Altera EP300 Design & Development Oral History Panel

Yiu-Fai Chan, Robert Frankovich, Robert Hartmann, Clive McCarthy, and Don Wong

Moderated by: Stephen Smith

Recorded: August 20, 2009
Mountain View, California

CHM Reference number: X5498.2010

Editor's notes:

1. Y.F. Chan on all CHM documentation ought to be written as: Yiu-Fai Chan.
2. Don Wong is referred by his nickname in the video (Sau) but requests that all CHM documentation use his given name: Don Wong.

Stephen Smith: Welcome to the Computer History Museum in Mountain View, California. Today is August 20th, 2009. Today we'll be recording the Altera EP300 design & development oral history panel. Altera is an early pioneer of programmable logic and today it's one of the market leaders.

I'd like to introduce the panelists. We have Robert Hartmann (Bob), Clive McCarthy, Robert Frankovich, Don Wong, Yiu-Fai Chan, and I'm Stephen Smith.

I'd like to begin by asking each of the panelists to introduce themselves, and give us some background to your industrial experience before Altera.

Clive, could we start with you?

Clive McCarthy: Sure. I began my career in semiconductors by working in an R&D wafer fab for Texas Instruments, straight out of college, which was 1969. From there, as was quite common, I went to different semiconductor companies, and that's how I arrived at Fairchild in the early 1980's. It was from Fairchild that I was introduced to the beginnings of Altera.

Smith: Now we have Don Wong.

Don Wong: My name is Don Wong. I started my career at Intel Corporation, first working on DRAM, CCD memories, later on working on non-volatile memories cell development. Since Intel, I worked at some other companies before joining Altera. After Altera, I worked at Mosel Vitelic and also co-founded Invox Technology. Now I'm semi-retired.

Smith: Next we have Bob Hartmann.

Bob Hartmann: I graduated from the University of Minnesota in 1965, a long time ago. I worked at, originally, at Rockwell when they had a MOS division. In the late 60's, I came to silicon valley and worked for a number of firms here, at Fairchild, at Signetics. Later, after the Signetics experience, started a consultancy company called Source III which was doing gate array design consulting. Subsequent to that, I was one of the founders of Altera. I was at Altera from 1983 to 1994 as VP Engineering.

Smith: Now we have Yiu-Fai Chan.

Yiu-Fai Chan: I'm Y.F. Chan. I graduated from UC Berkeley with a Master and Bachelor degree in 73 and 72. I started my career at National Semiconductor where I designed calculator chips and TV game chips. Then I went to Intersil, which was later bought by General Electric, for about 7 years. I designed all the CMOS products: microprocessors, SRAMS, CMOS EPROMS, and all kinds of CMOS stuff. Then I joined Altera in 1983, and was there for 7 years. After that, I joined a startup called Terra Microsystems for a couple of years. They ran out of money. I joined company called Rambus and stayed there for 7 years. After that, I sort of retired, but then came back to the workforce for a couple of more years at Nucore Technology. Now I'm still semi-retired doing consulting work on the side on a part-time basis.

Smith: We have Robert Frankovich.

Robert Frankovich: I started working in the IC industry in 69 when I was still in college. I started off working at Fairchild and stayed there for about 7 years. Went to one startup called Umtech, that lasted a year, then went over to Philips, and worked there until 83 when we started Altera. I met Bob in the early 70's at Fairchild. We worked on building microprocessors then.

Smith: I'd like to start off by looking at the precursor to the EP300 project, and Altera as a company. We're in the timeframe of 1980, so let's talk about the technology landscape of the day and look at alternatives, including ASICs, logic and memory. Would you like to comment on what you saw as available in the 1980's?

Hartmann: I'll kick it off a little bit. Altera grew out of a company that I founded called Source III, and the founders of Altera came out of that same consulting company, but the landscape of the time, our business at the time was doing gate array consulting, and after doing that for several years, it became apparent what the pros and cons of doing gates arrays were. But in terms of the landscape, TTL was probably very dominantly the logic family that was being used. General MOS, and CMOS was probably the dominant MOS technology in place. Microprocessors were, certainly, maybe not ubiquitous like they are today, but approaching that, they were a big design element. The technology was in the several microns, 5-6-7 microns, probably kind of 6 microns kind of range feature size, what else might you add?

McCarthy: The Apple Mac didn't exist, the PC did exist just barely. So, that was a little, what you could do in computing, what the technology was, we did calculations sort of between us earlier: it's shifted 100-fold since then.

Smith: What about programmable logic in the early 80's, including PLAs and PAL architectures. How developed where those and how readily available where those?

Hartmann: Prior to Source III, I worked at Signetics for 5 years. Signetics was an early developer of, even what they called at the time, FPLDs, PLAs I guess it would have been, these were bipolar technology. Progammable-AND, programmable-OR kinds of structures typically. They were on the landscape from the sort of middle-late 70's onwards, and about the same time, maybe give or take a year or two, Monolithic Memories (MMI), came out with their PAL product line. They were very well known I would think by that time by the early 80's, as a design technology.

McCarthy: These were all bipolar and fuse programmable.

Hartmann: Gate arrays were just emerging in, say, the 1980 time frame, 1981. LSI Logic started, I think it was founded in either 1980 or 1981. They were the first of the relatively large gate array companies that were substantial in its size, and its scope. That was the nature of programmable logic at the time.

Smith: Looking at these alternative technologies, what led to the idea of forming Altera? What was the driving force and the improvements over existing technology that you hoped to gain?

Hartmann: Harping back to the Source III work that we were doing. We were a design consulting company, and at that time, to do a gate array design, somebody, whoever the supplier was, be it LSI Logic, or Signetics, or Intersil, or whoever was doing it, AMI, they would say, here's our product family, and the design tools were, I would say, quite crude, certainly by today's standards. So, for an end user to work with the silicon supplier was a difficult task. We came in as a kind of intermediary to help an end user design product into somebody's gate array. Having done that for a couple of years, then seeing how the problems of making that kind of design system work, and the difficulties of it, and the cost of having to produce custom mask sets, to do a particular part type, all of these problems. We had the notion, in a survey book that we wrote, wouldn't it be wonderful, basically, if somebody could come up with something that is programmable after it was manufactured to do general-purpose logic. We felt that it would be a game-changer. So, subsequent to that, we said hey, why don't we see if we can launch off in this direction, and that's what we did. That was sort of the, out of that work, and thought process, Altera emerged.

Smith: Altera was founded in 1983. When you took the idea to the venture capital industry, how well was the idea received?

Hartmann: In a certain sense, I'd have to say it was received very well, in that at least the people who ended up funding us initially, which was Alpha partners. We made presentations as you do with venture capitalists, and in their case, they somehow felt that is was a great idea, and rather quickly agreed to finance us. They financed us with a rather tiny amount of money, and a big piece of the equity. Never the less, in terms of just getting it done and launching us, it went all right I would say.

Smith: How tiny was the amount of funding?

Hartmann: We raised, in our first round, a grand total of $750,000. That was actually split into a couple of pieces; somebody recollected that they thought the first was $250,000 in the first tranche. So out of that $750,000, OK we'll give you this $250,000 piece now, you go start and we'll watch your progress, and we'll subsequently feed you more, which they did do. That launched us. Even then, that was quite a tiny amount of money. It was just enough I guess to get us rolling.

Smith: Was that enough to cover the design and development of the first product?

Hartmann: Oh no, it wouldn't have gotten there. My memory is a little fuzzy on this, but these were sort of milestone driven financings, so OK we'll launch you, for example, we didn't have a CEO at the time. OK, so one of the tasks is that you have recruit and bring in a CEO, which we did. We had to show progress in the development of the first products. Gathering a team of people to work on this and that sort of thing, and given that we met certain milestones, more money would be forthcoming. But it was always understood that this $750,000 was going to run out pretty quickly, and that there would have to be a subsequent round of financing.

