COMPUTER
HISTORY
MUSEUM

Oral History of Aart de Geus

Interviewed by:
Russ Hall

Recorded: May 15, 2009
Mountain View, California

CHM Reference number: X5332.2009

Russ Hall: We have the honor of being with Dr. Aart de Geus and talking about his background and central role in the design automation, and just the impact that's had in the whole advance of computers. So Aart, tell us a little bit about yourself. Where did you grow up?

Aart de Geus: I'm Dutch, but I spent most of my youth in Switzerland. We immigrated when I was four years old. I do speak Dutch, and got the benefit of the Swiss school system, the Swiss Boy Scouts-- the first Swiss loves, I guess.

Hall: French side? German side?

de Geus: Good question. Actually, first in the French side, about eight years or so, and then moved to the German side, learned German and Swiss-German there, and then went back to the French side to go to college.

Hall: So you're trilingual.

de Geus: Yeah, at least that.

Hall: So college. Was that an easy decision?

de Geus: It was not a conscious decision, because at that time I was interested in playing guitar and blues and having friends, and college was one of those things you do because that's the continuation of school more than anything.

Hall: Hopefully you haven't lost that first love.

de Geus: No, absolutely. I still do play some electric guitar, and those are some of the best moments in the day.

Hall: When did electronic design, or even computers, start to enter into your educational side of things?

de Geus: It was part of the education the minute I went to college. I think it wasn't really conscious until I actually came to the United States. I did my undergrad and what now would be called a master's degree in Lausanne, Switzerland, but then decided after working for about a year and a half, to mostly make money to be able to afford to do it, decided to go to the United States, see the world, get out of Switzerland. It wasn't all that specifically driven, and going to grad school was, again, an easy mechanism to just escape Switzerland.

Hall: Grad school was easy to choose where you wanted to go?

de Geus: It was actually driven pretty randomly there as well. I got some recommendations. Turned out that some of those recommendations were for grad schools from the 1950s. Meanwhile, we were in the very late 1970s-- 1979, I think, is when I arrived here. It was also ultimately motivated by a girlfriend being in another college more than anything. But the randomness of that choice-- and it was truly very, very random-- turned out to be of incredible benefit, because I went to SMU [Southern Methodist University] in Dallas, and the day I arrived there-- actually, the minute I arrived there, I walked into the school. At the same time, the new chair of the engineering department, department of electricity, walked in-- electrical engineering. His name was Ron Rohrer. Now, in a minute I'll explain why Ron Rohrer is important. But we just arrived at the same minute. He just talked to me. I had no idea who he was. Unbeknownst to me, within a matter of minutes, he rewrote my application to where he turned out to be my advisor. Now, who is Ron Rohrer? Well, Ron Rohrer is really the guy who started the program SPICE. What is SPICE? Well, SPICE is the most long-lasting circuit simulator in the history of modern design. As a matter of fact, it still used today, and it has many different brands and names. Ours is H-SPICE, for example. But he started that as a graduate project at Berkeley, initially under a different name. The name was CANCER at that time-- Circuit Analysis of Networks and something or other. But clearly, Ron was an exceptional person. He was the youngest, I think, electrical engineering professor at Berkeley for a while, and he had just taken on the chairman job in the electrical engineering department. So he must have seen something in me, because from there, he arranged that I could do some teaching.

But most importantly, he had also cooked some deal with Texas Instruments whereby SMU received 10 or 12 of the early-- they were not called personal computers yet. The brand name was TI99/4, which was a real little 16K memory computer with some hokey keys. But it worked as a basic computer. The other nice thing is, you had no screen with it, so you needed to connect it to a TV. So I managed to not only get one of those and to take that home, ergo, I now had a TV, which was a whole new dimension to my education. Most people don't know that I learned English mostly by watching Mary Tyler Moore and the Jeffersons. So right there's a wealth of educational that came upon me. But what was really relevant to this discussion is that he said, "Why don't we see what we can do with those 10 computers, maybe connecting it to education." I managed to put together a set of educational programs on this little 16K memory computer that were the basic steps of learning electrical engineering. So the very first program was, "Here's a voltage source, here's a wire, here's a resistor. And now you can change the voltage, you can change the resistance, and it computes the current," or some combination thereof. So these became essentially labs for students learning basic electrical engineering, instead of doing the same by doing it physically. So again, completely non-conscious, I was part of the very first wave of creating computer-based education. So within one semester, I essentially put together about 10 of those.

But I have to say, there's something else that happened. I managed to, after writing the first program myself, to quickly hire four students to actually do the programming work. Looking back, that is really where I became a manager, and very, very quickly realized that if I could have some good students and some creativity, now there was leverage in that. So we created some actually very fun programs. I explained the simplest one, but one of the more fun ones was, "Here are three logic gates, and you get random stimuli, ones and zero. How many seconds does it take for the student to know what the output is going to be?" So this was actually not only an educational program, it was a little bit of a game. So educational gaming became a way to quickly learn about NANDs and NORs and inverters. So those 10 or so programs then became a lab section next to the basic 101 of electrical engineering course.

Hall: Do you remember the programming environment? Just sort of what the language, what kind of platforms were you programming back then?

de Geus: Sure. This was in BASIC, and the machine was, I said, a TI99/4, 16K memory. I don't recall what processor was in there. Probably some TI controller or something like that. Pretty simple.

Hall: Simple network, or any network yet?

de Geus: No networking whatsoever. It was so simple, yet there were 10 TVs in the room that were connected to some ad hoc connection to make it work. But the memory limitation was really the key thing. We used these down to the last 10 bytes or so. "If we make the picture just a little simpler, we save another 0.3 kilobytes," or what have you. It was literally an optimization problem in that sense.

Hall: So it sounds like Ron had a pretty profound impact on just your thinking, career-wise, and what inspired you. Maybe you can elaborate on just what-- other than maybe grabbing you as an advisor and putting you in this thing. How would you describe his impact?

de Geus: Well, Ron is going to show up in just a minute again. And you're right, he had profound impact. But the interesting thing is most of his impact was to take a kid with no clear direction and put him a situation where suddenly he caught my passion for something. He was showing me something I could build on, and the building on was both technically-- was both with an idea set out of nowhere, and this other thing, building and leveraging through other people, specifically students. Lastly, he did one more thing, which is he paired me up with a teacher who was just an extraordinary speaker. Notwithstanding the fact that I really didn't know all that much English, he helped me just completely overcome any fear of speaking because they were cool ideas. Well, if you have cool ideas it doesn't matter if your English is good or not. You're going to communicate that.

