Oral History of Sridhar Vajapey

Interviewed by:
Paul Sakamoto

Recorded: March 10, 2016
San Jose, CA

CHM Reference number: X7768.2016

Paul Sakamoto: Hello. Today we're going to interview Sridhar Vajapey on behalf of the Computer History Museum Oral History project. Today Mr. Vajapey is going to discuss his history and involvement with the Texas Instruments microcontroller development over the years. Currently, Mr. Vajapey is at Oracle, where he is Vice President of Hardware Development. So without further ado, I'll introduce Mr. Sridhar Vajapey.

Sridhar Vajapey: Thank you very much, Paul.

Sakamoto: So Sridhar, if you could, please, tell us a little bit about your early background and history and how you went through the Texas Instruments organization helping develop their microcontroller technology.

Vajapey: Surely. This was back in the early '80s. I graduated from University of Houston, and then stayed in Houston, joined Texas Instruments, which was located in Stafford at that time. At TI I joined the microprocessor and the microcontroller group, which was then an integral entity. Spent about 15 years there in various roles in the company. Starting from a systems design engineer through chip design and then became a design manager and applications manager. And later in life, 15, 16 years later, joined the Sun Microsystems, which was acquired by Oracle, and been there for another 16 years. And today I'm Senior Vice President at Oracle in the Hardware Development group, managing SPARC products.

Sakamoto: So perhaps along the way you met some people who were mentors to you or helped guide your career path. Maybe you can talk a little bit about some of your influences as you are coming up through the ranks.

Vajapey: Absolutely. So my observation is all the people that I've met who have actually achieved quite a bit, most of them have had some sort of mentors, some guides, people who show them the way. And I also felt that having good mentors was a good idea, since coming out of school I didn't know much anyway. And at TI, while at TI, walked into the company and Peter Linder, from MIT, he was my boss and a brilliant, absolutely brilliant person. So when I was in the microprocessor group managing their design, portions of their design for the 32-bit processor, Peter Linder was an excellent guide.

And then later I switched over to the microcontroller group, after the processor group wound up. And a person, Mark Stambaugh, who is now in Spokane, Washington, he actually kind of was a very thorough person, got into the intricate designs, details of the design, and kind of the thoroughness as designs got more complicated. That was just something that I hadn't done. It was more of reliance on tools, and now it was more of, "Let's get on the whiteboard and let's start drawing gates." And then let the tool go do the verification. So it was kind of - Mark came up in a different way, more detailed, hands-on, pen and paper type of guy at the time.

And then later on Richard Chang really was another mentor who kind of taught me the very basics of management. One day he caught hold of me and said, "You see all these people out there? They know how to do design. You don't need to get into this detail." Everything Mark Stambaugh taught me, he said, "No, no, stop. You need to go teach these people how to go do design and what methodology, the flows and tools to use." And showed me the basics of management. Today I look back and wonder, you know, it was a very simple message, but he got that message across to me.

And on the business side, most of it I learned at Sun and Oracle, the mentality was quite different. Anant Agarwal, who's today in Khosla Ventures, we have also done a oral history of him, who ran the SPARC group. [He] kind of pushed me into thinking about dollars and cents and why it did or didn't make business sense to do something. Even at TI, we did a lot of it, but this was more methodical since SPARC was so integral part of Sun and ran so much business for Sun, we started thinking at a much higher level, at a company-wide level. And then later, David Yen and Mike Splain – different mentalities but came from different angles. Everybody's different, and I learned a lot of things through my career through these mentors.

Sakamoto: Is there any other sort of figure in your career that kind of stands out that was kind of a particularly memorable sort of person?

Vajapey: Two people actually stood out. One is Rich Templeton, CEO of Texas Instruments today. He was not actually a mentor, per se, but more of a person who used to drag me every Monday afternoon at one o'clock to sit down and go over what I was doing. He knew what I was doing was more of a CAD flow and methodology at a time things were handcrafted, or they weren't good [design] flows, you know. How to think. And TI's CEO Templeton, an extremely well-thought out individual. And not only that, his memory is like a steel trap; always remembered things. And that was just fascinating. He was a breed apart. The other one was Gregg Lowe, who was my immediate boss at Texas Instruments. Became CEO of Freescale, and he was another amazing individual where most everything he did turned into a lot of money. Whether it was profit or margin, revenue. He was another one who stuck out and was an exceptional individual.