Smith: When you took the idea to the marketplace, and sought customer input, what kind of feedback, if any, did you guys receive?

Hartmann: You mean at the early development?

Smith: At the early development.

Hartmann: Well, in a sense we didn't. I would say we didn't go out to various end users and say "hey, wouldn't you like this kind of a product", because we could see from the marketplace for gate arrays at the time that were mask programmable, and for things like PLAs and PALs and things like that were bipolar fuse programmable, we could see that there was a usefulness to this kind of product, so, there wasn't much we could've done by going to a customer and saying how would you like these, yet.

Frankovich: We tried to keep it secret.

Hartmann: Yes, I guess that's true.

Frankovich: We were in stealth mode.

Smith: So, in stealth mode, another approach is to have what people would call a "killer app" in mind. Did the programming allow for ubiquitous applications as opposed to one killer app?

McCarthy: Yes, I think that's true. There wasn't any distinct single application that was going to make the difference, that was going to be "oh, yes", we'll penetrate this and we'll, that'll be, the success. No, the idea was, just like a gate array, that is was going to be ubiquitous. If you said there was a target market, I guess ultimately, it was, well we think we should take away the gate array market.

Hartmann: Exactly.

Chan: Actually, I remember that in 1984 at Westcon, Bob submitted a paper; the title was "Programmable Logic Replaces Gate Arrays". I was supposed to deliver the paper. I did deliver the paper after him. As I stood up to the stage, actually people started to walk out the room, I think that it's Bob's mission that really counts, not just user input, at that time.

Smith: What was the process of defining the architecture internally, within the company, what sort of insights did you bring into the first architecture?

Hartmann: I like Robert's report to this. We talked about this a little earlier.

Frankovich: The investors wanted us to do a test chip, for proof-of-concept. They didn't think anything like this was feasible. I guess, I find that hard to believe, they wanted us to do a test chip, and Bob wanted to hit the ground running with something that was 3-4 times bigger than the EP300. I didn't want to do a test chip because I know it takes 90% of the effort to do a real chip. So I went looking for the smallest real chip that we could come up with, and that was an emulation of one of the existing MMI parts. So, it was a small one, 20-pin part. And I thought, we can do this one. So that's how I remember it happening.

Chan: Well actually, the architecture ended up replacing 7 or 8 of their products, right? Same pin count.

Frankovich: That's separate, and after I looked into that approach, we found out that MMI had 10 or 12 different parts that they made from the same part. But even though it was a programmable part, they used metal to differentiate these 10 different parts. They actually had a hardware difference. One of the things I figured out, since these parts are programmable, we can make our part programmable to emulate not only these 10 parts, but a whole array of parts that didn't even exist. So with that, we started running.

Smith: So, what is it about the architecture that allows you to emulate so many existing and future programmable logic parts?

McCarthy: In a certain sense, the MMI parts and the others at the time, were unnecessarily, to be honest, unnecessarily restricted about sort of how they were arranged. And, so, particularly since our devices were going to be erasable and reprogrammable and so forth, I think there were probably other issues, in terms of testing the fused programmed devices, and all the various architectural modes, there may be aspects of why they did what they did. We don't know. What made sense for us in making one part, if you're going to make it somewhat programmable.

Frankovich: Make it all programmable.

McCarthy: Make it all programmable.

Frankovich: Maybe it was laziness. I know Yiu-Fai and I had the discussion. He wanted to make it metal programmable like they did. We had some brief discussion. I didn't want to deal with all that extra overhead. Just make the whole thing programmable. It was, in a sense, expensive in area, but I think it paid out in the end.

McCarthy: That's really been, given the scope of what programmable logic is, that's always been true: you make the thing programmable, which you could obviously make it a whole lot smaller if it were fixed, but that's the strange proposition: programmability is worth a huge amount in the long run.

Hartmann: You got to make it as general purpose as you can. As widely applicable to any situation.

Chan: I think, architecturally, there may be a reason for MMI not to make it programmable, you need a separate programmable path, to blow the fuse, for the architecture control block. I think what happens is that in the EP300, we did figure out a way to do it in a way, in an inexpensive way.

Smith: So now that you have the decision to choose a given architectural style, how long did it take from the decision to choose an architecture to its full implementation?

Frankovich: I think 4 months.

Hartmann: Yes. It depends on how you count it. We know we opened the door, mid-June, 1983 .

Frankovich: We had a fractured database by Christmas Eve, 1983.

Hartmann: Whatever that time difference is. 5 months, plus or minus a little bit.

Smith: That's 5 months to tape-out?

Hartmann: To full tape out.

Frankovich: No, the tape was done. The tape was ready.

Chan: Including fractured.

Frankovich: Including fractured.

Hartmann: We were ready to make masks.

Frankovich: Ready to make masks.

Hartmann: So now we're looking at whatever the mask and the fab cycle is after that.

Frankovich: The whole company went and got a place to do business at. We hired all the people. We did it all, in that amount of time.

Smith: So during that period, 5 or 6 month period, when did it start to feel like a real company in terms of processes in place, making progress on the architecture?

McCarthy: I think, you know, real company start around 1987.

Hartmann: All the bits and pieces are there of a real company around 1987. What's a real company?

McCarthy: I think the other thing is we all worked, all separately, as you said, and the rest of the people. We had all worked at serious companies before. So we kind of carried on working at serious companies when there was nobody here but us chickens.

Hartmann: You need to have what it takes to do things. And these guys know how to design stuff, what the cycle is like, and we just go on with it.

Wong: One thing I want to add is particularly that in the design and layout area, that each of us came from a different background, different experience, it really takes a while for us to be able to work together. Each one had a different kind of a methodology they are accustomed to. It's a matter of trying to get used to the Altera environment. It takes some time.

Smith: Once you did have the architecture defined, how easy was it to get everyone at the corporate management to buy into that architecture? That is, people who weren't part of this team.

Hartmann: That was easy.

Frankovich: We were a startup company and didn't have those problems.

Hartmann: I was corporate management so I can say yes. We're making a little light of it, but it's a fact, really this group of people probably made 95% of all the decisions that were made.

Frankovich: I think we were smart enough to go through off all the bad stuff that we all experienced at our previous companies. You go into a company with momentum like a Fairchild, like a Signetics, that's been around for 20 years. They do a lot of stupid things, and they keep doing them.

Hartmann: But whatever they are doing is sort of in their DNA, and they do those things.

Frankovich: Right. But it's not efficient.

Smith: Moving on to the design aspects, and the design environment of the day, what kind of tools were available in 1983 to create this new architecture?

McCarthy: Robert's got the principle ones in front of him [editor's note: pen and paper shown]

Frankovich: Right. Cypress was a contemporary company. They hit the ground running. They raised a lot more money. They bought two $1M computers. And we didn't. We shoe-strung it. So there was stuff out there. We just did it the old fashioned way with pencil and paper.

Smith: What kind of computer assistance did you have? If you had no machines of your own, did you have any time-sharing capability with someone else for example?

Frankovich: For the engineering we did.

Chan: We actually did Spice simulations, the circuit simulation. We bought or rent a terminal; a 300 baud terminal. Later we upgraded to 1200, our terminal, though an acoustic coupler. We dialed up service to an outside service for circuit simulation. We were actually one of the first ones, I don't know if we were the first ones, to do logic simulation on an Apple II computer. We didn't actually; we weren't able to simulate the whole chip using Apple II. We just took a slice, the logic slice, from input to output, and all the architectural control which we programmed with different kinds of programmable chips, using Apple II. The simulation time, if I remember, was probably a couple of days.

