Computer History Museum

AT&T DSP1 Oral History Panel
Daniel Stanzione, Richard Pederson, and James Boddie

Moderated by:
Shayne Hodge

Recorded: January 16, 2015
Atlanta, Georgia

CHM Reference number: X7410.2015

Hodge: So, this is Shayne Hodge. We are here in Atlanta, Georgia in January on a somewhat frigid winter's day, outside of our normal filming environment at the Computer History Museum in Mountain View. We are here with a panel to talk about the AT&T DSP1, the first, or one of the first, digital signal processors that was its own IC. And so, we will let the panel introduce themselves.

Boddie: Hi, I'm Jim Boddie. I was brought up in a small town in Alabama. And I graduated from Auburn University in electrical engineering in 1971. I got a fellowship to study at MIT for my master's degree and EE degree, which I received in 1973. I worked in the Research Laboratory of Electronics and did research on psychoacoustics. Particularly, how one localizes sound sources outside in 3D space. I went back to Auburn in '73 to work on my PhD. I did a lot of work as a teaching assistant there and thought that was really my calling, while I worked on my PhD-- my dissertation in speech recognition. While I was there completing my degree program, I got a call from Jim Flanagan, who was the head of the acoustics research department at Bell Labs Research. Jim was a well-known figure in the field of acoustics and speech recognition, speech synthesis and he offered me an opportunity to come to Bell Labs for a postdoc program. So, I jumped at the chance-- went to work at Bell Labs with Jim Flanagan in 1976 after receiving my PhD, and worked on problems relating to de-reverberation -- basically taking sounds from a room like this and removing all of the echoes and making it sound more natural like you had a microphone directly attached to your lapel like me. We did this work using digital techniques, digital signal processing. And it turned out to be using a refrigerator-sized array processor to do some fairly straightforward signal processing. While I was doing this work, there was an opportunity to become a full time member of technical staff at Bell Laboratories. And I, after experiencing all of the wonders of working at Bell Labs, I kind of changed my career goals from going back into teaching and decided to work at Bell Labs for my career. That's when I was introduced to Dan. One of the departments that I interviewed with was Dan's. And he offered me the opportunity to work on this new project that was getting off the ground floor--a single-chip digital signal processor.

Hodge: So, that's at [INAUDIBLE].

Stanzione: Yeah. It was a lucky day for me. It's good to be with you Shayne, despite the low temperatures here in Atlanta. And it's good to have time with both Jim and Dick here today. Thanks for setting this up. So, I was born in New York, but my family moved south. I grew up in Virginia and South Carolina and ended up going to school in Clemson, in South Carolina. I got my bachelor's in electrical engineering there in 1967. And through a series of serendipity events, I stayed on for graduate studies there and got my PhD there in 1972. And most of my research work was around computer architecture. My dissertation was actually in operating system design. And through further serendipity, I ended up at Bell Labs actually starting in Greensboro, North Carolina, a satellite location, in 1972 insisting that I stay in the South and insisting that I continue my software work. And two years later, I was designing integrated circuit applications in New Jersey. So, that's the way things work.

Hodge: But you in effect-- that was around the time UNIX was being developed.

Stanzione: I did. In fact, some of my early interactions after joining the labs were with Ken Thompson even though we were in not a related part of work. But Ken was one of the originators of UNIX, as I'm sure you know. And he gave me a login. I did some experimenting with UNIX. He had his chess program beat me soundly a time or two after I first joined the Labs. And it was good. And after different assignments in Greensboro, and I was in Colorado for a while, I was invited to join a group in Holmdel, New Jersey in 1974 that was doing microprocessor applications. And it was very exciting work, again, working with just extraordinary people, just a very exciting time professionally for me. This was about the time that the Intel 8008 had come out about a year or two earlier. And the 8080 was just being developed. And the question arose as to whether or not a microprocessor was appropriate for what was then the Bell System. All our applications, all our work, as both Dick and Jim know well, was focused on what was then the Bell System. By agreement with the government, we didn't do work outside of the Bell System. So, there was some study done as to whether or not a microprocessor should be built. And one of the things that we focused on at the time was whether or not the effort to build a processor would be warranted given what was going on commercially. And it was concluded by others that if we could really build a processor that was better suited for software development and specifically for which we could build a really good C compiler, then that would make sense. The key to that, and this is kind of a little bit of a building of the platform for what led to the digital signal processor work, was an architectural wrinkle where we took the registers that would have otherwise been on the chip and took them off chip, and high speed memory off the chip. And this allowed us to build, I forget the number, I think it was an architecture at the time that had eight registers, which was a big deal in the '70s. And we were able to build a C compiler then that worked well on this particular device. When-- about 1975 or so-- '76, the question arose on digital signal processors. And then-- and Dick can tell you a lot more about this than I, in order to get the number of multiply-add operations per second that we needed to do to do reasonable signal processing, it didn't seem like we had a technology that was capable enough. And the question was is there an architectural wrinkle that really makes sense here. And there were some people that we can talk about later that I met at the time, Jim certainly being one of them, John Thompson being another, who was a key architect in the early work, and Dick as well, that caused us to think that we could do this work. And we can talk a little bit more about that in a while.

Pedersen: Hi, I'm Dick Pedersen. I was born and raised on a farm south of Minneapolis, Minnesota about thirty miles. Went to a very large--

Stanzione: So, this is warm weather for you.

Pedersen: Yeah. Went to a very large school, had a graduating class of forty-eight. And I missed going to a one room school by one year. I think they saw me coming. I then attended the University of Minnesota majoring in electrical engineering. I got my bachelor's degree in '62 and then a master's degree in '64 doing a thesis, as required by the university, actually on thin film superconductors, field emissions from thin film superconductors. I joined Bell Laboratories in ’64 knowing they had a PhD program thinking that perhaps I would pursue while I was working. And I did take more course work at Lehigh University and decided it was too much of a distraction from the work, which I was enjoying. So, I never did complete the PhD.

Hodge: Okay. I always thought that all members of the tech staff at Bell Labs, I thought it was a requirement to have one.

Pedersen: No. No, it’s not.

Stanzione: No. At the time we were there, I think maybe a third of the members of technical staff had PhDs, and two thirds didn’t. But a very interesting point to me is you didn’t know. Whether you had a PhD or not really wasn’t important. We didn’t use--

Boddie: Didn’t use titles.

Stanzione: Didn’t use doctor as a title or anything like that. Really, your performance determined your position, not your education.

Pedersen: Right. And at Bell Labs, my first assignment was to actually characterize and kind of optimize the NikeX switching transistor, which was one of the early bipolar devices that went into production at Allentown, Pennsylvania after point contact and things like that. And I think it was in 1965 when people started talking about integrated circuits. And I was fortunate enough to be assigned to integrated circuit work. And I designed the 1A, the 1B, the 1C, the very first circuits that actually went into production at Western Electric. And from that point on, I worked in various bipolar technologies and various circuit designs, was privileged to be able to design many test masks to evaluate circuit designs, to evaluate technologies. I worked on such things as CDI, which was collector diffusion isolation, a technology in which we developed a three volt logic family that was used in all electronic switching systems I believe, the 1A processor and 4ESS. That actually stayed in production for about twenty years. It was probably one of the longest running production families. I explored such things as integrated injection logic. And it was about that time, I believe, that I ran into Dan and Jim and heard a talk about digital signal processors. And I was always curious to try to evaluate new things and to try new things. And so, I kind of raised my hand and said, “I’d like to get involved.” And that’s where we started.

Hodge: Okay. So, one thing, all three of you came from either the South or the Midwest. Was that-- I guess I don’t know if you [Dr. Stanzione] consider yourself a Southerner or a New Yorker.

Stanzione: Actually, I think of South Carolina as home. That’s where I grew up-- where my parents--

Hodge: So, was that typical of the demographics of Bell Labs at the time? Was there less sort of West Coast influence? Or less people? Obviously, the West Coast was less populated then.

Stanzione: Right. I don't know if I would characterize a typical demographic. A lot of people from the northeast for sure.

Pedersen: Yeah.

Stanzione: So, if you'd asked where the most of the people were from were from the northeast. But the group we had on the system side-- while we talk about the work we'll talk about the system side. Jim and I were on the system side, kind of designing the-- searching out who had applications, designing the processor, doing the architecture work, and the device side, which is where Dick led the effort. And on the system side, the group that we had had-- I remember, Renato was an Italian born in Argentina. Jimmy Tao was from Taiwan. We had someone there from Romania at the time. We had actually two or three people from Alabama, which is almost like another country, right? I'm sure Jim in New Jersey felt like he was from another--

Boddie: I still have my passport.

Stanzione: So, no I wouldn't-- certainly in our group and more broadly in the Labs, I wouldn't say there was a typical demographic, but it was mostly people from the northeast I'd say. What do you think?