Sakamoto: So it was a great, an interesting time, wasn't it? A time when there's a lot of things coming together. Had a few companies that would later turn out to be some of the Nexuses of development for the semiconductor industry, and certainly you've been at two of the really big ones.

Vajapey: Absolutely. Two big ones – name brand icons. And by golly, we entered, at least me and the group I joined at TI, was hiring a lot of people at that time. We joined there and, it was , like eight micron technology running there, five micron technology, and today we're running, what, designing into 10 nanometer technology. So come a long way. We've seen technology come a long way, and it's been very exciting bringing up new fabs, ironing out tough technology issues, been extremely, extremely rewarding and enjoyable too.

Sakamoto: So going back to the beginning of that, maybe you can take us through a history of the processors and controllers that you worked on at Texas Instruments, and then maybe go on from that to kind of compare that to where you saw the rest of the industry at.

Vajapey: So TI itself had been in some sort of computing business for a very long time. If you think about it, TI introduced calculators which I think almost everybody has, well, purchased one at one time or another. And they started it back in early '70s. They were very rudimentary calculators, but nevertheless it had a very simple limited function computing capability. And then there was the Speak & Spell. I know my kids carried them around, and that was a very simple set of codecs rolled together with some kind of a control function. You know, it could do Speak & Spell and then later it did Speak & Math. And those products sold quite a bit. It was almost like you told a kid, "Texas Instruments," and the first thing the kid would say, "Speak & Spell." So rather iconic products at the time. And we churned out a lot of those. I wasn't, per se, involved with it. I joined a little later, but when I joined TI, those products were big time in the market.

Later in time, around mid-'70s, TI introduced the TMS1000, which was a 4-bit processor. And it actually got into a lot of applications, very basic automotive applications. Of course, with 4-bit capability and a small chip, it did what it did at that time. And it was an old PMOS technology, consumed lot of power. But nevertheless it was a microcontroller and it did a lot of functions. And at the time it reached a record of something like 400 kilohertz. We talk in gigahertz today. And the follow-up from the TMS1000, the 4-bit controller, TI proceeded to do 8-bit controllers and processors. And one of them was the 9900. It was used in a lot of places. This was supposed to be the single board computer. It was very iconic. It got out there and, boy, I ran out and got one when I joined TI. All it was was a keyboard and it had all the functions in it. And all it did was connect to a TV and lo and behold, you had a computer.

Sakamoto: Is it fair to say that TI was a lot more focused than on sort of the consumer segment than, say, some of the competition?

Vajapey: Absolutely. And in fact, the Speak & Spell was in every kid's hands. The TI 9900, one of the things it was deemed [targeted] for is just to be a home computer, something you could do [use] at home. And I recall it had a basic compiler [interpreter] or something built into it. It did a few things. I actually never did any real computing other than play with it. And after that was the real general purpose microcontroller, which was the 7000 series controller, which was, which started out to be, an 8 micron or 5 micron something out there NMOS type device. Consumed a lot of power and later on it actually, the Japanese team, TI had a number of teams at that time, the Japanese team went and took that processor and made it a CMOS processor. The core was called Shogun and it was--

Sakamoto: Hm. Just for the record, where was the Japanese team located? I mean, in Japan. But was it at Miho or…

Vajapey: There was, TI had, a number of facilities. One was in Miho. And I believe this team at that time was in Miho or somewhere around there.

Sakamoto: Okay.

Vajapey: Close to Miho. And it was a good team, and the processor core itself was very compact. It was CMOS. It was very power efficient. And that's when I transitioned into the TI microcontrollers. But between the 9900 and the 7000 portion of my life, at least, what TI was trying to introduce was one of the first 32-bit processors. In fact, I was in that team, hired in, the team started around '83, '84. And I also started my career there at TI in the microprocessor group in '84. And we had two processors at that time.

One was a 32-bit processor with a 16-bit I/O bus going outside the chip. The other one was a 32-bit. In fact, it was a small chip. This is the 32032 chip, a rather compact--

Sakamoto: It's--

Vajapey: It was more square-ish chip.

Sakamoto: Hold that steady please and we'll try to zero in on it. Wow.

Vajapey: This was a 5 micron NMOS type technology. It was available in a CLCC package. I forget how many pin. I think it was 68-pin or something thereabouts. And the other one was the 32016, which I mentioned earlier. Also, a four-inch wafer. Boy, we thought this was a large chip at that time, and just for sake of comparison, today the product that Oracle just launched and showed off proudly is that size [a single Oracle chip is the same size at the aforementioned wafer of chips].