The teacher, a guy by the name of Leo Pucacco, had this amazing ability to convey thoughts, including his hands, which still to this day I think I do somewhat as well. He would say things-- "There are three fundamental things. There's this one and that one and that one," and then he had a fourth one. And I would say, "How can you do that?" "Well, you can do that because enthusiasm always makes you go one step further, right?" And he was infectious in terms of his ability to communicate with students, and I think I've inherited a little bit, some of that, or he just brought out my own passion for things. So that was sort of phase one with Ron. I don't think that he had planned it, but I think he had recognized talent or energy or passion, or whatever it is.

Then Ron, who has moved to many places, didn't stay particularly long at SMU. After about a year, year and a half, he moved on, and ultimately he ended up at General Electric. By the time I had finished my coursework, it was time to figure out what to do as a PhD thesis. He was still affiliated with SMU, and he said, "Why don't you just come spend a few days at my place?" At that time, in Charlottesville, Virginia. I ended up staying three months at his place, which gives a new meaning to the notion of mentoring somebody. It turns out, I think Ron has only had about eight or nine PhD students over his lifetime, and I think they all have had various forms of pretty good contributions. But part of it is this nurturing by virtue of just having a glass of wine together. But the other thing that happened, as I stayed there for these three months, he had meanwhile become a manager at General Electric, and essentially signed me up to start working at GE. He said, "Well, you'll do your PhD while you work at GE." Well, the reality is I really did my PhD at night and very quickly the GE work became very interesting, and I had an opportunity to build things there as well.

Meanwhile, both Ron and I moved to North Carolina because GE had decided to invest in the semiconductor business. This is now around 1981 or so, and GE had an objective to really get into high-tech. So they invested in semiconductors, they invested in factory automation, and the whole things. Ultimately they bought Calma, which was one of the leading EDA companies here in Silicon Valley. That story will come back in just a minute via disaster, building an opportunity for Synopsys. But before I get there, another thing happened at GE, which is I discovered that I could hire one or two summer students. Well, I had learned what to do with summer students. You put bright people, great ideas, and you just try out stuff. The one characteristic that good students have is they don't know what's impossible. Therefore, we will try. So the first year, I had one student. The second year, I must have had four or five. By the third year, I think I had 27. That's when GE noticed, because they needed tables and they needed computers too. But on the positive side, they noticed that I was able to attract some very good students, and I became a recruiter for the company. In the process, greatly helped by Ron initially to build connections, I built connections to some of the best schools in the world, because he put me in contact first with Berkeley and CMU and MIT, University of Illinois, and pretty soon I had both the research connection to those places and the recruiting connection to those places. Becoming a summer student or a-- MIT had this program over multiple quarters where you could come in a quarter or semester and then study a semester. I kept taking the best ones back. So that was an incredible filtering system, because after four or five years, I had a set of students that were top-notch in the intellectual department and potential department. Most people don't know that by the time I started Synopsys, six out of the seven people had been students in my group. So this was really-- we called it the Children's Crusade. And it was children, including myself, who were pretty uninformed about...

Hall: It sounds like there was a fairly porous boundary between the academics and business, and it's in part maybe an outgrowth of Ron and your background. But was the state of the art in science at that time, SPICE, was it just coming out of the university, and so it was inherently this connection between academia and industry?

de Geus: This is an excellent question because absolutely the state of the art at that point in time was that industry was sort of gearing up in terms of starting to use automation. But the state of the art of research and development was in schools. As a matter of fact, UC Berkeley was probably the singular most advanced or most at critical mass place, and they had many different programs, including he oldest one, which was SPICE, which was nurtured at that time by a fellow by the name Richard Newton, who had taken it over as sort of a project, and who actually-- Richard had immense technical talents, but also a lot of marketing talents. So he was essentially marketing this to the industrial community of, "Hey, why don't you team up with us? We'll keep the moles up to date. We'll listen to your needs. We'll make it work better and better. And give us some funding." So UC Berkeley had managed to get this affiliates program that essentially was able to very successfully get money and some equipment from companies, in the process invested more in these programs, that they would then give to the affiliates or make available in sort of standardized form. From of course simulation, it went to layout and to digital simulation, and a variety of other things. I would say the golden years of EDA development in this interaction between academia and industry was absolutely the late 1980s-- 1983 to the very early 1990s. That's really where this all happened.

Hall: Was there much of a role for the Bell Labs or the IBMs? Were they doing their own thing, or were they influenced by this? Did they have a part of the story?

de Geus: Yes, absolutely. They were still at a very high investment level, both of those, but the beginnings of decline were visible. But UC Berkeley specifically was very, very skilled at getting some of the best talent from Yorktown Heights, IBM Labs and from Bell Labs to come spend the summer. That's where Bob Braden comes in, for example, who I think to this day is partially a professor at UC Berkeley. Well, he was really one of the key gurus at IBM, right? Kurt Keutzer, who today is a UC Berkeley professor, was for many years at Bell Labs. So that community always tried to find which ones are the best talents, the latest, greatest ideas. Then of course the race for publication was on, between Don Peterson initially, and then Richard Newton and Ernie Kuh and Alberto Sangiovanni-Vincentelli...

Hall: And you have them at Berkeley.

de Geus: At Berkeley, right. Now, there were competing factions. There were competing factions-- MIT with some really good people; CMU went a little bit more to the more advanced synthesis direction. But at the end of the day, there were three or four places that just had a lot of movement. So partially helped by Ron, who had all the academic connections, partially helped by my connections via the recruiting of students, I then also became the affiliates' representative of General Electric at UC Berkeley. I still remember the first time I went there. I was a little young kid and here were all these people whose names I'd read on papers. Therefore, they must be God, right? And moreover, these were gods that knew marketing to the hilt, so they really manufactured themselves as being the gods of EDA, and their students must be walking on waters because they were on the papers too. And I was very naïve in many ways, but it was at the same time very stimulating.

Hall: How young did Rich Newton look when you first met him?

de Geus: They were very young, but they were in probably their 30s, just when you take off in your career. So they were very high energy. Everything they touched turned to gold. Being connected to them was the first assurance that you'd have a great job later, for many of their students. And some did turn out to be great, and some vanished into complete obscurity, as the law of averages dictates.