Boddie: Yeah.

Hodge: Who was your biggest competitor for hiring in terms of if people weren't going to go to Bell Labs, who were they likely to go to?

Boddie: IBM?

Stanzione: Yeah, it depends on the discipline. It depended on the discipline. So, why don't you-- from the device side?

Pedersen: In semiconductors, it probably would have been Fairchild Semiconductor at that time.

Stanzione: I think so. Then on the research--

Pedersen: It would have been the West Coast.

Stanzione: And on the research side, where many more of the MTS [members of the technical staff] were PhDs to your earlier point, IBM research, GE at the time I think had a very significant, very good research effort. I know I'm leaving somebody out. And I'm not sure if other-- you'll recall-- you won't recall, but perhaps you've read that at least at that time if you were in telecommunications in the United States, and in major development, you pretty much wanted to be at Bell Labs and Western Electric on the product side because that was the place where it was all going on because there was a Bell System that did almost all the work. And so, if someone was interested in telecommunications research and a variety of other research activities, as Jim was, you wanted to go to Bell Labs.

Hodge: Going on to the idea for doing a DSP on the chip, who first sort of came up with and ran with that idea?

Stanzione: Success has many fathers, right? I'll claim-- I'll say that I was at least one of them. And, as I said, we were doing this microprocessor work, which ended up a process was built. It was built with CMOS technology, worked fairly well. There was a development environment that went around it that took a long time to get working but finally ended up working well. And while that work was going on, the question arose as to whether or not building a digital signal processor really could be done for the kinds of applications that we had at the time in the Labs and in the Bell System.

Boddie: I like to think of the genesis of the single chip DSP as coming from the lab that Dan and I were in. The history of it was all about filters, building filters for the Bell System. And that meant that early on, resistors, capacitors, and inductors. And the lineage of the single chip DSP stems from that technology. How do you put more filters on a board in a system? How do you make the filters more efficient? And so, a lot of technologies were developed using analog techniques for making better filters, like switch capacitor technology that was also done in our lab.

Stanzione: Right.

Boddie: And around the end of the '70s, the telecommunications industry was going digital. And you had 4ESS, which is already a digital switch. And there was the genesis of the local switch, the 5ESS that was being conceived. So, how do you start integrating these filters in a digital environment? And it seems to me that is the one thing that kind of pushed it together.

Stanzione: Pushed us toward the signal processing. The department that Jim and I were in on the system side had, as Jim was saying, a history in filters. In fact, two or three groups were focused on designing filters for other organizations, mostly in transmission but not solely in transmission kinds of products. But there was another group for-- there was a group that did what was the system design for microprocessors. There was a group that did per-channel codecs, the system design not the semiconductor side.

Pedersen: Mm-hmm.

Stanzione: You remember Doug Marsh's group perhaps did that. But when we got to the digital signal process, Jim was saying the network was going digital. When we got to the digital signal processing, it was believed the technology wasn't capable enough. And much as in the microprocessor work, there was the notion that maybe there's something that can be done architecturally to give a boost to the performance. John Thompson, at the time, was a fellow in research that I was having lunch with and discussing this problem with him-- or issue. And he said he's the one who thought he saw a pipelined architecture way of doing this. And that's my recollection in any event. And that led to further work and ultimately led to a study group that said, "Okay, well if we build something like this what would it look like? And are there applications that would use it?" And that was the beginnings of that. And once we got the essence of a system that might work, we turned to the semiconductor people and said, "Okay, here's what we think might be able to fit on a chip. Is this something that's doable in current technology?"

Pedersen: Right.

Pedersen: I didn't think that we could get any yield at all. Plus, I didn't know how to make a memory. And CMOS, versus NMOS, the NMOS had the more history behind it because there was such a large effort on dynamic memories, DRAMs. So, we elected to concentrate on an NMOS technology using enhancement depletion NMOS devices, and started doing some simulations. I think I imported two people with some history of MOS design, one in logic and one in memory. And the rest of us were all rethinking how to do things with NMOS transistors rather than with bipolar transistors. And it looked like it was feasible to meet their goals. And we proceeded with the design project.

Stanzione: I remember a deluge of acronyms coming from Dick. SPICE, I forget what that stood for and so on.

Pedersen: It's a simulator.

Stanzione: Simulator names where a lot of simulation work was done I think for months, right?

Pedersen: Yeah, probably so.

Stanzione: Yeah, quite a lot too went to see if a device of the kind that we were contemplating, meanwhile Jim was further evolving the actual processor design, but to see if that design could be implemented on a single chip.

Boddie: I think another important concept here that was pulled together was the idea of a single chip that could be programmed for multiple applications. You always had the option for all the various filters or applications, tone generators, equalizers, to build full custom implementations. But that would take a lot of resources for each one of the applications. But if you could do this in a programmable single chip, then you'd have an enormous savings in R&D and also a faster time to market. And since the advent of the microprocessor, microcontrollers, having the same effect for some control applications, it just made sense that if the technology was at the right level, you could apply the same ideas for signal processing.

Hodge: When you say to market, do you meant to deploy internally in the Bell System? Or were there any external customers?

Boddie: No, it's all internal applications.

Stanzione: Yeah, we spent some time in 1977, as I recall, before Dick was involved on the system side-- Contrary to popular belief, I guess we were technology driven. But this work wasn't going to go forward unless there were developers who said, "We can actually use this."

Boddie: Yep.

Stanzione: "If you build it, we can use it." So, it wasn't just technologists getting together and saying, "Oh, let's build this." In order for the work to go forward, to get funded and to go forward, there needed to be systems that said, "Yeah, if you build something like this, there's a pretty good chance I'll end up using it." So, we actually had a study group across different parts of Bell Labs, and people from Western Electric, people who were building datasets, people who were building switching systems, people who were building transmission systems, who really helped evolve the architecture and said-- didn't commit to using it, but said, "There's a pretty good chance if you can build something like this that we would use it in our application." So, they were the market, if I could use that word. They were the market for the product.

Boddie: You led the study group, right?

Stanzione: I did lead the study group. And that was part of the effort to get the project going was to have different applications that would use these devices in the thousands or tens of thousands stand up and say, "Yeah, if you can build it, I'll probably use it."

Boddie: Yep, you have to have some potential customers. You have to have an understanding that the technology is capable of doing it and some idea of how you might do it if you were given the go ahead.

Hodge: So, what was the division between Bell Labs and Western Electric at that time?

Stanzione: Maybe-- could I interject something before I ask Dick to answer that? I think one point I'd like to get on the table right away is that the three of us at the table, we were really standing on the shoulders of giants. They were just an incredible number of talented people. I won't try to name them all. I mentioned John Thompson a little bit earlier. But there were an incredible number of talented people. And we may get to them as we discuss parts of this going forward, that we were working with. But the Western Electric interface was really through Dick and his team.

Pedersen: Yeah, I was located at a Western Electric factory location in Allentown, Pennsylvania. And we were charged with development as opposed to some of the people at Homedale and Murray Hill, who were doing research. And our job was to design product for manufacture by Western Electric. And we had a close working relationship with Western Electric, but they were not part of the design process. They were on the receiving end and had to be convinced that these were acceptable products, and that we had something that was manufacturable.

Stanzione: And also Western Electric provided some of the funding for the work to go forward.

Hodge: How long did it take to do the study to determine to go forward and make this?

Stanzione: Jim has the study.

Boddie: Yeah, there's the study group here.

Stanzione: The report was issued in October of '77. I think that went on for months. Do you have an exact date?

Boddie: January of '77 is what I have. The study group began in January of '77, and this paper was put out in October. And I think it was shortly thereafter when you became involved.

Pedersen: It was late '77, I believe.

Stanzione: Yeah.

Hodge: Hold that up [for that camera]. So, is there a warehouse somewhere full of all sorts of these?

Boddie: My basement.

Hodge: So, late ’77 you got the go-ahead. That’s when you started the project then?

Stanzione: We issued this report. My guess is we started around the first of the year. We certainly didn’t start until after Dick had become involved and was convinced something could be built.

Pedersen: I’m not certain the time we actually, formally started and got funded.

Stanzione: Probably the calendar year.

Pedersen: Could have been calendar year ’77 or 8.

Hodge: January of ’78?

Stanzione: Wait, we have our historian here. Maybe he--

Boddie: Well, a part of the study group document has a prototype architecture that John Thompson, I believe, was the key developer for. And so, I have the architecture specification work beginning in June of 1977. And it was completed in April of ’78.

Hodge: Okay. So, doing that tools question here, you talked about doing simulations, what was your development environment like? How much of it was still by hand? How much of it was done on computer, you’re running it on U-- you mentioned this is somewhat amusing in retrospect-- and this is page fourteen thirty-five of the Bell System Technical Journal, Volume sixteen, that the program is entered into a computer by a UNIX time sharing operating system text editor. There’s a little footnote there that UNIX is a registered trademark of Bell Laboratories.