Sakamoto: Oh, wow.

Vajapey: So a big [chip]-- we have come a very long way. This [the Oracle chip] is a 20 nanometer chip, which was showcased in the Oracle OpenWorld. And we've given out a lot of samples for a lot of customers as well. But we've come a long way. And these [the TI 32016] microprocessors were supposed to be the first one we second sourced from National [National Semiconductor Corp]. Texas Instruments actually didn't design this. It was taken from National Semiconductor, as a second source and between us, the thought at the time TI management had was that we could go and make big business out of it. It didn't work out that way. So eventually TI decided to shut that group down and move me back into the microcontroller group.

Sakamoto: Just as a point of interest, what were some of the obstacles for the 32-bit product at TI, and apparently at National too?

Vajapey: So the first one was… I think there are couple of notable ones in this business. There's that slogan, "Death to a New Instruction Set" architecture, right? And what National and TI brought to the market was essentially something that was different. Intel, on the other hand, had already had had the 8080, the 8086, getting over to the 80186 and the 80286. And it was a sequence of processors with a fairly uniform instruction set, although it was a growing instruction set and was going from 8-bit to 16-bit to then 32-bit. So the transition for those using that processor seemed a lot easier. TI also came up a little slow on this processor. I should know. I spent a lot of time in the fabs at that time. This chip was actually designed at National for the National fabs, and we were trying to adapt our technology so that we could take that database and make it. So there were a lot of circuits in there that it took us time to go work out. The clock skews, try to work out a lot of design- circuit design and layout design things, to get the yields up and get the speeds up. And at the time it was going to hit an all-time speed of 10 megahertz. Again, today we are at four-plus gigahertz. But it was an achievement at that time even to hit 10 megahertz. So that was, I think, the big obstacle. We were already playing with a very small market. We came into the market. Although we came in very early into the market, the people had to get new compilers and assemblers, linkers. They had to get new tools, and the Intel products already had a roadmap of development and products out there. So that was an impediment.

Sakamoto: Now, we'll get back to the microcontrollers here in just a moment, but I wanted to-- maybe you mentioned it and a missed it. I apologize if I did. Was this based on NMOS or CMOS technology at the time?

Vajapey: This was based on NMOS technology at that time.

Sakamoto: Okay.

Vajapey: TI did earlier PMOS, then NMOS, and then of course converted everything over to CMOS.

Sakamoto: Okay.

Vajapey: We never actually did the CMOS version of the 32000 series processors at TI, although National went on to do the 32332 and the 532 products, and I believe some of those designs converted over to CMOS actually.

Sakamoto: Okay. So kind of rolling on past the processor, the pure embedded processor world, I guess then you finally crossed over to work purely with the company on microcontrollers.

Vajapey: So [the] microcontroller business today, microcontrollers is a very large business. And even at that time when there weren't too many electronic gadgets around, you can believe how much demand there was for microcontrollers. It offered a very cheap method to get digital products into the consumers' hands. And the TMS1000 kind of laid the foundation for TI into the general purpose microcontroller space, and then the TMS7000 followed into it. TI made a good name in the automotive sector. There were lot of commercial products also available on it. In fact, the TMS7000, the product that I worked on, the first microcontroller, the TMS77C82 product, actually was an integration of EPROM into logic, into the basic microcontroller Logic. It was actually the second one. I don't know who came out with the first EPROM with a Logic integration, but I would believe the TMS77C42, which proceeded the 77C82, was probably one of the first. And at the time, I had to finish the C42 design and do the C82 design. It was just pretty interesting in itself. I came loose from a very large microprocessor team, and going into the microcontroller team I was surprised that my boss gave me three people. And said, "You don't have to design the core. You don't have to design anything. You need to rip out the ROM and plop in the EPROM [Erasable Programmable Read Only Memory – required UV light for the erase function], redo some of the blocks on the chip and tape it out." So we tried to do as best compaction as we could. One of our customers was Data I/O, was then doing the EPROM version, EPROM programmers themselves. Then we had Bayer as a customer, who later reused the ROM version. They were either doing the blood pressure meter or the blood glucose meter, one of the two, with the product. So we had a bunch of products roll out in the consumer space. It made a lot of noise in the market because it had EPROM integrated in to it.

Sakamoto: Mm-hm. Now, if I might, you know, although you and I are old veterans and we completely understand EPROMS and bipolar PROMS and other things, maybe you could just give a quick explanation of what an EPROM is, because someone watching this video might not come from the era.

