, the joints become the nodes of the graph. And the links become the edges of the graph. And the job of a joint is to coordinate the fullness and emptiness of links. And a joint acts, under certain circumstances where some or all of its input links are full and some or all of its output links, the ones it wants to use, are empty, and then it acts. And part of the action is to make use of input data, compute on it, deliver that data to output links, declare the input links to be empty and the output links to be full. That's an atomic operation of a joint. Now, there's a wide variety of kinds of joints you can imagine. But thinking of self-timed systems in terms of links and joints makes it all perfectly understandable. And more important than that, it ignores totally the question of what is the interface protocol between joints and links. It's just fullness or emptiness, fill and drain. It doesn't matter how that's encoded in wires electrically. It's all the same. Now, the asynchronous community has been fighting for as long as I've known it, two decades at least, over my protocol's better than yours. And it's well known that my protocol is in fact best, okay.

<laughter>

**Sutherland:** I know that deeply, but other people are misguided. But the important part of Marly's paper is it says, "Look, we all do the same thing. We have links that are full and empty. It's merely a matter of how we encode fullness and how we encode emptiness. Whether we use one wire, two wires, return to zero encoding, non-return, who cares? The only important thing is fullness--" And that up-leveling of the whole discussion makes it possible to mix and match different protocols, makes it possible to do many things. And that has become sort of the fundamental basis of our understanding of what we're doing. And I think it's a very simple and very potent idea. Now, again, when we submitted this paper, the reviewers said, “Well, this is all pretty obvious. Reject it.” Well, okay. Simple ideas all seem obvious in retrospect. But Marly gave a talk at Manchester at Steve Furber’s group who’s been in the asynchronous business as long as anybody--

**Sproull:** That would be Sir Steve Furber.

**Sutherland:** What’s that?

**Sproull:** That would be Sir Steve Furber.

**Sutherland:** No, he’s not Sir.

**Sproull:** He’s not?

**Sutherland:** No, he’s not.

**Sproull:** I thought he was. Oh, sorry.

**Sutherland:** He’s the next grade lower.

**Sproull:** Oh.

**Sutherland:** He’s a Commander of the British Empire.

**Sproull:** I’m sorry.

**Sutherland:** Not a Knight of the British Empire.

**Sproull:** I’m sorry.

**Sutherland:** Never mind. But, so as Furber said, “So, why did it take twenty years to figure this out?” Which is a very good question. Why did it take twenty years? I mean I wish I’d thought this through before micropipelines so we could have published it in 1988. It would have set the field off in a good direction. But it took the difference in point of view of Marly, a mathematician, and me an engineering type interacting for that idea to appear in Marly's mind. And I think it's a very important simplification of the whole field. Now, we reasoned with the organization committee. And they allowed the paper to be published. And remarkably enough, although it was not on the voting-- was not on the ballot for best paper award, it received enough write-in votes that it was given a certification, not as best paper, but as one of the best papers in the conference.

**Sproull:** Well, so the computer science community has a tradition of sorts now starting for years--

**Sutherland:** Tradition of?

**Sproull:** Years later, conferences sometimes give awards for "Test of Time" papers. So, maybe in ten or fifteen years you can hope that this paper will be given a Test of Time award. They're actually highly sought after awards. And they often have the property you're describing, which is they weren't necessarily "a new idea." They were a simplification and elegant design.

**Sutherland:** Well, I think the notion that fullness and emptiness was the only important thing was, in fact, a fresh idea.

**Sproull:** Okay.

**Sutherland:** And it's a potent one.

**Sproull:** So, I would--- unless you would-- there are other things about your life you'd like to talk about, my plan now is to switch and talk more about some topics, things that kind of went throughout your career. And maybe what we'll do is just launch right in and then at the end look for loose ends. Did you have any-- either of you have any questions you wanted-- okay, so, one of the things that you don't do so much anymore, but you used to do a lot of is service on government committees of various kinds.

**Sutherland:** No, I'm sorry I didn't hear.

**Sproull:** Service on government committees. You were on DSB for a while. You were on several National Academy's committees. And you co-chaired a somewhat famous one, the Brooks Sutherland report. Were you on the Naval Research Advisory Committee?

**Sutherland:** I was asked to join the Navy Research Advisory Committee some long time ago. I forget when it was. I had been cleared for classified information when I was a military person at NSA. And so, I had a clearance. And I knew Bob Frosch, who had become the Secretary of the Navy. And they asked me to join the Navy Research Advisory Committee, which I did. I was the youngest member of the Navy Research Advisory Committee. And the oldest member was a man named Gus Kinzel. And Gus Kinzel was probably into his-- in his seventies. And I was in my thirties. And Kinzel put his arm around me and said, “I will show you the ropes,” which he did. And we developed a very nice friendship. He had been at Union Carbide for many years. And he was an interesting character. So, the Navy Research Advisory Committee did a number of things. We had a visit on an aircraft carrier in the Mediterranean on one occasion. And one of the things you can do out at sea that you cannot do on land effectively is to have a supersonic airshow. So, we all went on deck, and they said, “Now, if you look towards the bow of the ship you will see one of our fighter aircraft coming supersonically towards the carrier.” And it flew over the carrier at, I don’t know, five hundred feet or something, supersonic. And it went bang bang. Okay, now Ed Heinemann was an aircraft designer of some note who was on the committee. And I asked him why two bangs. And he said, “Well, the first one’s the wing. And the second one is the tail.” They each make a shock wave, and they’re separate shockwaves. And of course, you didn’t hear anything while it was approaching. And then you had bang bang brawrrrrrr off it went off in the distance. It was quite fascinating. It was on the Kennedy. The Kennedy was conventionally powered. It was an enterprise class carrier, but it was conventionally powered. There was a wonderful painting in the engine room. Apparently, one of the folk who worked in the engine room was something of an artist. And they made a wonderful empty wall there. And he had a wonderful painting on it, which is great. It was a marvelous experience. I got involved with the Navy Security Group. And there was a man named-- the name has faded. Villard, Mike Villard. His real name was Oswald G. Villard. Now, with a name like Oswald G., what would you call yourself? He called himself Mike. And Mike and I traveled around the world visiting Navy Security Group installations. And it was quite a fascinating tour. We went to a place called Skaggs Island, which at that time was a listening post at the north end of San Francisco Bay. There’s a swamp at the north end of San Francisco Bay, a salt swamp, which has Skaggs Island in it, and makes an absolutely beautiful ground plane for antennas, salt water, flat, smooth, no waves, perfect ground plane. And they had an antenna farm there that was just par excellence. They had rhombic antennas facing in every direction. They had long-wire antennas of various sorts. They had a Wullenweber directional antenna, all this stuff there. And it was all fenced in. And you couldn’t go in the buildings because it was all classified stuff there. But they told me that on field day for the amateur radio operators they opened the gates and let the amateur radio operators on field day set up a camping. The rules are you have to be away from commercial power. So, you have to provide your own generator and so on. And then you try to communicate all over the world. The guy who was able to communicate the best gets the prize or whatever. So-- and Bert and I, as high school students, used to participate in this. So, they told me that on field day, they would pass antenna leads out the window from Skaggs Island and let the amateurs use these beautiful rhombics with great ground plane and everything for field day. I mean what a place to be. I wish I’d been able to participate. But these were very interesting tours. The remarkable thing-- Villard was an artist at going to a place and discovering the pocket of excellence. And everywhere we went, we found equipment that was not being used because nobody knew how to use it. There were problems everywhere. There are always problems in any big organization. But Villard always found the pocket of excellence. He found the few people who knew exactly what they were doing who knew exactly how to use the equipment and made perfect use of it. And these were the people who made the whole system work. And Villard always found them, and he always patted them on the back. It was quite remarkable. We went various places, and we met individuals who just knew their jobs thoroughly. And it turns out in every big organization there are such people. And they're what makes the organization work. And if management can identify them and reward them, what a wonderful thing to do. Villard was an artist at that. I remember when we were on the carrier, Villard said, "Now, I will bet that there are non-linear, electrical things around this carrier caused by corrosion or whatever." And he said, "And I'd like to identify them." He said, "So, what frequencies are you transmitting on?" And he wrote those down. And then he took the sum and difference frequencies because a non-linear thing will inter-modulate the frequencies. And if you listen at the frequencies that are sums and differences of the frequencies that are being used, you should be able to hear the result of these non-linear things interacting. And sure enough, there they were in the middle of the Mediterranean ocean where there's nothing else around to create those frequencies. It was quite remarkable.

**Sproull:** So, let me turn to another committee thing. I mentioned it, but I want to come back to it. You were co-chair with Fred Brooks of the committee to review the high-performance computing program. I don't exactly remember when this was. But this was the report that became the origin of the fabled tire tracks diagram--

**Sutherland:** Yes.

**Sproull:** --that showed how interweaving of government and industrial research and other activities would ultimately lead to substantial economic activity. But it took fifteen years or so for research ideas to make it to the marketplace. How did that become the focus of this review?

**Sutherland:** I was at a meeting in Washington. And the executive secretary of the National Academy Committee on Computing, whatever it's called. You were chair of it.

**Sproull:** No, no, no, not at the time. This must have been-- it will come to me.

**Sutherland:** And what was her name?

**Sproull:** Marjory Blumenthal.

**Sutherland:** Marjory Blumenthal. Marjory Blumenthal said to me, "We've been asked to set up a committee to review this--

**Sproull:** HPCC.

  

**Sutherland:** Well, it was-- no, it was pre HPC-- Yeah, so the computing-- the computing activities, which you chaired. And of course, I gave my impulsive no. I said, “Of course not. Don’t be ridiculous.” And on the way home in the airplane, I realized that this was an important thing, that this would be a guiding principle for the next decade of what we do in computing. I figured this was pretty important. So, I called Fred Brooks. I said, “Fred, I’ve been asked to chair this committee. I’m not willing to do it. But if you’ll do it, I’ll do it.” And he said, “Well, I’m not willing to do it. But if you’ll do it, I’ll do it.” And so, we agreed that together we would chair this committee, co-chairs. And so, it was. And we had called a bunch of people that we thought would be useful on this committee. And every one of them said yes. It was quite remarkable that they all said yes. There was only one person who couldn’t do it because he was going to be out of the country on sabbatical or something. And the members of the committee, I don’t remember who they were. Butler Lampson was on it. It’s a matter of record. I actually have the committee report if we really need to know. But there were-- it was a remarkable group. And we decided that we needed to show how industrial research, and government sponsored research, and academic research, the interplay between them, how that had an impact on the field. And essentially, the collective memory of the committee, Butler recorded. He recorded the collective memory. How did this happen? How did that happen? Who went from here to there? Who went from Berkeley to PARC? Who went from PARC to somewhere else and so on? And we recorded that as a chart. And that became the racing stripe chart. And it shows, over a period of time, the growth of I think eleven billion dollar industries from first seed of idea to multi-billion-dollar industries. And it’s a remarkable interplay that there’s some of it is done by government sponsorship, and then some of it is commercial. Some of it is academic. And there’s a flow between them. Now, I think the marvelous part about the U.S. research establishment is that it has this flow of people from one place to another. When Fred and I were briefing this report was shortly after the implosion of Thinking Machines. Thinking Machines Company was set up-- an MIT spinoff that built big machines. And it had gone broke. It had a lot of DARPA sponsorship, and then it went broke and disappeared. And somebody in the audience asked the question, “Isn’t this a giant waste of government money?” And I said, “Well, as far as I know, the knowledge that was gained in that research program was largely in the minds of the people. And as far as I know, not one of them has left the United States.” They’ve all gone to different employment in other places. At Sun Microsystems, we hired several of them. And we got our fair share. And other companies got their fair share. And this is technology transfer at its very best. You take the people who know the stuff, and you move them to some other organization. It’s a very high-bandwidth technology transfer. I said it’s too bad for the shareholders. I mean the shareholders of the company lose money. But in terms of the national welfare, it’s research money invested well and now transferred to where it can do some good. And I thought that that’s, in some sense, a unique property of the way government research and private industry research and so on interplay in the United States. And it’s the strength of the research establishment. We’re sitting here in Silicon Valley, which is traditional for doing this kind of thing.

**Sproull:** So, this is a great segue to the next topic, which is your various roles over the years in venture capital. And we’ve talked a little bit about that before. Last time, we talked about your meeting the Venrock people and Teddy when Evans and Sutherland was founded. And we talked a little bit about you and I doing due diligence for ATV and so on. But, I think I’d now like to approach it more from the-- your thoughts about venture capital, the various roles you played, ranging from an advisor to a limited partner with ATV, to an angel investor, to running one of the portfolio companies for a while. And I seem to recall at Sun, once a week, somebody would call you up or darken your door step expecting advice and probably money as being--you were the engineer who knew enough about venture world to be a soft touch, or an easy introduction, or somehow approachable. So, the various ways in which you were part of the venture community seem to me quite fascinating and broad. You've already told us, by the way, about the three-legged stool.

**Sutherland:** The what?

**Sproull:** The three-legged stool. You don't have to do that again.

**Waldo:** Or get to do that again.

**Sutherland:** Well, we've got Jim Waldo, who wasn't here last time.

<laughter>

**Sutherland:** And Bert. Okay, well Teddy Walkowicz was a remarkable guy. He was a PhD from MIT in aeronautics. And his job during World War II was to be the-- he was in the Air Force I think. He was in the Army, the Army Air Corps. And his job was to be the bag carrier for von Karman. And so, he was thoroughly trained in the research establishment and how research is done and so on. When he left the military service, he went to work for Laurance Rockefeller and spent the next twenty years, roughly, working with the Rockefeller people becoming eventually a partner in what later became Venrock. And he had been on the board of the Evans and Sutherland Company. So, he and I had gotten to know each other pretty well. And he decided to spin off from the Rockefellers and form his own venture capital fund, which he called Advanced Technology Ventures. And he asked me to be a founding partner, which I was. I helped him found it. And we started this thing. And it seemed easy at first. I mean this is in a period where almost any investment that you made would made good because the economics of the country was interested in that. And people were investing like crazy in new start companies. It was very much a bull market. And so, a number of things that we did turned out to be quite successful early on. And then the-- I don't know what to call it, a crash, or a setback was in-- what was it? When was it? 2000--

**Sproull:** Well, the dot com bubble was 2000, roughly.

**Sutherland:** The dot com bubble burst. And it became very hard. There were no public-- no initial public offerings for four or five years. It just became very hard. Teddy died leaving a huge gap in the venture capital partnership. And by then, Bob and Bert had been participating. And so-- and we did the due diligence. We thumped the tire, so to speak, of deals to see whether the people were sound and whether the ideas were sound. I managed to keep ATV out of a number of integrated circuit deals involving-- I've forgotten the name of the technology now. Help me out. Not silicon, but we used--

**Sproull:** Oh, well like--

**Sutherland:** We used gallium arsenide. Gallium arsenide integrated circuits had promise because they could be faster than silicon circuits. But there was an important defect in them. It's not widely known, but silicon is an important integrated circuit material in part because if you oxidize it, it makes glass. And glass is a fantastically good insulator and fantastically dimensionally stable and all that good stuff. And neither gallium nor arsenic have that property, which makes making gallium arsenide integrated circuits quite hard. And we stayed out of a bunch of gallium arsenide potential deals, I think wisely. Now, I don't think I'm very good as a venture capitalist actually. I see the promise of a technology much more clearly than the difficulties. The difficulties involve: will the products that it make be useful. And I can imagine all kinds of uses that simply aren't. And so, I think that there's a role in venture capital for dreamers like me. But it's not the central role. The central role is how do you make wise investments that, in fact, will return money, which is what the venture business is all about. And fortunately, in ATV there were partners who were better at that than me. And ATV managed to be a moderately successful venture firm. It wasn't wildly successful, but it was moderately successful. There's still some pieces of it running at the end of a long period.

**Sproull:** So, entrepreneurs would come to us and pitch investments. And a part of that of course is a business plan. And I seem to recall you had, perhaps not a unique perspective, but a sharp perspective about what the role of a business plan was.

**Sutherland:** Well, nothing ever goes according to plan. But if you don't have a plan, you can't go not according to the plan. And so, a business plan is kind of the outline of what you think you want to do. And then it has to be modulated as you get into it and figure out what you really want to do. And I think the people who are better at modulating the plan to do something that's useful are better or more successful, in general, than people who see a plan and then want to stick to it. Well, I don't have any great wisdom there. The important thing is, for people who want to start a business, it's not hard to start a business. It's much harder to make a successful business. And I think an important thing about venture capital as practiced in the United States today is that failure at one business is not necessarily a lifetime black mark, because people who fail at doing a couple things have learned important lessons. And that makes them more valuable as entrepreneurs the next time around. People who have been very successful may not be such a good bet on the second time around because A, they have too much money, so they're not hungry. And B, the fact they were successful at A does not necessarily make them successful at B. And so, there's some concern about how you balance this whole thing. I think engineering and science largely progress by failure, that if everything works, you didn't learn anything. If not everything works, then you learn what didn't work. And that's an important lesson, maybe more important than learning that well, we did these fourteen things and together it all worked. I wonder which one was critical.

  

**Sproull:** So, you ran Austek for a while in Adelaide.

**Sutherland:** Yes. Well, we had this program at Caltech called the Silicon Structures Project. And the development office at Caltech-- that's-- development in a university is a code word for sales. Development are the people who go and get money, right? So, we had this development office at Caltech that said, "Well, we could set up this deal with industry." And it was a well-crafted deal. We'll get a number of industrial partners to provide money to Caltech to make this project go. That will be the support for it. And then we'll also get them to provide people who will come for at least a year. People who are mutually acceptable to the company and to Caltech will come for at least a year and be a participant in the program. And the deal for inventions was, while they were on the Caltech campus, anything that they invented would belong to Caltech. But the participants would have a royalty free forever license to do whatever the hell they wanted with any of these inventions. And so, this was a way that a number of companies could send individuals who then could participate freely and have conversations. And it actually worked very well. We had participation from IBM, from Xerox, from Burroughs, from Hewlett-Packard. And there were two more. I forget what they are, from DEC, Digital Equipment. There's one more, but I'm-- and I may remember it.

**Sproull:** Yeah.

**Sutherland:** But these representatives came. Now, I had gone to Caltech to build a computer science activity. That's what they hired me to do. And then I found that I couldn't hire any faculty. I tried to hire faculty. And the people that I wanted to hire were pretty good. One of them was Al Perlis. I tried to get Al. I got Al Perlis for a year to Caltech on a Fairchild fellowship. But I tried to get him hired on the faculty as a tenured faculty member. Now, I can't imagine that, if you have the opportunity to hire Al Perlis, you don't just say, "Yes, grab him." I mean this guy's fantastic. But the engineering faculty at Caltech said, "But he doesn't have any publications." And I was not a wise enough and astute enough politician to say, "Would you believe seventy PhDs? Is that not enough publication record for you?" But I didn't realize that that's what I could have said. It's only later I realized that. I didn't know what to do. And so, we didn't make an offer to Al Perlis. What a mistake. I mean insane.