Hall: You teased us with GE/Calma, and an industry issue that kind of was the genesis of where the next leg of the story leads.

de Geus: It's interesting. Some of those things I have understood only fairly recently. So let me start with the biggest picture. The thing I've understood only most recently is as I started to graph the history of semiconductors-- which is really very much up and down. Semiconductors can go very much up and down as a function of the temporary economics. When I graphed all the peaks to bottoms, one after another, I saw that the biggest downturn was the 1985 one. That is until the 2001, which was much deeper, and now we're in a massive downturn, and it will remain to be seen how big that one is. But what I hadn't realized at that time was that in 1985 was a massive semiconductor downturn, and because of that, GE actually decided to get out of semiconductors. So that's piece of the puzzle number one. The second piece of the puzzle, now going back a little further, is-- I may have my years off by maybe one or two-- but Calma was probably acquired by GE in 1983 or so. Calma was a high-flier, one of the three. It was Applicon, CV and Calma were really the three, mostly layout-oriented, manufacturers of EDA solutions. And Calma was bought by GE. Part of this, "Well, we're going to put the big picture of automation together." Well, it didn't quite work out like that. A, Calma was probably not doing that well already. That's why they were probably for sale. B, managing a very fast-moving Silicon Valley startup is not quite the same as running GE. Now GE thought, "We'll do all the right things. We'll put our best management team on that," which turned out to be the aircraft engine team. Well, think of time constant. Aircraft engine, you start to design, and then if you're lucky 10 years later you'll be in some equipment. You think about EDA and Moore's Law and the change, and then Silicon Valley, hey, if people don't like it, they just walk across the street to work at some more promising startup, especially at that time. Things started to go very wrong with Calma, and this became a big program for Jack Welch, the CEO of GE, because here he is managing, whatever, 25 billion dollars, and every article says, "This guy walks on water." "Best manager ever seen in high-tech." Except for Calma, which is 200 million. So every *Business Week* article, that's doing so well on this 25 billion dollar company needs to have then one more paragraph explaining why Calma really is going down the drain. So this got completely his attention, and he put like task force after task force of "What do we do with Calma? How can we not fix this thing?" So I got involved in one of those as sort of a technical expert to help think through this, and because of that had the opportunity at least three or four times to go to GE headquarters. Well, pretty quickly you learn that walking into the palatial marble building with deep carpet floors, there's not that much there. It's just business. Smart people do smart things, and hardworking people move forward too. So what was a mere 15 layers above me in terms of pay grade, I did actually manage to, after two or three meetings, learn how to interact at that level.

So how do these play together? Well, they play together because GE decided to get out of semiconductors. Ergo, we were going to be laid off. And I actually interviewed with a couple of companies, one of them in Silicon Valley, and discovered much of the technology we had developed in my group actually is pretty promising, actually pretty advanced. Secondly, because of the interactions in the Calma interaction, I felt okay saying, "Hey, why don't we put together a business plan and go present it to GE?" as a way of, "Why don't you let us go? Give us the technology, maybe some investment. In return, you will get a stake in the company. And we are absolutely going to talk to GE because it's their intellectual property. This is an outstanding company that treated us and me very, very, very well, and that's just the only way to do these things." So because of the Calma interactions, I had an opportunity to literally have a one-hour interview to propose ultimately this idea to the vice chairman of General Electric. This the number two guy in a 26 billion dollar company, which was a lot of billions at that time-- and I'm, I said, 15 layers below this-- and in that one hour managed to convince him that that was a good idea. I didn't quite realize that when they said, "Why don't you raise the other four million, then we'll give you the million, including the technology," that sounded like, "Hey, done deal" for me. Of course, it was a much longer road to ultimately get the other pieces together. But it's really all of these circumstances that brought this about in a somewhat accidental fashion. And I think so many startups get created like that. It's because there's some disruption-- economic or technical or whoever you happen to run into-- and suddenly out of that comes, either by necessity or by vocation or by luck, a pathway that then you just work very hard on. And because of that, we happened to have the right technology at the right time.

Hall: So was this the birth of Optimal Design Systems? Is that the...

de Geus: Optimal Solutions. That is the birth of that. One has to go one more time back, which is, so why synthesis?

Hall: I was hoping you would get there.

de Geus: We'll get there. So let's go back a little bit, also about three years before that-- again, about 1982, 1983-- which is GE was in a thing called gate arrays at that point in time. What gate arrays were really fairly regular structures where you could have and/or inverter gates, maybe flip-flops as a combination of those, as a vocabulary of building blocks, arranged in rows, and then if you wanted to do a circuit, you would create essentially the netlist of how to connect those, and then the layout would essentially put them in the rows in efficient fashion. That worked reasonably well. Now, I had a read a paper, and actually later met the guy who wrote the paper, at GE, a guy by the name Shelly Akers, around this notion of binary decision diagrams, which was an alternative way to look at logic expression, instead of the standard NANDs, NOR, inverters, which was really supported very much by this information through Karnaugh maps. Binary decision diagrams, very simply put, are really the logic equivalent of multiplexors. Signal in, and you have either this way, or you go that way, and a multiplexor does exactly that. Instead of a transistor, it says, "Signal goes this way or that way." So the question that came up was, "Well, why would it not be more efficient if these binary decision diagrams were truly more efficient. Why do a logic in terms of multiplexors instead of NAND/NOR gates inverts?" So I talked to a buddy who was actually on the circuit design side, and he said, "Why don't we put together, instead of a gate array, a multiplexor array?" Well, pretty quickly we figured out that multiplexors don't regenerate logic, because they're passive. Whereas NAND and NOR gates actually do have voltage and ground, so they reset the signal. The conclusion was, "Why don't we do a combination multiplexor and logic?" and actually we called it the MLA-- the Multiplexor Logic Array. And he did a few circuits, and indeed they could be more efficient. The problem was though, all designers coming out of school use Karnaugh maps, know NAND gates, NOR gates and inverters. They don't know multiplexors. So designing with these things was actually difficult. That brought sort of the question of, "Why don't we try to automatically generate these things?" I didn't know this was called synthesis. I didn't know that what we were inventing was our local form of synthesis. Actually, I had no background nor knowledge of any of the stuff that independently was being done on synthesis. So in my group, now with students, or with summer students, we created this first program, at that point in time called SOCRATES-- Synthesis of Combinational-- fill in the blank. It was a cool name for essentially synthesis of circuits. It created automatically these pieces of logic from a logical function. Some of the designers tried those out inside of GE and found out that actually the results were quite good, and started to use it on their circuit design.