Vajapey: Ah. You're right. So today there are a number of different types of memories. And EPROM is a type of memory. You have the ROM, which is a read-only memory, which is just hardcode programmed and all you can do is read from it. There's no write possibility in it. Then you have a PROM, which is a one-time programmable read-only memory. So you're able to just raise the voltage up. I think we used to put 12 volts, and then it had a floating gate, and you would program a charge in it, and the charge remained in the middle there. And it would then act as a read-only memory once you programmed it.

EPROM was an erasable programmable read-only memory, where you could program the memory to whatever data you wanted. And then if you wanted to erase it and reprogram it, you put it under UV light for about 25, 30 minutes and all the charge would dissipate. You put it back into a Data I/O programmer and then you change your code and you reprogram that into the device and you put it back in your system. It was basically socketed in a ZIF [Zero Insertion Force] socket. You'd zero insertion force socket, ZIF socket, put it back. You'd boot it. If you found an error in it you'd go erase it again, reprogram and put it.

Today, of course, after the EPROM, it changed over to the E^2PROM, mainly used for data initially. And now you have the flash as well. These are different types of memories. So this was one of the early EPROM, usually in the early days, you had an EPROM side by side with a processor and with an external bus connecting to it. Putting the EPROM inside the die, now allowed a cheaper method, a lower-cost method to do a emulation of your final design before you actually went back to a company like TI and said, "Remove the EPROM and put this code into the ROM," which was a smaller device and a cheaper device and fewer process steps people make .

Sakamoto: So now at the time, I happened to have been at Intel, which was kind of the other side of the curtain.

Vajapey: <laughs>

Sakamoto: From your group. And I remember that those were like really key features, differentiating features of microcontrollers in general, is that they had both the EPROM and the permanent mask program, ROM version, of the device.

Vajapey: Right. And at that time, everybody in the planet seemed to have microcontrollers. Intel, you went on the other side, Intel had the 8032 and then the 8048 and the 8748, the EPROM version of that. Which I happened to use later in life, by the way, for another product that was designing <laughs> as a program memory.

Sakamoto: Yeah.

Vajapey: NEC was in the market, with processors. AT&T made their own controllers. Toshiba made their-- everybody seemed to want to make controllers. Of course, today that's come down to the few, as time has passed. But it was a very interesting time.

Sakamoto: So what were some of the challenges of designing and manufacturing the TI microcontroller line at the time? Because, you know, you mentioned the fab technology, which doesn't sound that challenging today, but that was a different era. There's also combining technologies of Logic and Non-Volatile Memory. And there's just managing the flow of mask program ROMs. There's all sorts of things that come up and they're different for different companies. What were some of the challenges that were high on your list?

Vajapey: So having been through so many generations of processors, you know, we've found that there's always something new you add and then you struggle through that development. In this particular case, in the case of the TMS 32000 series, there was the fab-to-fab, National fab to TI fab, mismatches in the product transistor and so on that we had to work out before we stabilized yield. In this case, it was one of the early processors where EPROM mixed with Logic. Until then the memory fabs were one side and the Logic fabs went on the other side, and then we had to mix it. So we went through quite a few process learning cycles before we could make sure that the EPROM charge from the floating gate didn't leak out.

There were some things we learned where in this particular case we kind of invented the low-power capability, meaning you could put the processor in sleep mode or standby mode. We had a couple of modes. In fact, I was forced to introduce that in the TMS C82 product. And it was key when we said low-power, it was literally had to go down to nano amps. And we found that where we put the wells, N-well to P-well, it was too close and we were leaking. So we had to go back and move those out. We did some of those learnings when we went and did that. Until then power didn't seem to matter until we went into these handhelds. And at that time, all of a sudden it mattered.

Sakamoto: Interesting. Yeah.

Vajapey: So later, of course, there were a lot more technology issues as technology grew. And I'll talk a little more when we did the TMS370 project and the SPARC project, which TI also made the SPARC processor, for a very long time. So TI was in the computing business for a very long time.

Sakamoto: Okay. So going past what sounds like was kind of the beginning of the mass market for these microcontrollers, what was the next evolution in both TI's roadmap and how that played off against the competition?