**Sproull:** So, I was trying to get you to talk about Austek.

**Sutherland:** Okay.

**Sproull:** You'll get there. Okay.

**Sutherland:** So, but what I did have was these industrial visitors. That was my faculty. And one of the industrial visitors was a man named Craig Mudge. Craig Mudge had been a student of Fred Brooks, wrote a PhD for Fred Brooks. He was an Australian. He lived in Adelaide. And he is-- he started a company there called Austek which was an Australian high technology fabless semiconductor company. And I helped him raise money for it, made a small investment myself and helped him raise money. ATV put some money into it. And a number of people got involved. We had a board of directors that included some Australians and some other notable folk. And it was moderately successful for a while. It provided cache memory controllers for the personal computer market. And it missed a turn of integrated circuits. We went from one technology to another. And Austek made a mistake about the models that they used for the design. They failed to notice that the polysilicon resistance had increased out of proportion to other things, or something like that, which caused them to miss one product cycle. And the company eventually died. But I was on the board of directors of Austek. And as a result, I went to Australia four times a year for almost a decade. So, I've been to Australia many, many, many, many times. I was propagandized as a youth that Australia was a pretty good place because my parents had been married there. And they thought Australia was a pretty nice place. And I'd had a couple of extended vacations there. One of the reasons I went there for extended vacations was, in fact, that Mudge was there and was hospitable. And so, I had some contacts. And so, while I was a Austek, one of the things we tried was a micropipeline, actually built some micropipelines at Austek. And they worked just fine. And people understood how they worked. But it was not something that could go into the commercial products.

**Sproull:** Okay, so I want to change the topic now and talk a little bit about a topic I have labeled "work habits." It seems to me that over the time I have known you, there are certain invariants in the way you work. And I want to talk a little bit about all of them. Maybe I should give you a brief outline. So, one is the daily dose of technology. We'll get to talk about that. Another is writing things down. We'll get to talk about that. And another, perhaps, is starting and finishing things. So, let's talk first about the daily dose. You've been quoted as saying, "When denied my minimum daily dose of technology, I get grouchy. Without the fun, none of this would go on." And it strikes me that you've been doing this certainly ever since ARPA. I thought your NSA, hiding out at NSA, was a minimum daily dose technique even while you were running IPTO at ARPA. And ever since, this has been an invariant. So, talk a little bit about why you need your daily dose.

**Sutherland:** Bob, I don't know why. I don't know. I haven't any idea why. I think it's Bert who first coined the phrase, "Ivan needs his minimum dose of daily-- daily dose of technology." But it is certainly true. And one of the pleasures of my life now is I have contact at the Asynchronous Research Center at Portland State. And I go there every day. And I do things. And sometimes, I get up early in morning, and I do something technical on the projects that we're working on. And that somehow makes me happy. I think there's something in me that's like the thing that's in artists that artists have to create something that they think is pretty. And I think that good engineering is pretty in a sense similar to the fact that art is pretty. And you see this in the design of bridges, for example. I mean a good bridge is not only structurally sound, namely it stands up, but it also is an asset to the community that--the big discussion when the part of the bay bridge had to be replaced was what should it look like. It is, after all, going to be an icon of San Francisco. It can't look like some dirty old thing that's just there and is an eyesore. It has to look nice as well as being functional. And I think of the things that I have done as an engineering sense as having a beauty to them. A simplicity with which you get the function to do, the fact that you could see in the form what the function is, the fact that it all works has got a beauty in it that brings me a great deal of pleasure.

And so, there's pleasure in doing those things. And the pleasure comes, I think, from doing them in a way that is aesthetically appealing. Now, one man's aesthetics is another man's belly laugh, right? I mean there are people who think that different kinds of art that I think are just nonsense are beautiful. But I think the minimum daily dose comes from the need to make something that's satisfying.

**Sproull:** So, I notice that even vacation has its minimum daily dose. I mean things like-- you always-- when you talk about the FIFO designs, you always say it was while lying on the beach in Australia. They're inseparable.

**Sutherland:** Yeah, that's right, isn't it?

**Sproull:** And your sabbatical in London, you didn't talk about the research project there. That was when you figured out logical effort.

**Sutherland:** That's right.

**Sproull:** And I don't think it was because you got up every morning to join the research group. But it--

**Sutherland:** I understand. And it's right. It is right. A vacation is a way to change the scene and have time to think about things that you wouldn't think about in the daily-- the urgent always pushes out the important. And if you're in your own office, if I'm in my own office, the phone rings and people come in to want to see me and so on. And the time gets frittered away. But when you're away from-- when I'm away from the office, I get to think for extended periods about stuff that is just stuff. And sometimes, some ideas come that are interesting enough to pursue. Sun Laboratories used to say, "Now, it's important to take your vacation because to the company, untaken vacation is a liability," which adds up on the liability side of their balance sheet. And my response was always, "I'd be happy to record my vacation if I knew when. I can't tell whether I'm on a vacation or not." And my favorite example is I was riding my motorcycle across Nevada somewhere. And I had this really good idea. And I stopped at the next public telephone. I didn't have a cellphone in those days, stopped at the next public telephone and called Bob Sproull and told him what the idea was in case I got killed on my motorcycle in the next segment of the trip, right? So, now was I on vacation or not? I can't tell. And the point is I think that in the idea business, ideas happen when they happen. And they tend to happen when you're not as focused on particular things as you would be when you're not on vacation. And so, it's quite appropriate for people who are in the idea business to be paid by the year rather than by the hour because you can't tell when you're on the clock and when you're not. And I think that's the distinction between professional employees and hourly employees is that professional employees are supposed to be partly, at least in the idea business--

**Sproull:** Okay, I'd like to move on now to writing things down. And again, you've been quoted as saying "it's not an idea until you write it down." And I certainly remember that one of the most striking features joining you when I was an undergraduate was the “display file,” as you called it with pun intended, in which all of the ideas, if you will, of group were written down. And you were the leader in doing that. And I actually, I happened to be, after our last discussion, looking through Ed Berkeley’s papers which are saved by Minnesota, is it? And you can go look at them online. And he has an index of the papers. And in there is a paper written by Bert and Ivan Sutherland which he collected and added to his collection. It was about one of your early robot-like gadgets. But so, you’ve been writing things down for a long time. So, talk a little bit about what got you started in that direction and sort of what’s-- I know you ultimately, by the time it came to Sun, there were even some principles about how to do this. And you certainly expect everybody in your group to do it. And you do it-- you don’t do it by lashing them. You do it by setting an example and every-- it follows. It’s really quite a compelling habit within the best possible sense of the word habit.

**Sutherland:** Hmm. I have no idea how it started, Bob, no idea. It’s a major asset to a person with a weak memory <laughs>. So, no the paper that Berkeley probably collected, we wrote-- it was actually an article we wrote. It was published somewhere, I think in Computers and Automation. Right, he published it. That’s right. He published it-- by Mugglin, I think was his-- Mugglin, right? Bert and Ivan and a man named Mugglin was Bert’s buddy. I don’t know where it happened. But what I find now is I can tell you that there are many, many, many times when I’ve sat down to write down an idea, and as I write it down, it turns into garbage. And that’s when I realize that the idea is no good is I cannot write it down. And the articulating the idea by writing it down makes it possible to look at it and question it. If you don’t write it down, and it remains part of the folklore like fishing stories, it may grow in length with the telling. It may grow in value with the retelling. But if you write it down, you could find the defects. Until the computer program is written down, it has no defects whatsoever. An unrecorded computer program is flawless. I know any number of people who are the non-author of the perfect novel. Okay, I have the perfect novel. It’s in my head here. I haven’t written it down and submitted it to a publisher yet, but it’s there. It’s perfect, right? Of course, it’s perfect. It’s still in your head. And until you write it down, the flaws don’t appear. And I don’t know when I realized that. I mean the Harvard display file was a communication means that was intended to allow the group to function on the same set of ideas, the same set of knowledge, that as we gained pieces of knowledge, we wrote it down so that the students could see it. The rules for the display file-- this is long before computer documents were known. The documents were written on pieces of paper. And an original document was stuck in the display file along with many copies. And you were allowed to take any copy you wanted. Now, how did you know if it was a copy? We had an embossing machine that would emboss the original with the word original in a way that did not copy on the copy machine. So, the original actually had raised print embossed in the paper that said original. So, you could copy it as many times as you wanted. They never copied the original, which was a good device. I wonder what happened to that embossing machine. <laughs> And that was the display file and it worked. Now let's see, at the University of Utah we did that too but I don't recall what they were called and at the Evans and Sutherland Company we did it but I don't remember what we called the files. But today I do it as a way to get ideas into a place that they can be shot at and often they are and often they are not right and that's good, you write something down, you can find its flaws.

**Sproull:** Well and in fact, one of the principles…

**Sutherland:** And I think, you know, I spell very badly, that is to say I spell very creatively. I have friends like Bob Sproull who look at a word and know whether it's…

**Sproull:** Used to.

**Sutherland:** …spelled according to the dictionary or not, I don't know how the hell they do that, it's perfectly amazing to me, I'm astounded that there are minds that work that way, I spell phonetically which causes me no end of trouble, under what letter do I look up cucumber, if I want to find the spelling of cucumber, okay, I need to know what its first letter is, I have no idea, so I sit here and talk to you, I have no idea what the first letter of cucumber, it's either Q or C but I'll be damned if know which.

<laughter>

**Sutherland:** So writing things down also exposes my poor spelling. Fortunately we now have spelling checkers which at least insists that the words be in the dictionary if not the correct word.

**Sproull:** So did you ever use-- so the competing technology here is lab notebooks which the fiction was that all scientists and engineers kept lab notebooks and they were initialed and signed every day, et cetera, et cetera, et cetera, did you ever keep lab notebooks?

**Sutherland:** No, not really. I think we had lab notebooks at the Evans and Sutherland Company but I never took it very seriously. The documents that were written by computer were so much easier to write. I mean, I don't know. We have a photograph in our laboratory at Portland State, we have a photograph of Grace Hopper's lab notebook in which she taped in the bug, a moth had gotten into one of the relays and so she captured this moth and taped into her lab notebook and that was the original bug found in a computer and we have a picture of that on the wall, it's a nice touch. It's in the Grace Hopper room, we have two rooms in our laboratory in the ARC one is the Alan Turing room in which we have a number of posters about Alan Turing and then the other one is the Grace Hopper room in which we have some photographs of Grace and some memorabilia from her including a delightful picture of a gray haired old lady in a Navy uniform with broad stripe on the sleeve. She was the oldest active duty officer in the U.S. Navy, by act of Congress, I think the Navy kept wanting to retire her. Congress thought she was a great role model and kept her on.

**Sproull:** So another writing aspect is you put together a set of notes for Applied Math 252R at Harvard which was sort of the beginnings of a textbook and didn't ever really carry it beyond that but William Newman and I did. When I went off to NIH after a while, those notes became an outline for sure, I don't think we plagiarized anything but it was definitely the inspiration for courses I taught at NIH and for the book that William and I wrote. So much so that in fact William later thought we should have listed you as a co-author.

  

**Sutherland:** Well I treasure a bound copy of that book in which I am listed as co-author which was given to me by its real authors some years later, I'm very appreciative of that. But, you know, Feynman is the author of six books, none of which he wrote and I think that's exactly the right way to be, writing a book is hard work, requires finishing, okay, it's easy to start something, it's much harder to finish at least for me, I'm a starter of things, not a finisher of things. Sproull and Newman decided to-- Newman and Sproull I think it's actually called decided to write this book on gravity, more power to them, I think it's wonderful that they did.

**Sproull:** So, but while we're on authorship, another aspect of writing that I think you have views on and certainly practice around is what it means to be an author of something and who the author should be on a paper.

**Sutherland:** Oh. This is a complicated business, I think there's a tradition in academia that the senior person is the first author. You know, the Millikan oil drop experiment is famous, it was an important scientific experiment to measure that charge is actually quantized, it was a wonderful experiment, it's probably done by a couple of graduate students but it's known as the Millikan oil drop experiment. And Millikan apparently was widely known for demanding credit for things that junior people did. There was a sign appeared on the Caltech campus that said, "Jesus saves," and underneath somebody wrote, "But Millikan takes the credit."

<laughter>

**Sutherland:** And somehow that doesn't seem quite right to me. I had the good fortune to have Claude Shannon as a thesis supervisor and so when my thesis was published, it was published as Ivan Sutherland, Shannon's name doesn't appear except in the acknowledgement, he didn't want any credit for work that I had done, he was a very modest man, he had more piles of credit, if he'd added one more thesis to that pile of credit, you wouldn't even notice. And so that was a place where I appreciated the value of being the sole author and I've tried to carry that forward with the students that, and fortunately I have plenty of credits so I don't need one more paper added to the list of publications that I have. I have, did you know this, that I have six, in terms of patents, right, I have six dozen, six dozen patents, right, which is more than enough credit. Patents are a wonderful way, wonderful form of publication, they're the only form that's stamped by the U.S. Government, “original,” stamped by the U.S. Government, original, right, as a fine form of publication and it looks nice on a resume. Students often ask me, "Should I apply for a patent for this?" and I said, "Why, do you want to protect your research so other people won't use it? I don't think so, but it looks terrific on a resume if you can get a patent, so more power to you." Why do I have six dozen patents, because IBM came to Sun and said to Scott McNealy, "We have this many patents and you have this many patents, we think you should be paying us some fee for not suing you." And so that tax was charged to Sun for years and years and years and McNealy hated it and so he had a very aggressive patent program to increase the number of Sun patents. By the way when Sun purchased Sutherland, Sproull and Associates, they tripled their patent portfolio in that acquisition. We had four, they had two, the total is six which was three times what they had had before the acquisition. Now, since then, they've done better at the patent business.

**Sproull:** Well, so just a closing footnote to this, back to Shannon and authoring your thesis, do you recall that you listed me as the first author of the clipping divider paper and this struck me as an extraordinary move that made a huge impression and as far as sort of carrying on the tradition, certainly stamped it on my memory, so.

**Sutherland:** The original clipping divider paper was Sproull and Sutherland?

**Sproull:** I think so.

**Sutherland:** I wouldn't be surprised, I wouldn't be surprised.

**Sproull:** Not that I would claim to have been the inventor or the guy who did the most work or any of that but I was the first author.

**Sutherland:** Well, you know, the walking machine paper in Scientific American is Raibert and Sutherland and I actually wrote the paper at least I wrote the first draft of it and of course the important thing in it is Raibert's hopping machine but I insisted that it be Raibert and Sutherland. He came to me and said it should be Sutherland and Raibert, I said, "No, Marc, I already have tenure," which I think is the right argument as to why it should be the other way around. But, you know, there's a very funny thing because credit is important and senior people need credit also. Many executives fail to make an adequate plan for replacing them should they get run over by a truck or something, a succession plan. Succession plans are hard to put in place and why is that, because as you put a succession plan in place, you wonder, "What use am I?" And I felt this keenly, I went off to Berkeley, left Sun Labs and went to Berkeley, I was there and I had a wonderful time at Berkeley, it was the right thing to do. Sun paid my salary so I had the run of the university with an industrial salary, how can you do better than that, right? But on the other hand, I began to question the value of my own contributions, "What is it that I do, what is my contribution to the company?" And I think succession plans are hard because they bring into question the person who's setting up the succession plan, what value is he, if he now has these successors who are as good as he is, why shouldn't he just retire now, what is his continuing contributive-- why is he important to the company? And the balance between being important to the company and contributing and not standing in the way of the next generation, it's a very difficult balance to strike and to strike it from the inside, for individuals to strike it for themselves is almost impossible, in some sense the board of directors has to put its arm around the executive and say, "Mm, I think it's time that you let the younger generation take over." I hope some people will tell me when it's time, I mean I'm at the ARC and I have partners there who are wonderful partners and the question is when will I no longer be the contributive person, so it's a hard question. Perhaps the best way to solve that problem is to have a heart attack.

**Sproull:** Yes, well we've seen that.

**Sutherland:** Then the problem gets solved for you, right?

**Waldo:** Or it becomes someone else's problem at that point.

**Sutherland:** I'm sorry?

**Waldo:** It becomes someone else's problem at that point.

**Sutherland:** It becomes moot, right, the whole thing becomes moot.

**Sproull:** So Ivan you mentioned earlier, and this too is an older topic about, you think you're a better starter than finisher and I don't know if there's anything more you'd like to say about that. I will tell you that the most graphic perspective that I had on this was while I was at Carnegie Mellon as an assistant professor and we were doing, you and I and Ed Frank were doing some micropipeline designs and you were the layout guy working with Magic which was a graphical layout thing and you could see every day the design which sort of climb farther up the screen, the rows of the micropipeline were horizontal and the summit would keep going up. And Ed and I were starting at the bottom trying to make sure it was right in doing design rule checking and simulating the circuits and trying to put in some testing stuff and and so on and we were proceeding at, I don't know, a tenth the rate that you were. So that was for me, the most graphic example of starting and finishing, you laid the ground work, not only with the golden shovel but all of the rules and sizes and exactly what we were doing but it was a struggle to get the finishing done on that. But that's not always been, I think you do yourself a disservice by saying that you're not a good finisher, you certainly inspire others to finish or to help you, you set up the structures that let things be finished, chips done in your research projects get finished, they don't get abandoned. So talk a little bit about starting and finishing.

**Sutherland:** Well, Bob, you don't see the ones that are abandoned.

<laughter>

**Sproull:** Well they get abandoned early maybe is the…

**Sutherland:** I don't know. Bob Sproull once said to me about me, something which maybe is true, he says, "You have an art for picking the right things to do." And it seems to me that picking a project to work on, one should pick something that you can see through, that you can see to the end. Now I have great admiration for people like Jonas Salk, who said, "I am going to make a polio vaccine." Well it took him 20 years and maybe at the start he could see that it was perfectly doable and so he said, "I'm going to do this," right? But I don't have the courage to start something that looks to me like it's 20 years with no end. On the other hand, picking something that you can sort of see it can be done and you can see that it's perfectly plausible to do and maybe one that other people haven't noticed yet, and I like to think it should be easy, that is at least it should look easy at the start. Now I've been involved in this asynchronous stuff, self-timed stuff for a long time now and you might ask why. And I think the reason rests in physics, you see the synchronous design style has as its fundamental assumption that a large number of things will happen simultaneously over space and Mr. Einstein explains carefully that time and space are intimately related and the integrated circuit industry has now gotten to a place where it's noticed that, it's noticed that a major chip is so large that communication across the chip takes more than a clock period and so simultaneity over this chip is very hard to maintain. It used to be that the village bell, the church bell in the village was how you told time, at noon you got 12 strokes of the clock and they traveled at the speed of sound throughout the village so time was later at the fringes of the village than it was in the middle but nobody cared. But when trains started going from New York to Chicago and the fastest communication between them was by horseback, it wasn't possible to run a railroad so the telegraph had been invented thank heavens and so New York could know what time it was in Chicago and vice versa, at least to the second and so you could run train schedules. Now on the integrated circuit you're limited by some speed limit that's related to the speed of light somehow or something and simultaneity you can't have, you just can't go that fast. So there's something fundamentally wrong with the synchronous design style and I think self-timing is the way to fix that. I talked earlier about a link is a thing that stores data and transmits it and tells you when the data has arrived and that allows you to coordinate the actions of things that are spaced apart in space in a way that doesn't require imagining that they're simultaneous. Now unfortunately sort of all of the training and all of the thinking that computer people do is based on this notion of a clock, that there are ticks of the clock and on each tick of the clock, the entire state of the machine changes from state one to state two and you can tell in spite of the fact that the machine is so big that that can't happen. And the folk on the computer science side of the hall at Portland State and other places are struggling with how to do parallel computing, how to coordinate the actions of multiple machines on the same task where the timing of the different machines is not understood. And the folk on the electrical side of the hall are struggling with the questions of how do we do clocking over regions that are bigger than you can communicate a clock pulse. And these are the same problem and they involve escaping from the clocked paradigm and that's what we're trying to do in some fundamental way that's an important thing to do and we're learning important lessons about how to do it.