Boddie: Are you talking about the programming environment? Development environment, or the development environment for the chip?

Hodge: Yeah, the circuit, chip development.

Boddie: Okay. Yes, a lot of it was by hand and completely inadequate for the scope of what we were trying to do, which got improved over time. But yes, we started out from the architectural specifications to logic drawings that were on paper. These were diagrams that were done using building blocks that, I guess, Allentown had said we should be using for the particular design. But the various sections of the chip were divided into-- divided up among various developers. And these logic drawings were written up on paper. This is a reduced size, but the paper that we were drawing it on is not much bigger than this. From these logic drawings, we had a person Renato, who translated the logic drawings into a gate level description using a program called LAMP, which was a program running on a time shared machine at Indian Hill, I believe. So, he developed a gate level simulation. The individual designers of each section were responsible for test vectors, which Renato ran and passed back overnight a printout so that they could scan and see the ones and zeroes matched what they expected. Once we were satisfied that the logic was correct, we would send the drawings over to Dick.

Pedersen: Thank you.

Boddie: And you can take it from there. My understanding was circuit designers, who are assigned for each section, also drew up-- interpreted what we had drawn in logic, drew up logic drawings, and then handed those over to layout specialists, who actually did the layout work.

Stanzione: There were a lot of tools that you used in that regard.

Pedersen: Well, we did a lot of simulation using SPICE, which was a transistor level simulation.

Hodge: I know-- is it Larry Nagel, was from Berkeley. And I think he went over took Bell Labs around that time. Was he with you guys?

Pedersen: I don't know. SPICE came out of Berkeley. Don Pederson was involved in it. Larry Nagel is a familiar name. But we weren't the development team for the internal version of the SPICE. We were the users. So, I really don't know how much interaction there was there. In terms of actual layout, there had been quite an activity in logical design using NMOS-- using what was called polycell approach. And it was a somewhat automated approach to lay down building blocks and then wire them together via the use of computer. We determined that some parts of the DSP, which were not speed sensitive, could be done that way. And they were done that way. But a large section, most of the DSP, in particular the multiply, accumulate memories were all hand laid out by engineers and then translated into mask shop information by draftsman. Personally, I spent many hours tracing things and going through layouts to check that the layout was correct because we didn't have any automated techniques yet to really check the interconnect of custom layouts.

Boddie: I remember a session with a layout, colored pencils, and--

Pedersen: Light tables.

Boddie: Yes, and marking up the layout.

Stanzione: As I recall the arithmetic unit was particularly sensitive to-- it was particularly performance sensitive, and therefore sensitive to layout certainly a design-critical piece in order to get enough performance--

Boddie: That was about a quarter of the chip was the arithmetic unit. And that was all a full custom circuit design layout. The addressing units, the control, and I/O used the polycell approach.

Hodge: So, you were still laying out then what looked like a sheet of Mylar and tape?

Stanzione: Shayne, you have to remember that it was only a few years before that some of us had reluctantly left our slide rules behind. I'm serious. It was in the early '70s I think before we began using the calculators that we all remember well to do some of this. So, we were just a few years beyond that. So, a lot of this was hand work as well as large time share systems that were used for simulators.

Hodge: Just out of curiosity then, when you talk about the polycell building blocks, was that like physically a library when where you could go and get pieces of it and drop on the Mylar so that you had the piece of logic?

Pedersen: Yes. Yeah, they could be. I don't know if you can see them on the picture there, but some sections are rows of cells and then a column of interconnect between the rows of cells. And that was largely done kind of semi-automatically. But the speed critical parts of the chip couldn't be done that way.

Hodge: Well, to get to your point, those of us who came later are very grateful that you did these layouts by hand so we didn't have to.

Stanzione: You did them well. In fact, this is a-- I think now-- what do fabs run now, twelve inch wafers, somewhere in there?

Pedersen: I have no idea.

Stanzione: I think they're twelve inch wafers. And there's some work going on eighteen inch wafers. But I think that's a long way off. This [holds up for camera] was a gift that Dick gave me in 1979. This was a wafer then of DSPs. And so, you can see that compared to what's going on today, this was just early stages. Very advanced at the time but--

Pedersen: It was the biggest chip we'd made at the time.

Stanzione: It was a large-- this says above here, "A hundred plus sites. A hundred and thirty-seven active sites." Dick wouldn't have given me this if there were any working devices on it I know. So, yield wasn't very high, but a hundred plus sites on a wafer.

Hodge: What was your overall yield in those days?

Pedersen: That's a tough question. I mean--

Stanzione: It depends on the size of the device and--

Pedersen: It was a tough start because it was the biggest chip. And it--

Stanzione: I remember the celebration when you got one working device.

Pedersen: When we got the first one, yep.

Stanzione: No, seriously. Getting a handful of working devices was a big deal.

Pedersen: Getting the masks all correct wasn't easy. We had a few iterations. You know some color line didn't connect correctly or whatever.

Hodge: Yeah.

Pedersen: But eventually, we started getting samples out. And that would have been 1978 I guess.

Boddie: Yes, '79. The first silicon was a bust because of a design rule violation and some mask errors. The second silicon was in--

Hodge: May I ask, what do you consider a design rule violation?

Pedersen: We were using what was called a four and a half micron design rule at the time. So, if something got too close and created a short, that would be a design rule violation.

Hodge: Okay.

Stanzione: So, there was a process that was capable of producing circuits if you didn't get things that were too small or too close together. And you had certain separations that were required, and certain widths that were required, and so on. And, as Dick was pointing out, a lot of this was hand done.

Pedersen: And within the--

Boddie: And we didn't have a tool that would actually check it.

Stanzione: So, if that was violated, then the process, when it produced wafers, would produce devices that didn't work.

Hodge: So, did you figure out there was a design rule violation because devices didn't work, and then you had to post mortem and figure out where the problem was?

Pedersen: There was a lot of cooperative work with Jim's people and my people on the test sets trying to diagnose what the problem was.

Boddie: Yep.

Pedersen: A lot of work.

Hodge: Who was involved in that because, as you said, we didn't have a chance to bring everyone in one room so, what people were most involved?

Pedersen: Well, my designers were involved. I had a full-time engineer that wrote the programs based on the vector sets that they came up with. And he designed the probe cards, and performance boards that went with the probe cards and whatnot to try to actually make physical contact to the devices and connect them to the test set. But Jim's people would have been there. And they would do analysis of the type of failure to try to figure out exactly where's the failure occurring in the chip. And it could have been one transistor where two metal lines cut too close together, and they were shorted.

Hodge: So, Jim how is it-- sort of the last we left off with you, you were sort of happily doing math somewhere. And now, you're troubleshooting silicon. How did that happen?

Boddie: I'm sorry, which--

Stanzione: How did you get involved in the signal processing? How did you come from the research work to the signal processing?

Boddie: Well, my first assignment on this project was to look at the arithmetic unit for this proposed DSP.

Stanzione: Can I interrupt?

Boddie: Sure.

Stanzione: I heard Jim giving a presentation. He was doing this postdoc fellowship kind of work in research. And I heard him giving a presentation. And I think I went up to him afterwards and in so many words said, "Jim, I have an assignment that would be a lot more interesting than what you're doing. Would you like to do this?" And he said, "Yes." And that's how he got involved in the work. Is that a fairly accurate representation?

Boddie: Pretty much it. I'm always interested in how you can do things on a smaller scale. When I was at Auburn, I did some signal processing work. I built a speech synthesizer on an array processor [PEPE] that was designed to track ICBM missiles coming over the pole. It was donated by the Army to Auburn. So, this is [the size of] a couple of refrigerators. And then at Bell Labs research I was doing this echo-cancelling, de-reverberation on another array processor. So, it's kind of intriguing to see how can you now-- is it possible to do this on a practical scale? And this is what the first DSP offered was the ability to do some of the signal processing, speech recognition, speech synthesis, the de-reverberation, all of these things on something that you could afford to hold in your hand.

Hodge: Was an array processor-- is that like a bit slice processor, or is that--?

Boddie: Yeah, the bit slice would be the technology that would be used in it. I forget some of the manufacturers of the array processors at the time. It was some company in New England, I believe, that made the array processor that we bought. But it was also attached to an Eclipse minicomputer, which had to drive it.

Hodge: So, how did you go then? You started off on this from a signal processing perspective. How did you move from signal processing to caring more about what the silicon was doing? Did you wind up becoming fairly expert in device physics by the time this was over?

Boddie: Absolutely not. On the system side it was mainly architecture, logic design. And my-- I guess my hands on to the actual silicon would be in the debugging processes when I would go over to Allentown and participate in the debugging of the test set.

Hodge: And was the arithmetic unit, was that designed by John Thompson?