Hall: And their benefit was efficiency and ease of layout, or what was the...?

de Geus: The first benefit was you would write the function and 20 minutes later, you'd actually have a netlist. So right there, that's called automation. The second benefit was compared to you doing it manually, it typically used fewer gates. Right there, that's a big benefit because fewer is better, because fewer ultimately ends up in smaller area on a chip. The benefit that came later, as we evolved this, is we also managed to start looking at, "Well, where is the longest signal through this, and can we make it shorter?" i.e., "Can we make the circuits faster?" That in itself evolved dramatically over the years ensuing and really became a differentiator later in this story. But essentially it was the automation of a step that humans could do, and humans were very good if you had three, four gates. By the time you had 30, it's really hard. By the time you have 300, it is completely impossible to do manually.

Hall: This is the birth of SOCRATES.

de Geus: This is the birth of SOCRATES.

Hall: And the kind of designing, programming environment-- refresh us on that. What was the...

de Geus: Well, that point in time it was initially all in Fortran, and then we were sort of moving towards Pascal and C. Actually, I forget which one we ultimately ended up with.

Hall: And pretty much not anywhere else? Were the Kurt Keutzers of the world working on this at Bell? Was there a sharing of knowledge?

de Geus: Initially not. And then we published the first paper on SOCRATES, and slowly discovered, of course, there's all this body of people at Berkeley doing things, people at IBM doing things, people at-- a little lesser degree, but Kurt Keutzer came a little bit later at Bell Labs. Then there were a bunch of Japanese companies, but mostly you could barely read the paper and certainly not talk to them. Suddenly, I got invited to present at conferences and at seminars, really not knowing anything these other people's work. I always felt very awkward about that, because here you have these academics, and they not only have the paper but they also have the mathematical proof why it was better. Whereas we had just things that worked. To this day, if I meet with Alberto Sangiovanni, I make damn sure I never get into talking about complexity of algorithms, because he's going to run rings around me. On the other hand, our software today does 98 percent of all the circuits in the world. So success comes in many forms, I'd say.

But it was important, actually, both of those aspects, because by the time we now moved towards the next generation, I started to recruit and incorporate some of the academic line of thinking. And there was truly awesome thinkers and capabilities there. As a matter of fact, by the time we're arriving at, okay, we're going to be laid off at GE, we meanwhile are recognized as at least one of the synthesis groups in the world. We've built a set of connections to people and schools. Now there's slowly increased interest of, "Oh, all these guys may spin out." I had I guess recognized intuitively, if not only the technical wizardly, but certainly also the marketing potential of specifically Rich Newton and Alberto Sangiovanni-Vincentelli, and managed to get them as advisors in our spinout, which of course immediately gave more credibility because they really put a stamp of approval of "This is good stuff," and they had connections to many companies. So it created critical mass beyond sort of the yokels from North Carolina that were essentially trying to build something. From a GE point of view, this only added something else too, which is, "Here are some academics that think this is good stuff. There are some other companies that think it's good stuff." We managed to set up a funding scheme that said, "We'll have a few industrial partners, and ultimately some venture partners." Well, the venture partners, they knew Rich and Alberto well, because these guys were in those circles anyway, and by the way, they had participated in creating SDA systems that then became Cadence a few years earlier. I literally personally reverse-engineered the SDA business plan, financial plan, because I had no idea how you do this. So I sort of took their P&L and sort of said, "Well, if we did one of those, what would it look like?" I had no idea what the differences between revenue and orders, and sometimes our own sales force today still is struggling with that. But leaving that aside, the body of these pieces of knowledge all got sort of learned backwards. I think my entire career is based on learning by doing. I'm a street learner. I always think, "Oh, I just should have a year or two to MBA," or "If I had just been at a school like UC Berkeley I would know the algorithms too." But there are other ways to get there, and get benefit from the great skills that many of these other people brought. So that's sort of the origin behind Synopsys.

Hall: So they were sitting there. You need four million to launch this thing. You thought it was a pretty easy thing. Rich Newton and Alberto are giving you maybe some contacts? Pick up the story there. How easy was it? First-time CEO, got a team nestled in North Carolina...

de Geus: The first four weeks were a piece of cake, because GE had said yes to the equivalent of a million. I think it was 600K worth of technology, 400 in cash. We then when to Harris Semiconductor, which had done something similar with SDA, so they said yes to a million. I was thinking, "This is going to be a piece of cake." Then started nine months of cold calling, literally calling up companies, some connected via Rich and Alberto, others by essentially going through the phonebook for semiconductor companies. "Hi, I'm Aart de Geus. We have this great idea, this technology. It's already proven. All we need is one million." Invariably, it would be the same thing, which is the high-level guy would be very positive but have no clue what we're talking about. Or the low-level guy would be super-interested-- "We'll do a demo, we'll do a benchmark," and the guy has no authorization to even pay for lunch. So you're sitting in this high interest but no ability to really close on money, and that is where the venture arm came back in as, "We really should sign up some venture capital folks." Really I should be thankful to Rich Newton, who introduced a fellow by the name Jeff West, who had been one of the partners at Oak Investment, but had left Oak, but still had the connections there. He essentially said, "Hey, this is what you have to do. Let me introduce you to some of the VC companies. Here's how you clean up a little bit your presentation," and he introduced us to Oak and to TVI.

Hall: That was Bandel and...

de Geus: It was Bandel Carano at Oak, and it was Bob Kagle at TVI. Exactly. From there, one started to essentially build a connection system, and literally nine months later, in September of 1987, we rounded out the first round of, I think, 4.7 million, which was a lot. But some of that came from industrial investors, and about 2.7 million from venture capital investors. I still remember the day that the 2.7 million from Oak and TVI came in as two checks, because I personally took those checks and brought them to the local branch of the Carolina bank, or CCB, something like that. I walked to the teller and I put them on the teller and say, "Give it to me in used twenties." The lady sort of looked at the checks, looked at me, looked at the checks, looked at me again and said, "One second, Sir." She goes in the back, and less than 20 seconds later, the local branch manager-- the branch manager probably had six people reporting to him. He whisked me into his office, offers me a cup of coffee. The lesson I've learned out of that that I always share in business schools is if you want to get a free cup of coffee at a bank, all you have to do is deposit 2.7 million dollars, and you'll get that. But it was obviously a fun moment. So literally the minute we had the money, we moved the company here.

Hall: And the company at this time is you and...?

de Geus: And about nine employees. Something like that.

Hall: And most of those, the bright bulbs that you found from the university program and nurtured them along?

de Geus: Yes. Essentially the six others were all from my group, and then we had signed up two people to join us once we would move here. One of those is still with Synopsys today, is our senior vice president of all of technical services and support, has the largest group in the group.