**Sproull:** You've talked about various topics that are going to come up again, that's fine, but I want to collect them. And I've labeled this section as your activities, as an engineer and designer. Because it strikes me all of your projects are about building things, you don't prove theorems, you sometimes write software, we could argue about whether that's building things, let's call it building things.

**Sutherland:** Of course it's building something. Of course it's building something.

**Sproull:** All right, all right. Building companies, building research groups, building departments and so on. So there's actually three themes to the technical part of this that I want to bring together and explore and I've called them geometry, symmetry and structure. So let's start with geometry. Now these are topics we've talked about before in one way or the other. So I think if people watch you work, maybe I should just say, I notice that geometry is a huge theme that runs through things you've done, you have uncanny knacks for it I think in ways that most people don't, obviously it was a piece of Sketchpad, you've talked a little bit about formulating some of the constraints there in a symmetric way and so on. At Caltech, I think you started and then a bunch of students took over, a polygon package which was a piece of sort of advanced geometry, at E&S you did the polygon clipper with Gary Hodgman, so geometry's been around. And moreover as a chip designer, I don't know how commercial people do it, I've never seen them but watching you work is unique, it's fascinating. You start a project by figuring out sort of what a geometric modulus is going to be, so many wires, so this kind of wire is going to horizontally, this kind of wire is going to run vertically, they're kind of going to be spaced about like this and the module sizes are going to be quantized in the following way and so on. And you start out and maybe some revisions to this happen but by and large you just go and maybe it's different than last time and you learned from last time or something but this is an approach to the geometry of laying out chips that I don't think most people A, have or B, can do. Certainly I can't, I know when I sat down and tried to just design some little circuits, they looked like a mess, there was no ability to think about it in a kind of a general way geometrically. So I want to start you talking a little bit about…

**Sutherland:** All right. All right, Sproull, I'm going to turn this back on you, see. My jaw drops open when you say, "Well this is how this word is spelled," right, this is something which I have no ability to do whatsoever, right? And you don't think of it as a big deal, right, it's what you do, right? Now you just said I have this skill about geometry, I don't notice it as a skill.

**Sproull:** Okay.

**Sutherland:** Okay? It's just what I do.

**Sproull:** So does it get better with practice? What kind of errors, is there a feedback term somewhere there, what trips you up?

**Sutherland:** Chuck Seitz is the guy who said to me some long time ago, "What you are is a geometer in the tradition of Pythagoras," right? And I'm kind of proud of that, if that's true, that's something to be proud of, I mean that's a pretty good start, right, Pythagoras, he's remembered 2,000 years later, okay, what a goal to achieve, right, I mean if you've got an ambition, that's a pretty good ambition, right? Now I don't think of it as a special skill but I do notice places where it's showed up. I remember when I was a kid, they had these tests where they said, "This arrangement of black and white blocks in this picture is most like this one, this one, this one, this one or this one." Okay? And I would look at those and the answer was always obvious, it wasn't a hard thing to think about, just you look at this, you look at that and the answer is obvious, right? Now how come is that, I have no idea, okay, but it is. I noticed this at Carnegie Tech when I was a senior, I think I was a senior and we had this course called Engineering Analysis and one of the problems that we had was a weighing machine for weighing trucks, and this weighing machine had a series of levers that supported the truck and then supported that on something and supported that, finally ended up with one of these balance bars that you had a weight that you moved along and you could weigh the truck by moving that. And the question was, "How much weight did you have to have here to balance a weight of a ten-ton…," I forget exactly what the problem was. And I looked at the problem and the answer was obvious, there was not a problem here, you just looked at the lengths of the levers and the answer popped out, I didn't see why they even give this problem in the class, I mean why bother, the answer is trivial, trivially obvious sort of thing. I stuck my hand up and said, the answer is whatever it is, and, you know, you're not supposed to do that.

<laughter>

**Sproull:** No, that's infant Gauss behavior.

**Sutherland:** What's that?

**Sproull:** That's the infant Gauss.

**Sutherland:** Well maybe so, I don't know. But I said, "Hey, wait a minute, there must be something that I'm seeing that nobody else is seeing, okay. And I've never really articulated that and you just said more about that in the last talk that you gave that I've ever thought about, it just seems obvious to me that this is what you can do. And there's a certain beauty in making stuff fit, I mean there are people who do crossword puzzles and they make the-- and there's a certain beauty in how the crossword puzzle-- and I kind of think of this as recreation like crossword puzzles, that you do this stuff and it fits together and you can get it fit together and it's as much fun as a crossword puzzle but it has economic value and so it's worthwhile doing. And I get a big kick out of that. Now, I suppose that's a special talent, I don't know. Now there are stories about childhood, my father would bring some thing home that had to be assembled and my older brother always grabbed the instruction book and read about what to do and said, "No, Ivan here's what you do," right, and by then I had it assembled. I mean it's pretty clear from the stuff how it has to go together, there's not much choice. And I don't know how he put up with me, but…

<laughter>

**Sutherland:** …somehow we're still pretty good friends. Now if that's a talent, okay, I've got it, okay, and it's geometry, there's no question about it, it's geometry. And I'm doing some layout as we speak and I think about it, I think about it at night, I think about it on the airplane, I think about it, how is this going to go together with that, I can kind of visualize these things in a pretty vivid way, I have images in my head that are pretty strong and I can see that that'll go and that won't go. Now they're not perfect because-- but yeah, that's a strong theme. And part of geometry is symmetry, I mean symmetry, some wise person said, "What could be more like my right hand than my left and yet more different?" Okay? There is a symmetry to the world, lobsters come with a big claw and a little claw, the Atlantic lobster has a big claw and a little claw and sometimes the big claw's on the right and sometimes it's on the left and it doesn't matter but they're not symmetric for a valuable reason, the two claws serve different functions. And people aren't symmetric, okay, I happen to know that Bob Sproull is left-handed. I was once being interviewed for a student who was going to get his security clearance and the interviewer said, "Is he right or left-handed?" And I said, "What the hell does that have to do with a security clearance?" And he said, "It's a measure of how well you know the person." If you know a person left-handed, you know that person better than if you have no idea, it's a hint. Another hint is do you know their mother's maiden name, if you know somebody's mother's maiden name, you know them pretty well because otherwise you wouldn't know that.

**Sproull:** So you've talked about sort of a lot of the intuition and intuitional aspects of geometry, but the other aspect it seems to me is some analytic things and especially in this business about how important wires are. And we've talked a little bit about that but I just want to reprise it in this context. So things like laying out printed circuit boards at E&S giving rise to the how big should a printed circuit board be, which has to do essentially with looking at geometry in a more abstract way and saying, "What are the constraints, what can we do?" And then similarly of course in the VLSI arena, worrying more about wires and scaling issues that arise from the geometry. Those are the obvious ones anyway.

**Sutherland:** I understand, there's-- right. How much information can you cram through a square millimeter opening, I have an opening which is a square millimeter, I want to get the maximum amount of information through it that I can, how do I do that? Well you have some options, you can have a single communication channel, you can have multiple communication, if you have multiple communication channels, they'll be smaller and so they may have higher resistance and so you may not, or-- but if you use optics, you can go up in frequency so you'll get more information per unit area and so on. And there's a geometric aspect to computing that I think is often ignored, the beauty of the computer abstraction that computer science people use is that every piece of information in memory is equidistant unless it's on disk. But the random access memory has this model that it's equally easy to access every part of the memory and that's a very useful abstraction because it avoids a whole bunch of geometric questions that you would have to answer. When we were working on the HPCS program at Sun, we planned this sea of memory thing in which communications messages would go through the sea of memory to find the memory element that it wanted and fetch it back. And Proebsting[2] observed that in that environment, DRAMs are faster than SRAMs. Now SRAMs have a flip-flop for each storage element and consequently the momentary access to the flip-flop is faster than a DRAM because you don't have to sense anything, you just get the answer. On the other hand, they're bigger and so for the same amount of memory, you occupy a bigger area and at some point the speed with which you can communicate across that area is

[2] Robert Proebsting.

the limiting factor not the actual access time of the memory and consequently DRAMs, which are much more compact in large size are faster than SRAMs. That was a very interesting observation.

**Sproull:** Yeah, yeah.

**Sutherland:** And it depends on the geometric observation. And of course, all of this assumes that they're flat, that they're two-dimensional. Okay, but space is three-dimensional, so you can use a third dimension in some effective way if you can figure out how to do that. And people are now talking about stacking chips in various ways to get the three-dimensionality to improve things and they all face the problem of where does the heat go. Because the third dimension in most integrated circuit systems is where the heat goes, out through the third dimension. It's very useful, that dimension is well used to absorb the heat. Now, these are sort of fundamental geometric questions that deserve treating as geometric questions and answers to them can be very useful guides to what you should and shouldn't design. We did this stuff at Sun for proximity communication, the idea is very simple, if you take two chips with their circuitry on the palm side, circuitry on the palm side and the silicon on the back side and you put them together face-to-face, the electrical behavior of one can be felt through the capacitive coupling to the other. And so it is possible to get information to flow from this chip into that chip very rapidly in very small areas with the information never noticing that it left the face of the chip. And that turns out to work just fine, we've built tests at Sun and they worked fine and a bunch of patents got filed and it was good stuff, but it didn't get into use. But it'd be very valuable to use because it lets you double the amount of circuitry that you have access to on the one face. There's a collection of mechanical problems associated with it, not the least of which is that unlike plywood which doesn't warp, okay, chips do. Why does plywood not warp, the plywood doesn't warp because there's always an odd number of plies and they're always symmetrically distributed, if there's a thin one this face, there's a thin one on that face and of course the grain goes different directions. But it's a balanced design and it's symmetric, you can't buy four-ply plywood, it's always an odd number of plies. Now integrated circuit chips are two-ply plywood, they're silicon and they're silicon dioxide and so they warp, okay, and that makes the face-to-face problem a little bit nastier than it otherwise would be but no reason you couldn't put circuitry on both sides of a piece of silicon.

**Sproull:** Actually, so I next want to turn to symmetry which is a perfect lead-in from plywood. So this is another theme that has come up and again, I just want to rehearse slightly the context in which we've heard about it and a few that we haven't mentioned. One was you mentioned about the test for colinearity and Sketchpad being symmetrical in the three points which is not the way it's often done. I recall that when we did the clipping stuff at Harvard the treatment of the coordinates there was all X, Y, Z and W was all purely symmetric, that may not be so surprising since that's a straightforward linear algebra type proposition but it certainly meant that we were designing only one thing, just replicating it four times. You mentioned that you often do it because it's simpler and saves work but also because it's intrinsic in the structure. I think it's the case, you've told me that the ideas about logical effort came to you by studying symmetric integrated circuit designs.

**Sutherland:** We were just lucky.

**Sproull:** <laughs>

**Sutherland:** I mean we were using Muller-C elements and we were using Exclusive-OR gates are the two basic things that we used. They happen to be two functions that are totally symmetric between ones and zeros. And low and behold, because of that symmetry, logical effort became clear and if we had worked with AND gates and OR gates, might never have found it. But the symmetry helped enormously and once we got the ideas then it was possible to apply them to the asymmetric functions.

**Sproull:** And another example you've mentioned was the near symmetry of micropipelines in terms of bubbles and full and empty traveling and so on. And finally another one which is actually somewhat of an-- always struck me as unique is the TRIMOSBUS. So this was, you may recall is a thing that you and Charlie and I did at Caltech and the idea was to use the bus as a storage element and to be able to transmit successive transactions along the bus and the question is why are there three?

**Sutherland:** Yeah, well that's interesting, the electrical engineers at Portland State sometimes come in to talk and I often ask them, why our high voltage transmission lines come in threes. And first of all, many of them have not noticed that that's true.

**Sproull:** <laughs> Oh dear.

**Sutherland:** It's sort of an important idea that they should come in threes and there's this trigonometric identity that says if you take the power or the energy that's transmitted in a sine wave, it goes to sine squared and sine squared trigonometric identity is twice the frequency, one plus cosine wave at twice the frequency. And if you take three sine waves spaced 120 degrees apart and you calculate the energy transmitted in each one, you get the sum of three double frequency guys and there's trigonometric identities that show you like magic the total energy transmitted in a three-phase transmission line is not just approximately constant, it's exactly constant, the trigonometric identity show you that. And so why are there three transmission lines, because the flow of energy is continuous and constant, no variation at all. Also true of DC by the way, which is very interesting. So, you know, there's symmetries there that have a sort of a beauty to them, those trigonometric identities happen and I don't know what magic the mathematicians had that made it come out this way but that's how it comes out and it's magic, it's amazingly beautiful.

**Sproull:** But yes, all this is great but it seems to me, the interesting thing about three is, one of the great things about three-phase is it gets motors to go in the correct direction.

**Sutherland:** Yeah, that's true, you can switch the wires.

**Sproull:** <laughs> The reason the TRIMOSBUS had three paths was so you could tell which way was forward.

**Sutherland:** That's right, that's exactly right. Three is the smallest number of legs you could put on a stool and make it stable. There's something intrinsically important about three. But nature didn't discover that, you know, why are there no animals with three legs? Marc Raibert made this cheetah he called it that is the fastest legged robot known to man and it uses the cheetah's gait, the cheetah puts its front feet on the ground and then it puts its back feet on the ground then puts its front feet on…. It's the only animal I know of that has this double flight phase gait and Raibert built a robot that does that. And I faulted him for it, I think it should have three legs and it should bounce on that one, bounce on that one, bounce on that one, while turning around in the meantime. Now be hard to imagine how a living thing would do that and keep track of where it was and--although it's perfectly possible it could but we don't have any examples in nature of animals that run on three legs consecutively. And by the way, if you run on three, you could go on that one and then you could turn 240 degrees and bounce on that one and then turn 200 and you could-- so you could have different gaits for different forward velocities, which would require various things... And I told this to Marc, "Why don't you do this?" And he's, "Well, I-- " mumble, mumble, mumble. So now he's built one with wheels, its shoes are wheels and the video is worth watching, it's a splendid video, okay?

**Waldo:** Can I break in?

**Sproull:** Yes, yes.

**Waldo:** You keep coming back to this notion of beauty.

**Sutherland:** I'm sorry, you have to speak a little louder.

**Waldo:** You keep coming back to this notion of beauty in what you're doing, that the magic happens, you did this because it just seemed to make sense and it fit together, it was more-- and that's not a phrase that most people expect to hear from an engineer. So what has the role of this sort of aesthetic been in what you're doing?

**Sutherland:** What I'm doing?

**Waldo:** Or what-- yeah, what you've done. You obviously think it's important.

  

**Sutherland:** Oh boy. I worked with a guy called Glen Fleck for a while who was basically a designer with a capital D. A designer is a person who figures out how things ought to be to be most useful and what we worked on was a museum and how do you make a museum that will teach people. He had worked in the Charles Eames office and Charles Eames is a well known designer with a capital D who designed all kinds of things including furniture and an amazing collection of stuff. And I think Glen Fleck understood that beauty was important in all of the things that he did, that somehow the aesthetic is important in the design of any object. He said, "If you want people to keep the object on their desk, it's got to be attractive." So if you have an advertising piece that you want to hand out, you should make it attractive so that people will want to keep it on their desk, that's what you want, you don't want them to just throw it in the wastebasket because it's ugly. And he understood that full well and he said, "Now, the difference between an artist and an engineer is that an artist makes something beautiful where beauty is in his head, he makes something that he thinks is beautiful and an engineer makes something which many people will think is beautiful because it's useful. It is some sense got a similar characteristic but it's more widely obvious that it is, that that characteristic is there." And in some sense, beauty is a thing to be sought that a beautiful program has exactly the right stuff in it to do its job and not one iota extra and it has a notation that's clear enough for everybody who reads it to understand. Computer programs are read much more often than they're written, so you have to make them readable and that helps you make them better because you'll remember how to read them a few years later when you come back to them. And I think the same is true of any artifact that there's a beauty in the artifact that's important. And I think that this is something that Apple has understood very well, better than nearly any other manufacturer in the computer business that, the attractiveness of the artifact is an important part of making it saleable. And so by and large Apple's products are attractive just as objects and I think they put enough effort into the design and the testing of the design and so on to avoid ugliness, to avoid awkwardness in the use and whatever. Now I don't know why that's important to me but it is, it's important to me that the beauty of the artifact along whatever axis you measure beauty and it's different for every artifact is somehow important and having a symmetric representation, Bob mentioned a symmetric formula for the area of a triangle given the coordinates of the three vertices of the triangle generate a symmetric algebraic function that tells you the area of the triangle. And by symmetric I mean that if the three corners are A, B and C you should be able to interchange A, B and C in any way you choose and still get the same area. Now there's one exception to that, because it's a triangle you could look at it from the top and it would be A, B, C or you could look at it from the bottom in which case it would be A, C, B or the sequence A, B, C matters, so maybe one sequence would give you a positive area and the other a negative area, that would be okay. But, or maybe it doesn't matter so you might want to have some-- in mathematics we have this absolute value of what you put with bars which takes the positive and negative symmetry away and maybe you would need that to get such a formula. But simply writing that formula down is pretty in some sense, it has a beauty to it, it has the beauty of symmetry. And somehow symmetry is attractive, okay, if you see people whose faces are vastly different on one side than the other, you don't think they're attractive, it's a common trait. Everybody’s face is a little different on the right and the left, and that turns out to be important too, and having a mole in just the right place on your cheek is a sign of great beauty, right? You might even want to grow one if you want to be, you know, very beautiful and notable person, but anyhow.