Wong: We simulated from the input, the speed path, from the input to the output, it took a long time.

Frankovich: In my previous job, I had worked on 16bit microprocessors. I thought the EP300 was really simple, and we did a lot of logic simulation by pushing ones and zeros on the schematic.

Hartmann: Yes, it's a mental simulation, I guess, you're walking through it, you talk through what happens when you set this bit. And are we doing the right thing things, following the logic paths on the circuit diagram. I can remember days of doing that, with all these guys, just walking through the logic paths.

Chan: I think the logic simulator, we only simulated once or twice as the final verification. I don't think we could even afford the time, to go through all the combinations on the computer.

Smith: A lot of this work was clearly manual. How many engineers did you have on the silicon design team in total?

Hartmann: We had myself, these two guys [editor's note: Bob indicates Don and Yiu-Fai], and two more. So, four plus myself. On the layout side, we had Robert plus three others. Right?

Frankovich: Right.

Hartmann: So, essentially 9 of us on that, doing that piece of the work.

McCarthy: So Robert, you've got the names on the chip, you have the layout?

Frankovich: We can do that. [editor's note: Robert displays the EP300 micrograph]. We have a micrograph of the EP300.

Smith: Can you make out the names on the bottom?

Frankovich: You were asking about the people. Here's the four people that drew this picture [editor's note: Robert points to the lower right hand corner of the micrograph]. Here's the six people that did the underlying circuit design [editor's note: Robert points to the lower left hand corner of the micrograph].

McCarthy: Tell us who they are.

Frankovich: OK. The engineers are Yiu-Fai Chan, Haugh Suo, Robert Hartmann, Jim Sansbury, John Oh, and Don [Sau] Wong. The layout designers are myself, Mark Belshaw, Susan Falk, and Kathy Hopkins.

Smith: Staying on that topic, what was the process used to accomplish the mask design. What were the design techniques of the day?

Frankovich: To address your earlier question, in 83, it was probably right at the cusp where people were hand drawing these things, and beginning to design them on computers all the time. The dimensions were still small enough that we hand drew everything that you see there [editor's note: Robert refers to the EP300 micrograph], at a thousand times actual size. So there's four of us spending months and months drawing, and then having those drawings digitized, which means turning them into something a computer can represent. From the digitized drawings, we then had the computer plot the thing to show us that yup, we had captured it. Once we knew the computer had captured it, that's when we went to fracture. We broke it down into a pattern that could be used to make a mask.

Smith: The mask making was clearly contracted outside of Altera. The mask making process itself was contracted outside?

Frankovich: Correct. We contracted out everything else outside. That was also part of our claim to fame; to try to do the most with the least.

McCarthy: How was DRC done?

Hartmann: You guys were pioneers in DRC, for sure.

Frankovich: Well, because we were trying to save money, people's time was cheap, we got to use the old fashioned way too, and we checked everything by hand.

Hartmann: You guys were very early users of DRC.

Frankovich: That's true. DRCs had been around for maybe 2 to 2.5 years. We hooked up with another startup company, ECAD. Once again, trying to do things on the cheap, I negotiated with them to borrow their computers, I would take our database, physically bring it down to their shop, and we would run a computerized check of our database on their equipment. We would just pay them by the hour. The deal was, we would pay them, in little bits and pieces because all that software was really expensive at the time, and some of the money we paid to them by the hour, would go towards a future purchase, when we got bigger. That's how we did it.

Hartmann: Did ECAD eventually become Cadence?

Frankovich: Yes. ECAD eventually became Cadence. Many years later.

Chan: We also ran LVS. [editor’s note: LVS is a layout vs. schematic consistency check]

Frankovich: LVS was definitely new.

Chan: Very new.

Frankovich: A computerized check of the database for the logical content and the electrical connectivity was maybe a year and a half, to two years, old. DRCs were maybe three years old. We helped debug their software tool. So we were one of the first people to use that.

Hartmann: ECAD was probably a leader in that whole field on that side of it. We were probably a lead customer of theirs.

Frankovich: We were a lead customer of theirs. They were looking for people to buy and use their software. There were one or two other big companies, and they were trying to break in, and their claim to fame was that they were cheaper, they were faster, and it took less room on a computer. Sounded like our kind of guys.

Smith: Staying with process technology for a moment, what was the CMOS EPROM process capabilities at the time in terms of feature sizes, reticle sizes, and so on.

Chan: It was a double-poly EPROM process. We got away from a single-poly already. The feature size if I can remember was 5 microns. A single metal process. Double poly, one metal.

Hartmann: So, we’re a hundred times more dense today?

Chan: Yes, at least.

Hartmann: It’s funny to think about stuff like 5 microns.

Frankovich: For those that are in nanometers, our feature size was 5,000 nanometers.

Smith: What kind of performance could you get from that process? Pin-to-pin for example.

Hartmann: Particularly slow!

McCarthy: The combinatorial input to output for the device was spec’ed at 90ns.

Smith: That’s just over about 10MHz. 11Mhz.

McCarthy: The datasheet said 13.3

Hartmann: That was probably to do with some slight of hand, an internal path or something.

Chan: Actually, it’s not that bad if you remember the memory technology around that time is also in the 60-100ns, access time.

Hartmann: Consistent with that.

Wong: But that was not the EPROM technology. EPROM was inherently slower. That was one challenge we needed to face. At that time, from the circuit design point of view: is how to speed up the internal array delay.

Chan: Actually, what happens is, one reason EPROM is slower than regular memory, is that the high voltage, a combination of high voltages, so the channel length of the transistors has to be bigger, or you stack it up to increase down voltage. It slows things down. The other thing, the ERPOM cells are inherently slower than a ROM because it's double-poly.

Hartmann: That was a constant battle. That never went away.

Smith: Is this the so-called slow bits versus fast bits?

Chan: The slow bits I remember were what we used for the architecture control to configure the chip to do seven different types of equivalent MMI PALS.

Hartmann: Maybe you can expand like this: the so-called slow bits are for static control lines, in the sense they are slow as they don't change at all, they are very slow, they are static. Versus everything else where you are doing logic, they are expected to respond quickly, is probably a way to think about it.

Chan: We made the logic, the slow bits, very simple. So we don't use many transistors to save area, we don't need the speed, it's basically static.

Smith: What about pin counts and packages? Was there a reason for selecting the pin count that you did, and what was that pin count?

Frankovich: It was a 20-pin part, and the reason we selected that, was that it was the most ubiquitous MMI part at the time. So we emulated that.

Smith: Was the part deliberately pin-compatible in any way?

Frankovich: It was definitely made to be pin-compatible because I thought we'd have an easier sell, if we can knock on their door and say here's this part that everything your current part does, and we do it lower power, and if you make a mistake it's reprogrammable. It's identical conceptually to the part you're using. The only downside is that it'll be slower, but it'll be one-tenth the power.

Smith: So it's lower power, and lower performance because it's CMOS versus the bipolar parts that existed in the day?

Frankovich: Yes. I mean, We go way back knowing that everyone doesn't need the fastest stuff there is. When we did that 64K memory way back early on. Everybody doesn't need the highest speed. There's a whole raft of people who could use that part fine.

Hartmann: Looking into the future, we can say, OK, this technology will work for much larger parts where there's no competition. Now it's not a case that you're slower than someone else. They can't get there with bipolar technology. We can reach out from the EP300 to do any bigger, much bigger parts, we felt anyhow, and that did eventually prove to be true. I guess I believed, maybe it was a little naive, but as you go larger we could afford to say we're maybe not quite so fast, but we are larger, so we win on that scale.

Frankovich: Bipolar wasn't going to scale to be three times bigger. It was going to break in power. Where-as we could go a hundred times bigger, easy.