Hall: Deirdre?

de Geus: Deirdre Hanford, who was a student of Richard Newton. So you can see how these things come about. I'm always incredibly proud of her, because this is somebody who started as our first application engineer, and today manages, at the top exec level, the largest group in Synopsys. Actually was president AEA as a side thing. We do these things too. So you can see that-- and I'm just as good an example of having had the luck of this machine that brought forth, year after year, of challenge and learning, and just also the opportunity of success. But the biggest gift has not been the success; it has been the stimulus of challenge and learning over so many, many years.

Hall: You've just relocated to California. You've got a board. You've got funds. Tell us a little about the state of the industry at that time. You were a young startup. Was it Daisy, Mentor and Valid? Who was active and what sort of environment were your customers and competitors looking at, at that time?

de Geus: It's interesting looking back how little attention I paid to the state of the industry. It was clearly Daisy, Mentor, Valid was coming to an end because Daisy was really going down fast at that point in time and the up and comer was SDA, that then tried to go public I think in 1987 on literally the day of Black October or whatever it was called and couldn't, and therefore merged with eCad and then became Cadence. Then Cadence acquired Valid a couple years later. So the Daisy, Mentor, Valid really atrophied and Mentor for awhile was very much king of the hill. They have had their own challenges in the late 1980s I believe. So there was a change of the guard around that. So there was definitely this first wave of Calma, Applicon, CV (Computervision). They essentially vanished. Daisy, Mentor, Valid had been there for a number of years and that changed, and if you look today, Mentor is the only one the three still alive. Cadence is obviously alive and we have been the one that not only came in but then became the leader of the industry.

Hall: Tell us about the customer challenge at that time. You've got this product that you think helps. How do you get the word out to get people to use it?

de Geus: To understand the customer challenge, let me put it in perspective where design is at. When we started probably around 1982, 1983, design was sort of in the 5 micron, 3 micron and by the time we are now a startup company, we're heading close to between 2 and 1 micron. One of the benefits of moving to Silicon Valley was that there were a lot of people paying attention and some we approached and some approached us because they'd heard about it, plus, of course, UC-Berkeley's pontificating like crazy about how this was going to revolutionize everything. So this is a very vibrant environment. Most important one of the customer visits I distinctly recall was to visit Sun. Remember, Sun is right at the moment where they're taking off, that move from I think the Sun-1 to the Sun-2 and it's now going to be UNIX and it's going to be workstations, which is another change from Daisy, for example, that sold a complete solution.

Hall: So the year is 19…?

de Geus: We're now talking late 1987, early 1988. We have taken SOCRATES and are doing a complete redesign to essentially take advantage of all the lessons and really productize it to the next level. But the great benefit is we have SOCRATES in its next incarnation, which we can use as a demo vehicle and interaction with a company like Sun. In that meeting is a fellow by the name of Andy Bechtolsheim, which I didn't know from Adam, and was sort of a quiet, sit-in-the-background guy but who had some good questions. They decided to do a benchmark and a benchmark meant at that time that people would give us a circuit that they had already designed and see if we could improve on it. We put the circuit in and literally in a matter of minutes, we have something that is 30% smaller and 30% faster. Of course, the first reaction is, "This is wrong. There's no way. I've slaved on this circuit for three weeks or three months, there's no way this is right." Okay, we'll go home and check it out. Then about two or three weeks later, they come back and say it was right. "We checked it out. It's really working."

Now we have a different problem which is now the level of expectation has just gone completely nonlinear on us because one of the crucial ingredients any form of optimization and synthesis or automation is the fundamental premise that you need to simplify things enough to make them automatable, and that's really what we had done particularly well. So certain constructs, for example, are not suited for synthesis and others are. The benefits were so large that many designers were quickly willing to give up on some of the other tricks they could pull, like different sized resistors or transistors inside of the gates, for example, in order to get that automation benefits and 30% smaller and 30% faster is actually quite substantial. Faster was key. The other thing that I've neglected to say is, during all this time, we substantially improved the state of the art of static timing analysis and build it into the synthesizer. If there's one thing that really distinguished us from not only the academic passage, which were mostly just logic optimization and the couple of other startups that had started also -- there were two of them -- is that we nailed timing, and timing is the reality of circuit design. Sure, everybody wants to make them smaller but the hard thing is to make circuits faster. And they trade off in many ways because if you can make bigger transistors, they run faster.

Hall: Can we dwell on that for just a second. Did the insight that timing was important, did that come from the customers? Did that come from the team? Did that come from just experience? Tell us a little bit more about it.

de Geus: I think that came from the very fact that from day one, we were never trying to invent synthesis. We were trying to automate creating circuits with people sitting literally in the next cube row doing design. In the early days, part of the way we approached it was actually with a rule-based system where you would say here's the logic we have. You look at this and the designer would say "I can do better than that. If you take this aid configuration and replace it with that, it's going to be faster." We call that a rule, right and we kept adding rules. Notwithstanding the many issues with that approach, it was taking actual learning from designers and immediately turning it into an improved solution that could be reused on many different circuits. So that interaction with the customer, that groundwork was already laid before creating Synopsys. To this day, we're absolutely committed to always work with the state of the art companies because we want to be exposed this "That's not good enough because..." and then fill in the blank and try to take the because and turn it into automation. I think much of the success of Synopsys to this day is linked to the fact that we've combined that attitude with actually about 30% revenue, still today invested in R&D, driving the state of the art forward. I absolutely think, maybe at bigger scale, that if you look at why computation has been possible -- and we're here at the Computer Museum -- so what has made this possible? They're two things: the evolution of technology and manufacturing and the evolution of design automation. Then designers have taken those two pillars and built these fabulous computers that we see, the same computers by the way that we now use to automate design with, the same computers that we use to now simulate manufacturing.

Hall: It sounds as if there's a balance here, and if one pillar were to get out of size—weren't to keep up, you wouldn't be able to advance the field.

de Geus: Absolutely. Absolutely. I think that Moore's Law is predicated on three things. One is drive the state of the art of technology and manufacturing forward and that mostly deals with how can you scale things, both scale in terms of doing more of them and in making manufacturing cheaper and cheaper. So there's an economic scaling and there's a technical scaling there. The other thing is how do you actually make that scaling usable. That is by automating the technology scaling, and that started all the way back to creating spice models for this thing that's called a transistor, to the using more and more transistors and gates; and that is simulation and synthesis and automatic place and route and so on. If one of those two gets out of whack, Moore's Law slows down, immediately slows down. Now I want to give full credit to the design community. Those are the guys that bring this together.