  

**Sproull:** So the third part of this that I wanted to talk a little bit about was structure, and again, to repeat, reiterate, some of the ways this has shown up. Obviously a key element of Sketchpad that was part of what made it so impressive is this wasn't just a drafting system, this was a way for a computer to represent structure in drawings. Some pictures, for example, are relationships between different lines and curves and points and so on, so that was one aspect of structure. Another, which you mention quite prominently in your thesis, is looking for and finding structural similarities in the way you treated various things. To delete, to insert, to merge, which is what led to some of the feel of the object-oriented method of coding where a particular kind of object like a point or a line had routines or procedures associated with it that could, that were, had common or similar characteristics with procedures for other geometrical objects. And finally, and maybe this is the one that I find most interesting, and you alluded to it a bit before, is the structure in all of your async chip designs. Some of it is part of the micropipe--the structural aspects of micropipelines. Some of it though is what turns out to be the modules that you design. We've talked about the geometric aspects, but there're also electrical and protocol aspects of those, and you mentioned the crossword puzzle. They fit together. They fit together in simple and elegant ways, and so that thing-- and your discussion even of the joints and nodes is like that, where the structural boundaries or the structural definitions turn out to be revelatory and organizing in the way you put the thing together, and finding better or simpler or more elegant structural descriptions is a wonderful end in itself, because it gives you a more elegant, more simplicity, and a design methodology or a design pattern, if you will, that can be widely adapted. Does that…

**Sutherland:** Well, finding the patterns in things is tricky. We're working now on a thing we're calling Hexnet.

**Sproull:** <laughs>

**Sutherland:** Now, you know, we built a chip called Weaver, and Weaver basically has 72-bit words and it has a crossbar switch that is non-blocking, eight inputs, eight outputs, each data element going into one of the inputs says which output it wants to get out on, and it goes there, and each intersection in this crossbar has a first-come, first-serve aspect to it and so information can flow through the crossbar in any way. Never blocks. One piece never blocks another, but everybody gets through and I started working on this when some of the guys at Sun Labs said, "Well, we have a crossbar switch that has a latency of less than a nanosecond via an electrical trick, clever electrical trick." I don't want to detract from the trick. It's very clever trick, but I looked at that and said, "Hey. If we build this in a self-timed way, we can simply go through the number of stages that we have to go through in less than a nanosecond. So we don't have to use any electrical trick. We just build it and it'll work at that same speed. Why would you want to use a clever electrical trick?" and so that was the origin of the Weaver. We built a test chip on it. Works like a charm, and we've done a whole bunch of experiments on it that are interesting in their own right. But what's next? Now, in the Weaver, each data element says what its path through this eight-level switch should be. So it devotes a fair number of bits to saying, "Turn left here. Turn right there. Turn left here. Turn right there. Turn left here. Turn right there." But wouldn't it be nice if it could give the address of where it wants to go? Wouldn't it be nice if you could build a network where each data element carries with it its destination address and every place that it gets to in the network knows where its address is relative to where this thing is going and sends it in the right direction to get towards the address? So we thought this might be an interesting project to work on and we proposed to DARPA, and I'll talk about that, what that proposal really is after a while, but we proposed to DARPA that this might be one of the examples we could use as the grist for the mill of what we're really working on, and so that's what we're doing. We're deeply involved now at the ARC, we're deeply involved in designing a thing which can provide this function. It can be the nodes of a network, which will communicate 72-bit data words, self-timed. So they'll take however long they take to get there to get there, but each data element will carry with it an eight-bit address, which is the address where it wants to go, and each element in the switch will say, "Oh. To get there from here you have to go that way," right? And so there's not a path here. There's a destination. Path is much easier. It says, "Turn right first. Turn there." Okay. A destination is a little harder, so that's what we're working on, and it turns out to be a terrific example, because it exercises all of the things that we know about self-timing. You want to make it-- I want to make it very fast, because, you know, I think that's an important thing to do then. If it can be fast enough, you can turn the power supply down. It'll consume a lot less energy but it'll go at the speed you really want it to go at, but the energy goes with $CV^2$, $V^2$, $V^2$**.** That's a big deal, right? So we're working on that. So symmetry comes into this, you see? The thing we know how to arbitrate is first-come, first-serve between two. Which of these two horses is faster? Not which of these seven horses is faster? Which of these seven horses is fastest? Okay, but not-- don't put them in order, okay. So two. So wouldn't it be nice if every junction in this network, every input, has only two possible outputs? Seems like an obvious thing to do. Well, the network that you look at most often, that has this property, is your bathroom tiles. If you look at the cracks in your bathroom tiles, an ant crawling along one of those cracks comes to a junction. He has only two ways to go, right? So the hexagonal topology has this very nice property that it's a binary decision. Every time there's a decision it's a binary decision. If you're in Manhattan and you have a Manhattan geometry, every time you come to an intersection you have a trinary decision. Do I go straight, left, or right? That's too hard. Let's make a hexagonal topology so we have only a binary decision at each intersection. Well, that's fine, but laying out integrated circuits with 120-degree wires is a little tricky, so we'd like the wires to all be north, south, east, west, and so what, what geometry do we have that has that same hexagonal topology but it's all north, south, east, west? They're cracks in a brick wall. An ant is crawling up the crack between two bricks and he gets to the intersection, he can either go right or left. He can't go straight ahead because there's no road there, right? So instead of Manhattan geometry, we use the geometry of a brick wall. It's the same topology as your bathroom tile. But we're calling it Hexnet because that's kind of memorable and it's, you know, short, so we're hard at work on Hexnet, and it's bringing together a whole bunch of pieces of how do we do links and joints, how are we going to make it fast, what can we do for the amplifiers, blah, blah, blah, blah? All the kinds of things that we've learned over the past many years will come together in this test chip. We're hoping that Oracle will build it for us. They built Weaver and I've talked to them about it. They seem inclined to maybe do something like this and it would be terrific if we could, and DARPA is footing the bill. I don't think they'll foot the bill for the actual fabrication, but maybe Oracle will do that and gain the knowledge how this thing works. Now, there's geometry in that choice of thing. Why did we choose that? I've given you the arguments for why we chose it but where did those ideas come from? I don't know. You know, it seems obvious to me that a binary choice is better than a ternary choice, and, you know, you look around the world, what do you see? You see hexagonal bathroom floor, and I noticed that the hexagonal bathroom floor has the same topology as a brick wall. Now, I told all this to Chuck Thacker across the street here and Chuck, “I never noticed that.” How could you miss that?

**Sproull:** <laughs>

**Sutherland:** I mean, you look at a brick wall, you look at a bathroom-- it’s the same topology. I mean, that seems perfectly obvious to me, but Chuck Thacker’s not a stupid guy, and he didn’t notice it, and many other people have said to me, “I never noticed that,” okay? Well, I didn’t know that you couldn’t notice that. <laughs> Right? So that’s what we’re doing. We’re working on Hexnet. Now, maybe that’s a good choice; maybe it’s a bad choice. But it’s a choice which has a sound reason for it and I don’t think it’s any worse than a Manhattan geometry. I suspect that the distances may be slightly longer or something, but, you know, we can adjust various things to make that come out, and if that really works, maybe it’s a good choice to do for many other networks other than the one we’re working on.

**Sproull:** Well, and an interesting aspect of that is that the reason you had to solve this problem is the binary choice is what we know how to do with electrons and we don’t know how to do anything else.

**Sutherland:** Well, we tend to have binary computers and they tend to like binary computers. <laughs>

**Sproull:** No. But I’m-- the arbitration problem especially.

**Sutherland:** The arbitration problem especially. Okay. There have been various papers written on how to do arbitration between three and they’re all kind of messy. They’re not pretty and so on. We just use the Seitz arbiter, which, you know, been around for nearly 30 years. It’s published in Mead & Conway and we have a variant on that, which we simply use. We’ve built a bunch of chips that tested 17 ways from Sunday and we’ve never observed a failure, so we’ve developed confidence that it’s a thing that we can use. It’s a building block of what we do. The trick is a flip-flop can go metastable, and it’s quite easy to detect the exit from metastability. The two sides of the flip-flop have some constant value and if it’s metastable, those two values will be close together in voltage. As it exits metastability, those two values get quite different and what Setiz said is you can use the threshold of a transistor to detect when that difference is big enough and then metastability’s over and you know what the decision is. So just wait until the two voltages…. That’s the Seitz arbiter, and that’s what we use. It’s a very simple thing to do.

**Sproull:** So I want to close off my symmetry structure, design, whatever, and you’ve actually unwittingly led me into a particular piece of the asynchronous world that I want to discuss. Do you remember the Q-flop?

**Sutherland:** Yes.

  

**Sproull:** So the Q-flop was a variant of the Seitz arbiter, essentially. It says you can sample the state of a signal. You can ask for a sample of a state of a signal and get told when it has been reliably sampled, which in effect, well, in effect there's an arbiter inside there, and you can build state machines out of Q-flops, and you could build computers this way, much the way you could with stoppable clocks, which is what Chuck Seitz taught us how to do with the Harvard multiplier and clipping divider. But no one has, this has not been an area that people have pursued. You have not really pursued it. People don't, when they even build separately timed regions on a chip, they don't do it this way. They use synchronizers between segments that are running off the same master clock. Why is that?

**Sutherland:** I don't know.

**Sproull:** Is it inherent? Is there some inherent--

**Sutherland:** Bob, these are hard questions. Why do people not do X? I don't know.

**Sproull:** Well, so--

**Sutherland:** I mean, it's as hard as why do people believe X?

**Sproull:** Well, my naïve view was if you were willing to solve the bundling problem and run bundled data, then your only problem was to build simple state machines for control. Why not do it this way?

**Sutherland:** I don't know. Maybe we should try it.

**Sproull:** You have to now generate a clock.

**Sutherland:** Maybe we should try that.

**Sproull:** You have to generate reasonably well-crafted delays in CMOS. We know now, we kind of know, we did, you know, some people figured that out kind of. Maybe not enough to production quality, but it's, it's a whole avenue that doesn't seem to me has been pursued.

**Sutherland:** So to me-- yeah. Okay. Well, I take your point.

**Sproull:** Okay. But now here's a flip observation. So at the time that we started all this asynchronous work there was a body of academic work on how to build asynchronous state machines. It was complicated and hard as hell. Steve-- do you remember Steve Unger's book? at Columbia? and these were arbitrary state machines, mind you, and Charlie was the expert at all the manipulations required to make this happen, which I just reeled from. I don't know about you, but it struck me, it has always struck me, that your micropipelines idea was in some sense a reaction to that, saying, "I don't want to have to design complicated asynchronous state machines. I want the asynchronous protocols, if you will, to be very simple and to make the complexity in control or something, be, come from, some other, be done in some other way."

**Sutherland:** Well, and when we talked about the links and the joints, the joint is, in fact, a synchronous machine. It has an action, which it does, and the joint, the size of the joint is limited so that that action can be simultaneous over the entire joint and so a conventional state machine can be used in a joint and in a conventional way. It is a clocked system in some sense, and the asynchrony is all between the joints. Maybe that's the simplifying assumption that we've taken. But your point about the Q-flop is a good one. That's something to revisit.

**Sproull:** I mean, perhaps at the time, I mean, you've developed huge confidence in the Seitz arbiter, and at the time the Q-flop was being discussed, I don't think there was such confidence available.

**Sutherland:** I'm sure there wasn't such confidence, but…

**Sproull:** And it's worth--

**Sutherland:** I think one can build Q-flops today that are sound.

**Sproull:** Right.

**Sutherland:** Right?

**Sproull:** And it's worth pointing out that today's designs, correct me if I'm wrong, in the commercial world, involve somewhat locally synchronous, globally asynchronous designs, but they are connected by connections that use synchronizers, not arbiters.

**Sutherland:** That's true.

**Sproull:** Because basically all of those parts are running off the same clock, hence they cannot use arbiters, because they can't wait. They don't know how long to wait for the arbiter.

  

**Sutherland:** Oh, yeah. I think the fundamental problem with the clocked design is it doesn't know how to wait. Now, you can make a pausable clock that says that the next clock pulse doesn't happen until everybody's ready for it. That works, but then you have to have feedback between the clock generator and the customers of the clock, and that's hard because amplifying the clock signal to deliver to many customers is an electrically hard problem, which is solved by a clock distribution network that has to be carefully engineered in its own right.

**Sproull:** Right.

**Sutherland:** That takes time.

**Sproull:** That's right, and that certainly is a thing that separates synchronous from asynchronous designs.

**Sutherland:** Yes.

**Sproull:** But it needn't separate stoppable clock designs from, for example, your style of micropipeline design, because you still, you also are generating those control signals. We don't call them clocks, but they--

**Sutherland:** That's right.

**Sproull:** --talk to the same transistors, right, that clock-- excuse me-- latch the data. <laughs>

**Sutherland:** Now, now, speaking about clocks, Jo Ebergen's son Tom was an intern at the ARC for four months. He's a student at Waterloo, and they have a work-study plan where the students all do internships in industry for one quarter each year. So he spent a quarter with us as an intern, as an industrial intern, in the ARC at Portland State, and one of the tasks that he undertook was looking for unexpected synchrony in the Weaver. Now, the Weaver has 10 circular buffers, circular FIFOs, into which you can put any number of data elements. So they, they turn out to be 48 stages long, so you can put any number of data elements from 0 to 48 in them. If you put 0 in them, you get no throughput. If you put 48 in them, there's no space, no throughput. Somewhere in between, you get reasonable throughput. So he put half a dozen in one ring and he put a single guy in another ring and he chose two rings which were geometrically adjacent to each other or at least nearby, and then he counted how many times these groups went around the rings. With counters. He ran an experiment for a second and the counts turn out to be a billion or so trips around the loop in this time, and he got the two counts--totally logically separate rings. The two counts were identical within two or three counts. Not two or three percent, two or three counts in a billion. So there is every reason to believe that these two actions, logically totally separate, were happening in synchrony, that for some reason the guy in one ring was influencing the guy in the other ring just enough to cause him to go around the same number of times. Very interesting phenomenon. Never before observed, as far as I know. I don't know of anywhere in the literature that this has been observed, and so one could ask the question, "Why?"

**Sproull:** <laughs>

**Sutherland:** "Why does this happen?" and that's a very interesting question. My hypothesis is that the several data elements in one bunch draw enough power from the power supply that they cause the voltage to droop a little bit there and because there is close to where the other guy is coming along, the other guy can't get past because as soon as he gets close, the power supply drips a little bit, droops a little bit, and then his, he's slowed down, and so he's sort of stuck by-- this is like being stuck in one lane of the freeway behind this massive set of trucks in the other lane that are providing so much wind across to the next thing that you, you can't really get by or it's a rainy day and they're sloshing rain up into your--

**Sproull:** Snowplows. It's snowplows?

**Sutherland:** What's that? Say it again? Snow--

**Sproull:** It's snowplows.

**Sutherland:** Snowplows. Yeah. You can't get past the snowplow, even in the other lane, because it's pushing snow into your lane.

**Sproull:** You can but it's very dangerous.

**Sutherland:** Right? That's a good example. So that's my hypothesis of why it is. But the phenomenon is real. We've measured that phenomenon. Now, does it matter? Well, in the self-timed world, we don't care. The fact that you've been slowed down a little bit by the action of this other guy shouldn't make any difference to the correct function, okay? So in some sense we don't care. There's a second-order phenomenon, which is interesting to observe, but do we want to get rid of it? I don't think so. I don't think we need to know the root cause to get rid of it. It's just an interesting thing to observe.

**Waldo:** Yeah, you want to know.

**Sutherland:** What's what?

**Waldo:** You want to know why this is happening. Even if it doesn't matter.

**Sutherland:** Well, we have a perfectly good hypothesis.

**Sutherland:** What measurements would you like to make to confirm or deny the hypothesis?

**Waldo:** <laughs>

**Sutherland:** I don't even know what to measure to figure that out. Okay? I mean, maybe we'll make another chip that has intended to measure this particular phenomenon, but, you know, making a chip's a big deal. We have this chip. We're milking it for all the things that we can get out of it. This is one that we got out of it without it being an intentional-- that's not an experiment we intended to try when we first built the chip. But having the chip, of course we tried this experiment, and it's quite a remarkable thing. He's got nice graphs, which show this ring interacts with that ring and under what circumstances it does and how much and all this kind of stuff and geometric proximity needs to be nearby in some very real sense. They need, the two rings' intrinsic speed needs to be not too much different, because of, you know, a superstar in a ring that's far away from us doesn't care. Okay. And then the amount of loading that the half a dozen data elements, if you make them data elements that don't have any data, that just all transmits zeroes, they consume less electrical load and so they don't exhibit this phenomenon as much. That's bolstering the evidence that it's a power supply droop issue.

**Sproull:** Mm-hm. Yeah. Yeah.

**Sutherland:** But it's not solid evidence.

**Sproull:** No.

**Sutherland:** Just--

**Sproull:** Evidence. You gave me as a Christmas present-- Utah. That was-- it was the Christmas I was at Utah you gave me a garbage can, a wastebasket, which was in the shape of a Coke can. I still use it. Every day. So we're going to resume and Marc Weber--

**Sutherland:** I have two Tiffany lights which say Coca-Cola around the side. One of them I commissioned from a guy who had made the Coca-Cola Tiffany lights for the ice cream company from Seattle. What are they called? I forget. They have a branch in Berkeley called Farrell's. Farrell's Ice Cream. Very good ice cream. And they had these Tiffany-- and he had the pattern on which they were made, so I commissioned one. I have that and I have another one I got somewhere. Unfortunately, the new house has not got, not yet, got a place allocated to hang <phone> it.

**Sproull:** Hold on. Uh-huh.

**Sutherland:** Yet is the operative word here. Yet.

**Sproull:** I'm sure this will be solved.

**Sutherland:** They need to hang, and the oriental rug under which my multiplication table--

**Sproull:** <laughs>

**Sutherland:** Or oriental rug that ate my multiplication table repeatedly also needs a place in the new house.

**Sproull:** I'm sure that will get solved too. Will the kitchen--

**Sutherland:** Hm. That's probably--

**Sproull:** --have a hole in the floor for the periscope?

<laughter>

**Sutherland:** No. But we have a number of features in the new-- you have to come to the open house and see. We have a number of features that are there strictly for historical or educational reasons.

**Sproull:** Ah, okay. All right. So we're going to start again and Marc Weber is going to start with some questions.

**Weber:** All right. So can you talk about--

**Sutherland:** You have to speak up, because my ears are not very good.

  

**Weber:** True.  So at Lincoln, can you talk about when you met Severo Ornstein, impressions of him at the time?

**Sutherland:** Severo Ornstein?  No.  I think I heard Severo Ornstein long before I met him.  He had a piano and he played beautiful piano music at lunch time, and he was not known as Severo Ornstein.  He was Randy Ornstein.

**Weber:** Really?

**Sutherland:** Like Robert [French pronunciation], you can place where people know him from by what they call him.  Okay?  He took the name Severo after he left Lincoln.  It was, I think it was, his given name but he went by Randy at those time, that time.

**Weber:** And any other impressions of him in that period?

**Sutherland:** No.  The Lincoln group was quite a remarkable bunch of people.  The important person in the Lincoln group was Wes Clark.  I've talked a little bit about him.  He died recently.  But the important thing he did with TX-2 was to keep the club small.  He understood that online use of computing was important and it was the future, and he was determined that TX-2 was going to be used online, so it was not batch processing, which is totally unusual in those days.  I mean, everything was batch processing.  You submitted your deck of cards, you came back after the run-around time, turnaround time, I'm sorry, and you got your listing back, and that was the way computing was done, and Wes said, "No.  Computers are tools to be used by people at people's pace," and so he, he said, "Ivan, you get, you know, a reasonable amount of computer time."  Now, admittedly, it was at four in the morning, which started a habit of getting started early in the morning that I've kept lifelong.  But, you know, it was Wes Clark's drive that made that possible and I think it had a huge amount to do with the online use of computing.  That he had an exemplar of a machine that was many years ahead of its time in terms of computing power that was used that way, and, you know, Sketchpad was perhaps the most widely known result of that, but there were many others, and those results I think led the way to encouraging people to use machines online, and I think Wes is unsung for that contribution.  It was a very important contribution.

**Weber:** Yeah, and my next question's what-- the team that he had there, describe kind of the way they interacted with each other, the way he brought that team together.