Vajapey: So the TMS7000 kind of laid the foundation into a lot of automotive business and lot of other businesses. And these were little calculating machines and so on, so forth. We knew that the 8-bit processor that the 7000 series had was just a little short in horsepower. And as the feature set needed by the customers grew and the technology moved from generation to generation, there were more features capable to be added in the same real estate for the same cost. We were driving the cost down very rapidly. So the 7000 itself, I believe, was a five-micron technology. We took it down to three micron technology at that time, and oh, boy, we could put a lot more features in it. But then at some point we just needed to change the core of the processor. So the TMS370 product was then introduced, which was an 8-bit controller, which was actually started by an advanced development group outside of our group, by one Dr. Mohan Rao, who actually ran the TI's Memory group as well. Very brilliant gentleman who has achieved a lot over time. And the TMS370 was also an 8-bit processor, a full dynamic logic, no CMOS static Logic. So it consumed power. And the aim was to go unseat Motorola from, the automotive [business] hold it had. And it was quite successful. I was made the design manager of that group.

We had about 16 or 17 different families of products. The families were defined by the amount of memory size, the number of serial communication ports, number of I/O ports, and so on. So that's how we differentiated it. And each family came in a type of package so that a customer could go and replace the package with another chip of the same type of package, get a few more features like an added timer, like an added communication port and so on, so forth. It was very successful. We actually shipped a lot of product, and we made a lot of inroads into Delco Electronics, which was the automotive supplier. And Ford. And [we] got into a lot of automobiles in Japan and in Europe as well. So we did very well. And TMS370 itself was a plain logic chip, differentiated on memories and logic itself. So from a process technology perspective, it didn't throw us new challenges like the 7000 EPROM integration did. It was more of a, "Let's get it out to the market." It was a new, beefier core of that day.

Sakamoto: Sure. Now, did those computer cores of those processors once again use a proprietary TI instruction set? And if so, how did that get handled?

Vajapey: At that time there was no such thing as an open instruction set or a proprietary instruction set. Every company had their own instruction set built in, and it was built into the compilers and assemblers, linkers, everything was built on top of an instruction set. And they carried that forward. And I think in the case of TI, the 7000 and the 370 broke that instruction set continuity, although I can't say the difference was too large. But it was a different architecture, even though both were 8-bit. So for us, the challenge was creating development kits, creating application notes, and lot of that was newly done, so to speak, for the 370. And then moved out into the market.

We did a lot of business on that processor and that core itself. I think that we completed that design, that and TMS, again, 370C16, but Prism architecture, overlapped over time from going to market. And, of course, the 16-bitter had a lot more horsepower. It was fully static design. And that one gave us a lot of technology challenges. A very huge amount, because we said we were going to pull in a lot of board components and it's going to solve a lot of things. So, for that processor, called internally Prism at that time, and called CASM at our supplier, which was configurable, application-specific microcontrollers. We tried to pull in components like the cluster driver transistors in an automobile cluster. So they were loaded in. So it had logic, it had analog for a lot of A to D. And then it also had power transistors inside the chip. So it threw us on a spin. Getting such a complicated process was quite difficult. Went through many, many learning cycles on this one. Eventually it was very successful. We got it out to the market, but it didn't come without a lot of blood, sweat and tears.

Sakamoto: Interesting. So, it seems like your fab technology challenge was more centered around having multiple technologies on the same die, almost like a mixed-signal device, if you will, while some of the competition was really just chasing photo lithography and scaling?

Vajapey: Correct. We actually felt that bringing this mixture of technologies, mixed-signal, as you said, putting it all on one chip, helped the customer reduce the board cost. And I'm sure, although I can't, I don't actually have the numbers of what the customers did, the few exercises we did on subtracting component costs and so on, so forth, certainly showed a lot of benefit to the customer. In fact, it was one of the key differentiators we were taking out there to the market to actually unseat the incumbents. And Motorola certainly at that time was the 800-pound gorilla sitting in the room, and we needed to shake that loose.

So pretty much between when we started designing this somewhere in '91 and then got it out to the market '96, it was a new core, new process technology, a lot of new things there in that chip and in the technology. We managed to actually get design wins all across the board in the automobiles. So, when we first introduced it, our first design win was at Delco Electronics. Then we proliferated it into Ford, and then into Japan, with a lot of customers there in Japan. Took it to Europe and then got a lot of design wins, including we were at Bosch in a number of applications. In fact, that's interesting too. Then-CEO of TI, Jerry Junkins, visited Bosch and then he actually died in the car after the business meeting. After our microcontroller business meeting. And that was, that was quite a thing for our group at that time.

Sakamoto: Yeah.