Boddie: I designed the arithmetic unit. John designed the control logic for the DSP. And he was the architectural lead for the project.

Hodge: And we all looked for him prior to this panel, but he seems to have, in his retirement, done a good job of going off the grid. So, none of us could find him.

Stanzione: Yeah, but he was clearly a key contributor on the architecture. No question about it. The actual design that Jim and John-- I'm not sure the shared credit. It was a multi-stage-- four stage I guess, pipelined design and had to be done that way in order to get enough processing capability to do the DSP activities, but made the programming of it a little bit difficult if not arcane to the initial programmer. So, there was some work to be done there as well.

Boddie: Yeah.

Hodge: It sounds to me sort of like you came up with the RISC architecture before they really named it that way. Would that be correct?

Stanzione: I wouldn't have called this a RISC architecture.

Boddie: No. But there are certain aspects of the instruction set that are RISC-like for sure. There are dedicated fields for controlling multiple functional units. It's an architecture in which every instruction takes the same amount of time. There's no multiple-cycle instruction. It's all-- every instruction is executed in four clock cycles, controlling, arithmetic unit, the address generation, the memory access, and the I/O.

Hodge: Skipping back just a little bit, going to the fab, so I heard someone who was at Bell Labs around that time, or maybe a little later, he said that if Bell Labs was to build a microprocessor, the first thing they were going to do was buy a beach. So, since you were sort of on the fab side, how much of the fab-- how close did you get to the beach on the fab side?

Pedersen: I was not involved in the actual fabrication process.

Hodge: How many fabs did you guys have at the time? More than one?

Pedersen: We had one in West [ph?] fab that was concentrating on DRAMs. And that's where we put this device.

Stanzione: That was in Allentown.

Pedersen: That was in Allentown, yeah. There was also a fab at Murray Hill for CMOS at the time.

Stanzione: Right.

Pedersen: And there was a bipolar fab at Allentown, and one in Reading.

Stanzione: In Reading, Pennsylvania as well.

Pedersen: Yep.

Hodge: What made you do your own DRAMs instead of the Intel or a little later I guess the Japanese DRAMs that were on the market?

Pedersen: Well, I'm not sure I can answer that.

Stanzione: I don't think I could either.

Pedersen: There wasn't an effort at the time to commercialize it. It was to use it.

Hodge: Okay.

Stanzione: The thinking at the time, generally-- so I can't characterize the DRAM decision per se. But the thinking at the time generally was at least in two parts. One, do we have unique needs? And often times when we used outside devices, we got military spec devices. And we actually had an organization, it may have been in Allentown, that qualified outside devices if we used them because the Bell System had a little bit broader temperature range and a little bit different kinds of requirements on devices. So, do we have unique needs that merit doing something ourselves? And a second thing beyond the unique needs, is there sufficient volume to economically justify doing it ourselves. So, I'm sure that kind of thinking went in to whatever the decision was on DRAMs as well.

Boddie: But I think also DRAMs, in a sense, also drove the technology to strive for the highest density circuits, the fast circuits. DRAMs is not a bad start. And then the technology that is derived from that can be applied to other circuits as we did for the DSP.

Stanzione: I think even-- you'll know better than I, Shayne. But even in the Intel case, their principle technology drivers were the memory devices. And the microprocessors, I'm pretty sure the 4004, 8008, maybe even the 8080, were initially custom jobs that they were doing. I'm not sure they always came to fruition in the application for which they were designed. So, Jim's point I think is a really good one that DRAMs drove the technology. Wouldn't you agree, Dick?

Pedersen: I agree with that completely, yes.

Hodge: So, DRAMs work this sort of way to push the technology forward on the process side to I guess forced you to innovate for a lack of a better way of putting it maybe?

Stanzione: Say again?

Hodge: It was kind of forcing you to innovate, forcing you to develop your processes to make them work?

Stanzione: We felt at the time, I think, certainly in the Labs, that we had to be in the semiconductor business. It actually originated in the Labs. And that in order to fulfill our mission, which was providing technology for the Bell System, we had to be in the semiconductor business. I think earlier in the '70s, there'd been kind of a stumble in DRAMs. And so, there was a major effort to come back and build the technology for memories. And for a while, they were competing technologies. But it was clear NMOS was really the technology for dynamic random access memory.

Pedersen: Right. It was around 1970 or '71 there was kind of a turnaround.

Stanzione: Right.

Hodge: So, let me ask you then, in the mid 1970s, may have had an uncomfortable conversation or two with the DOJ up to that point. What was your situation with the consent decrees? What did it allow you to do, not allow you to do, etc.?

Pedersen: That was 1984.

Stanzione: The ‘84-- well, the conversations may have started in the ‘70s certainly. I wasn’t involved in those.

Hodge: Let me redo that question. I thought even back into the 50s they were some--

Stanzione: In 1956, there was a consent decree. In fact, right now, if you look at an old movie that was made say in the 1940s or so, in the credits you’ll see at the end of it sound by Western Electric. And so, even then technology from what was then the Bell System was beginning to have commercial applications. And there were no rules around where that technology could be sold, and so on. And so, I believe it was a consent decree in 1956-- sometime in the ‘50s, that restricted the technology and other commercial activities to the Bell System.

Hodge: And how, in general, would you guys characterize microprocessors or DRAMs as being useful within the Bell System particularly in the early ‘70s?

Stanzione: As being useful in the Bell System?

Hodge: Yeah, or for that matter even something like UNIX as being useful to the Bell System.

Stanzione: Well, certainly-- I think the date for the Intel 8008 was 1972, I believe. It was the early ‘70s in any event. So, they really weren’t used much before then. And, in fact, I came to New Jersey in a group that was really just beginning to experiment with how microprocessors might be used in applications. So, in the ‘70s not really that much at the time, but again as was being recounted earlier--

Boddie: Yeah, digital PBXs, digital central offices, the trend was clear. There’s going to be a digital foundation. And microprocessors and memory are going to be at the center of it.

Hodge: So, was one part of the thinking that you had a lot of billing and other more traditional business stuff, and that things like UNIX could be useful for that irrespective of the fact that the business underlying it was telephone? Or did things actually need to have a specific telephone purpose and not just a general help you run your business purpose?

Stanzione: With regard to UNIX, this is question with regard to UNIX? I'll take a stab at that. Maybe Jim might want to as well. The fellas, extraordinary people, that developed UNIX came out of a lab that was focused on computing research. Their mission was research. And it was to advance computing. Thompson and Richie, I think I credit them appropriately as the UNIX inventors, were really focused on how we can advance computing. I'm sure they thought it would be useful in all kinds of applications not just things that were in the Bell System at the time. And so, that was their work in UNIX. And as you know, during that period, especially since the Bell System was restricted to Bell System Telecommunications kind of activities, there were a lot of other variants of UNIX that came out, some of which were more commercially successful.

Hodge: Why was it approved though above their level? Why did someone say that computing research is important to what we're doing here?

Stanzione: Oh, I have an opinion there, but I wasn't part of that. I don't know the answer to that.

Boddie: Because research was able to do a lot of things that you might not, at first glance, think have application to products here and now. But, in the future, they will.

Stanzione: But I can clearly kind of see how someone making that decision at the time would say, "Look, is computing important to the future of the Bell System technologies that are needed?" And the answer would have been clearly yes.

Hodge: So, your research department that had sort of a freedom to experiment fairly broadly, even if it wasn't, you know necessarily, clearly going to be applicable immediately?

Stanzione: The research group did, yes.

Hodge: So, coming back to the DSP, you guys were working on-- I know you talked that-- you talked about you were starting to go to digital telecommunications and the PBXs and things. Were your initial applications thinking you were going to work with already digitized signals or that you would put these in with A to Ds and D to As around them to replace it-- when we're thinking analog? What was sort of the expected use case?

Boddie: Well, I think it's a little of both because we were in a transition period from a mixed digital analog environment to what's now almost a hundred percent digital. The 4ESS at the time was already a digital switch.

Hodge: Could you go ahead and tell a little more detail about what that was?

Boddie: The 4ESS was a switching system that's primarily used for switching long distance signals as opposed to the 5ESS, which would be a local digital switching office, a central office, a piece of equipment for switching local telephone lines. When we first started the project, we had the digital switch. So voice data, for example, was transmitted around long distance in a digital format. It was basically eight bit PCM data that was compressed into a format called mu-law, or A-law for international. And it was transmitted digitally over long distance lines. At this time, on the other hand, the central office switch, the switch that handled your local telephone calls, was completely analog. And we were designing the 5ESS switch, which would be an all digital circuit. So, in the case of 4ESS, there was a need at some point to be able to decode tones that would be transmitted over the line. And before the DSP, I believe they had actually converted digital PCM to analog to run into an analog tone detector back to digital. Of course with the DSP, you could replace that with completely digital tone detectors.