**Sutherland:** Well, it was a very congenial group.  The character of the group, every group has a character who is so memorable he has to be recorded, the character of the group was a man named Tom Stockebrand.  Tom Stockebrand was a Caltech graduate who rarely wore shoes, and one time I was going to give a demonstration of Sketchpad and he said to Fred Frick, who was the boss of the Lincoln Laboratory, he said, "May I attend?" and Frick looked at him and said, "Well, you're wearing shoes today, so yes, you can come," and right. Stockebrand subsequently went to work for DEC. I went to visit DEC one time and they had partitions that were made of two-by-fours with a panel inside that were partitions and a loud voice from the other side of the room said, "I think that's Ivan Sutherland." "Oh," I responded, "Oh, I think that's Tom Stockebrand." Said, "Stay where you are, Ivan. I'll be right over," and he came over the partition.

**Weber:** <laughs>

**Sutherland:** But the group was informal. It was, people were informally dressed. It was a friendly group. I don't know what Wes did to instill that environment in the group, but--

**Weber:** But they were--

**Sutherland:** --it was clear.

**Sproull:** They were sort of all working on their own individual problems, right? There weren't teams of people working on the same problem, were there?

**Sutherland:** No, there were some teams, but--

**Sproull:** Some.

**Sutherland:** --by and large there were bunch of separate problems. I don't know how the problems were picked. There was a speech group that was working on speech, and Larry Roberts came in, worked on image processing of various kinds and not everybody was as computer savvy as they might've been. I watched one of the speech guys change the location of one of this global variables. Four or five times. It was in location 125,247, and he changed that to, you know, 248 then he changed it to 246, then he changed it-- I said, "Why don't you give it a symbolic name and let the assembler figure out where to put it?" "You can do that?"

**Weber:** <laughs>

**Sutherland:** And I watched him change the number multiple times as he's unwilling to trust that the operating system would, in fact, assign him a location and he didn't need to know the number of the register that was in. You just need to know the name of the thing, right? This is, you know, this is the eternal problem of the thing and the name of the thing and the name of the name. There's this wonderful thing in "Alice in Wonderland" where the victim is told the song. The title of the song is called "George."

  

“Well, then the song is about--“ “No. No.” The song is entitled, “Mary.” But the title is called “George.” But the song is about Pete. Okay. I mean, there’s the title and the title of the title, right, and there’s indirection, which is sort of the lifeblood of computerology, but it can be terribly confusing.

**Weber:** And--

**Sutherland:** Computer History Museum is seeking history, but you only get the labels of history as interpreted by people like me, right?

**Weber:** <laughs>

**Sutherland:** You don’t get history.

**Sproull:** <laughs> You’re right. You get a remembrance of things history.

<laughter>

**Sutherland:** That’s right.

**Weber:** The husk. And the Lincoln WAND, that was Larry Roberts’ project, or were you--

**Sutherland:** That was Larry Roberts, right.

**Weber:** Okay.

**Sutherland:** We fussed about that to measure the head position for the head mounted display and there was some effort-- was done at Lincoln, this is later-- that an ultrasound system was built to measure head position, which turns out to be moderately hard, because sound goes about a thousand feet per second and drafts in air in an office building are a foot or two per second. So measuring, using sound to measure range, it’s not a very good game.

**Weber:** You’ve talked about the sword of Damocles at length, right? Yeah.

**Sproull:** We did, a little.

  

**Weber:** Thought so. Yeah, but you've covered it pretty well. I didn't read that part of the transcript because I figured you would.

**Sproull:** Okay. Ask whatever you want.

**Weber:** Well, I don't want to repeat on that.

**Sutherland:** In some sense, in some sense, the beauty of the TX-2 machine was that the machine was built to learn about the reliability of transistors in that kind of service, and so the machine had no other purpose. So it was not built to do the accounting for Lincoln Laboratory or to do the projections of future radar systems or something of that sort. It had no purpose except to be a machine that would teach about the reliability of transistors, and so its purpose could be whatever the people who were there wanted it to be, and I think this is an example of how unfettered research money can pay the biggest possible dividends. That if you know and you target a piece of research, you're less likely to get exciting results than if you have clever people and you let them explore what they want to do. This is the hard part to explain to the bean counters. You know, "We can promise you a breakthrough in this research project five weeks from now. We'll have that breakthrough." Well, I'm sorry. We don't know that.

**Weber:** <laughs> Do you remember when you read-- I'm assuming you read Vannevar Bush's article, "As We May Think"?

**Sutherland:** About? I'm sorry?

**Weber:** The article, "As We May Think," by Vannevar Bush?

**Sutherland:** Yes. Yes.

**Weber:** You remember when you read it?

**Sutherland:** I have no idea whether I read it or not.

**Weber:** Did it have any info-- <laughs> Okay.

**Sutherland:** I'm not a big reader, and even less a rememberer of things read.

**Weber:** Well, you've answered the question.

  

**Sproull:** <laughs>

**Weber:** And first, you guys did ask a little bit on this, but the first exposure to Licklider's networking ideas and I was going to ask you about his book, "Libraries of the Future," but maybe that's not the right question.

**Sutherland:** I don't know the book.

**Weber:** Okay.

**Sutherland:** I read the thing about the Sloan Foundation piece, about Licklider, which-- and I read it carefully-- and the parts of it that I recognized seemed to be about correct. Seemed like a well-written book. What was it called?

**Weber:** "Dream Machine."

**Sutherland:** "The Dream Machine." Right. "The Dream Machine." Right.

**Weber:** But did his networking ideas, I mean, you've already talked about that the galactic, intergalactic network was--

**Sutherland:** Yeah.

**Weber:** --sort of percolating.

**Sutherland:** Right.

**Weber:** But how did you feel about those ideas? Were you excited by them? Was it--

**Sutherland:** Oh, it seemed to be a good thing to try, and I tried and failed. I mean, there were three large computers at UCLA and I encouraged the people there to submit a proposal for linking them into a network, which they did. But without great enthusiasm and without great ideas, and whereas they could make a network that worked, nobody really cared. Right. And the message I learned from that is that as the director of the IPTO office, I could say "yes" or I could say "no," I could not say, "This is what I want you to do." Proposals have to come from below. Very hard to direct research. You can find it and nurture it, but it's very hard to direct it.

  

**Weber:** But were--

**Sutherland:** I think what Bert did at Sun Labs was to say, “All I can do is hire people that are relevant,” and, you know, you hire the people that are relevant to the areas you want to go into and it’ll happen, and if you can’t find those people, you can’t do it.

**Weber:** Do you have any recollection of the trip to Hot Springs with Licklider, Larry Roberts, in late 1964, which is where apparently a bunch of things gelled around networking?

**Sutherland:** As I recall, the Hot Springs trip was done on a train. The train left Washington and went to Hot Springs and that seemed like a good thing because it was a linear search for anybody on the train. Wanted to talk to somebody, you had only to walk the length of the train and you would run into them, and there were no cell phones and so on, so there were no distractions. Lot of conversations happened on the train. I don’t remember the meeting much other than that. Hot Springs is a very nice place.

**Weber:** <laughs>

**Sproull:** So was that organized to have the discussion on the train? Was that the MO?

**Sutherland:** Oh, yeah. Yes, oh, yeah. The train was clearly a part of the meeting.

**Sproull:** So it was a private train.

**Sutherland:** Yeah. It was a train occupied solely by the people who were going to the conference.

**Weber:** Ah, okay.

**Sutherland:** Conference started when you got on the train.

**Sproull:** Yeah. Hot Springs was a stop off, I mean, a--

**Sutherland:** The excuse.

**Waldo:** The excuse. It began with the hallway track?

  

**Sutherland:** What’s that?

**Waldo:** It’s a conference that began with the hallway track.

**Sutherland:** Yes.

**Waldo:** You just would wander and find people.

**Waldo:** I see.  Brilliant.

**Sutherland:** <laughs>

**Weber:** Was best part of-- yeah.

**Sutherland:** Licklider was good at-- I don’t know whether he organized it.  I wouldn’t be surprised if he did, but Licklider was quite good at getting people together and getting them to work together.  I mean, he was a psychologist by training, and a good one.

**Weber:** And you funded some of the networking experiments that Larry Roberts did.  Can you talk--

**Sutherland:** I funded some of the network--

**Weber:** Didn’t-- I believe you funded some of the networking, early networking experiments, Larry Roberts did?

**Sutherland:** I have no idea.

**Weber:** Okay.

**Sutherland:** The one that I recall is one at UCLA which basically was a failure, and the reason it was a failure was my proposal that they’d picked up.

**Weber:** But there was also the connection between Lincoln and-- where was it?  Somewhere in California, but it was somewhere else.

  

**Sutherland:** That was not my doing.

**Weber:** Okay.  It was not.

**Sutherland:** That was not my doing.

**Weber:** Okay.  And many people have said-- I have no idea if it's correct-- that you had told Larry that he would succeed you at IPTO.  Or implied that.  Does that ring any bells or…

**Sutherland:** No idea.

**Weber:** Okay.

**Sutherland:** You can ask Larry whether I told him that.  If he remembers.

**Weber:** <laughs> I will. <laughs> But you did say in your earlier interview here that you saw Bob Taylor more as a kind of interim person, as not necessarily being the long-term leader of the IPTO group.  Can you explain why?

**Sutherland:** I hired Bob Taylor from NASA as my deputy and he was clearly my deputy director.  He was called that, and I had not hired him as a succession plan.  I hired him as a deputy.  But, you know, life is whatever it is, right?  And you can ask Taylor what his version of it is and you'll get his version of it. That's fine.

**Weber:** But the fact that he did not have a computing background, was that part of that?

**Sutherland:** No.  He had known Licklider and worked with Licklider on various NASA projects, I believe.

**Weber:** No, I know that.  And when and how do you remember first hearing about packet switching?

**Sutherland:** I'm sorry, I didn't hear the words.

**Weber:** When did you first hear about packet switching?

  

**Sutherland:** Oh, packet switching. I have no idea. I do not recall the first instance in which pocket switching crossed into my vocabulary, sorry.

**Weber:** But what did you think of it early on?

**Sutherland:** I have no idea. I went on to do other things and that was not in my sphere of interest.

**Weber:** Okay. Anything else on ARPA days that you want to say or that-- anything particularly for Larry Roberts when I talk to him soon?

**Sutherland:** Well, hm. There is an issue as how you make organizations lively. DARPA has been pretty lively over a long period of time, but it has decreased in liveliness as it's grown. Becomes large enough to be noticed and then political forces and other forces encourage it to do things or not to do things for reasons that aren't purely the science or the engineering behind it, and small organizations tend to be more nimble than big ones. I don't know what the magic is of making an organization that's effective, but freeing it from excessive administration is clearly a part of it, and to do that you have to take risks, you have to bet on the people, rather than providing them this tight set of rules that they have to follow to prevent them from cheating. If you bet on the people, you have to bet that they're reasonable people and will do reasonable things without having too much enforcement to go with it. I was struck at Caltech when the Ford Motor Company came in to use the Caltech wind tunnel to test its new body shapes, and the Caltech wind tunnel had been used for classified, militarily classified stuff for many years. Airplane shapes were tested there on a regular basis and there was reasonable security but not excessive. When Ford came in, huge security, and the point was that the U.S. citizens tend to be loyal to the U.S. country, but not to the Ford Motor Company, and so you needed more security for Ford's secret stuff than for national secret stuff. Perfectly reasonable thing to do. But that was the first time I noticed that that's true and the reason the U.S. Defense Department works at all, the reason that the security clearance process can work at all, is that the population is fundamentally loyal. If the population were, you know, 80 percent disloyal, you could never figure out what the 20 percent that are loyal are. Okay. If it's, you know, 99 percent loyal, then you have some hope of finding some of the 1 percent that are not. It's a peculiar thing, that to get the best results you have to trust the people who are getting the results, and it's remarkable that not everybody does. I met a woman one time who was sure that all of the moon landing stuff that NASA put on was fake. That it was all just a scam to extract tax money from a population. Now, perhaps, perhaps it is, but I think that's unlikely. I know some of the people that were involved and they seem to me to be honest, loyal citizens.

**Weber:** Can you talk a little bit about Jim Clark and Alan Kay and the--

**Sutherland:** About Jim Clark and Alan Kay?

  

**Weber:** Or did--

**Sutherland:** Well, they're both-- they're both--

**Weber:** Alan I think you've-- Jim Clark.

**Sutherland:** They're both likable folk.

**Weber:** But when did you meet them? What was your--

**Sutherland:** I met Jim Clark fairly late in Jim Clark's career. He came to the University of Utah, presumably to study in the graphics world that I was in, but he came after I left. So he and I did not interact at the University of Utah.

**Weber:** Ah, okay.

**Sutherland:** Okay. Alan Kay was, of course, one of the leading graduate students at the University of Utah and I interacted with him quite a bit.

**Sproull:** Well, in fact, I remember, he visited you at Harvard. He came to visit the graphics project. I remember Alan vividly from that first meeting.

<laughter>

**Sutherland:** You remember that visit?

**Sproull:** Absolutely.

**Sutherland:** I don't, see. You had a different perspective.

**Sproull:** Well, he was chattering away, asking lots of very interesting questions, totally engaged.

**Sutherland:** He was Alan Kay.

  

**Sproull:** He was Alan Kay, right.

<laughter>

**Sutherland:** Pure Alan Kay. Right? Even then.

**Sproull:** Pure Alan Kay.

**Sutherland:** <laughs>

**Weber:** And then so Bob Sproull said that he remembered sitting with you and Larry in San Francisco when Larry was frustrated getting protocols done for the ARPANET, and he-- do you remember anything about this, that Larry--

**Sutherland:** No.

**Sproull:** So we were in San Francisco because that was the Fall Joint Computer Conference at which Engelbart gave his famous demo. You were giving the papers on the Harvard project. Larry, I presume, he was already at IPT, I believe, at the time.

**Weber:** ’68.

**Sproull:** I don’t remember. He was certainly--

**Weber:** Oh, he was there, but he was not yet the head of it.

**Sproull:** Oh, is that right? Okay.

**Brock:** I think Taylor left in--

**Sproull:** But at any rate, we were sitting outside, I don’t know, drinking coffee or something on a sidewalk café. And you and Larry were swapping frustrations, kind of what you were expressing before, that you can’t tell people what to do. You can hope to find good people who will do good things and enable them, but it’s kind of hard to get them to do their best. And I think Larry was kvetching about these various graduate student committees that had formed themselves to do ARPAnet. Host to host protocols we’re way before TCP here. And I think it was Larry, but to some extent, it was the both of you concluded that you two could do this this weekend.

**Sutherland:** And we could do it over--

**Sproull:** You could do it over the weekend.

**Sutherland:** I see. Well--

<laughter>

**Sproull:** I just was-- two IPTO directors commiserating about the wet noodle approach to research.

**Sutherland:** I don't remember the thing. But yes, it is a frustrating business. And I think Licklider understood it really well that when Licklider was put in the IPTO job, he basically assembled the cast of important characters and asked them what the office should do. He didn't tell them what to do. I think he asked them what to do. And you get a group as good as the group that was there, you get Perlis, Newell. You get-- at MIT, you get Corbato, you get a few other people from MIT. You get those people together, a corresponding group from UCLA and Berkeley. You get them together, and you ask them what to do. They'll tell you. I mean it's incredible. Now, that, of course, is elitist science, which is very naughty you see. You should give equal opportunity to everybody to participate in the decisions of what should be done. It's a little like the time standard. The world time standard is a very egalitarian thing. It turns out that every nation is allowed to contribute to the international time standard. But the deal is your contribution is in proportion to the accuracy of your local time standard. So, if you have a very accurate local time standard, you get to contribute a larger fraction of the international time standard than if you have just a very weak one. And the result is a very good time standard. And a large fraction of it comes from the U.S. And a large fraction comes from the U.K. And other countries contribute, but only in proportion to the accuracy of their national time standard. And that's undemocratic. But it's scientifically correct. So, perhaps, that could apply to a political system in which you have to take knowledge tests in order to vote, or in which votes are tied to taxes. Why is it that we have one man/one vote? Why don't we have one dollar in taxes/one vote.

**Brock:** We do.

**Sutherland:** This is how corporations work, right? You put more money into the corporation, you get a bigger vote. Okay, that's a perfectly good system. Everybody can understand it. And it works extremely well. Okay? And perhaps, taxes should be voluntary. And in fact, the amount of tax that you pay allows you to have more votes.

  

**Weber:** So, you're saying from each according to his abilities?

**Sutherland:** Oh, I'm not saying that at all. I'm saying for political power, it should be money. That's an approach that one could take. I'm not suggesting it's the right approach. I'm just saying that there are many approaches one could take, not necessarily the one that we have. The one that we have is extremely effective. And part of the reason that it's extremely effective is it's been in place for two hundred years. And the population accepts it. And so, one can have a change of power in the federal government without a shot being fired. Okay, Nixon got kicked out. We had a whole new thing happened, right? Nobody was killed. There was no coup d'état. There was no-- nothing that wasn't by the rules. That's amazing. And it's very powerful. And so, I believe in the system we've got because we've got it, not because I think if you started ab initio it's the one you would want to set up. But somehow, the founding fathers agreed on something. And it wasn't easy. I believe that the constitutional convention had plenty of controversy in it. They didn't get it right right off. They had to have amendments to get it right. Some of the amendments were even bad and got taken out. And but the process has the conviction of the population. And that's the most important thing. What's the right programming language to use? It's the one you know.

**Weber:** And the last question for me, talk about meeting Danny Cohen and any good stories about him.

**Sutherland:** Oh, Danny Cohen was a source of many good stories. He came into my office wanting to take my computer graphics course. And I said no. But he didn't hear that, or he didn't believe it or something. He came anyhow. So, that's how I met Danny Cohen. And there was-- do you remember the Six Day War? Israel and the Arabs got into some kind of altercation. And Danny was Israeli. And so, he signed up to go back to Israel. But he didn't get back in time for the war. The war was over before he got back. And he was quite philosophical about that. He said, "They took pilots and doctors first." He says, "I wasn't a pilot. I wasn't a doctor. So, of course, they took me last." He was very good about credit. He said, "There's plenty of credit to go around." He said, "Don't fuss about getting the credit for any particular thing. On average, it'll all work out. There's plenty of things that you'll get credit for. Accept the ones that you get credit for, and don't bemoan the ones that you didn't get credit for." And I think that attitude is quite a good attitude. The trouble with Danny, of course, was you could never tell what he was going to do next. And that's, in some sense, a strength and, in some sense, a weakness. I heard Tony Oettinger say something about Marvin Minsky which I'll never forget. He said, "Marvin Minsky never does what he says and consequently, isn't nearly as bad as he sounds." Now, I mean Marvin Minsky's a terrific guy. And he has many bright ideas. He had many ide-- he's dead now, but had many bright ideas. I've got the greatest admiration for him. But it was hard to predict from what he said what he would actually do. I think Danny had that in spades. So, you had to kind of tolerate the noise that Danny generated in order to believe in him. And fortunately, there were some people who would believe in him. And so, he was able to accomplish quite a lot. I have to tell you another Danny story. So, my daughter Juliet worked for Danny at ISI, which was, at that time, in Marina del Rey. So, I arrive at L.A. Airport without a vehicle parked there. So, I was going to take a taxi to my home in Santa Monica, which was right past ISI, about equidistant on the other side of ISI from LAX. So, I thought maybe Juliet has the car at ISI, and I could take a taxi only half way and then ride home with her. So, I called her on the phone. I called her up, and the phone rang. And Danny answered it instead of Juliet. So, I said-- he said, “Are you calling your daughter Juliet or my employee?” I said, “Well, I’m calling Juliet.” He said, “Parents of employees are not permitted to call,” and hung up. I had to call back to get Juliet.