I always think of EDA as our job is to connect physics to function. The designers have one interest, the function. The manufacturing guys deal with the reality of physics. We make the connector between those two. So that's the trio that has carried for Moore's Law and of course, Moore's Law is predicted on one dramatic simplification, incredible simplification, the mother of all restrictions -- digital. Without that restriction, there's not to the 10, to the 12, to the 13 or whatever the number is of expansion that we've seen over 40 years. That doesn't mean that analog is very important because physics is analog. But our job is to free you from knowing physics as much as possible, make you be able to deal with ones and zeros, including now the whole other world of software that's sitting on top of that. That is why I think EDA is on par with the incredible accomplishments in technology and manufacturing, especially also in lithography that sits in the middle of that, as the two pillars upon which ultimately every compute system, every phone system, every game system, Google, Yahoo and so on system is built.

Hall: I bet this has had profound implications on where Synopsys has gone. Conquering one challenge and looking ahead to making sure you're capturing the next. Let's save that for something slightly different. Go back to just the birth of a company, the birth basically of a new technology and maybe a little bit of a major change in the industry. Before asking for some lessons learned, I want to drill a little bit into what you said about—the benefit of being the first one out there is that you get the input from all these state of the art designers, which is valuable. Are there any dangers that they might take you in unusual directions that may not be mainstream, or does that just take wisdom to weigh together being the first and foremost and getting maybe some extreme designers taking you in new directions?

de Geus: In general, one always will aspire in high tech as it's moving so fast to be the first. But it's also well understood that very often, the fast follower does better from a business point of view. I think we were, again, astonishingly lucky that we recognized early on that there was a necessity to understand some of the business aspects. I was extraordinarily lucky that in understanding this business necessity, I ran into Harvey Jones. Harvey Jones had been involved in Calma a number of years ago. I think he had been VP of Marketing initially at Daisy, then became at some point in time the president of Daisy, and he had recognized independently the power of synthesis. He brought of course an immense amount of business knowledge, experience but also acumen. He is just a fabulous business savvy person. He saw this technology, this group of people and this was like he could almost have a field day of "How do I bring this to market?" I give him so much credit for having known how to take the next steps, how should we go to design automation conference, how should we early on sign up some salespeople in this area and then in other parts of the US, when to go to Europe, when to start working with a trading company in Japan.

So by the time we're now in the middle of 1987, so literally 18 months after the formal departure, we are now on three continents, we are the Design Automation Conference. We have meanwhile developed the next version of the product now officially called Design Complier, and we have many customer interactions. It is true that if you don't watch out, customers will drive you to some destination that's only good for them. I think we were able to balance that early on pretty quickly and because we had many customers and a high degree of interest, we were in what Jeffrey Moore and his book "Crossing the Chasm" calls the tornado, which is you have more demand than you can actually satisfy. Harvey understood very well that "If that's the case, just keep selling." On the technical side, I knew very well keep this thing together because otherwise, we won't be able to support it. Because we were blessed with just outstanding people, all the functions evolved in a reasonably coherent fashion and we did not get, I think, off the deep end in terms of following just one or the other customer. At that time there were more design companies than today. Right now you see a consolidation of more powerful companies. At that time, there were not. The most powerful already then, Intel, they had their own thing. They did their internal synthesis. Yes, they would bring us one every six months because they were looking at buying; they were really training their internal team to work harder. So there were at a lot of forces at play but it was reasonably balanced.

Hall: Is there any element of methodology shift, of needing to sell a new theology of design? Did you have to not just sell a product but also sell an approach?

de Geus: Actually, the terminology "Selling a new theology" is very appropriate because for those that thought that, they were stuck because there were a number of people that said "This is going to take away our jobs. How can these programs be good enough." But because we were in a tornado, there were these other people that said "Hey 30%, this is a no brainer, let's go now. How do we use this to our advantage." So the Andy Bechtolsheim's of the world, they moved quickly. Wilf Corrigan, at LSI, who by the way had its own internal effort, actually saw "I can help me grow my ASIC business because [for the] ASIC business this is perfect. You're mapping two gates; we'll do the rest." The people that moved were so successful with it so quickly, that the ones that wanted to debate theology would just go to the next customer and not spend a lot of time with them. But there were absolutely these fears and I've seen this multiple times over the last decades, that if change is difficult and if the results are overwhelming, some people just will miss the boat. And in the process they will kill their companies or certainly slow them down.

Hall: You were the beneficiary of almost a Darwinian the success of the people you were able to sell to.

de Geus: Absolutely. Very quickly, it was a winner takes all. We had two competitors, Silc and Trimeter. We benefited from what in hindsight were mistakes they made. These were not dumb people. Not at all.

I'm not making that point. Trimeter actually had followed us on this rule-based system approach and that sounded so good because you could go to customers and say "We don't know everything. I'm sure you know more. You can add your own rules and then it will get better." That sounded great in theory; in practice it's not like that at all. You had a rule, it gets better. You add a rule, it gets better. You add one more rule, it gets worse because the rules interact. So you get into a nightmare of both maintaining the program and customers not being able to add good rules. So Trimeter ultimately went down the drain and I think got acquired by Mentor.

Silc was different. They were saying "It's pretty cool, that Synopsys stuff. There's gates, but really, really, the big benefit is to go to a much higher level of abstraction. We'll take behavioral description and do the synthesis from there." What they completely missed was that state of the art do move but as much as we have complete discontinuities in electronics, we have no discontinuities whatsoever. Everything needed to be backward compatible and the one thing you cannot jump the gun on is the quality of the circuits, specifically timing. So yes, they could synthesize from the high level down. The circuits they synthesized were just horrible compared to what a human could do in enough time, and you had to be at least better than a human. We were in that sense taking a smaller step. We were taking a natural step, and it was so natural that still today, behavioral synthesis, with very few exceptions, is not anywhere close to what would be necessary to do full automation. These waves do tend to last 10 to 20 years actually. I think it's back to this question of realism while being connected to real designers because at the end of the day, when the real designer says it doesn't work, it's not good enough, it doesn't matter how much you pontificate and sell and market around it, it doesn't work, it's not good enough, it's the end of the road for people designing multi-billion transistor chips

Hall: As we summarize sort of the launch, the startup, the success, any lessons—you benefited from some competitors self destructing or choosing some wrong paths, and then the shifting industry dynamics. Any more generic lessons learned, as you look back on that chapter, because I'm sure you wrestled with some things. Not everything might have gone right. Other lessons that you picked up.

de Geus: There are thousands of sort of business and growth lessons that are all retroactively obvious. But looking forward, they're mostly learned by either being lucky or by having run into the wall and trying again and again. So it's just amazing how some of the basics still today are completely valid. The quality of your team determines 70% of what you will be able to achieve. By the way, it's interesting that some of the last 10, 15 years, business books have emphasized it different than what was before. Before people always said -- and venture capitalists always said -- it starts with the market. You need to have a growing market. Today, built to last or best in class -- whatever the names are -- will say it starts with the people because if you're in the wrong place, good people can change, whereas not so good people in the right place will keep going wrong. So we were lucky, and maybe through that filtering system of the students and through the complete luck of being connected to Ron Rohrer who was connected to Rich and Alberto to being connected to some of the CMU people, I had the benefit of interacting with these best in class people while not knowing what best in class was. Then lucky again by being connected to some venture capitalists that absolutely fit the children's crusade, too. They were lucky but bright and hardworking. This combination of the harder you work, the luckier you get is true. It is also true that we were on the right technology at the right time. That's one insight.