Stanzione: I can add a little bit there, and help me with this Jim. In the Bell System and telecommunications in general, the first systems that went digital were long-haul transmission, so-called T-carrier. T-1 being the first one and so on. And so, that's where the digital technology first hit. Then the 4ESS was a switching system that switched those signals. But there was the need to do things, perform functions like the filtering Jim's talking about. Echo cancelation was another thing became an early user of doing things digitally. But then that creeped out to the edges of the network. So, Jim was-- as that moved out, there were things Jim was reminding me just earlier today that one of the early applications were called DTMF, dual tone multi-frequency receivers.

Boddie: Right. The touch-tone receiver.

Stanzione: Yeah, sure. Touch-tone receivers, when you press a touch-tone, that has to be decoded. So, those were some of the first applications for the DSP. Later on datasets, at the time, datasets were analog. In fact, many of them were, again in the '70s, when we were still using slide rules were acoustic couplers. You actually had a box that sat beside your desk. One end of it plugged into the computer. And then you took a handset off the telephone and put it in two cups.

Boddie: Suction cups.

Pedersen: Two cups, yep.

Stanzione: And transmitted-- I actually remember when we made the big step to three hundred bits per second using those acoustic couplers.

Hodge: By dataset, do you mean modem?

Stanzione: The modem. It's what we would call a modem today, and in fact there was one that was called the 212 that was first twelve hundred bits per second, then ninety-six hundred. It had a lot of digital signal processing. So, later on in the '80s, they were big users I believe of DSPs. So, what were some of the other applications though that were really big users of DSPs in the Bell System?

Boddie: In the Bell System, there was--

Stanzione: Or telecommunications at the time?

Boddie: A need to develop test systems that would test the reliability of the circuits. So, they would generate tones. You would detect tones. You would do circuit analysis. It wasn't until much later when the level of integration increased and we got the power consumption down that we got into an explosion of digital signal processing technology that we see today in modems, and cell phones, and cellular infrastructure, cordless phones, digital cordless phones. But I'd say in the early days, the application for this original DSP were largely in switching systems and PBXs.

Stanzione: Meanwhile, Dick and his teams were-- these first devices we built were about a million instructions per second, or maybe a couple of million instructions per second and two to four micron line widths. And that's sort of, as I recall--

Pedersen: First one was four and a half micron, and I think it was one and a quarter million instructions per second.

Stanzione: So, one MIP machine is what we're talking about at the time. So meanwhile, his team kept making these things. Design rules got finer. Devices got faster and faster and consumed less power. And so applications, many applications, found this an attractive technology.

Pedersen: And as the CMOS technology became more mature, we evolved to CMOS for those devices.

Stanzione: And therefore the power consumption dropped considerably.

Hodge: A question I wanted to ask was from what I've read about Bell Labs, or AT&T at the time, was that you actually had some fairly strict standards regarding network quality like what voice quality needed to be on the network and things like that. A, is that true? And B, how did that impact your design rules?

Stanzione: Well, first of all, clearly it's true.

Boddie: Right. So, we had system customers that had requirements on the algorithm performance for certain things. Take the-- it's not quite the same, but I think in the similar vein, the digital touch-tone receiver, touch-tone receiver, which was one of the first applications of the DSP. The big problem for touch-tone receivers is talk-off. When you pick up a phone, and you talk, and a radio's playing in the background, it's possible for the receiver on the other end to falsely think that a tone is being sent down the line when you're talking or if the music is playing just the right pairs of tones. So, there were Bell System standards for the operation of a touch-tone receiver to reject random noise or tones as opposed to accepting the legitimate touch-tone signals. So, one of the aspects was developing the algorithms for the digital touch-tone receiver. And we spent a considerable amount of time testing our receivers to make sure they met the Bell System requirements. I'm not sure how that impacted the architecture per se. One thing that certainly did was the performance of voice quality. And we felt that we needed to have an arithmetic unit and a data format that was large enough to accommodate the signal processing that we anticipated for voice signals to maintain the standards of Bell Systems.

Hodge: So, were you running internally at a higher bit depth than you would ultimately communicate to the outside world in order to manage round-off error and things like that?

Boddie: Yeah, we adapted a basic word size for our data as twenty bits--basically twenty bits of data and sixteen bits of coefficients. We felt twenty bits would be sufficient for most signal processing applications for voice at the time. Sixteen bits would have been fine for tone processing. Thirty-two bits would have been overkill. So, to be as economical as possible, we settled on a twenty bit format for our data.

Hodge: And I guess this might go more towards the A-to-Ds and D-to-As you were using, but I know at least some early compact discs were released without the recordings being properly dithered when they were done. And so, you had ugly stair-step quantization errors in it. Was that something I assume that you knew about and accounted for in order to maintain quality?

Boddie: Most of our data was not CD quality. Most of the data was from codecs came in the form of eight bit, mu-law compressed PCM. That's an eight kilohertz sample rate. And the data is compressed into an eight bit word, which is like a floating point number essentially, which has a data range, in a fixed point world, equivalent to thirteen or fourteen bits of fixed point data.

Hodge: So, who was designing the A-to-Ds and D-to-As that were your interfaces to the outside world if you know?

Stanzione: Actually, similar arrangements in the early days to the one you just heard Dick and Jim describe between the systems area. There was a kind of a sister group in our same department in transmission, which as I said earlier was where some of the early digital technology evolved. Doug Marsh was the head of that group. And he had some system level circuit designers. But he worked with people who were in the same organization that Dick was and who actually implemented the semiconductor designs for mu-law codecs that were used in this country.

Pedersen: My recollection is most of the codec work was done in CMOS.

Stanzione: I think ultimately in CMOS. I don't know what the first codecs were, but I think the early ones were clearly CMOS, right.

Pedersen: We had an experimental one in I^2L that never went to production. But I don't remember any NMOS designs. I thought they were CMOS.

Stanzione: Yep, I don't think there was any NMOS. But that was obviously a critical device as well because one went on every line when we finally went to digital switching.

Boddie: Yeah.

Stanzione: Every single telephone line had a mu-law codec it, mu-law in this country.

Hodge: If I can ask you Dan-- so, you stayed on with Bell Labs and wound up being president of Bell Labs at one point.

Stanzione: I did, yes.

Hodge: Currently, there's a way of-- I'm going to generalize and editorialize in the same breath-- a much looser engineering standard, particularly around Silicon Valley. One famous motto for a while was move fast and break things. Did Bell Labs or AT&T ever start going down that path and loosen up on the standards? Or did it always remain fairly strictly, these are performance metrics that have to be met for things that we're doing?

Stanzione: Well, first let me point out that my time at Bell Labs ended in the last century. So, I can't speak for what currently is going on in Alcatel/Lucent and Bell Labs. I'm just not qualified to speak to that. Let me divide what you're saying into two things. One was kind of a strict kind of waterfall methodologies that were used at the time for very large projects, especially, software projects. So, one is the development methodology, and the other is this quality standards for the end result. I don't think there was ever any letting up on the quality standards for the end result at least during my time there. So, I think even today if you're-- I don't know how many digital switches get produced anymore, but the reliability standard was it's got to be five nines and-- you know ninety-nine point nine, nine, nine, five percent 99.9995% reliability. That's probably still the standard. I doubt that changed. But methodologies change considerably, at least on the software side. So, much more rapid iteration, fast prototyping, rapid iteration, methodologies changed considerably. And I'm sure it's still changing. But even in the '80s and '90s, the methodologies were changing.

Hodge: Okay. So, how-- looking-- I'll bring this back to you then, Dick. How-- in that waterfall method, it seemed like you were sort of being tossed like a schematic and told to build it to some degree.

Stanzione: And get it right.

Pedersen: And get it right.

Hodge: Yes, and please fix whatever bugs are accidentally introduced. How difficult was that to do? How much back and forth was necessary to--?

Pedersen: I wouldn't say that we were tossed anything. I think we had a very good, strong team effort, a lot of cooperation back and forth. And basically we knew their forte was the architectural design, the logic design. And we tried to implement that as efficiently as we could. And we worked closely on simulators. And I think--

Boddie: They provided early feedback to us on what the capabilities of the technology were in terms of the density we could expect, propagation delays, and memory access times. So, we had to work with the device people very closely to come up with the initial architecture that we thought would be feasible.

Stanzione: I'd say, Shayne, that while compared to nearly forty years ago, devices are a hundred thousand times faster, and you've got over a million times more transistors on a device, and so on. The switching speed of a neuron hasn't changed, and nor has human behavior changed that much. So, the successful activities were very collaborative. And that was a real strength of the Labs at the time. Certainly, we had our squabbles as any group might. But the real strength of the Labs was to have so many capable people in different disciplines who could collaborate together.

Hodge: And you weren't physically-- you were in Allentown, correct?

Pedersen: Right.

Hodge: And--

Boddie: We were in Holmdel.