**Sproull:** So, we’re now actually going to turn-- get back in the more or less the flow we were in before. And the next sort of general topic that we wanted to talk about was testing. And for this I’m going to--

**Sutherland:** Was testing?

**Sproull:** Testing.

**Sutherland:** Testing.

**Sproull:** So, let me start this off--

**Sutherland:** Testing. Testing.

**Waldo:** Let me start this off. The-- almost the religion now in Silicon Valley is that testing is something that you don’t have time to do. But I always remember you testing things very, very carefully. Yeah, you didn’t sort of throw things-- you always had a test plan in mind about everything you did. So, can you talk a little bit about why you were what would now be considered a fanatic on tests?

**Sutherland:** I have no recollection of being a fanatic on testing at all.

**Waldo:** You do it.

**Sutherland:** I have no recollection.

**Waldo:** You do it.

**Sutherland:** Well, that may be true.

**Waldo:** That makes you a fanatic.

  

**Sutherland:** But I have no recollection of being a fanatic on testing.

**Waldo:** But you test. You find out things about--

**Sutherland:** The most impressive piece of testing that I think I ever saw was when my father took me to the Denver Test Laboratory of the Bureau of Reclamation, or somebody, who was testing the strength of concrete. And they wanted to test the strength of concrete pillars, which were like three feet in diameter. So, they wanted to crush these to failure. Now, a concrete pillar three feet in diameter is a pretty strong item. So, they had presses, which were two or three stories high that would squeeze these bloody things until they broke. Now, that's testing. I mean that's really testing, right? And the remarkable thing I remember is they always broke off on a forty-five-degree angle. It was years later I learned out why.

**Sproull:** Okay, why?

**Sutherland:** The maximum shear strength.  Maximum shear force is forty-five.

**Sproull:** Is forty-five, okay.

**Sutherland:** And you always break off at a forty-five-degree angle. The top piece slides off the bottom. And I remember seeing that and noting that. I didn't learn until many years later in a strength of materials class why that was. But that's my first experience with testing with a big T, capital T. Now, a part of Marly Roncken's paper in Async 2015, which you should read, is that in the title it says, "And testing." And one of the tough things about self-timed systems is how do you test them. You can't speed the clock up and see 'til-- when they fail. You can reduce the power supply, but they still work. They just work slower. And how do you test them? And an important part of her notion is you can't test it if you can't stop it. And so, she devised this notion, which we call MrGO, which is a circuit that is part of every acting element, which is giving permission to say, "You are allowed to go or not." And it has a MrGO circuit in it, which will or will not give it the go signal or the stop signal. And you can set the go and stop signals by means of the control-- the external control mechanism. So, you can now isolate a piece of the system to test it. So, for example, this is a real example. The first time I ever used that was we have a counter, which is supposed to count data elements as they go past. So, if one goes past, it should count one. If two go past, it would count two. How do we test it? I mean data elements go past at a hundred picoseconds per stage of advance rate. And how are you going to test something that takes less than a hundred picoseconds? You can use a sampling oscilloscope, but only if it's a repetitive event. How do you test a single event that happened there? So, I had written the test code. We've had the Weaver test under-- the Weaver chip under test. And I said I can write a test that actually tests this now. So, this is a piece of first in, first out data flow pipeline. One of the stages has a counter on it. So, we tell the stage two stages before the counter, "You can't act. You're not allowed to go." And three or four stages after the counter, we have one that's not allowed to go. So, now we have an isolated piece of railroad track, if you will, where the train cannot escape. And behind this blockade, this red light, this stoplight that prevents you from going on the freeway too soon, we have a queue of things that we're going to release, say a queue of two. And now, we release this barrier and say, "Okay, you can go now." Those two guys go down to the next barrier where they stop. The counter better have counted to two. And since you can read the counter before and aft, you can see whether it did, in fact, count to two. And this entire experiment takes only a few hundred picoseconds. And I did this experiment. And I read out the answers. And holy smoke, I have observed a single hundred picosecond timescale event that worked: one, just one. Now admittedly, it took months and months to get the chip ready. And it took weeks to write the test code. And it took milliseconds to get the test loaded into the machine. But when you said go, two hundred picoseconds later, the test was all over. And you could see that it passed. Okay, and that's a very important contribution that that paper of Roncken's makes is it says, "You can't test it if you can't stop it." And so, there's a circuit called MrGO in each and every action item in the Weaver that in fact-- in fact--

**Brock:** My bad.

**Sutherland:** Allows that testing to happen. And it's worth its weight in gold.

**Waldo:** But this shows a-- you put a lot of work into allowing you the ability to test.

**Sutherland:** Oh, you bet.

**Waldo:** So, this is different than some people who will just do--

**Sutherland:** I think all modern chips today have what's called built-in test. You have to build some stuff into the chip to allow you to observe that it is working properly. And it takes a whole bunch of different forms. And the cost of testing is a central cost of manufacturing chips. You can't sell it if you can't test it. And the machines for testing chips are enormous and expensive. And the time that a chip spends on a test machine is part of the cost of the chip.

**Sproull:** But I think what Jim's driving at is: that's true for commercial chips not for your asynchronous experiments until MrGO, especially. The chips you designed and built were designed to do a particular experiment.

**Sutherland:** Yes.

**Sproull:** And part of that design included how you were going to carry out the rest of the experiment.

  

**Sutherland:** Yes, of course.

**Sproull:** After the chip came back.

**Sutherland:** Of course.

**Sproull:** And there was often a tremendous amount of thought in test programs, in boards, for example, to mount the chips so they could be run at speed. A substantial amount of the project was designing and executing the test.

**Sutherland:** Yes, that's certainly true.

**Sproull:** This was not a casual let's build a chip and fuss around with it kind of pattern.

**Sutherland:** Well, we're building this thing now we call Hexnet, which will be a chip if we get to build it. And an important part of the design is how is it tested, how do you know it's working. And of course, every action item in that chip will have MrGO in it. So, we'll be able to isolate pieces and test the pieces separately to see that they're doing what they're supposed to do. But the overall test plan says, "How do we know how fast it's going?" So, we have to build counters in that count how many things go by in what period of time. And there's a fair number of things that we have to build in in order to make the thing testable. And making it testable is a part of the experiment. Of course, it's part of the experiment. Otherwise, you couldn't, with a straight face, say, "Yes, it works. And I know it works because." You could say, "It works. Now, you show that it doesn't." Right? This is kind of the religious approach.

**Waldo:** Maybe this is the difference between hardware and software.

**Sutherland:** Yes?

**Waldo:** Where often in software the second approach has been taken.

**Sutherland:** I see. Well, okay. That's fine.

**Waldo:** No, no it's not. But--

**Sutherland:** Now, test is too expensive to get the bugs out of the design. And the contract we have with DARPA is, in fact, to learn how to do formal verification for self-timed designs. The formal verification tools that exist today depend on the synchronous paradigm. They all say, "I'm now in this state. Can I prove that the next state the machine will get into will be the one that I expect?" And so, I do elaborate mathematical proofs. I mean the typical one is I have a IEEE floating point adder. Okay, and I want to prove that, in fact, it adds. So, I look at-- I write an algebraic question equation for every bit of the output as a function of algebraic names for the input bits, which I can do by looking at the circuit diagram. And I do a similar thing for what I know division is because division has a mathematical meaning. So, I can do the same thing for a theoretically correct division. And then I compare those two algebraic results. And they better be the same. And if they are the same, I can be assured that, for every possible input, I get the desired output not just for the examples that I run through the actual test. And that's formal verification. It's a very potent idea. It's applying algebra instead of arithmetic to the problem of verification of the design. Now, the art of that today depends on clocking. It depends on having this fiction that things distributed in space happen at the same time. And so, the contract we have for DARPA is really to figure out how to do that for self-timed systems. And the Hexnet, and whatever other examples that we choose to use are the grist for the mill to learn how to make those proofs. And it appears possible, but it's not easy.

**Waldo:** So, what do you think is the balance between what you put into a design for test and what you put in for-- that's not just to test it? How do you decide what the effort is here?

**Sutherland:** Another way to ask this question is, "How much design effort do you put in before building a prototype and trying it?" Well, it depends on what the expense of building a prototype is. And my notion is you want to put enough design effort in to warrant spending the money on the prototype. But you don't want to put in a hundred times the design effort of the cost of the prototype because then the prototype-- if the prototype went bad, you'd be losing one percent of your-- somehow there's a balance between how much design effort and how much the cost of a prototype is. In my shop at home, how much effort do I put into a test jig to test that the hole is going to be in the right place or whatever it is? Well, depends on the cost of the operation that's going on.

**Waldo:** So, you talked a lot about this having to do with the hardware you've been building. When you started off doing more software, things like Sketchpad, did you have the same sort of building the test framework for what you were doing there?

**Sutherland:** Well, the beauty of graphics is you can kind of see whether it's working. And the visual test is very potent. Now, there were lots of times when things that you saw were wrong were obviously wrong. But how do you pin down where the error is? That's the art of debugging. How much infrastructure do you put in to determine where the bug is? And that's, I think, a very individual question. Now, your comment is that software systems don't get tested enough. I think that was the implication of your statement.

**Waldo:** It may have been an implication, yes.

  

**Sutherland:** Software systems are notoriously full of bugs. And you allow the users to use them until the users find the bugs and complain about them. That's one way of testing is let the users do it. I don't know what you can do for software to help. It would be wonderful if you could do formal verification on programs to demonstrate that what they do is what you expect. But the hard part of all of these things is what do you expect. Is this thing actually working? Is this a bug or a feature? Now, who was telling me that they saw a VW bug with the license plate, Feature?

<laughter>

**Waldo:** That must have been around here.

**Sutherland:** The perfect license plate for a VW bug, Feature. Now, it's only a bug if it doesn't do what you expected. And if you want to do that formally, you've got to write down what you expected. And one of the reasons I think that DARPA is interested in what we proposed is that they're concerned that the systems that DoD buys are specified in a thousand pages of English text. There's no way of proving whether an object meets that specification. There's no way of proving what the specification is. People-- perfectly honest people can read the specification and come up with different ideas. So, I think what DoD would really like is a way to specify with mathematically correct text of some kind-- or something, what the system is actually supposed to do. So, when you go to rebuild it fifteen years later and you can't buy the chips that it was built in originally, you can be assured that what you get now is the same thing that you got before and doesn't have some other quirks that the new users are going to have to learn. I mean the worst part of using shrink-wrapped software is they keep changing the interface. So, I've learned-- I've gotten really good at using version X. And now they give me version X plus one and it's just enough different that the things that I used to know and used to love are gone. So, how can I be assured that version X plus one is the same as version X? Only if the specification is simple enough and mathematical enough that I can prove things about it. And I think that's what DoD is really about in this contract that they've given us. But it's not an easy thing. And there's a cultural piece to that, which is what should be the specification for a complicated piece of equipment.

**Waldo:** Yeah, it's a--

**Sutherland:** You look at the architectural plans for a building, and the plans do not specify everything. They always say something about good standard practice. These are plasterboard walls assembled to good standard practice. Everybody who puts up plasterboard walls professionally knows what that means. But maybe there's variation. Maybe some of them are better than others. It's not clear to me that it is possible to specify everything or even that writing such a specification is less expensive than making the thing.

**Brock:** Yeah, actually-- yeah, precisely.

  

**Waldo:** There are certainly those who would claim that, for a lot of software, the specification is the source code, which I always find problematic because that means you can't have a bug. But it's a-- because it's always-- it always functions to specification.

**Sutherland:** There are some places where you can show that, if you do addition, there ain't no way to do something simpler that will check if the addition is right.

**Waldo:** Yes.

**Sutherland:** Casting out nines, that tells you something about whether the addition is right but not everything. And I think there's a fundamental problem. There are things that are just hard. And getting things right is hard in part because what does "right" mean. There's always a specification in somebody's head. This is what I expected it to do. But if that specification isn't clearly written down, who else can agree?

**Sproull:** Yeah. All right, should we move on to--

**Waldo:** Yeah-- yeah, so-- can I-- speaking of non-sequiturs, we're going to move to something different. So, one of the things that I find still most interesting, perhaps distressing, when I look at Sketchpad is how much seemingly new stuff you put in there. Alan Kay talks about you having invented graphics object oriented programming, constraint based programming, and half a dozen other things. So, were you aware that you were doing all of these new things, or were you just doing this?

**Sutherland:** Oh, come on, Waldo. Of course, I wasn't aware. I wasn't aware of anything, right? I was just trying to get interesting stuff to work.

**Waldo:** Well, but wait a second, Sutherland. Don't give me that, because there were things like recursive traversals of the data structures that you had, the abstractions of some of the graphic operations on different kind of things, so circles and lines, that was really different than how things were being done. Well, things weren't being-- they were really different than how things were being done ten years after Sketchpad. So, was this just because it seemed to make sense at the time? Or was it a constraint of the machine that you were working on? What led to this sort of design?

**Sutherland:** You know, recursion was a relatively new idea. And I remember having endless discussions with my fellow graduate students. Len Kleinrock was an office mate of mine. And Larry Roberts was an office mate of mine. And somebody else was there. I forget. And we had endless discussions about pushdown stacks. How did a pushdown stack work? TX-2, I think like many other machines of the time-- Bert, you can correct me if I'm wrong here, I don't believe it used a pushdown stack for subroutine returns. I think it copied the address that it was to return to into a local register. So, it was not position independent code, and so on, which was typical of the time. PDP-1 certainly did the same way. When you did a subroutine call, the address-- the calling address was put somewhere where the subroutine could store it somehow to return to it. There was a jump and return instruction of some kind. But it was not a stack. And stacks were quite a new idea. And we had endless discussions of how they would work and what they were and whether that was a good idea or not and so on. So, the notion of recursion and data structures was in the air and was fresh. And it seemed the obvious way to do things. And I subsequently realized that in a structured graphic system, there are two different kinds of stacks. There’s a stack for the subroutines that you call, some of which may put structural information from the objects that you’re dealing with into an object stack, so that when you get finished dealing with this object, you can go back to the object stack and figure out what the next thing to work on is, as different from the next subroutine to call. And it’s hard to intermingle those two stacks. Now, those were new ideas at the time. They weren’t necessarily my ideas, but they were in the air. There was discussion of it. And I simply did the things that seemed the right thing to do to make the code as simple as possible.

**Sproull:** So, while we’re on the stacks, if you wanted to pass arguments to subroutines in TX-2 and Sketchpad, how did you do that, especially if it was more than would fit in the general registers available? Or that never happened because subroutines took very few arguments?

**Sutherland:** Well, TX-2 was an accumulator machine. It had a big accumulator. And the typical subroutine call put some information in the accumulator or put it in some global register somewhere and then called the subroutine. It was not-- the idea of a stack frame was not there.

**Sproull:** Right, okay.

**Sutherland:** Now, was that an omission of Wes Clark’s? I suppose it was. I mean later machines were more sophisticated in this way. But that wasn’t TX-2. TX-2 was a-- in many ways a very standard, state of the art machine of that time. Where it was unique and weird was that it had thirty-two program counters. And so, it had concurrency built in to the fabric of the machine. And these program counters basically competed for the use of the arithmetic elements of the machine.

**Sproull:** So, Ivan, I think just to venture a guess, you mentioned before that the subroutine call linkage schemes at the time stored return addresses. That meant they were not-- and that was linked to the fact that recursion was not yet all that popular. You would never store a return address if you were trying to make recursion easily.

**Sutherland:** It would be impossible to make--

**Sproull:** Right. So, the notion of reentrant code had not-- was not really around.

  

**Sutherland:** That's right.

**Sproull:** Because you don't need reentrant code until you have recursion or some other thing like concurrent processing. And so, the stack frame isn't going to happen until you have reentrant-- until you want reentrant code. So, it's just-- this was an era before any of the final ramifications of recursion and reentrance had percolated?

**Sutherland:** That's correct.

**Sproull:** Yeah.

**Waldo:** So, later-- Bob actually sent me a package that you had written I guess when you were at Caltech in Simula.

**Sutherland:** In what?

**Waldo:** Simula.

**Sutherland:** Yes.

**Waldo:** So, how did you come across Simula?

**Sutherland:** There was a graduate student who said, "You ought to know about Simula." And he started telling me about Simula. And that seemed a bit familiar to me. And so-- and it had the right flavor of stuff. And so, I used it. I wrote quite a lot of Simula code. It was a wonderful language. But it was a graduate-- I forget his name. I could probably regenerate it.

**Sproull:** Was it Tony Barton?

**Sutherland:** No. No, it wasn't Tony Barton.

**Sproull:** Okay.

**Sutherland:** I don't remember the graduate student's name, but that's what graduate students are for. Graduate education is where the graduate students teach the faculty.

  

**Sproull:** And by good fortune--

**Sutherland:** This is a fine example of graduate education.

**Sproull:** And by extraordinary good fortune there happened to be a Simula compiler for the PDP-10.

**Sutherland:** I don't-- I guess so. Why was that?

**Sproull:** I don't know, but you used it a lot.

**Sutherland:** Yeah. I used it a lot, yeah. I don't know why, but yes indeed. And then later on, I had learned Java. I learned Java from somebody at Sun.

**Waldo:** Either it was--

**Sutherland:** Who was--

**Waldo:** Ted Goldstein, I think.

**Sutherland:** What's that? Ted Goldstein.

**Waldo:** Ted Goldstein, he certainly takes credit for it.

**Sutherland:** He absolutely-- he taught me Java.

**Waldo:** Good.

**Sutherland:** And he said, "You asked the damndest questions." He said, "You asked questions about the syntax. You don't ask questions about the underlying structure." And that's right. I understood the underlying structure from having used Simula, having done similar structures in Sketchpad. So, I didn't need to ask those questions.

**Waldo:** Did you have much interaction with the formation of Java?

  

**Sutherland:** No, I had none. I think Java may have actually been named by my older brother. But you'd have to ask him about that.

**Waldo:** Well, that is steeped in legend.

**Sutherland:** I'm sorry.

**Waldo:** The naming of that language is steeped in legend. But we will ask--

**Sutherland:** Steeped in legend?

**Waldo:** In legend, yes.

**Sutherland:** I see, well--

**Waldo:** But--

**Sproull:** So, did you write a significant amount of Java as well?

**Sutherland:** Say it again?

**Sproull:** Did you write a significant amount of Java as well as Simula?

**Sutherland:** Oh, you bet.

**Sproull:** What kinds of things?

**Sutherland:** Well, all of the code associated with the Weaver is written in Java. Now, all the test code is written in Java. The simulation code was written in Java. What else did I write in Java? Oh, we wrote-- at Sun, we wrote a bunch of special purpose simulators of various kinds were all written in Java.

**Waldo:** So, you're still using Java as your language.

**Sutherland:** It is the language of choice for me. The right language to use is the one you know. And so, it's the language of choice. There is a high school student who has applied to us for an internship. We're not able to pay her for visa reasons. But we will provide her the internship. She seems to be a very bright lady. And she lists under the things that she can do Java. She's learned Java somewhere and is not yet polluted with C++ or other competitive schemes. And so, I think what I will give her as an assignment is to take over the Java code associated with the Weaver. Why not? There's-- what a lesson to a high school aged person who's just learned Java to inherit probably five thousand, maybe more, maybe between five and ten thousand lines of reasonably well-written Java to learn, understand, and try and figure out how to improve. It's a role model, a teaching activity, par excellence. So-- and we'll get some good out of it I'm pretty sure.