Smith: Looking at package selection for the EP300, what did you select for low volume versus high volume production packages?

McCarthy: There was no high volume, so the issue of making high volume parts was a dream. A dream. I think the big issue was how we'd get a windowed package, a 20-pin windowed package, because no such thing existed. I don't remember how it came about, but it was obviously a necessity.

Hartmann: Yes, it was driven by necessity. We have this EPROM part, it's erasable with UV light, you better be able to get UV light into it somehow. Yet, we were driven by this 20-pin package. I don't even remember how it was developed, but we went to someone and said, hey we need a quartz lid on a 20-pin package, can you do it, kind of thing.

Chan: Actually, at that time people do have a quartz lid package on a wider, 24-pin package.

McCarthy: Yes. 0.4" pitch then, but we wanted, was it 0.2 or 0.3?

Chan: Yes, too much detail. One thing, the package house was afraid of was the moisture leakage because the package was really narrow, and it's hard to put a lid, not a lid a quartz window, without environmental leakage, whatever they were by that time.

Frankovich: A good leader on what to do on package approach. We did put them in plastic. Those were one-time programmable. We figured that in volume a person would have already figured out what their application was, and could afford to do that.

Smith: Talking about the architecture itself, can you comment on the logic cell structure, and was it a superset of an existing cell structure, or was it something completely brand new?

Hartmann: It was what Robert alluded to earlier, we did a superset of I suppose all the 20-pin PAL devices that would have been covered by MMI, that MMI had, plus we could do things probably that they couldn't do just by the nature of the architecture.

Frankovich: Once again, that's why we picked that one, because it was simple: it was one flip-flop, one I/O, one "macrocell". It was in a sense, conceptually very simple, and they were all the same.

McCarthy: The other important characteristic was the fixed architecture devices that MMI were producing, commonly; all eight of the cells would be fixed the same way. So, when you got one and it was version three, every cell had to be version three. In our case, you could have any combination in any order whatsoever, and it was still one device. So it offered programmable flexibility. That again was a differentiator. In other words, the user might have to buy two parts because they wanted half of one and half of another, as there was no such device that could meet their needs. Our device could have all of those possibilities.

Smith: Staying on flexibility, the other people into programmable logic is programmable interconnect, how did you arrive at the topology for the programmable interconnect?

Hartmann: Well, we didn't have programmable interconnect per se. We had universal interconnect.

Frankovich: Everything went everywhere.

Hartmann: Every logically generated output became an input to everything else on the chip. Conceptually we knew about segmenting interconnect, because that did appear in later devices, but early on the EP300 everything went everywhere in the parlance of programmable logic. That's both a blessing and a curse. Certainly, it's a blessing when you're trying to produce logic because everything is there. You can always produce a function, with any signal you generate.

McCarthy: There was nothing, at least, in the silicon, that having a programmable interconnect, or having a segmented architecture or whatever, there was nothing to prove about that, if you think about that half of the task of this first chip was being a test chip to prove the technology feasibility, that was an important aspect of what it was doing. There's only a conceptual feasibility about segmenting an architecture, and Bob had already figured that out.

Smith: What about the I/O structures at the time back in 1983, and choices or any standards you had to adhere to?

McCarthy: The choice was so obvious that I don't believe we even had the idea that there was a choice. Since the 1960's TTL logic levels had been the de facto, the absolute standard. Odd things like ECL had fallen into disuse anyway, as far as I know there really wasn't much else. There was occasionally, there'd be some high voltage, 15V CMOS, but that too had faded. Everything had become sort of, the ubiquitous use technology, was low power Schottky, and so 5V rails and TTL compatibility was sort of mandatory and there were no, really, any other rational choices.

Smith: Could you mention something about the size of the final die, transistor counts, metal widths, for example, just to calibrate the technology that you developed in 84.

Frankovich: It was a 5 micron, single metal, dual-poly CMOS EPROM process. As far as the transistor count, there are about 8,000 transistors on that die. The die size, it's over thirty years ago, but I remember 140 x 128 mils. About 3mm x 4mm. To put that into perspective, a die 250 mils on a side was pretty much the practical limit at that time. There weren't dies much bigger than that at that time. A quarter of an inch on a side, in 1983.

Smith: A quarter inch on a side was the leading process for CMOS EPROM?

Hartmann: No, for probably any chip.

Frankovich: Bipolar was probably significantly smaller.

McCarthy: There was the issue of how did it relate to the reticle and stuff like that. You have to recall that the defect densities back then were so high, that you wouldn't dream of filling the reticle. You would just not get any yield. Die sizes were very distinctly defect density limited.

Smith: Was there anything built into the architecture that helped with defect density?

McCarthy: No, redundancy came much later. It wasn't until we were working with a much more defective, foundry defective, process technology from a company that we shall not name, who we despise. Their process was so screwed up, so horribly defective, that you really had to have redundancy to get the damn thing to work at all.

Frankovich: In this case, the die was so small, to put something in to yield better would make it bigger and yield worse.

McCarthy: That's probably true.

Smith: Given the lack of sophisticated tools, and the early CMOS EPROM process, what was the greatest challenge in getting this architecture taped-out?

Chan: We were talking with Don [Sau] a bit earlier. One of the biggest challenges to get the speed through a high-voltage process. Basically, I think that the power supply voltage was 5V, and then for programming we had to put 21V, to program the chip. Usually you make the channel length of the transistor much, much bigger, but then that slows down everything. So the challenge is to get the speed while getting the high voltage through. The other thing is that, since this chip has only 20 pins, we have to multiplex almost all the I/Os to have high voltage to go through, without increasing the die size because you need bigger transistors. Sometimes, we had to stack two transistors on top of each other to make sure that, in device physics, you can reduce the breakdown voltage if you stack it that way.

Wong: Voltage divider action.

Chan: We had to sense the voltage so the 5V input suddenly became a 21V input and it would know by itself. We did a lot of circuit tricks to get the speed we wanted, with the die size smaller too.

Hartmann: I think this whole notion you're alluding to, the whole notion that at one level, conceptually, you have the user, and the user sees the chip just as a logic device, but we are designing it and have to account for another level, which is how do I program the various architectural paths and the bits and so on, and that's a whole different scheme. I suppose it's up here as it's 21V [editor's note: Bob raises his hand], instead of 5V. So there's that notion, the other is that we had this array that is a logic array for its normal use, but you want it to be like a memory when you're trying to program it. In other words, you're trying to do some type of row and column addressing for programming, but during use it isn't used that way. It's really two entirely different sets of circuits going on in the chip. One is the one the user uses if you will, and the other is the ones the programming circuitry uses. For me, I remember that was really, how do you do this in an efficient way, that was really a struggle.

Frankovich: Coupled with that we were trying to do this to a fixed pinout; something that emulates an existing part. That was relieved when we went to our own parts.

Wong: One more thing with the high voltage, beside the speed, you know we need to worry about the input to output delay, is the CMOS latch-up that is associated with the high voltage, that we need to watch out. Both from the circuit point of view, as well as the layout point of view. We need to pay a lot of attention to it.

Hartmann: It's kind of funny that you were asking earlier about the state of the art in technology at the time, and CMOS was really just, just emerging at that time, and this whole notion of latch-up, what a serious problem that was. We all forget about that, nobody thinks about that anymore. It was a big deal.

Wong: It was a bit deal.

Smith: Would you say, apart from these innovations you just mentioned, the other significant one was making a reprogrammable programmable logic device?

Hartmann: Certainly conceptually, yes, that was the goal from the outset. That was part and parcel of what we were saying we were doing: this is reprogrammable logic, if you will. So, a user can make a mistake, and say oops, and erase it and reprogram and reuse it. That's something one can't do with a fuse programmable device where you blow a fuse. You can't un-blow a fuse. It just doesn't work.