Stanzione: Holmdel, New Jersey. About-- what is that, two hours, two and a half hours?

Pedersen: Two hours probably.

Stanzione: About a two hour drive. And I'm sure driven a good bit during this work. So, it wasn't an over the wall.

Hodge: Okay. My apologies for characterizing it as that.

Boddie: Well, just-- I'd like to amplify a little bit on the notion of the collaboration among all the people at Bell Labs. It was one of the great things about Bell Labs in that era that you could walk into a world class-- a researcher at Bell Labs and say, "Look, I have a problem here. What do you think?" And you would not be turned away.

Stanzione: He or she would put down what they were doing and help you.

Boddie: Yeah. It was great. I worked with a Bell Labs researcher in mathematics, Debasis Mitra, early on to validate some of the ideas that I had about a floating point arithmetic. And he took on the problem right away. And he generously allowed me to be a participant on a paper that he wrote for an IEEE conference. It was a great environment, a lot of cooperation. Certainly there's competition, but the competition was in ideas. And I think in that era, things seemed to me to be decided based on their merits rather than politics and connections.

Hodge: So, was that basically a cultural norm about Bell Labs that if you're going to work here you help other people when they ask for help. You have people come to you, and they want assistance in your area of specialty. You're free to go help them and that sort of thing.

Boddie: I think it was encouraged in research. Research had enormous freedom to think about new, far out things. But they also knew they had an obligation to help developers in solving real world problems.

Stanzione: Yeah, I don't remember. I'm sure it happened, but I never remember an instance as an MTS where I was discouraged from stopping from what I was doing to help somebody else. And I never remember as an instance as a supervisor supervising other people telling somebody you can't help somebody. In fact, as Jim showed you this study group report, there was probably a dozen people on this study group from all parts of Bell Labs. And I can't remember someone at this stage from Western Electric was involved. It wasn't a big effort to get people freed up to study this and see if they could use it in their applications and what the best architecture might be. Yeah, Jim's pointing out, it's really more like, what would you say, twenty, twenty-five people from throughout the Labs who all helped in understanding is this a good idea and can it be done.

Hodge: Yeah, I recall reading my history at the labs that it seemed the only person anyone ever remembered who kept their door closed on a regular basis was Claude Shannon. I guess he was considered sufficiently brilliant to be allowed that eccentricity.

Stanzione: Well, I don't know. Again, I wasn't there at the time. I guess the question might be what might happen if someone knocked on that door.

Hodge: So, none of you guys then unicycled down the hallways while juggling?

Stanzione: No, no, no. I wasn't one of those. It would be a sad-- I'd need medical attention.

<laughter>

Hodge: A question I have from going through the technical journal is you said the logic design work produced, if anyone's following along [page] 1434, 'logic design board produced a gate level description of the processor. A logic level simulator program was used to verify the design.' And this was the part I had a question about. 'A TTL prototype of the device was also constructed, which could emulate the DSP running at full speed.' So, did you simulate the entire DSP in discrete components?

Boddie: We did.

Stanzione: Yep.

Boddie: Nadia Sachs, an MTS and our department translated--

Stanzione: She was from Romania, by the way.

Boddie: She translated this diagram into a TTL design. It was-- we called it the "breadboard." And I guess it had about five large wire wrapped circuit boards in it, two for the arithmetic unit, addressing unit, a control unit, and probably one for the memory. We built three of them. We had one that we used to develop-- kind of debug the architecture, develop the touch-tone receiver. We sent one to Allentown, I think, for to help develop the test program.

Stanzione: And this is what they sent back.

Boddie: Yeah. And then we sent one to a customer up at Whippany.

Stanzione: Oh, yeah I didn't remember that.

Hodge: So, these were like pull out the TI manual and the 7400 chips?

Boddie: Exactly, the orange book was exactly what we used.

Hodge: So, five boards and the transistor, I realize that you're working with ICs and not transistors, but it said the final circuit was approximately forty-five thousand transistors. How on Earth did you get something with that many gates to work doing it by hand? I mean I could never in school get things with more than about five chips to work reliably.

Boddie: Well, Nadia was very good in doing the translation. I helped in some of the debugging of the board. And we had a wire wrap shop that we kept busy making iterations.

Stanzione: It didn't hurt to have smart people.

Boddie: Yeah.

Hodge: Well, let's go then a bit into the actual design of the DSP1 itself. I'm not going to go that deep into it because the literature has a lot of technical detail, but just some of the things in looking through it I was curious about. One, and we were talking about this before filming started a bit, you used separate data and program memories. But you used the same bus. So, what was the design considerations behind A, separating those memories, but then B, using the same bus?

Boddie: It's hard to recall exactly the rationale but my thought was-- my thinking today is that it was because it made the design simpler. I think we were concerned about the cost of having multiple busses and all the connections to multiple busses. And it just made a more elegant design in my view for having a single bus that was multiplexed, multiplexed in such a way that every instruction was broken into four [clock] cycles. One cycle was used for transmitting the instruction over the bus, another cycle for data from the data RAM, another for the coefficient, which would come from the same ROM that the instructions were in, and then another cycle reserved for getting data back to the RAM.

Stanzione: One thing, and my memory's not clear on this. So, you guys will have to help me. First of all, the instruction memory was ROM, not RAM, right? And so, you may not be aware of that. It was a read only memory. It was a different memory. So, clearly you're going to have two memories. You want a ROM so that when the power was off the program was still there. And then you had a random access memory. So, to start with, the design was two different memories. So, you had to have two different busses to the device. And then it becomes a question of how to best implement a design when you want to get to processor from two different memories. How many pins do you have on the device? What kind of packaging are you doing as well as the silicon design itself?

Hodge: And what-- it seemed to-- yeah, it was a forty pin dip that was the packaging. Was that-- you wanted a forty pin dip and you designed accordingly? Or that was what worked out? How was that decided?

Pedersen: I think in the end that's what we needed. It was a matter of need.

Stanzione: Forty pin I think was about the largest--

Pedersen: It was.

Stanzione: That we could use at the time. But I remember the ones that are hermetically sealed package. And the forty pin, there may have been larger pin outs at the time, but I think forty pin was the largest.

Boddie: Right, we had to have something that could dissipate the heat.

Pedersen: Yep.

Stanzione: So, all the design questions. And then given the forty pins, how to best use the pins became an issue and also I think reads to your question about how the memory bus was used.

Hodge: I don't have the pin out here. So, did you wind up having to reuse pins as both inputs and outputs or multiplex outputs on pins and things?

Stanzione: I don't know that.

Boddie: Not so much. I think it was pretty much-- the address and data bus were multiplexed on the pins on the outside.

Hodge: Okay.

Boddie: And that was used for prototyping. We had the ability to replace the on-chip memory with external memory for prototyping.

Hodge: That's actually a question I have where you described about your support tools. You have a big box, the DSP mate. And then you had an emulator prototyping board. And then it said, "Finally, the system developer can request fabrication of a DSP with an on-chip raw coded with the debugged program." But on some of the other ones you mentioned using EEPROM. But what was the decision to use a hard coded one in the final versus an EEPROM or something that was external where you could just have the DSP be prebuilt and hook it to an external program that kept its memory?

Boddie: Well, we thought that in almost all cases all use cases, the DSP would ultimately be a device with the on-chip memory programmed. But you would need these other tools for developing the application.

Stanzione: And you wouldn't, at the time, electrically alterable read only memories, or EEPROMs or--

Pedersen: Didn't have it.

Stanzione: It wasn't a compatible technology with what we were doing. And it was a very large device. In fact, the first ones you actually erased using ultra-violet light-- or some of the first ones, you actually erased using ultra-violet light. And it was a complex process. Then you'd reprogram it. So, you think of it this way. You've got a signal processor you're going to put it in one of the touch-tone receivers that Jim was talking about earlier. You've got to write the program for it. So, the DSPMate was the development environment for which you could write a program for the DSP.

Boddie: Sure. We had-- well, before that step even, typically an application programmer would write the program using our assembler and simulate it using a simulator.

Stanzione: Right.

Boddie: The assembler was written by John Aggesen.

Stanzione: Oh yeah.

Boddie: Charlie Simmelmann wrote the simulator. Both of these were tools that were written in C and ran on the UNIX operating system.

Hodge: So, C was mature enough at that time to--

Stanzione: Oh yeah.

Boddie: Oh yeah.

Stanzione: And so, you'd complete the simulations. Then you'd--

Boddie: Then the next step would be to use the DSPMate, which is this box here. And this was-- PCs were just becoming available. And this was designed to be a standalone box that would interface to the user via a terminal, not a PC. So, it had its own microprocessor, a Bellmac-8 microprocessor. There was one board that had the DSP and RAM program memory. And it had one board that was essentially a logic analyzer to capture all of the signals coming in and out of the DSP so you could set break points, capture a whole string of ones and zeroes on the address bus and data bus for debugging your application. One of the outputs of the box was a forty pin plug that you could plug into your target system. The box then essentially emulated the DSP in your target system. Once you've debugged your program, at that point, you would program your EPROMs and use a box that's essentially this size. It would have a DSP and EPROMs on it for integration into your system.