**Waldo:** So, moving from the sort of languages-- moving away from the languages and more to how you build a research group because you've had a succession of them that have all been in various ways between quite successful to spectacularly successful. How do you build these? This is a question everybody asks all the time. How do you put together a good group? You seem to have a great track record. What do you do?

**Sutherland:** Well, the most important thing is to have very smart friends and relatives. None of these things are my fault or my creations. They're the creations of groups of people that have been assembled. And one of the good fortunes of my life is to meet and work closely with and like a large number of incredibly bright people. I don't know why it's happened that way. Okay? I'm a grumpy old man. But people seem to tolerate me for whatever reason. Maybe I'm just bumbling enough that they say, "Oh, this guy needs some help. Let's see if we could help him," right? I love the story about the fable about stone soup. Remember the stone soup fable? I used this at Austek. I was the president of Austek. And I said, "Now look, we need to make this company successful. I clearly can't do that myself." So, I called the key guys together. And I said, "We've got stone soup. I have the stones. You guys have to provide the vegetables, and the chicken, and all the other stuff to make this a successful deal. And I will try and placate the shareholders. But you guys have got to get to work." And so that's what we did. They gave me a T-shirt that said "stone soup" on the front. And in some sense, that fable is the secret to success of every research group, that somebody has enough vision to say, "This is the direction we're going to go." I see the direction. I can articulate the direction clearly enough that you can understand it. And then, let's get together and make it happen. Now, I actually took a formal course in leadership. Not many people that I know have ever actually had a formal course in leadership. I took a formal course in leadership as part of my Reserve Officer Training Corps education at Carnegie Tech. And I remember the lieutenant who taught it. His name was Jarvis. Lieutenant Jarvis was one of the instructors. And he taught this course in leadership. The military, quite correctly, thinks that leadership is important. It wins wars. It's a very important property. They haven't a clue how to train it, how to teach it. I'm not sure they even have a clue as to how to identify it. Okay, but they try. And you've got to give them credit for at least trying. So, this course we studied great military leaders. And the one thing that we learned that they all have in common is they allow their humanity to show. And so, Eisenhower is famous for a grin. And he's famous for the Eisenhower jacket. And he's a human being. And it's clear he's a human being. And it was clear to everybody that he was. Churchill smoked a cigar, famous for his cigar and his victory sign. But he was a human being. And people knew that he was a human being. And he allowed his humanity to show. So, that's one characteristic that leadership has. And I've allowed my humanity to show. The people who've worked with me know that I'm a person. Sometimes, I'm grumpy, and sometimes, I'm not. Sometimes, I'm happy. And sometimes, I'm sad. And I'm not afraid to allow the people that work for me to see that. And that's important. I think a second aspect of leadership that the military described is articulateness. And I think over my career, I've selected for articulateness amongst the people that I've counted as friends. My mother, bless her soul, was big on speaking with correct diction and correct grammar. And I had lesson, after lesson, after lesson. I had to memorize poetry. And I had to give speeches with correct enunciation and correct grammar and so on. And thank you, Mom. That has been a huge asset over my career. The ability to express in English what I'm thinking I think is important to let other people join the trip. How are they going to join up if they don't know what the trip is? Well, I don't know what the trip is when we start. But you try and articulate it. Maybe you even try to write it down. And you at least you have some notion of where you're going. And I've sort of-- I got asked the question by the Harvard Business School, "Why was the University of Utah research project so successful?" The question really was how can we duplicate it.

**Waldo:** Oh yeah.

**Sutherland:** But they asked the question in the polite way, "Why was it so successful?" And I've decided there are sort of three parts to a successful research project. And the first one is you need something worthy to work on. A vaccine for Polio is a fine, worthy thing to work on. It's been done now. So, we don't need to do that again. But putting a man on the moon and bringing him back safely, that's a worthy goal that everybody can understand what the goal is. And it's clearly a worthy goal. Putting a man on Mars and bringing him back is worthy goal today. It's much harder. But it's conceivable that we could do it. And so, it's a target to shoot for. Matter transportation, I want to send a man to the center of the Sun, find out what it's like and bring him back again, that's not a worthy goal. It's unlikely to happen. So, a worthy technical goal is important. What we had at Utah was the worthy technical goal of making realistic pictures. We want to make pictures of three dimensional objects that look like three dimensional objects. And at the time, the power of computers was not adequate to allow that to happen. So, we had to struggle to do it in spite having inadequate computing. It was just barely possible to make some pretty good ones at that time. And so, we did. So, those are the-- the first ingredient is a worthy technical goal. The second ingredient that you need for a research project is money. And at Sun labs, we had the good grace of Sun investing money in research. And so, we had a fairly unfettered source of funds. We didn't have to spend time raising money and so on. So, we had a source of funds that would allow us to do the things that we wanted to do. Money is the second ingredient. And the third ingredient for successful research, which is the one that is hardest to identify, is leadership. And I think the reason that the University of Utah graphics program was as successful as it was is a man named David Evans. David Evans was as honest as the day is long. If he said he was going to do something, he did it. And everybody trusted him. And he was a very personable fellow. He's a very likeable man. He had seven children. He was a Boy Scout leader. He would take kids camping. Okay, he was a pillar of the Mormon Church. He was well known in the community as an honest and able guy. And so, he was able to get people to follow the direction that he led. And he said, "We're going to do realistic graphics." He got DARPA to put money in. And so, we had the three ingredients, a worthy technical component. We had money. And we had leadership. And around Dave Evans, there collected a collection of people that was-- amazing collection of people. Tom Stockham was there. He figured out that you could reconstruct Caruso by listening to what the musical instruments sounded like through the bad recordings and unwinding that. And then you unwound Caruso at the same time. And lo and behold, Caruso had a pretty great voice. And there were a number of people there. He recruited me. And there were a number of people there that were able to flesh out the idea that he had into something valuable. But I think the leadership is everything. Now, if I've been able to be in research groups that have been successful, it's in part because there was some leadership, partly provided by me, but in large part provided by other people. Now, I went to Caltech. I was a green behind the ears professor. How did I get to Caltech? Well, that's an interesting story in itself, right? Caltech set up a Search Committee to find out who could get them into the computer business. And the Chairman of that committee, or leading element of that committee was John Pierce. And John Pierce and Dave Evans had a conversation about whether Caltech should hire me. And I think Dave Evans convinced John Pierce that they should, and so they did! And they said, "Make a computer science department happen." That was my charter. And I had exactly two faculty. It was me and two other faculty, right, to build on. And when I got to Caltech, there was this guy called Bob Cannon, who was the Dean of Engineering. He was terrific! I mean, the best deanship story I can tell has to do with our volleyball game. <laughter> Right? We played volleyball at lunchtime or in the late afternoons, we'd play volleyball in the parking lot. And we had a couple of, you know, pieces of pipe stuck in concrete in rubber tires, that we'd roll into position to hold the net. And then we marked out with chalk or something, we marked out the court, and we played volleyball. Now the next building over was Buildings and Grounds. And sometimes their cars parked in that parking lot would be in the way. And so we'd just pick them up and move them! <laughter> And sometimes the volleyball would go astray and it would leave a dust mark on a car. Never damaged the car, but it leaves a dust mark. So you know a volleyball had hit the car. Now one day there appeared in the place where we played volleyball a sign, which said, "No ball playing in the parking lot," signed, "Buildings and Grounds." Well, the very next day there was a guy from Buildings and Grounds out painting a stripe for our volleyball court in the parking lot. Now that's deanship at its finest. So I went to Cannon and I said, "What in the world did you do?!" Said, well, he said, "I took the head of Buildings and Grounds to lunch and I explained to him who works for who! So I think that volleyball game is an important part of your research program." And indeed he was right! It was the place where the faculty and the graduate students got to know each other as people, as opposed to as faculty and graduate students. And it contributed enormously to the research. And somehow that people equation is very important. Now when I went to DARPA, had it already been done. I mean, Licklider did something at the start of the IPTO at DARPA that was magic! He got the best people in the country involved. And you know, here I was 26-years-old, not a clue in the world as to what I should do, and here were this collection of wonderful people whom I could draw on for counsel. I remember meeting-- I called a meeting, because ASCII hadn't been invented yet. So I got all of the principal investigators together and said, "Wouldn't it be nice if we had a standard character set?" And Al Perlis put the whole thing in perspective by pounding his hand on the table and saying, "I'm in favor of a standard character set! Ours is available!" <laughter> Now, that in a nutshell tells you the problem about a standard character set. We never achieved anything in that, right? But finally ASCII took over the world, and that's a good thing. Somehow the standards bodies grind exceedingly slowly, but by and large do pretty well. The standards that the volunteer organizations, like the IEEE, and so on come out with tend to be pretty good, because they're done by people who want to use them. So that's my piece on research. Three things it takes.

**Sproull:** So, Ivan, okay, but I want to-- so your three things are good. We've heard them before. I think they're superb, but I think there are things about your groups that go beyond that. And one quote that I remember from an earlier discussion on this is, "Cherish and develop young minds." And so you-- one of the things that I think you do very well at is bringing very junior people in who may not have much experience or confidence or anything, and not only are they able to join and function in the group, but you give them room to become a specialist at something, to become unique, or the go-to person would be the Sun cliché, about a certain aspect of what's going on. So you don't crowd them out by deciding what works and what doesn't, or what's most important or least important, or what needs to be done tomorrow. You know, the group has room in which to grow.

**Sutherland:** Well, thank you. I take that as a compliment.

**Sproull:** Absolutely!

**Sutherland:** Take that as a compliment. Thank you.

**Sproull:** And I think Ian, for example, is a perfect example of somebody-- not the junior-est person ever, but who grew fabulously in your group!

**Sutherland:** What about this guy called Bob Sproull.

**Sproull:** No, no, we don't talk about him.

**Sutherland:** Whom I've watched from when he was almost a teenager until he now become part of the Council of the National Academy of Engineering.

**Sproull:** Ah, whatever.

**Sutherland:** Right?

**Waldo:** You're good at picking people.

**Sutherland:** Say it again.

  

**Waldo:** You seem to have a pretty good track record picking people.

**Sutherland:** Well, you know, the people that you know that I've picked have demonstrated that they're pretty good. The people that I've picked that aren't demonstrated to be pretty good, you just don't know who they are. <laughter> You don't know how many of them there are, you don't know anything about them. So you can't say I have a pretty good track record. What you know is there are some examples that were pretty good. <laughter> And I agree, there's many examples that are pretty good.

**Waldo:** Well, I think you have a pretty good track record.

**Sutherland:** We've got a couple of young people involved in the ARC now that are pretty good. And what is it about them that's good? Is they're interesting people. Dave Evans said, "You know, when you admit a graduate student to a graduate program, you should always admit some outliers." And Alan Kay was the typical example of a Dave Evans outlier. <laughter> Okay? Alan Kay was clearly very clever and very bright, but he didn't have the kind of track record that would get him into a graduate school without any question. But he was clearly a good pick! I didn't do that. Dave did that. And I've just found some people more interesting than others. And there's a young man working for us now who wandered into the office when Marly and I first opened the ARC. And he just wandered in and said, "Hello," and started talking with us. He turned out to be a very interesting guy! He'd been an apprentice to a printer at one time. And he was a buff about locomotives. So he took me one time to the Roundhouse, where the locomotives that are kept by the steam-powered clubs in Portland are kept. And there was this ring of steel about six feet in diameter leaning against the wall. I said, "What's that?" He says, "That's a tire for locomotive." Now think about his, a locomotive has wheels that are about six feet in diameter. What are they made of? Cast iron, of course! What else would you make a wheel that size of, then? Cast iron isn't a very good wear material. So you have to put a tire around the outside of the cast iron, as the wear surface to run on the rails. And this was a locomotive tire. You know? So how do you get it on? Right? How do you put the tire on? Well, you heat it.

**Sproull:** You heat it up! Yep.

**Sutherland:** And it expands, and you slide it on, you let it cool. That's not the interesting question. The interesting question, how do you get it off? Okay? Now, I'm not sure I know the answer to that. <laughter> But you know, and so this was just a very interesting guy. And he came in six or eight months ago and said, "I'm coming back to school, I want to finish my degree," and, you know, can we get involved? I said, "Hell, yes, we can get involved!" And so he's now working with us. And I'm so glad we spent the time. He's just an interesting guy. He's the outlier, but he'll be a key element to what we do. And I don't know, Glen Fleck once said to me, "You ought to try and identify in everything you do what's interesting. What is it that you would talk about with your brother at the dinner table? What is interesting enough to use as dinner table conversation?" And I remember him saying that. And there are always things that you're doing that are interesting in their own right for whatever reason. And I think I've always liked to work with people who are interesting people, who have some characteristics that are a sense of humor, a sense of irony, a variety of tasks that they've undertaken, a variety of things that they've done that make them interesting people. And interesting people produce interesting results. But there are people that I don't find interesting who produce good results anyhow. Bob Barton worked at Utah, he was a faculty member at Utah, and he was quite an amazing guy! I never understood what he said. I couldn't understand what he was about, but he had been key in the Burroughs Corporation in making interesting things happen there. That's why Dave hired him. I wouldn't have hired him, because I didn't find him interesting. But he turned out to be important, and made important contributions. And so I don't think that the characteristic of being interesting is all-encompassing. And I don't pretend to be able to pick all the winners. What I can say is I pick some people that I find interesting, and they've been successful.

**Sproull:** So one last question maybe about the research group. So junior people sometimes I think have trouble-- might have trouble joining a group of yours for fear that the great man is the one who's viewed as running the group, and all the ideas are his, and he's going to take the credit, etcetera, etcetera. And yet, I don't see that any of your colleagues-- and I use that term now as distinct from junior people-- have ever-- that's never a problem. It's not been a problem with Jo, or any of the people who have joined the group and worked alongside you easily.

**Sutherland:** I don't know. I'm not the sole source of ideas. I mean, this is one of the issues about where does research happen. And I think Bill Joy had it right. He said, "New developments will happen in the computer industry. Most of them will happen elsewhere." We don't have any monopoly on brains. I don't have any monopoly on brains. Okay, I think in a certain way, and I have some good ideas sometimes, but other people's ideas are as good or maybe better than mine. And picking up on those ideas and running with them is important. I think it's also important to try and give credit where it's due. One thing I do about reducing people's fear, is I have a very unusual first name. My given name is Ivan, which in this part of the world is rare. And I learned from Ed Berkeley-- Ed Berkeley insisted on being called Ed, not Mr. Berkeley, when I was in junior high school, and Bert was in high school. He always wanted to be called Ed. Danny Cohen came to our house when my children were in grade school, or before grade school even. They were properly brought up children, so they knew he was a generation older than they, so they called him Mr. Cohen. And he took to calling them Miss Sutherland and Master Sutherland, which they didn't like! So after a while he became Danny. And I took that to heart. And I've tried to get all of the students and people that I work with to call me Ivan. I write Ivan on the board at the start of a class, whenever I teach a class, say, "This is my name. Please use it." And some of the foreign students have a really hard time with that. They really want to call me Professor Sutherland, and they want to call me Dr. Sutherland. And they don't know what to do, so they call me, "Hey," and that's okay. I keep saying, "My name is Ivan. Why don't you just call me Ivan." And finally they do. And I think that's an egalitarian move that pays off well. Because Ivan is less frightening than Dr. Sutherland, PhD, PhD, PhD, PhD. <laughter> Right? At Portland State, the tradition for faculty members is to put their degree after their-- on the outside of their office. At ARC, we have signs on the outside of the office exhibiting everybody. And one of them has a degree after his name. Only one. Chris Cowan, M.D. This is our way of thumbing our nose at the local custom. <laughter> Somehow research is an egalitarian thing, and that volleyball game was indeed important to our research. Why? Because you don't have time to say "Professor Sutherland." <laughter>

Okay, you have to say, "Hey, Ivan! It's yours!" right? And you have to say it quickly, or you lose the game. And that's an egalitarian thing, which makes the-- which lubricates the research communications.

**Sproull:** So I think another aspect of our leadership style, especially in the async groups, and it relates part to this point that I've tried to make that I think you're superb at selective problems, but the leadership part is partly selecting problems, but it's partly also sustaining the enthusiasm about going for the next piece. About you know, keeping people motivated and optimistic when the chip design isn't going so well, or you know, goodness knows what has happened. And I'd like to use that as segue into the last piece here, which is to talk a bit about "Technology and Courage," and this is, of course, the title of a little monograph you have done, which I think is under-circulated and under-appreciated, which talks a bit about the fact that it's often hard to summon personal courage to do-- to work for long periods of times on these projects, and the tradeoff between making them tractable, and having them be important, difficult enough to be worthy problems-- so perhaps you might say a little bit about all that. I'll give you-- okay.

**Sutherland:** Well, I can define a worthy problem. A worthy problem is one which will sustain the interest of the people working on it for a long period of time. I have a similar definition of wisdom. It's—as a philosopher you may like this-- wisdom is the ability to make decisions in the absence of adequate information that stand the test of time. And it follows that it's not an operational definition. So it doesn't tell you how to exercise wisdom. It tells you how to recognize it after the fact. Too bad it's after the fact, right? But many things are like that. And a worthy problem is worthy if and only if it sustains interest on it for a period of time, after the fact. Now, Bob Sproull said to me one time, "The thing that you do best is to pick the problems." And I've thought about that a great deal since you said that, Bob. And yeah, I think I've picked some pretty good problems. And you know, we've got this Hexnet going now. And it's a worthy problem in the sense that if we can crack it we've got something that may be of considerable value. And yet it seemed easy enough because we simplified some of the things away, seemed easy enough that we can actually do it, and so that makes it, to me, interesting. And I guess you say I've always had the ability to sustain interest in the problems is not true. Because on a number of occasions I've given up on those problems and gone to something else. Okay, I did graphics for a long time, and then Sutherland, Sproull and Schumacher published a paper called "Ten--," no, "A Characterization of Ten Hidden Surface Algorithms." And what we realized was that hidden surface algorithms are just sorting. And once I realized that, I said, "Holy smoke! Do I want to spend the rest of my life working out better and better ways to sort things?" And I think the answer's, "No, that's boring!" So I quit! So yes, I've been able to sustain interest for a long period of time, but then I simply turned my back and walked away. And you know, the sojourn to Pittsburgh and the walking machines was in some sense a turning away from a set of computer problems to a set of mechanical problems, as a sojourn, if you will, or as a sidetrack, as a vacation, as a recreation or something else, that reenergized the interest that I couldn't take. And there's a worthy problem I'm attacking now on a broad front which is: how do we get rid of the synchronous design style? And I can't do it alone, because there's a training problem, there's an educational problem, there's a tool problem. All kinds of problems. But maybe I can show the way, and show some examples of some things that you can do if you do things that way that people will pay attention to. And I gave this talk at the Turing celebration from ACM, and it's published in the ACM, and it's called something about the Tyranny of the Clock. And they-- it says there's many things to overcome, education, you know, tools, all kinds of things. And at the end I said, "My fear is that the people who have courage to do this, do not share my language. That the folk who will have the courage to do this are not on this side of the two great oceans." And that's scary! That's scary. That says, "Unless we can do something important here, this technology may escape from us." And the risk we run, I think, in the advancement of technology is that we will fail to invest adequately with the freedom that it takes to go where the technology leads, rather than to direct things in directions that may be wrong. You know, research has this property that you find the right things to do by allowing a large number of people to, in essence, vote. People who are educated and steeped in what they're doing to vote as to what they are going to do personally. And those votes are critical to getting people to be most productive, and so on. And unless we can do that nationally, I think we run the risk of falling behind. I don't believe that the Chinese are any smarter than the Americans. There's more of them! Hey, there's more of them. That's a risk, okay? But on the other hand, we have a tradition of freedom of thought, freedom of speech and so on, which is a very important tradition, because it allows people to do things that they want to do, rather than that the collective wants them to do. And that's a very potent idea. So maybe we have a leg-up on how to do this. But if we don't do it carefully, we may lose out.