The second insight is when starting the company both in terms of recruiting for money and then talking with initial customers, even when we didn't have the product, we had a prototype. A prototype used by some real users, that's a succession of risk reduction mechanism for any investor that's actually substantial. As a VC, you know very well how many people come with ideas where the idea is so great, it can only be successful, but for that fact, only 5% is successful. The prototype looks so great, it can only be successful but for the fact that only 5% of the prototypes actually get used in reality. Then comes the real example that's so successful, how can it not be successful? All but only 5% of the real success examples make it into a product that's successful. That decline or that filter is very, very tough. So always being a couple steps ahead has really helped us in the early days dramatically reduced the risks for all the participants. But for the people that did it, of course, I call that luck. I call that thinking ahead. It was a little bit be on the next step every time. I think if I look at our strategic thinking for the last now 23 years, we have always done that, which is always try to understand systemically what would be the key trends and how would they interact with the future, and it has governed much of how we built Synopsys, which is really the next segment in this story here.

Hall: You had talked about how important capturing timing has part of the synthesis product was. Can you just refresh us on why are the two—it's not just capturing the gates or the compression and translation, but timing is an integral part of synthesis. Why is that?

de Geus: As we introduced synthesis, there were really two key components to it, not only the traditional synthesis of how do you map essentially function into a netlist of connected gates and make as few gates as well connected as possible, mostly to reduce the area. The other just as important component was timing, meaning how fast is the circuit, and that's closely linked to a few things; one is the pathway of the longest signal and the other is what are the things on the pathway that are the characteristics of each of the gates. Most of the R&D work that had been done at many of the other places had all focused on this functionality. You get the first thing right. We were lucky to early on, by virtue of interacting with designers, get an understanding of how fast it is really matters too. As a matter of fact, making a circuit fast is harder than to make it smaller, and there's actually a tradeoff, meaning that you can make something faster by using bigger transistors, for example.

So de facto what we did is really move the state of the art of static timing analysis forward, build it into the synthesizer, and every time we would look at a configuration of gates, we would actually check out the timing as well and judge are we making progress on timing or are we making progress on area, or even better, on both. One of the things that was interesting with the early version, which was mostly a rule-based system, you go to the designers and they say "That's really good but here, you'd never do it like this because it's too slow. Do it like this." We would encode that as a rule that if you saw this situation, replace it with this better thing. And that was very enticing because immediately you say if I just had more rules, the thing's going to get better and better and better. The reality was, you get more and more rules and suddenly the rules start to interact with each other and it doesn't get better. So ultimately, it's a very sophisticated set of interactions.

If I can fast forward to today, we have added, at a minimum, one more just as important aspect as timing, which is power. As a matter of fact, not only has timing become more complex from those days to today because most of the timing is in the interconnect, not in the gates; but secondly, I would propose that today, the single most difficult limitation to circuit design is actually power utilization and power dissipation. The combination of timing, power, area, which by the way all trade off -- and now you can amend other things such as reliability in it, yield. Those are all dimensions that make what used to be just a scale problem more and more a systemic problem; in other words, the interaction of many aspects.

Hall: When we left the conversation, you talked about the early days of Synopsys, and coming together as a company. You had laid out one of the lessons was the centrality of getting great people, getting the right people on the bus. As you look ahead, are there techniques or things you've been able to do to keep that in the DNA? You're blessed with it back in North Carolina because of your integration with the university program. How have you kept that focus going forward?

de Geus: It's a good question and I think the answer is yes, specifically right now this minute, which is we're in 2009 and I think we have a fabulous team. If I look back over the last 25 years, I can see waves and each one of the waves invariably brings about some changes in the team or be necessity, the step up of some members of the team to their version and plus one. I love to use the terminology "version and plus one," which is well suited for software because it's equally well suited for people, for organizations, for companies, and it is plus one. It's not a version plus .1 and another .1, .1. You have that too, but invariably, they're sort of step ups where you go to a broader set of problems, where you go to the next level of complexity, not just incrementally but in mega steps.

From an organizational point of view, there are only two passways. You have to have people that grow, that have that capacity. That includes the CEO and I often argue that on one hand, I may be an enabler of n+1. I've also often been the bottleneck of to moving to n plus one and another person in that role at that time could have moved the company faster. At what point in time do you give up and select somebody else versus try yourself to go for n+1? I think the great opportunity living in high tech for many of the people that have grown up in this age is that the sheer high tech evolution has given us a personal n+1 trajectory that is just unmatched in history. How is it possible that we have learned so many new things and accomplished so many changes in such a short amount of time? That is why high tech is drinking from a fire hose and being CEO for a number of years is-- there's only one downside. There's no on/off button. The positive is this constant need for learning. You can instrument a number of those things by virtue of investing in the management curriculum. I teach the management course at Synopsys, the first three hours, and there are many, many more hours and most of what I teach is actually about values, about the fundamental ethics of doing business and the fundamental drivers of success, not necessarily how do you fill out timecards.

Hall: Integrity?

de Geus: Integrity I think is an absolute necessity foundation to building anything of any lasting value in any domain, not just business. It can be an art. It can be in so many domains. As a matter of fact, literally from the early days, we formulated a little bit the Synopsys DNA as being three layers. The first and most fundamental layer is integrity and you cannot 100% define it. We simplify it in terms of do what you say and say what you do because fundamental honesty in communication and actions is the basis for things.

The second layer on top of that is what we would call execution excellence and that has to be measured in terms of how does the customer see it. The success metric is the success of the customer, but it is sort of all the nuts and bolts of doing things well. So if you have integrity and execution excellence, you have a good company.