Vajapey: So, We pretty much got designed into a lot of majors. And that played in our favor quite a bit, proliferating our TMS370 architecture. So the key differentiators, again, getting analog, mixed-signal power, into the chip, and getting it out there. One of the differentiators that we focused on was just using a low-cost watch crystal, 32 kilohertz, then being able to run up to 10 megahertz, and whatever speeds later, as it progressed. We introduced that processor in around 1.2 micron technology. So, technology had taken many clicks forward from 3 micron, where we did the TMS7000.

Now we had reached 1.2-micron technology, and then we took it shrink down to 1 micron and then to .8 micron technologies. So it ran quite a bit [faster], and I think if you go out today, I don't know, to any cars around, you know, the years '99, 2000, all the way to 2005, maybe 6, they all had this architecture in there. And my role in there initially was to do the system architecture and get that designed in.

And the second big job was to have a unified test methodology used for production. It was a big thing. We had teams at Houston, Dallas. We had a team in Portland, Maine, doing some of the peripherals. We had teams in India doing peripherals. And then we had a team in Japan and then a team in Europe. So if you wanted to do a quick integration and tape out of a new chip, then you needed it all to have uniform design methodologies, and CAD flows. So after the system architecture design, the role changed. And I think this is the time era when all companies started focusing on productivity gains. Our goal was to have a chip team consisting of maybe four people, or max five people - New college grads with a senior, go tape out a chip flawlessly, after integration. And do that job within four to five weeks.

Sakamoto: Seems pretty ambitious.

Vajapey: It was very ambitious. So when we couldn't do it, I remember, the boss came around. Actually, Rich Templeton came around and said, "You need to go get the CAD team together and run the CAD group. And you need to put in place a flow that would enable, as was termed by another boss of mine, a high school football team to go get a tape-out done," because we're taking way too long. And we needed to support a lot of design wins at that time. Somehow we pulled it off. Perhaps we didn't achieve four to five weeks. Maybe we achieved six. And perhaps we had a few glitches in the CAD flows earlier. Where scan chain would be open. But finally, boy, we nailed that one.

Sakamoto: So that's interesting. I guess what I'm wondering is, along the way, there's always two different ways to arrive at your targets. One is the supplier, Texas Instruments, in this case, comes up with all the answers and then says, "Here you go, customer. This is what you need to have." The other way to do it is you go out and you do a lot of research and as one of the architects of some of the chips, which did you find was the TI methodology and how did that compare with your competition?

Vajapey: There actually were not one set of products. First was we needed [CAD] backplane to design quickly. And we needed sufficient design blocks, which we said would be high-level. Then there were two routes we took to the market. One was, if you wanted to do a car radio or you wanted to do an HVAC system for the car. In those days, if I recall correctly, the HVAC system formed a communication hub with lot of things, microcontrollers in the car.

So automobiles were making a big transition from pure cables of wires to simple microcontroller and 80 cent or an inexpensive at that time, $5.00 component, and a simple serial communication cable between processors. So to cut costs on those high-volume products, we had to work with the customer to actually define what they wanted and make a custom, a configurable application-specific controller. Then we had general market products. The general market products were full-feature products, which you could go to Avnet or somewhere else and you could go buy that product. So the definition of those products were more straightforward. Lot of the attributes were carried over from the earlier TMS370, 8-bit processor, forward and creation of general-purpose processors. And, of course, we needed to be competitive. We knew we needed an extra timer, we needed a watchdog timer, we need-- so it was more that sort of definition of products and where we took it.

Sakamoto: So it's interesting. Like you said, you had sort of a "back plane" or base level of component that would be maybe broadly sold.

Vajapey: Right.

Sakamoto: And then if the big automotive or other makers came in, then you would make something special for them.

Vajapey: Right.

Sakamoto: But sounds like it was mostly in the peripheral interfaces.

Vajapey: Most of the changes were peripheral. We didn't want to touch the CPU. The CPU core is a difficult thing to design. It's a difficult thing to validate and remove all the bugs. And running on the CPU core and the instruction set were compilers and so on, so forth, and you didn't want to touch those. So everything else around the chip, At that time there was no concept of the terminology as SOC, systems on a chip. Which we use today. But at that time that term didn't exist. But that's what we were doing. We're taking a CPU and then throwing a bunch of cores around it. But it was an 8-bit CPU-- sorry. We used to take a CPU core and throw a bunch of peripheral interfaces around it. Whether it was an 8-bit processor, microcontrollers, or 16-bit controllers. Today that's what we do, actually. There are a lot of ARM processors out there. You have different ARM cores just like TI had different microcontroller cores. And around which we built a family of products for different applications.