McCarthy: For the evolution of the technology, the same things become true. The objective was to compete with gate arrays. Of course, if you make a mistake with those, that's really bad. That remains true. If you went to a fuse programmed device, or anything that's not reprogrammable, if you went to the fuse programmable device and simply scaled it up not only is the issue that Robert pointed out that the power would become completely untenable, but also, since now you're also building a big chip, and it's going to be expensive. A fuse-programmed device blown is again an expensive item relatively speaking, not as expensive as a gate array, but in those days, people were still annoyed at mis-programming the fused devices or discovering, as was the case, that not all the fuses, even though all their design was correct, not all the fuses worked. So you then get bad programming that was not correctable. Therefore, for each chip, you actually had to have an application-specific test for the fuse-programmed devices. So they carried the problem of every chip, when you did the design, had to create a test program for your particular design. Programmable logic, the EPLD approach, didn't have that problem either, because the devices could be tested, reprogrammed, whatever, generically. That issue just went away.

Smith: With this technology and reprogrammability, was there any sense of reprogramming in-situ on the customer's printed circuit board, or was it always the plan to have a programmer on the side.

Frankovich: We talked about it, but it was too early on.

Hartmann: Yes. I think the issue there is that the user has to have some high voltage logic circuitry somewhere on his user board.

McCarthy: Just as Bob was saying, not only would you have to have two levels of activity on the chip, but the customer would then have to have two levels of activity in their application, and that starts to get less and less practical.

Smith: Clearly, programmable logic devices have to be programmed, so let’s move on to development tools that Altera offered to the customer base. Perhaps you could start off by commenting on what sort of platforms were available to the customers in the 1984 time frame.

McCarthy: We know that there were Daisy workstations, or they were beginning, Mentor Graphics was still in the business of making, remarketing probably call it, they bought?

Frankovich: Apollo.

McCarthy: They bought computers from Apollo. People were buying, there was a very sort of confused state of the CAD industry because they were selling hardware bundled with software. So you bought either, you talked about buying a Mentor system and you actually got hardware, and so forth. But these were very expensive, and for the purposes of programmable logic, certainly of the era of the EP300, people were not used to spending that, those kinds of monies, that was for people with large budgets and planning to do initially gate arrays. The idea was the IBM PC was available. It had been out for two or three years. It was reasonable to imagine that people would have one of those, or if not have one, engineers would like to have one of those. All the tendencies going in that direction that people, that would be a good platform. The Macintosh wasn’t available so that wasn’t even an issue. The Apple II was pretty common, but it didn’t seem to have the robustness perhaps, and it didn’t, in an industrial context, it didn’t have the same feel to it. Whereas companies by then were already, large corporations were already buying PCs. It was sort of the IBM thing had happened. So, that became the platform, then the question was what forms of design entry would take place.

Smith: Can you remember what kind of memory on board, on the PC, that you had available, what kind of display capabilities you had for design entry?

McCarthy: The PCs at this time, it was unusual for them to have color displays, so they were all monochrome, character displays, 80 column character displays, and you were lucky if someone had 256K of memory, or maybe 384. Out of the maximum, all running DOS of course. That was the environment, the working environment that you could expect.

Hartmann: All floppy disk based.

McCarthy: All floppy disk based initially.

Smith: When you started to develop the development tools, which tools did you use yourself to develop the development tools?

McCarthy: One of the first tools for the EP300 was a tool that was built called Logic Map. It was compiled in C. I’ve forgotten the name of the compiler manufacturer, long since gone probably, and so this particular tool was, you could get native compilers that would run on a PC for compiling C code, and that was used. So you had both a development and test environment in a single and very inexpensive computer for those days.

Smith: Did you manage to get the development tools ready in the same time frame as shipping the EP300? Was there a delay between one and the other?

McCarthy: The first of the tools, Logic Map, existed before the silicon arrived. So there was actually some programming hardware to program a chip, a prototype version, sort of a one-off, along with this Logic Map program, but of course there was no way of knowing that is was correct until you actually got the chip in hand. It was coded up based on the specification of the chip. So that existed. But the real plan, even in 1983, was that there would be a whole suite of different software modules, tools and so forth, which were to come out in late 84. In particular then, because there was going to be a much larger chip than the EP300, that would line up with the needing of a larger design environment, akin to gate arrays. We all knew in those days that gate arrays were being designed with schematic capture, hardware description languages really weren't that common, or hardly existed at all. Rather curiously, the fusable programmed logic devices, they were programmed with Boolean equations, and the world went around again and coming back to hardware description languages, in a sense. So, when the EP300 arrived, well you guys should talk about the EP300 arriving first because it sort of makes sense.

Hartmann: What do you mean? It just arrived! We plugged it in and it works. I was trying to remember what didn't work. Essentially it worked. Some other minor thing or something, maybe it wasn't so minor to someone. Someone thought it was a power on reset circuit.

Chan: It was. It was not consistent. It depends on the, on how fast the power comes up.

Hartmann: I was thinking of this notion when you have a logic replacement, and when you power up you expect the logic to be there. It's hardware. When you have a power on reset that enables the logic that you expect to be there disappear, it's kind of a problem. Anyway, I guess that was the flaw which we beat on for years after that. But yes, the circuitry was there, we had chips that worked. I remember we did have the Logic Blaster hardware and that stuff so you could go in and program a device, we as engineers. It probably wasn't a comfortable tool for a customer at that time.

McCarthy: Yes and no. I think that back then the fused programmed parts that people were familiar with, yes there were people writing Boolean equations and doing it symbolically and so forth, but I think there was also, if you go back to the Signetics stuff, they had little charts, and people, they were not punched cards, but like a voting card you actually circled the dots in black somehow or other, and there was some arcane way of getting that into the [programmer], so that the chip got programmed. People were used to sort of a very physical level of things, so the Logic Map software that drove the programming hardware that drove the and so forth, was not completely out of the question for the start. I think there was a sort of, if not surprise, a delight when the chip arrived, and now what are we going to do now, because the main software that was designated wasn't going to arrive until much later in the year. So that was that. I ended up writing; I had been fiddling around anyway writing a little translator, that would generate programming JEDEC files that the Logic Map software could read, than could program a chip through the programming hardware. We sort of decided that, that set of things was kind of OK enough for us to be able to introduce ourselves to the market.

Hartmann: I was just thinking that another part of the corporation took over at that point and said, hey, we better start selling something.

McCarthy: How about revenue?

Hartmann: We need some revenue. They launched the, I think the desktop silicon foundry idea got launched at that point.

McCarthy: Right. We realized that there was no third party programmer support, there was no third party software support, there was only first party everything. So we figured we had to do everything. Part of the issue was we need, there we were, a bunch of guys in what I always thought were insurance offices, on Hamilton Avenue [San Jose]. As the company grew, we simply rented different rooms. The test center was on a ground floor office, the design group was upstairs, the CEO had his offices at the front of the building and another part of the thing, and as the population grew, we just rented more rooms. The air conditioning would go off in the summer. The test group, they had test hardware that needed compressed air.

Hartmann: They needed two things: they needed compressed air, and for some reason they needed nitrogen. So that was the bottles.

McCarthy: Yes. The usual gas cylinders would be delivered, and this was not really an industrial district.

Hartmann: Individual CPAs.

McCarthy: Individual CPAs, insurance companies, little old ladies in tennis shoes, and then these big bottles of gas would get dropped on the concrete, and they ring like a bell. We'd wake up our neighbors.