Hodge: How closely did your software simulation match that hardware simulator?

Boddie: I don't know of any issues.

Stanzione: If it didn't match, it was a bug, right, so that the simulator would have get fixed.

Boddie: Yeah.

Stanzione: Oh, did you say to match the hardware simulator or the chip itself?

Hodge: Well, I noticed you had a lot of tools. You have the software simulator. You had the big DSP mate for it. You had the smaller DSP emulator. And I was wondering how closely do these things match each other.

Boddie: I think they matched very well. The only (I think) issue of concern there was the DSP didn't have built-in hardware breakpoint capability. So, to insert a breakpoint, the software would actually modify the code to cause the DSP to halt at that point.

Stanzione: Then examine--

Boddie: And examine the results.

Hodge: Okay, so you could sort of suspend the execution, examine it, and then--

Boddie: Correct. Yeah, we didn't do it from a hardware point of view. We did it by inserting software breakpoints.

Hodge: And Dick, were you responsible then for helping to build these emulators as well?

Pedersen: No. Not at all. We were--

Stanzione: Fully occupied getting the devices working.

Pedersen: Fully occupied working on that.

Hodge: Okay. Getting the real ones working.

Boddie: Yeah, this was designed in Bob Cordell's group. Let's see, Frank Torres--

Hodge: Cordell is now known for doing audio amplifiers. Is that the same Bob Cordell?

Stanzione: I don't think so. How old is your Bob Cordell?

Hodge: He's probably around seventy I'd imagine.

Boddie: It could be. He was into audio.

Stanzione: He was a big audiophile, so--

Hodge: The Bob Cordell I know wrote a book "High Performance Audio Amplifiers" I believe is the title or something.

Boddie: It would not surprise me.

Stanzione: Yeah. I don't know. I don't know. But Bob was certainly into audio, very capable, just extraordinarily capable.

Hodge: So, going back to the design of the chip itself. And certainly add into this if you thought there was something noteworthy that I'm skipping over. But when I think of DSPs, sort of the defining characteristic for me is that they had a combined multiply-add instruction. So, how did you guys (a), go about determining that you needed that, and (b), implement it? And what did you think was novel, interesting, etc.?

Stanzione: Well, A was easy. B a little bit more difficult.

Boddie: Yeah. We had a starting point from this design study group, which had as list of applications that we might be able to fit into. So, we looked at those applications and extracted from the applications the key algorithms that we'd want the DSP to implement. And we studied how would we implement these algorithms in terms of multiply adds. And in refining those algorithms or boiling those algorithms down into prototype instructions, we were able to refine the architecture and instruction set.

Stanzione: The A part-- your A question of how did we determine multiply adds, we didn't invent digital signal processing. It was around for-- people knew how to do digital signal processing. And it was clear multiply adds were part of that.

Boddie: Well, when I came in--

Stanzione: So, that was a given. How best to do it is what Jim really focused on and focused on successfully.

Boddie: My original proposal was to consider a floating point arithmetic unit. And I guess my first assignment in Dan's group was to do that. Floating point arithmetic had a lot of advantages. When I was doing all this array processing work, most of the signal processing was done in floating point. So, it's easy, first of all, with floating point arithmetic you don't have to deal with all the scaling operations. Another consideration from the practical point of view of implementing a single chip DSP is that the multiplier was considered to be a fairly expensive part of the circuit in terms of speed and area. So, floating point had the opportunity to allow you to use a smaller multiplier to get the same effective performance for dynamic range and precision. It had-- floating point arithmetic had some noise advantages, limit cycle advantages. But the disadvantage of course was of the complexity of floating point. And so, ultimately we decided that for the sake of complexity and just getting it done, we would use fixed-point arithmetic with a word that was larger than sixteen bits.

Hodge: So, on that part about the high precision arithmetic capability, you talk about one of the applications is line treatment. And you talk about the treatment of voice frequency metallic circuits. What were those, and what was the DSP doing for those?

Boddie: I think that the application, if I recall correctly, is there are certain customers in the Bell System that wanted audio quality on their telephone lines that were higher than standard POTS voice over your handset. This would be like a radio station for example. And the line treatment is a way of equalizing and essentially doing signal processing that made the quality of that circuit better than standard POTS.

Hodge: So, is that along with-- you talked about doing transmission system measurements. Are those related to what would later be done in DSL to sort of equalize-- I don't know if it's to equalize the line or to equalize the transmission to correct the-- sort of pre-correct the signals, or pre-distort them?

Boddie: It's not my area of expertise, but--

Stanzione: I don't know if the two are similar. Certainly DSL requires a lot of signal processing, but I don't know if it's similar applications to what you referenced.

Hodge: Okay. Another thing you mentioned was speech synthesis. So, around the time you were developing this, I think the TI Speak and Spell came out. Was that something you took notice of, had any impact, anything?

Boddie: I don't know the impact, but we were certainly aware of it.

Hodge: Well, those were some of my main questions regarding the DSP itself. Is there anything sort of from this timeframe that I may have missed that you'd like to add on to, people that we have yet to mention, things like that?

Boddie: Before we get to that let me just elaborate on what we just touched on in terms of speech synthesis. One of the benefits of our DSP development was getting this technology in the hands of Jim Flanagan's department in the acoustics research. So, we gave the research department at Bell Labs a platform in which they could further develop all kinds of speech applications, speech recognition, speech coders, music coders. And instead of having to rely on array processors, they could now demonstrate the feasibility of pretty sophisticated speech processing using this platform. So, back to the people?

Hodge: Yeah, I know we've sort of touched on some people by names of function. And when we were originally going to put this panel together, we had a lot of people that we discussed. But we have to keep the panel fairly small. So, just who else was really instrumental in this in terms of their contributions?

Boddie: As far as the device design, we divided the design up into several team members. I did the arithmetic unit. The address arithmetic unit was designed by Gobind Daryanani. John Thompson had overall responsibility as lead designer.

Hodge: Sorry to interrupt your train of thought, do you know how to spell that Gobind?

Stanzione: Gobind, G-O-B-I-N-D.

Boddie: Yeah.

Stanzione: Daryanani-- I have the directory here so--

Hodge: We'll cross-reference that later.

Stanzione: India by the way, back to this flavor of the demographics of the technical people involved.

Hodge: Sorry, go ahead.

Boddie: Steve Walters did the I/O design.

Stanzione: Here, this may help you.

Boddie: Yeah, sure.

Stanzione: Remember people as you're going through it. But while the-- I know Howard Kirsch was one of your significant circuit designers, was he not?

Pedersen: Howard Kirsch was the advisor on the RAM.

Stanzione: On the memory design?

Pedersen: Yeah.

Stanzione: But you have a number of key design people.

Pedersen: I had a-- my lead designer was Jack Kane.

Boddie: Oh yeah.

Pedersen: And Tom Bartoli, Ron Freyman, Jack Grant, Bob Kershaw, and Don Cutress was responsible for testing.

Stanzione: As you go through the names, I remember the names.

Pedersen: And then there was another designer brought in for the memory whose name escapes me at the moment.

Stanzione: There were just general people without getting into specific names. It's just when you're at that stage in your career, early, you just assume the world around you is the way the world is. Having lived a few more decades beyond there, I just realize what an incredible array of talent we had available to us as we did this. It was more than just this project I guess. It was Bell Labs at the time.

Pedersen: Frank Barber.

Boddie: Yep.

Stanzione: Is that the name?

Pedersen: Frank Barber's the name.

Stanzione: Just an incredible array of talent of people that we worked with. And so, while we're representing the effort, both the intellectual capacity as well as the ability to collaborate of the people was really extraordinary. Jim--

Boddie: Ismail Eldumiati came on.

Stanzione: An Egyptian.

Boddie: Yeah. He came on and helped Renato on the--

Stanzione: I remember that well, he was--

Boddie: Fault coverage analysis, and later on he took over the I/O.

Hodge: Dick, were you working on other projects as well at the time? Were there other chips that you were--?

Stanzione: None more important.

Pedersen: None more important, that's right.

Hodge: Talk on any other projects were you also trying to --

Pedersen: Oh, I don't remember the details. We were still doing bipolar design, too. But this was the main effort.

Hodge: How different was it doing the bipolar versus the MOS designs?

Pedersen: Not very. You start understanding the characteristics of one device versus the other. And you look at some basic circuit design techniques. And I think the biggest difference was used-- my recollection is thirty-seven years old. But I think we used a unique fast carrier technique that we couldn't have done in bipolar with through transmission gates.