**Sproull:** Right, in fact, the argument you just made is more broadly one for diversity, and research generally. You could argue that that's why it's important to have both academics and industry working on it. Not-- partly they have different objectives, but they certainly have different perspectives and think contributions they can make.

**Sutherland:** Yeah, well, there are three research establishments in the United States. There's the ones that are industrial; there's the ones that are academic; and that there's ones that are research centers, largely government-run. And they all have a role. They all can do different things. And fortunately the people flow back and forth between them. And so there's a cross-pollination which goes on, which is very potent. And if you worked in a government lab, you're never going to work in industry, that would be insanity! You want people who've got government lab experience working in industry, and vice versa. And I think there are parts of the government which appreciate this and understand its value. There's this program in DOD to train rising officers in mid-career in how industry works. And you know, we had some of these people visit at Sun. And Bert became friendly with a number of them, who are now rising military officers, who had learned how industry behaves from the inside, which enables them to-- from their DOD positions-- to work with industry better. And engage industry in solving the problems that industry can solve.

**Sproull:** So just a personal comment, that paper, the paper you actually wrote talks about developing just sort of the a-ha! moment of logical effort and so on when you were in London, and working on the more symmetrical XORs and C-elements and so forth, in which you wrote a first draft of something, as you said, "Trying to articulate the idea." Showed it to me, and I read it, and we had conversations. And then I did some mathematics and extended it a bit more. And then we started writing a book. And then your conv-- your description stops, because that's where it was I think at the time you wrote the paper. What really happened is then we got stuck. You remember that? Nothing happened.

**Sutherland:** It was ten years.

**Sproull:** Well, we got stuck for a variety of reasons. And we got rescued by David Harris.

**Sutherland:** That's right.

**Sproull:** Who ca-- <laughs>-- talk about that.

**Sutherland:** David Harris came along and he had had the same ideas. He had them less well-articulated, I think, than our drafts did. But he had the same ideas. And he said, "Let's write a book about this." And he was a budding assistant arofessor who wanted a book to his name, and so he said, "I'll finish the book." And we had a bunch of notes about that thick. I have some copies of them. And we had taught a class to a number of industrial places about asynchrony, and about logical effort. And so David said, "Well, I'll finish the book for you." And he did! And so it's a fine book, but he deserves a large part of the credit for finishing. Making it into a complete thing.

**Sproull:** Yeah, that was a place where courage failed two of us. <laughs>

**Sutherland:** Well, that was very seldom. I think motivation failed.

**Sproull:** Well, all right.

**Sutherland:** I'm not sure that it was courage.

**Sproull:** But that's-- there were--

**Sutherland:** I don't recall feeling at any time that nobody would care, which is the failure of courage. I felt that a number of times. The paper about "Technology and Courage" itself, okay, I thought, "Who will care about this? This is a bunch of musings of a guy who's done some of this research, but it's not of any value." And so with great trepidation, I finally published the paper. I wouldn't have, except Juliet was on my case. She said, "You got to publish this stuff, daddy." And so finally I did. And a lot of people have said, "This is an important paper to read." So it's a non-technical paper. How can a technical guy like me write a non-technical paper? I don't know. In this discussion I have had a fair number of non-technical things to say. I don't know if they're any good or not. I mean, history would have to say. But there are some places where the view that I've had of what the national scene is makes sense to me in some important ways. And what makes sense to me is that the research establishment in this country is not run by a research czar. It's run by a diverse collection of people who have all kinds of ideas about what the right thing to do is. And so there's no way that any one individual can block progress. There's no veto in the research establishment in the United States. And the lack of a veto is perhaps *the* most important thing.

**Sproull:** Well, let's hope there never is a veto.

**Brock:** I was just curious, and you know, we've talked for a quite a bit this afternoon about problem selection, your ability to choose a problem. You describing it as being able to see through the problem, to maybe some sort of tractability of some kind. And I was just thinking about if your long-standing commitment to the self-timed paradigm, if I may call it that, is that typical? Or atypical of your problem selection in your career?

**Sutherland:** I think it's unique. I think it's unique, because a) it's a selection of a problem, which is really a long-term problem, and which I cannot see a way to the end. There are these insurmountable barriers, okay? In order to let the engineering community deal with asynchrony in a friendly way, one needs the tools to make it happen. One needs the education to train the engineers that can do it. And one needs the management courage to actually adopt these ideas. And any one of those is insurmountable. The three of them together is clearly insurmountable. So I consider what I'm doing kind of like Don Quixote. Okay? I'm tilting with windmills, and I will lose repeatedly. But the importance of the problem and its central-- the central physical correctness of it is so obvious to me that it seems to me a worthy thing to work on. And besides, it's interesting. I have in my wallet a copy of a card that Al Davis published at the first asynch conference, and it's ten reasons to use asynchrony and ten reasons not to! And one of the ten reasons to use asynchrony is, "It pisses my boss off!" <laughter> Okay? And you know, there's a number of things to that sort, you see? Now it's really quite fascinating. The notion that you could design a machine without a clock is heresy in many circles. But it's perfectly doable. It's just hard. And it's hard, in part, because you have no idea how to do it. And it's hard because how do you test the damn thing? Okay, well, we've solved that problem, right? I think we've solved that problem. And there's all these pieces that are hard. But you keep working on it, and it starts coming together. Glen Fleck was a designer, and he used to design things. And he said, "Well, what you do is you make a design and then you keep working on it and it keeps getting a little better. Because if you make a change, you decide whether it's a better change or a worse change, and if it's a worse change, you don't do it." And so gradually things improve. And indeed gradually things have been improving. I think we know what we're doing now. For the first time, I feel confident that we know what the ingredients are, to make a testable system. And I'm hoping that Hexnet is going to be a successful thing that paves the way to a collection of uses that might be effective.

**Brock:** To follow-up on that, I recall that-- or I have learned that William Shockley used to exhort his group not to fight the physics, but to go with the physics in the same way that the self-timed paradigm, as you describe it, is kind of going with the special relativity rather than fighting the special relativity. But you know, I just wonder in fighting this impossible fight, is there a vision that you can articulate of why you think the fight is worth fighting, if it is successful, what do you see as the consequence, broadly conceived?

**Sutherland:** Hm. That's a great question, which means it's hard to answer.

**Sproull:** So one possible place to start from is your ARPA proposal at the moment.

**Sutherland:** I'm sorry?

**Sproull:** One possible place to start from in answering that would be the ARPA proposal that you're working on at the moment.

**Sutherland:** Yeah.

**Sproull:** I mean, that's part of the vision, right?

**Sutherland:** We need tools.

**Sproull:** You're articulating in there why it's important to DOD.

**Sutherland:** Yeah. Well, I think DOD's reason to be interested is that the first thing you do when you design a clocked system is you look at the technology use you've got, and you decide on the clock frequency. Okay? And clock frequency has been increasing over time, and now they've met the, you know, heat barrier, and the clock frequency has flattened out at a few gigahertz. Clocks of a few gigahertz are now popular, and one doesn't see clearly how to make them faster than that. That's, you know, what we're going to have. And then there are technologies in which clocking at the frequencies that you expect to go at, Josephson junction technology, for example, or any of the other super-conducting kinds of technology that might replace silicon technology, it's not at all clear how you would make a clock that runs at the frequencies that they need. And so you know, you can sort of see that the clock's paradigm has been very useful to us for 50 years, but its utility is coming to an end. And what do we do next? So there's that, the future promise. There's also the thing about energy consumption. See, the one thing that self-timed systems are really good at is doing nothing when there's nothing to do. And you know I think of this as a post-op. You see, a modern computer, it has a-- it goes to reference memory, and that's going to take it a thousand cycles. And if I had been climbing a mountain, and I saw that I now had to stop and take photographs, I would think, "Thank goodness! There's a breather! And I don't have to climb anymore for a while." And so you have to think of every pause in the operation of a modern computer as an opportunity to cool off! Because what limits its performance is the general rate in which it generates heat.

  

And asynchrony has some property that you only do things when there are things to do. And so you don't waste energy pedaling in place. Right? I met a guy who worked at Intel, and he was an expert in formal verification. And he was applying formal verification to the-- what do they call it? The circuits that avoid clocking unnecessarily. There's a word for that.

**Sproull:** What do you mean, like conditional? Conditional clock--

**Sutherland:** Yeah, well, in modern synchronous systems, to save energy, you turn the clock off if you can figure out when it's not needed. And that occurred to me as a very strange thing to do. Why don't you do it the other way around, and turn the clock on only when it's needed? I mean, the idea of omitting clocks when they're not needed, that's backwards. You should think about making clocks only when you know that they're needed. And he was working on the formal verification of these circuits that would omit clocks. And it seemed to me that's backwards. That's not what you should be doing. And the self-timed stuff generates clocks only when they're going to be used. And so, you know, omitting clocks is not something we would ever think about. We just don't think about it. They wouldn't appear unless they were necessary. So it seems to me getting those things of the right sign is important, but it's hard. Because if you are steeped in the synchronous technology, you think, "Well, I've got this clock which is a crutch I can lean on. And without that crutch I'm lost." Now, with what do we replace that crutch? Right? And I don't have an adequate answer. You know, why is it worth working on? Because I guess there are two laws that you want to avoid violating to go into business. You want to avoid violating the laws of the land, because if you do you'll go to jail. And the other is the laws of physics, because if you violate them, it won't work. You know, and the laws of physics are firmer than the laws of the land. The judges of the laws of physics are, you know, not people that you can talk to. They're, you know, the laws of physics are there. Unfortunately, they change from time to time, right? Did you know that the speed of light is-- in meters per second-- it's known to nine significant digits, exactly? Not just with error possible in the tenth significant digit. With *no* error in the tenth significant digit. Did you know that?

**Waldo:** Well, it's by definition.

**Sutherland:** What's That?

**Waldo:** It's by definition.

**Sutherland:** That is the definition of the speed of light.

<overlapping conversation>

**Waldo:** Well, the definition of the meter.

**Sutherland:** It's how far--

**Waldo:** So the definition of the meter now.

**Sutherland:** Yeah, it's the definition of meter. Right. It's how a meter is some fraction of how far light goes in this many seconds.

**Sproull:** Right, right, so in a vacuum.

**Sutherland:** Well, the speed of light in meters per second is an exact number.

**Waldo:** Yes.

**Sutherland:** Because that's the definition of the meter. It's very interesting turn of events, I think.

**Waldo:** And no longer one that we can verify.

**Sutherland:** I'm sorry?

**Waldo:** No longer one that we can verify.

**Sutherland:** No longer one that we can verify?

**Waldo:** Yeah. It can't be wrong!

**Sutherland:** It can't be wrong. <laughter> That's right. It can't be wrong. Well, isn't that verification by definition.

**Sproull:** Yes.

**Waldo:** Ehhhh...

<overlapping conversation>

  

**Sproull:** Yeah, for technology.

**Waldo:** It's no longer one that we have to verify. We will need alcohol to continue this discussion. <laughter>

**Brock:** Did you want to-- Bob?

**Brock:** I do have a couple more. It strikes me in this review of your career that all of your engineering projects were experiments to learn something new. Which I don't think is maybe the case for all engineering projects or maybe most engineering projects. So I was just intrigued by this idea of an engineering project as experiment, engineering for new knowledge. Does that resonate at all? Does that sound like a fair description?

<overlapping conversation>

**Sutherland:** Oh, I think so. The Evans and Sutherland Company, we built products to sell. There's a little different game.

**Brock:** True.

**Sutherland:** I wasn't very good at it. <laughter> I mean, the Evans and Sutherland Company was quite successful, but it wasn't my cause-- I didn't cause that. It was caused by people who joined after we started the company, yeah. And they were very successful in selling product training and equipment, which was the success of the company, which was not the part of the company that I was deeply involved in. So I don't know. Engineering projects, you know? I don't know. In Hollywood, they don't talk about *making* a movie. They talk about *doing* a movie. "We're *doing* a movie." And I think that's a very interesting choice of verb, because it says the activity is something interesting in and of itself, independent of the result. Now I think research is like that. You know, Salk set out to do a polio vaccine, more power to him, and he had a target in mind. But I'll bet that the course of reaching that target had its own rewards and problems. And I think that what we're doing is *doing* research, not making research, not making-- we're *doing* engineering projects, not making engineering artifacts happen.

**Brock:** Right.

**Sutherland:** The *doing* is in some sense the reason for being engaged. I've said many times and I think it's true, if you're doing the right thing in research, it should be fun! I mean, we have a number of strange things in the ARC offices. You go into the ARC offices, you find that the tables are trapezoidal. They're not rectangular. And why are they trapezoidal? Well, we found a place that we could buy trapezoidal tables, and I said, "That's neat! We can put them together into triangles, we can put them together into circles, we can put them into all kinds of shapes. And that'll be kind of neat. Let's get that." And then they came with Formica tops with whatever you want on the Formica. And they had Formica tops that had grass on them, and they had Formica tops that had pebbles, and that had gumballs, and that had all kinds of things. And the table that's in Marly's office has coffee beans on it, all over it. And it's, you know, this high a table. And I think if you have a table that has coffee beans on it, it should be a coffee table! <laughter> And only that high, right? But never mind, I don't get everything my way, but there it is. And the place where I sit has grass on one side, and some solid color on the other side, and it's sort of a half-grass desk, right? So we have things that are fun. And one of our neighbors is a quilter. And she made a quilt by asking many of my friends to offer a square to put in the quilt. And this quilt hangs in the front-- there's a glass front to the offices, and it hangs in that glass front. And it has all kinds of weird things in it. And you know, it has a quote from various people that I've known and liked, and it has some pictures from some of the young folk who've drawn me on a motorcycle and said, "This is--," you know? And so it's weird. And I mean, and there it hangs, and people walk by and see this thing, and they stop and peer at it for a while, and then they go on. And so the place where we work tries to be fun. Tries to be a fun place to be. And I think that's valuable. It makes people more willing to be open about their weird ideas.

**Brock:** I guess another question, or invitation I would have is I was struck by your comments when talking about leadership and also within your own research groups, this idea of revealing your humanity. And I was wondering if there were any particular things you might care to share about your humanity outside of research in other parts of your life, things that you have found particularly important or activities that you have pursued?

**Sutherland:** I don't know. What I do professionally is a very large part of my life. There's no question of that. And my wife, Marly, is like that, too. She's really devoted to what she's doing. She is a terrific role model for female students, and she's very approachable, and so many people go to her and ask her things and so on. And we recognize that in some sense we act *in loco parentis*, in part as parents to some of the younger people who are-- with whom we're involved. One of our graduate stu-- two of our graduate students have had children while they've been there. They've had babies while they've been there, and we've been involved in trying to encourage them, and trying to say, "This is an important thing to do. Be sure you spend time with the kids and whatever else." And that's a valuable part of the research group is there's a humanity to that. Now leadership comes in different ingredients, okay? If I've been successful as a leader, it's been successful in groups of half-a-dozen or so people. Much less successful as a leader of many, many people. There are leaders, political leaders who are able to engage the hearts and minds of nations. That's a different kind of leadership, and it requires a different set of skills. Skills that I don't aspire to. And my son, Dean, understands a lot about who his father is, and how his father behaves, and I sought his advice on an occasion when I was asked if I would be a candidate for the presidency of a university. And I got the best advice! I called a number of people, including Bob Sproull's father, who was all for it. He said, "We don't have enough technical presidents at universities. You should do it!" "*Should* do it," okay? I don't listen to "should" very carefully. <laughter> And so I asked Dean whether I should do this, and he said, "Well, dad," he said, "Does it sound like fun?" And I said, "No." He said, "Don't do it! You're no good at things that you don't think are fun!" Okay! He understands me very well. I mean, there's a sense in which I haven't worked a day in my life. <laughter> I've done the things which I found fun. And Russ Kirsch worked at the Bureau of Standards when I was in DARPA. And Russ Kirsch said to me--it came as an amazing surprise to him at some point in his career, that people would pay him very well for doing exactly what he wanted to do! And this is the comment of a happy man.

**Brock:** Hm.

**Sutherland:** You know, if people will pay you for doing exactly what you want to do, what could be better in life, right? And in some sense, I've never done things--certainly haven't done them well, things that I didn't think were fun. Now I do some amount of, you know, the garbage stuff that you have to do to stay alive. I'm the one who balances the checkbook. I'm the one who writes out the checks. I'm the one who, you know-- and there's a certain number of things that you have to do to survive. And some of them I do, because if I let other people do them, I'm not happy with how they're done. You know? Dave Evans explained to me, "There's only two ways to get things done. Yourself and wrong." <laughter> How true this is, right? Right?

**Sproull:** Yep.

**Sutherland:** We're going to write some code, right? It's you and me, we're not going to trust those other guys, because they won't do it right. It might work. It might even work better than our code.

**Waldo:** Nah.

**Sproull:** No! Might work. Might work.

**Sutherland:** Do you have the same experience that I have with writing code? So it always works the first time?

**Waldo:** No. <laughter> Well, it always works the first time it works.

**Sproull:** So I don't know. Maybe this is a good place to stop. You know, we've just heard that there's a happy man who's been paid well to do what he loves.

**Sutherland:** Well, I think it's time to say that I have actually taken the title of Research Professor at the Portland State University.

  

**Waldo:** Okay!

**Sutherland:** For the first time in a long time, I am actually a professor. Now my position at Portland State University was as Visiting Scientist, unpaid, great position to have! I had the run of the university. I could offer comments and advice anywhere I wanted to. But I didn't have to put up with anything I didn't want to, because I was a volunteer. But now with the DARPA contract, we actually have a contract, and the Federal government is paying the bill, it seemed to me inappropriate to volunteer my time to the benefit of DARPA. And if I did, the university would collect less overhead than if I actually accept a salary. So by accepting a salary, I actually increased the fraction of money that they university is able to get. That seemed to me a worthy thing to do. So I've recently actually accepted, and signed on the dotted line. So I'm now properly called Research Professor.

**Sproull:** And that's because the sign of the flow of money changed.

**Sutherland:** That's right. <laughter>

**Waldo:** Welcome back! <laughter> Welcome back! It's like you never left!

**Sutherland:** Well, in some sense I haven't. <laughs> It's not going to change my habits deeply.

**Waldo:** I'm shocked!

**Brock:** Shall we stop?

**Sproull:** Sure! I think it's a--

**Brock:** Thank you, Ivan.

**Sutherland:** Thank you.

**Waldo:** Thank you very much.

END OF THE INTERVIEW

  