Hartmann: They had a little vacuum pump, that's what I remember about testing the chips, so they created a vacuum for the wafer resting on the probe station. We needed a vacuum, normally in an industrial setting you have a vacuum line but we didn't have that, so we had a little vacuum pump, but the vacuum pump was kind of noisy, so I remember them sticking the vacuum pump into a drawer of some bureau to keep the noise down, but generating a lot of heat. Our industrial facility!

Smith: How many engineers were on the development team for development tools?

McCarthy: At that point, there was probably, the group working on the main set of tools was probably 4 or 5, when we were still at Hamilton Avenue.

Hartmann: Yes, I suppose.

McCarthy: And then I just wrote the translator at home.

Smith: So, the main [development tool] product was called A Plus?

McCarthy: No, A Plus didn't exist, A Plus didn't come out, it was intended at the end of 84. I think actually got shipped beginning of 85.

Chan: Didn't you write the minimizer too? The logic minimizer?

McCarthy: Yes. I wrote one for this little translator thing.

Chan: Altrans probably?

McCarthy: Yes. So what we ended up with, I'm surprised at all of us, we were selling these boxes, these cardboard boxes, with a programming card in it, a cable, a plug-in thing with a 20-pin and a 40-pin socket (though the 40-pin socket wasn't wired up because we hadn't figured out what the chip was going to be, but it showed promise), floppy disks, and two EP300s. For this, we were selling it for I think $1250. When we came up with the next generation thing, it was $2500.

Smith: Talking about programming, a modern programmable logic device can have 100Mbits of configuration space, how many configuration bits were there in the EP300, approximately?

McCarthy: I think about 3,000.

Hartmann: It's funny to talk about configuration bits. There are configuration bits that simply steer signals, say, I'm going to be a registered output versus a combinatorial, I'm going to be an inverter or a non-inverter. Those are purely static lines.

McCarthy: In that case, for the configuration of the output structure, there's about half a dozen for each cell, six 8s are 48.

Hartmann: A hundred or two hundred or something like that.

McCarthy: Yes. Less.

Hartmann: When you think about it, if I hook this output back to some input, is that a logical input or a configuration bit? Well, it's kind of both. It's part of the logic path, but to get it to be useful, it's like hooking up a configuration bit.

McCarthy: One of the best analogies for thinking about programmable logic, it certainly didn't occur to me back when this was happening, is the way to think about human neurons and synapses. There is no point, you can count the neurons, but that doesn't do you any good, because even though there's rather a large number of them, it's how they are wired up that really makes all the difference. But the wiring is simultaneously what connects to what, but it's also the fan-in for the logic, and that dictates what it does. Just like an EP300 in a sense, what fans into a particular cell, is part of the definition of the logic too. You can try and tease them apart, but.

Hartmann: In the final counts of programmable bits, it was a few thousand.

Chan: I just made a back of the envelope calculation, since we have 20 inputs, each input has two's complement, you have 8 terms plus OE [output enable], and you multiply that and it's about a few thousand. A little bit less than 3,000.

McCarthy: You can still do the math after all these years!

Hartmann: Nowadays, you can probably do a few hundred million, or something like that.

Smith: Given that you have a few thousand configuration bits, and using a new process, the CMOS EPROM process, how long does it take to configure one complete EP300, for example?

McCarthy: Just a few seconds.

Frankovich: Less.

Smith: You mentioned earlier about programming machines [programmers] and at Altera you made your own one called the Logic Blaster, what about third party support? When did the third parties get on board?

McCarthy: Well, it's a fundamental chicken and egg problem. We had chips but they weren't in any significant volume, and back then, the major provider of programming hardware is a company called DataIO. They were sort of polite but we were kind of nothing. No one was beating down their door to ask them, other than us, to make programming hardware for our chips. On their business sense, they would be waiting until there was sufficient demand, but of course, as a startup company, you can't wait for anything. Though there was programming hardware, and even development software for fused programmed devices, we had very little leverage on the third party providers. So it was an evolutionary process, of us developing our own, and only when we got much larger would they pick up the threads and then start producing programming hardware for the chips we made.

Smith: At what point in the process did you bring distributors on board to help proliferate the technology?

McCarthy: That was pretty quick.

Hartmann: You probably know that better than I.

McCarthy: That was happening in 1984. As soon as we got something to sell, we did have the beginnings of a sales organization, but we immediately went out and there were the commissioning of distributors, and the beginnings of the commissioning of reps. [company representatives], to get the product established.

Smith: Did the distributors offer anything that you didn't offer directly, for example, volume configurations; you could ask for a thousand parts to be configured by a distributor?

McCarthy: That was already well established as something they did for fuse-programmed parts, so they, it was sort of a value-added service that they were already doing and understood.

Smith: The distributors presumably had field application engineers. Did you go through a training program and develop educational materials?

McCarthy: Not in the early, that sort of emerged later, maybe a year later. In the early part of the EP300 there was a limited degree of that.

Smith: Moving on to the actual fabrication itself, clearly programmable logic mainly has been a fabless semiconductor play, although not entirely, can you say something about which fabs offered services to Altera?

Hartmann: I can talk about this a little bit. Altera was I think the first truly fabless semiconductor company, and I was talking earlier with these guys, that emerged out of a vision that Rodney Smith had, the first CEO, who recognized the brilliance of that early on, that we should off-load this big expense for us. We initially did work with Ricoh in Japan, that was our first supplier, and that was driven largely by who was out there. There were very, very few suppliers of CMOS EPROM technology, available to anybody, to the public lets say, to a party like Altera. Obviously, Yiu Fai came from Intersil, and Intersil had developed CMOS EPROM technology and had a fab. Intel did by that time probably have some CMOS EPROM.

Chan: Not much.

Hartmann: There were very few places we could go to, and Ricoh was the one, that's where we started.

Smith: What kind of wafer size were they offering for CMOS EPROM?

Chan: I sort of remember 3 inches, because I remember this big; I used to hold it with a pair of tweezers no problem. Anything bigger than that, and you probably need special, bigger pair of tweezers!

Smith: What about packaging? Once the die came back, was there a third party packaging house you used?

Chan: I sort of remember that when the first part came back, we had it packaged locally, but without the quartz window, we just had the white ceramic package, no lid, and the lid was taped on physically with Scotch tape. There was a piece of metal you put on top and you taped.

Hartmann: These were not for sale. These were for engineering work.

Chan: We need to open it to probe and also to erase, the piece afterwards.

Hartmann: Early production I'm quite sure was probably Amcor in Korea that did early production packaging for us.

Smith: You mentioned test and that brings us on to a great subject here. What kind of success did you have with the first engineering samples? Were they all perfect or what kind of yield?

Hartmann: We dreamed of 100% yields. I don’t remember the yields.

Chan: I know for sure that the chip worked the first time.

Smith: Any issues from going from engineering samples to full production?

Hartmann: I think there was a mask turn.

Chan: Yes, we did a mask turn.

Hartmann: After that, that was production mask set for that version of the EP300.

Smith: What about testers, going from functional test to speed testing? What kind of test equipment did you use for these parts?

Wong: I think Altera was probably the first one to use the PC for doing the wafer sort.

Hartmann: For doing functional test.

Frankovich: We didn’t do that out of the chute though.

Hartmann: I think now the speed testing was, someone said, Megatest.

Wong: Megatest, yes.

McCarthy: But there hadn’t been something before.

Hartmann: Some functional test. Yes.

Chan: I remember a tester this size, this big [editor’s note: Yiu Fai indicates approximately two feet wide and four feet tall].

Hartmann: This was a Megatester.

McCarthy: The ones I remember were the size of a locomotive.

Chan: Those are the bigger Megatesters.