Another huge differentiation we did, and it was for the first time, it didn't actually take hold too much at that time, but we pulled it out. We could have two cores on the same chip, and that was one of the first we went and did, which is having two TMS 370 16-bit cores on a die. And it was for some mission critical applications where they wanted to be sure the processor was doing the right thing, so we actually compared the outputs of both and made sure they were in sync.

Sakamoto: Oh, that's interesting. So it's a little bit different than the way we might think of multi-core computing today. It wasn't really a throughput issue.

Vajapey: It wasn't a throughput issue. It was more of a critical application issue. However, we learn a lot from it. We actually now had multiple-- we did heterogeneous cores. Those were two same cores, homogeneous cores. Then at TI I went on to do another chip, which was called the MAD chip, the microcontroller and DSP integration chip. Used for cell phones where a lot of DSP compute horsepower was needed for the communication interfaces. And you needed a small microcontroller core to actually do the man-machine interface control functions and so on, so forth. However, in Prism we actually did that first one. And now, of course, having multi-cores on a die is pedestrian. Everybody does it. And in the SPARC chip that we announced, we got 32 of those sitting in there.

Sakamoto: Interesting. It's kind of interesting to see concepts like multi-core computing showing up a while back.

Vajapey: Right.

Sakamoto: And then finding out that there was an evolutionary path. From, "We were able to actually put two of these on the same chip," to, "Oh, my gosh, we should probably do this a little better."

Vajapey: A little better. So at the time, we didn't know where it was going, computing was going. The thought in everybody's mind in all the companies was, "Build a beefier chip. Build a beefier core. Just put more horsepower that way." The concept about having multiple of those and getting more throughput in that manner was still not there. It just wasn't there.

Sakamoto: Do you think that was largely a limitation of the hardware or was it more of a software limitation?

Vajapey: I don't believe there was a hardware limitation, per se. Since we showed we could do it. But you need an operating system to be able to handle it, some kind of scheduler to handle it, to be able to do some kind of a small SMP type stuff. Cache coherency was still in evolution phase at that time. You needed to get multi-port cache memories so that both processors could hit on those on the same memory and yet not trod one over the other. So there was lot of innovations yet to happen. But at least we pulled it out in a small way. And then evolution. You know, once you seed it, it happens. So I don't know when we did it. Probably we taped it out in '95, '96. Probably even '94 time frame. But it was not until 2001 and 2 that really high-performance computing started introducing two cores on a die. In fact, I was involved with that as well while at Sun Microsystems. And that really ran general purpose computing in a big way.

Sakamoto: Mm-hm. Well, still, it's interesting to think of early, or relatively early microcontrollers, having multiple cores on them, albeit for a different purpose. What was the next evolutionary branch of the TI microcontroller family after that?

Vajapey: So this was a time when a lot of computing was needed on structured arrays. Digital signal processors started coming about in the world. And TI was one of the early players, along with AT&T at that time. And then joined in later by analog devices with their SHARC processors. So a lot of R&D money was being funneled into the DSP area. And one of the things, the road that the management took at that time, was to go have a licensing agreement with ARM, advanced RISC machines, from Cambridge. So TI licensed the ARM core. In fact, the MAD chip that I referred to earlier had actually an ARM 7, with thumb instruction set, along with a DSP attached next to it, for cell phone applications. That at least was one of the first. Now, together, we didn't go out to the market calling it the MAD chips. It was called the C2000 line. The C2000 line is a very, it turned out, a very popular line used everywhere, where some kind of heavy computing was needed, transforms were needed, that sort of stuff was needed and a control function was needed. Those chips, that family of processors, I just went into TI website and poked at it. There are a lot of automotive devices. ADAS controllers and so on, so forth, have these processors in there. Of course, [today] those are probably 28 nanometer technology and so on, so forth. But at that time when we integrated it, it was .8 micron technology and 1 micron technology moving forward.

So the road for the microcontrollers after TMS370 to 8-bit and 16-bit line, seemed more headed towards licensing of a microcontroller. ARM was out there selling the core, along with the development kit and the entire chain on top of it, stack on top of it, with application to everybody. So now, the VLSI Technology, LSI Logic, Philips, a lot of people out there, now NXP, of course, were licensing the ARM core. So it seemed like the most obvious thing to do, to not reinvent the wheel on a microcontroller side, but focus on differentiators such as DSP, various DSPs, and invest in integration, fast tapeout, fast productization, and take it out to the market, do cost effective production.