Hodge: What was the state of semiconductor research at the time because obviously you invented transistors at Bell Labs and then sort of your later history is just to the West Coast with Shockley and Fairchild, but how were you guys progressing internally at that time?

Pedersen: Well, there was still research being done at Murray Hill on semiconductors. CCD invention was there.

Stanzione: Oh, there's a lot of different technologies at the time. Certainly charged-coupled devices were there at the time.

Pedersen: Charge-coupled, yeah.

Stanzione: You may remember bubble memories, which were magnetic not semiconductor. They were magnetic. A lot of innovative technologies.

Pedersen: And then of course there was III-V research too, III-V compounds, which I became involved with in 1990.

Hodge: This is where I show my ignorance of chemistry. Is that gallium arsenide?

Pedersen: That is. Yep. Gallium arsenide.

Hodge: So, back after a brief break. Some of the things that we didn't get to talk about on the architecture of note, one, the DRAM refresh. So, DRAM tends to just-- will this memory die if you don't refresh it. So, what was-- you had the innovative solution for that refresh.

Boddie: Right, which was essentially to allow the memory to be refreshed by the program. In typical applications, you're hitting every DRAM location of interest every hundred and twenty-five microseconds. So, for most applications, most of the memory got refreshed automatically by reading and writing it in the course of the [application sample] cycle. Occasionally, you'd have an application that'd have a memory location that you would need to put away for a while, but it's just a matter of the programmer being aware of that and adding something that kind of touched that location in the course of every cycle [sample].

Hodge: So, if I put something in memory and then forgot about it, it would forget about me.

Boddie: Right.

Stanzione: And the purpose was to save the silicon space, right?

Boddie: Absolutely, we-- for DRAM, the-- we didn't have to have logic to do the automatic refresh. And I believe that most of the time the refresh would take up time if you had it.

Pedersen: Plus, we certainly didn't want to go with a static RAM.

Boddie: Right.

Pedersen: Just could not afford the silicon area.

Boddie: Yeah.

Hodge: Another thing, you had done some interesting stuff-- so, this processor had to be programmed to the assembly language-- or in assembler. But you changed the mnemonics around of the instructions from the normal sort of I guess add/move whatever.

Boddie: Yes, most of the programs for the DSP were going to be small to begin with. So, it didn't make much sense to develop a full bore compiler for it. I'm not sure it could support a real compiler. So, application program would be in assembly language. Developing our-- the instruction set to begin with, we used symbolic notation for indicating data moving from one register to another and for doing multiplies and the adds because we didn't know what the instruction set was going to end up being in the first place. At the same time, the Mac 8, the Bellmac 8, which is our eight bit microprocessor, used a C like mnemonic [notation]. Instead of using three and four letter mnemonics for move, add, etc., it used actual C notation, so symbolically. We decided to adopt that for the instruction set for our DSP as well. It made the code very easy to read because most of the applications are arithmetic in nature. And you're just looking at the equation on paper as the code.

Hodge: Was it related to any of the more formal notations that had been developed, I want to say like-- I'm going to mess this up, BNI that was like--?

Boddie: No. it is more like traditional C. we used the notation of a star before a registry name to indicate that registry uses a pointer to addressing.

Hodge: Okay. And you have a little--? [refers to programmer's reference card shown in video]

Boddie: Yeah, we have the--

Stanzione: Oh, you still have one of those.

Boddie: The instruction set card here. So, every instruction-- because the device is highly pipelined and a lot of parallel operations, a typical arithmetic instruction would have several fields for controlling the arithmetic, for controlling the addressing of X and Y data, and controlling the data back to memory. And each of those fields were filled in with symbols that looked like equations or C-like notation.

Hodge: Okay. And then when someone, despite that, inevitably messed up and had to go through the debugging process, what was your debugging tool chain like?

Boddie: That was I think one of the fun things in my view.

<laughter>

Hodge: I've never heard anyone call debugging fun before.

Boddie: I had the opportunity to man the hotline, which customers would call if they thought they had a problem. So, they would call, and my job would be to take their problem and boil it down into a simple set of instructions that caused the failure using the DSPMate, and so on. My first job was to try to duplicate the problem in a reduced form, and then next, make some hypothesis about what could be causing this problem, what is failing in this device--is it a gate on page thirty-nine of the arithmetic unit?, and then write additional programs that would be used to kind of verify the hypothesis. Next, we'd go to Allentown and get on the probe machine. And now, they would have an idea of what to look for in terms of potential failure modes. And sometimes, we'd go at Allentown and have to modify test vectors on the fly to try out different things to see what the problem was. Wafer probing-- it was actually possible to put down probes at the middle of these circuits. The geometry is large enough that you could actually probe individual outputs of gates to see what the potential problem was.

Hodge: Did you use your own probe? Was your silicon test equipment-- was that-- did you build it yourself? Or did you buy it?

Pedersen: No, it was commercial equipment with custom design probes that fit this chip. And then we had special probes that could be moved around very carefully to hit lines internally. But it was a commercial, I believe Century, test set.

Boddie: Yeah.

Stanzione: Did you have as much fun as Jim did doing this?

Pedersen: No.

<laughter>

Boddie: There were some great guys on Dick's team that would come in and say, "Oh, yeah." And then they'd get under the microscope and say, "Oh, I see it. I see it. This is a design violation here that we missed."

Stanzione: Really?

Boddie: Yeah. And then back, after collecting a bunch of those, we'd be going back to the mask shop for another iteration. Our first-- I guess it was the second silicon that we received had a problem in reset. The reset mechanism was broken. And Dick's people figured out a way to work around it by wire bonding from the reset pin to somewhere in the middle of the die. So, we had working chips that had this strange little wire that went from a pin to the middle of the die.

Hodge: So, just a couple things I want to touch upon quickly as we have just a couple minutes left. So, you gave a lot of-- not a lot, you gave some of these away to universities. For you, did much development work spring out of that?

Boddie: It's hard to say. But at the time, there was not much else for universities to use. And I'm sure we did benefit them in getting their DSP programs off the ground. We gave numerous development systems, emulator boards, to universities all over the country.

Hodge: And how much use did this wind up getting used internally at Bell Labs?

Stanzione: Gee. So many-- you mean how many applications, really? Developers really built this into their systems.

Boddie: The first silicon [devices] were distributed to over forty customers in Bell Labs.

Stanzione: I would guess there was a point where if you were doing digital signal processing in Bell Labs, it'd be unusual to not be using the DSP to do it.

Boddie: Right.

Stanzione: Would you say that's a fair statement?

Boddie: Sure, their alternative at the time would be to not use a DSP at all and do full custom design or to buy a DSP from the outside, which they're perfectly free to do. So, there is real competition. Our customers were not locked in to us. But at the time that we introduced the product, there wasn't much for them to use. There weren't many alternatives, see.

Hodge: And you have a chart there of follow-on designs. Can you hold it up? So, you had designs going forward several years into the future. How long did you wind up continuing to do DSPs internally at AT-- at Bell Labs?

Boddie: As long as Bell Labs was Bell Labs. We-- the microelectronics portion of Lucent Technologies was spun off into a company called Agere Systems. And they continued DSP development until their-- they were acquired by LSI Logic.

Stanzione: In the '90s.

Boddie: In the '90s, late '90s.

Stanzione: So, the effort that we and others began became a business unit in the semiconductor part of AT&T and then Lucent. And Jim, you actually led that for a while if my memory serves correctly.

Boddie: The design, yeah.

Hodge: So, certainly a very interesting legacy that came out. I know whenever you say the first, there's always some dispute over, debate over what makes a first or not. But I know a lot of people credit you guys as having either the first single chip DSP, if not the first, certainly one of. And you have a very interesting history and very interesting set of requirements that prompted it. Unfortunately, we're running out of time for today. One of the things, anyone reviewing this transcript is encouraged to look at the Bell

Labs Technical Journals that describe all of this. They're freely available online to anyone who is interested.

Boddie: This is the September 1981 [issue].

Hodge: September 1981 has a lot of design detail that we did not go into here. And as we heard earlier, Jim's basement has a lot of papers that you're welcome to go to his house and pick them up.

<laughter>

Hodge: Is there anything else I missed briefly here then that you guys would like to bring out?

Pedersen: I don't know I just feel fortunate to have been part of a great team. And I think that's what it was. It was a great team.

Stanzione: My sentiment is exactly the same, just we were really lucky. We were lucky to have each other and the other many, many people that we were fortunate enough to work with.

Boddie: Yeah, it was a great time to work.

Stanzione: Yeah.

Hodge: Well, thank you gentlemen for making time out of your schedules to gather here. And hopefully, you've all found this interesting.

Stanzione: Yeah, thank you.

Boddie: Thank you.

Pedersen: Thank you.

Stanzione: Great.

END OF INTERVIEW