McCarthy: When we moved out of Hamilton Avenue and moved into Monroe, first of all, there was one of these. They looked like chopped off locomotives. Diesel locomotives. I always dreamed of that we’d see, I remember the first aliens movie coming out, I always imagined, they were so dark and mysterious these things, once expected that when you went into a test room, you expected Sigourney Weaver to come out in her grey underwear and a flame thrower. It never happened.
The whole issue of building in redundancy in subsequent devices, that is so algorithmically beyond even functional test, you need a, there was no tester out there that could behave in those kinds of ways.

Smith: The product is now back, it’s functioning, it’s fully tested, and moving into the marketing cycle. Tell me something about the T-bird tail light demonstration platform.

McCarthy: We needed something of course to be able to show people. You can discuss programmable logic is conceptual and should do all these things, but you got to have demo. An applications engineer that worked for me, his name is Don Faria, he some time earlier than this, had come across a little state machine that would emulate the sequential tail lights of a Thunderbird. We debated that was it really a Thunderbird, and what model. It doesn't matter really, and it became known to us as T-bird taillights. There were 8 LEDs on the box, which can correspond to the 8 outputs of the EP300. So you had three LEDs for the right turn and three for the left turn. We had an ignition light and we had a brake light. And then you had simple controls: a switch to turn on the ignition, a push button to press the brake, and then a left and right turn signal stick. This could be programmed and into the EP300. You could take a blank EP300, show somebody, show them in various forms as years went by, but initially you could just show them the logic equations, and you could program those logic equations, and then program the chip, plug it in and it would work. You could take the chip of course and erase it, and do it over again. There was a sort of instant proof of that this wasn't just an idea, because it was a brand new set of technologies being put together, so people were not inappropriately skeptical that it really worked. He you had something quite simple that was convincing.

Frankovich: Did you guys have this when you went down south and did the Leonard Nemoy thing?

McCarthy: Yes. Robert's talking about, who's got the picture. You've got to see this.

Chan: I've got an autographed picture [editor's note: Yiu-Fai holds up a picture of Leonard Nemoy holding an Altera development kit box).

Frankovich: It's a marketing ploy because Altera's tag line was the logical alternative. Who better spokesperson than Spock? It couldn't be Spock because.

McCarthy: Paramount or someone owns the rights. But as a celebrity, there was a Westcon conference in late 84, it was Halloween, and Don the inventor of the T-bird taillights, went out and got thoroughly drunk, and could only just manage the following day to turn up at the booth, and we all love this story. Mr. Spock was there, Leonard Nemoy, and great throngs of engineers came to the booth. While that was happening, we were also introducing our next chip, the EP1200, but anyway, part of the T-bird taillights demonstration, and we had an early version of our A-Plus software that wasn't ready for release, nor the chip had come back (the next chip), we were doing the same demonstration at this particular thing with the EP1200, or it's prototype at that stage. So that was the end of 84.

Hartmann: Mr. Spock was our spokesman and chief, sort of, at Wescon.

McCarthy: Yes.

Chan: We also had the license plate: Altera, the logical alternative.

McCarthy: Yes. I don't know what the tag line is now.

Chan: Do we have one now?

Hartmann: I don't know.

Smith: I believe that when you first came out the tag line was "desktop silicon foundry".

McCarthy: That was one of the ad campaigns for this box. One of the lovely graphics was a man, it's always interesting, he wasn't portrayed as a nerdy engineer, he was a rather suave looking guy, but he wasn't be-suited, he wasn't wearing a tie, but he was pretty cool. He was looking at this box, and the tag for the ad was the desktop silicon foundry. That was the $2500 version.

Hartmann: That was this blue box that looked about 2x of a standard shoebox, and in there was the programming hardware, and the floppy disk, and whatever number of parts we had in it at that time. That was a big splashy ad.

McCarthy: Although we were, we knew we were a semiconductor company; the kind of curious thing was, for the first year of course, all the revenue came from selling cardboard boxes.

Smith: Can you remember, approximately, what the average selling price was of an EP300?

McCarthy: That's a good question. I don't know.

Hartmann: I don't know.

McCarthy: I want to say $5. It's probably $10.

Frankovich: Much more!

Smith: When you put this device into customer hands, did you get any unexpected applications using the EP300?

McCarthy: We were just happy they were using them. We didn't mind if they paved the street with them.

Hartmann: Gary Schwartzfigure is the guy I remember. He sold to somebody. I remember him coming back, and it was like some semiconductor equipment vendor application, and for whatever reason they displaced, I don't know, an MMI part or something like that. That was kind of the first application that sticks in my mind. The first feedback; hey I've got this order for an EP300. That was probably, you know, May or June of 84. Really early on.

Smith: In 84, I believe the revenue was about $60K for the company, in 85 about $6.5M, and in 86 about $24.5M. Were you surprised by the ramp rate of the company?

Hartmann: I don't know. Probably surprised. Probably elated. Probably disappointed. All of the above.

Frankovich: That's what we have planned.

McCarthy: From my individual perspective, back then, no one discussed whether we were surprised or what. It seemed remarkable.

Hartmann: Those are big numbers now when you think of them. Those are startup numbers. Those are pretty big numbers.

McCarthy: Those are pretty big numbers. The fact that we were selling anything was kind of amazing.

Chan: Three years and complete $24M.

McCarthy: Pretty good, and those were 1985 dollars. They are not the contemporary, treasury, watered down dollars. Print some more dollars. There was no, something easing, quantitative easing. They weren't doing that then.

Frankovich: Considering the profit margin. Considering what, $7M went into the company? So we already made back all the money that went into it in profit.

McCarthy: It was cash flow positive.

Smith: In two years after releasing its first product?

Frankovich: Yes.

Smith: When you started to talk to customers, what was the biggest item they wanted that wasn't available on the EP300?

McCarthy: They wanted the EP300 to go faster.

Frankovich: Right.

Smith: Bipolar speeds?

McCarthy: Yes. They wanted that, but that was no surprise to us. It wasn't some "oh my gosh, how did we ever not know that?" We understood that was from the going out position.

Smith: Looking at final thoughts at this point. What would you have said you would have done differently on the EP300 project, having time to think about it?

McCarthy: Nothing.

Frankovich: I don't know.

Hartmann: It's always easy to look back and say, we could have done this thing or that thing better, but I always felt that it felt like a lot of hard work at the time. That's how it always felt to me, like we were just really tying to get this thing done. Maybe conceptually it's easy, when you look at it conceptually in terms of number of transistors, and all kinds of things, it's not that huge, but just in terms of solving problems, day after day after day, I though it was a lot of hard work. I don't know. What do you think Don [Saul]?

Wong: There's a lot of details, particularly the design tools at that time were so primitive, that we had to do a lot of manual checking, and the layout.

McCarthy: You wouldn't have changed that, would you?

Hartmann: No, it was just a fact of life. That was life at the time.

Wong: You have to watch out, there's modeling, you cannot really trust the transistor model.

McCarthy: That hasn't changed. Has it!

Hartmann: Yiu-Fai says it has.

Chan: I think now things, especially the EDA tools, and the models now, instead of just the model, they give you the whole design kit. I've been consulting for other semiconductor companies in recent years. Things changed quite a bit. But it's surprising that even though things change, the fundamentals didn't change that much. It's just that the regular engineers got shielded from the actual transistors. Further and further away.

Hartmann: Abstracted away?

Chan: Yes.

McCarthy: Since the EP300 did more than, perhaps it was intended to, in a certain sense, why would you want to change it?

Hartmann: I think it did what it set out to do.

Smith: What do you feel about the future of programmable logic? Do you see any trends, or do you have any expectations for the next 25 years of programmable logic?

Hartmann: I'm overwhelmed how big they are. I really am. I'm just stunned how big they are.