Sakamoto: So it seems then that the original TI microcontroller core had been sort of obsoleted by the end of the '90s and was then replaced with the ARM core. What actually happened? Because, you know, usually products that are shipping millions of units like these don't just die and go away. There's some sort of legacy that carries on. Is there any legacy that carried on past that point, and if so, where did it go?

Vajapey: So I left TI in 1997, but I do know that the 8-bitters were in full volume production. The 16-bitters were taking off, in volume production. And stayed well into the early 2000s in mass production. So you're right. While I, my time frames in lot of things that I'm talking about, refer to the design period, going out into the market, get designed into various applications, and then there is that lifetime production. And then there's the EOL [End Of Life]. The product lifetimes for microcontrollers is very long. Of course, there's the Moore's Law which kicks in and you reduce cost. So as generations go over, you probably saw these controllers have two or three generations of lifetime over which they got their money back.

Sakamoto: Okay. So how about today, here in 2016? Like here-- fact, I should've mentioned at the beginning of the video, we're doing this on March 10th, 2016. Is the TI instruction set still being fabricated or is that line pretty much gone now and they're just shipping ARM? I realize you don't work there anymore.

Vajapey: It's been about 15, 16 years. My guess is that, of course, there are many, many running on the streets. My old car, probably still has one of those or five of those. Not to forget, at the time we started, automotive electronics made a huge change, there was something like two controllers in there, one controller. Today there are probably 100 controllers in a car. And it's only growing. So lot of the cars between 2000 and 2005, 10, they probably got a lot of these parts that are still sitting in there. I'm sure a lot of those early processors have now end-of-life'd. But they're probably now gone down to no production, end of life, many of them. But who knows? There might be something out there. But we know there are lot in use. DVD players, I'm sure that people have 10-year-old DVD players and other players and <laughs> those all have those processors in there.

Sakamoto: Okay. Well, do you have some other thoughts about the TI microcontroller history?

Vajapey: Actually, while that is a microcontroller line where you had first the 4-bitter and then you had the 8-bitters, two generations of 8-bits, which went, yeah, much more powerful 8-bit. And then you got to 16-bit and then you introduced the ARM 7 with THUMB instruction set, so it was basically a 32-bit with 16-bit instruction, I/O and so on, so forth. And 16-bit extensions to keep the code base compact. And then you had the regular ARM 32-bits and all of those that are available on the web.

TI in parallel, in the same group I was in, was also doing SPARC processors, 32-bit microSPARC processors. And SPARC processors are there today. In fact, this is a SPARC processor that I'm working on today. So my boss, Paul Nixon, who's now at NXP, was one of the managers, early managers at TI, managing SPARC designs. So TI and SUN Microsystems jointly designed SPARC processors, both took it out to the market, for general purpose computing. And these drove the early workstations, a lot of workstations, and low end servers, and then as the ultraSPARC 64-bit processors came about at Sun Microsystems, we started doing the two-socket, 4-socket, 8-socket, 16, all the way up to 64-socket systems, which actually drove the .com boom. It was the biggest .com processor at the time. And Sun Microsystems made a lot of money on those processors.

And today, TI microcontrollers themselves, TI has moved away from SPARC, and Oracle and Sun Microsystems, as has been announced, now make it at TSMC and make it in other fabs. Moved on, TI has moved on from the SPARC processor line, and I happened to go from TI to Sun and Oracle and I'm still on the SPARC processor line. TI itself, I notice today, has their own MSP 430 line of products. And those were, those are in meters, smart meters. They're in a lot of places, I looked up online. That seems to be one of the applications it's in. And again, TI is staying, seemingly, with the ARM processor. And it's got, of course, its robust DSP line. And has branched out now into, big time, into analog space as well. So that pretty much has been the TI's business, as far as I know. I've been involved with the TI, and my continuation of that SPARC line inside Sun and Oracle today, into these big chips and SPARC roadmap, of course. It's online and everybody can go see it. And TI also has published their recent roadmap as well. I see their processor line. I Googled it just before I came, and seems to be a lot of products out there.

Sakamoto: Great. Well, unless you have any more last thoughts, I guess that kind of concludes our interview today. So I want to thank you very much on behalf of the Computer History Museum.

Vajapey: Thank you very much.

Sakamoto: For sharing your thoughts about the TI microcontroller line with us. And with that, we'll close.

Vajapey: Thank you very much.

Sakamoto: All right. Thank you.

END OF INTERVIEW