McCarthy: Yes. That's sort of the, it's still, I still think in a sense, it's in its infancy. How will things like Larrabee [editor's note: Larrabee is a multi-core general purpose processor and graphics processor from Intel], intersect with programmable logic, I don't know. Some of the promises of reconfigurable computing engines and so forth, will the world ever come around again to, a sort of pre-Von Nuemann sense about how computers might be? Could you go back to Eniac hardwired approach, where the programming takes place in the wiring, which is what programmable logic is, as apposed to sequential fetching of instructions. Traditional computing has now got multi-cores, because it can't fetch the instructions any faster. So we're going to end up with just vast numbers of cores, but we've got vast numbers of programmable logic elements. Is it a horse race? I don't even know if that's the case, between those two things. Of course, I haven't been doing it for money for a dozen years, so what the hell do I know?

Hartmann: One thing for sure, programmable logic has largely displaced gate arrays, at least gate array companies. There were 30, 40, 50 gate array companies in the early to mid 80's. I don't know how many there are now, but a very small number. That whole industry has changed. Anything smaller, I don't know where gate arrays fit anymore, but programmable logic has displaced a great amount of that. I would think that's going to continue, because the arguments for programmable logic, the basic argument that you could do one and program it for any application, and you don't have the development costs, and you don't have the burden of mask costs, and obsolete inventory, and on and on. All those arguments.

McCarthy: Have remained constant.

Hartmann: They're the same. So that march will still continue.

Smith: The three most successful programmable logic companies today.

McCarthy: No, there's only two.

Smith: Started in 1983, 1984 and 1985. Two and a half [companies] perhaps. Do you think it's surprising that no one has actually been successful since the founding of those three companies?

McCarthy: I don't think it's surprising. We put a lot of work into patenting everything we do.

Hartmann: I think it's surprising that VCs still fund startup programmable logic companies.

McCarthy: It's so stupid. I mean, is it surprising that Intel makes microprocessors and apart from you know.

Hartmann: And no one can break in.

McCarthy: Is it surprising that Microsoft supplies most of the operating systems, no! The task of every capitalist it to try and obtain a monopoly. That's our job. It's the government's job to stop us. Ultimately, the sad thing is for us, is that it's turned out to be a duopoly, because we would prefer a monopoly. So, is it surprising? No. Once you get to a certain sufficient size, you can fend off with intellectual property, and sheer market acceptance, all kinds of things. There may be some game changing things in the future.

Hartmann: That's true. There could be a game changer and we shouldn't be smug or anything.

McCarthy: I can be smug today, and sad tomorrow.

Hartmann: Right. Tomorrow somebody could have a game changer and wipe out programmable logic. We don't see that.

Frankovich: The fact that there's two is keeping each of them honest.

McCarthy: That's true. I prefer money to honesty though.

Frankovich: They're fighting each other, so that's probably enough to keep the new guys from coming up, because they are forced to leapfrog each other.

Chan: The one thing that I think that Altera was very successful, and the other company too, is not just the hardware. We were hardware, silicon designers, but one of the things that made programmable logic successful was the software. When people use the Altera parts, or the competition's parts, they don't really think it's a chip.

McCarthy: They are buying an aggregate.

Chan: It's software. It's designing logic with it. We don't do hardware anymore, the software is probably more.

McCarthy: That model, at least in my mind, is already well established.

Frankovich: That was stated.

McCarthy: Yes, but it was also evident that Intel had so successfully combined chip development across multiple technologies, with a software development, and that it was a killer approach. People that then pulled back on their software development, found their chips not as well accepted.

Smith: As a final point, for each of you, are you retired, are you working on a new project, building anti-gravity machines? What are each of you working on today? If we could go round the table.

McCarthy: Three years after I quit Altera, I decided to go to art school. I went to art school, naively perhaps, I figured if you want to be an engineer you get an engineering degree, so I went off and got a degree in art. Since then, I like other people, my friends from art school and so forth, it's always difficult to say you are an artist, but I try to make art. Most of it now involves art that has a computer in it somewhere.

Wong: Lately, I've been involved in multi-level flash, analog flash, and I'm semi-retired but still involved in IP, that is, patenting and so on.

Hartmann: I think I'm fully retired.

McCarthy: You're a project manager for a very large construction project.

Hartmann: That's true. I'm having a house built. We're competing with China now on our use of concrete. Not quite so much on steel, but quite a lot of concrete. I'm doing that, and adjacent to that property, I have some acreage, and have an 8-acre newly planted vineyard. So in some sense I'm a farmer. I'm returning to my very early roots in Minnesota. I come from farming country, and probably farming stock in a certain sense, and now I'm returning to my roots. That's where I am.

Chan: I tried to retire full time a couple of times and failed. Now I'm actually semi-retired, and have been consulting with different companies and helping them. Helping startups getting started. Also helping other companies to reorganize. Looking at the engineering structure, and helping them with things they don't know, because like I said before, today's engineers are too far away from the real technology. The real devices. People tell me it's hard to find people that have gone through the whole thing from the device level transistor, all the way to, I'm sort of familiar with the newest EDA tools also, even though I don't know how to operate them.

Frankovich: I guess I'm mostly retired. My biggest time sink over the last five years has been battle bots. I don't know if you guys have seen robot wars, but we have a 340lb robot called Ziggy that's ranked first in the nation. And our 60lb flame thrower bot, Texas Heat, that's ranked 8th. All of that's pretty much on hold because the entire robot team is working on coming up with an electric car for the Progressive X prize. They have $10M offered for a car that can get 100 miles per gallon gas equivalent, and win a 1200mile race. That race starts April next year, so we're furiously building a car. The car's been built up on Canada, and all the software and electronics are being built here in San Jose. We'll be shipping it out here in a couple of months. Lately, the last thing I've been looking at is green power, and coming up with a highly efficient generator. That's the latest thing that's on the back burner.

Smith: The Computer History Museum would like to thank everyone for their participation in the Altera EP300 Design and Development Oral History. Thanks!

END OF INTERVIEW

APPENDIX A: THE LOGICAL ALTERNATIVE

Frankovich: This is a picture of Leonard Nemoy. As a marketing ploy, Altera hired him at Westcon [conference] in 1984, to be at the booth, and sign these pictures to generate some interest. Obviously, with Leonard Nemoy there, a lot of people came over to the booth. That was the pull. The reason we got Leonard Nemoy was because Altera's tag line at the time was "The Logical Alternative". Obviously playing on his Mr. Spock and being logical.

APPENDIX B: EP300 MICROGRAPH

Frankovich: This is a photomicrograph of the EP300, the chip that we were talking about. It's approximately 4mm x 3mm. Here in the center is the programmable EPROM bit, in this area, these are the macrocells, here's the outputs, here's the inputs. Everything you see around this is support circuitry to justify this programmable logic.

APPENDIX C: EP300 PACKAGE AND DIE

Frankovich: If you want to see the actual die. This is an EP300 encased in plastic. It shows the 20-pin part, with the window that we had to develop for this package, so that the part could be reprogrammed. It was reprogrammed by exposing it to UV light.

APPENDIX D: FIRST ADVERTISEMENT

McCarthy: So this is the first advertisement. Before this, the company hadn't really announced, had kept quiet about what the company was doing. This was a very big event for us to come out with a two-page spread in an electronics trade paper, announcing the EP300. If you look at the thing very carefully, you'll see it says EP300-ES. These were engineering samples. The big event was: it was the public announcement of the product.

APPENDIX E: ALTERA LOGOS

Frankovich: What you're looking at is Altera's present logo. I'm going to take credit for that, because here's our original logo down here [editor's note: Robert shows the EP300 photomicrograph]. Similar, but Rodney [Smith] thought it looked too Star Trekie, or something like that. I'm going: that was the whole idea. We sent it to a fancy company and they altered it a little bit to look like that [the present logo].