To have a great company, you need the next layer, which is leadership. Leadership implies setting the trajectory for the future, which invariably implies understanding the trends around, having a vision for where they're going to go and how to impact, and then putting in place the mechanisms to execute within the framework of that.

I always cheat and I put a fourth one on top of those three, and the fourth one for me is a flag, and it's the flag of passion. If we come back to a theme close to my heart, which is this notion of really good people, they all have passion. Whenever I look at a resume, I look for where did this person have passion. Frankly, I don't care all that much if it was in basket weaving or theoretical physics, maybe somewhere closer to us than others. But people that have passion learn so many things because they learn to persevere, they learn that there is such a thing as excellence, they learn that giving energy returns energy. They exude momentum and confidence. Any of the people you've talked to in these interviews, I bet every single one stands out being passionate potentially about many things, many things not related to the business we're in.

Hall: Do you have stories that would be useful to pass on?

de Geus: We sort of jumped fairly rapidly from 1988 to 2009, a mere 21 years later. We actually did do some work in that time. Also, there was an evolution around us, an evolution within us. Around us, of course, what we've seen from those days around numbers, was between 1 and 2 micron, to essentially bulldozing through the micron part limit that people have said lithography will never get underneath that. Today, we have customers working at 22 nanometers and we have tape outs at 32. So we've come a long way from that perspective but that has had huge impact. First, as you go smaller and smaller, physics do matter again, so it bubbles up every which way and we see a lot of that right now. So the costs are much higher to be in the development of new technology for example. Secondly, because things are much smaller, you get many more of them. That's what I would call scale complexity. Frankly, I think all of high tech, we're masters at that. We have licked this. Give us another zero, 10 to the 12th and to the 13th, we'll handle it. Don't worry about that. Scale complexity we can manage.

The challenge is, that when you put scale complexity together with physics and together with utilization of the scale, you now get systemic complexity. I alluded to that earlier by saying it's not just about fewer gates and more transistors. It's also about timing and it's also about power, and it's also the interaction of that with yield and the interaction over time with reliability with products that are, let's say, in cars for 25 years or so. So now suddenly, you have a set of interactions that make both design incredibly complex or you transfer that really to EDA being incredibly complex. So we wouldn't dream today about having any products that don't also understand power. That's a lot to be said because all of these things have to work together. So one of the very fundamental shifts we're seeing as we speak is that for many domains after 50 years or so of digital design, systemic complexity is forcing a variety of consolidations. You can no longer get a complete design flow done by just assembling tools. These tools have to play together really well. You cannot assemble a phone system without having many players work together actually without playing with all of the software guys. By the way, they need to be financially connected to the advertisement guys because the financial food chain is just as important as the technical food chain. So interdependencies or what I call system complexity in our field has, especially in the last 10 years, gone up very radically.

Another aspect that just now is going completely nonlinear is the fact that every chip is really not only a computer, a massive set of computers. We're going to go more and more cores. With that comes embedded software and on top of that, massive amounts of software. And by the way, connections to the real world -- phone systems, net books, you name it. How do you verify this all works? The notion of systemic correctness has just gone completely nonlinear on top of scale complexity that is still going nonlinear. Ergo, we still have a lot work to do so there's a lot of opportunity.

The other ramification is we are seeing this in manufacturing, in technology development, in architecture development, in EDA, by necessity, there's a gradual consolidation because many small parties cannot afford the R&D necessary to solve the interconnectedness of the problem. I think we're yet in another wave of change and now 2009, being simultaneously in a massive economic downturn. What downturns do is mostly accelerate things that were going to happen anyway partially because change is hard, but if you're in deep trouble, you may as well take some risk anyway. So it moves things forward, or by necessity. Just like by necessity, we needed to start interviewing as GE got out of semiconductors. So by accident, we discovered that there was an alternative, like form Synopsys. So that's what downturns tend to do in high tech. It actually accelerates things. One of the things that's accelerating is really a reconfiguration of the semiconductor industry, of the design industry and EDA industry. So that's how you can explain it from a technology point of view or from economics.

A term I've been using for a number of years is you build a tech-o-nomic that says there's constant interaction between the technology and economics and the one that's driving is always the most recent, bigger change. So if there's a big technology change, the economics will follow and will follow where the new opportunities are. If the economics change, the technology needs to invent something new because otherwise, we're in trouble. So tech-o-nomics I think is the way one needs to learn to understand trends and when I talk about leadership in a company as sort of the top of the pyramid, it is really understanding the tech-o-nomics of the field that one is in. If you're a small startup, the tech-o-nomics may be a small area. When you're a larger company or a pillar of an entire industry, you need to understand sort of all of these trends.

Hall: Go back 21 years. Are there any particular times that would be good examples when areas were wrestling with one of these dislocations and changes?

de Geus: There's actually one that is the alignment of, in my opinion, multiple dislocations from the physical, all the way to the macroeconomic, which is around 2001. Let's start with the physical. What happened then? You look at circuits at that point in time. You wanted to know the speed of a circuit, you sort of looked at the longest path through the gates, you add up the delays in the gates. That just changed. Because the transistors became so small that most of the delay was not in the gates, it was between the gates, the wiring in between. By the way, if you have more gates, they're going to be further apart, relatively speaking. So almost overnight, and it really happens between I would say 1.8 and 1.3 micron. That's really where it manifested itself markedly. It forced a complete rethink of layout, of the synthesis tools to deal with the fact that the delay's in the interconnect. That's one thing that happened. The second thing that happened is that between 1.8 and 1.3, suddenly people said "Let's deal with that interconnect and why don't we go copper," copper metal because the currents are faster. You can essentially have a faster signal there.

So copper and local dielectrics, two new materials occurred simultaneously. It threw off the evolution of the next node by about five quarters, which is very substantial in something where every next node comes on a two year timeline. Third, around that same time, the entire industry is moving from 200 millimeter to 300 millimeter fab. So you're saying "That's just a little bit bigger wafers." No, it's not bigger wafers. Every piece of equipment is new. Every piece of equipment is as sophisticated as a computer or a software program. In other words, it has bugs and so there's a massive debugging while simultaneously, the price tag goes up from about 1 billion to 2.5 billion to 3 billion per fab. So now you're starting to touch the economics, which is the price to play has suddenly gone up massively. At that very moment, lo and behold, the end of .com. The end of the ultimate high tech party in 2000 was Y2K. It made people spend a lot of money. It's not about revenue; it's all about clicks and eyeballs. All of that came to a crashing halt and we saw at that time what was by almost 2x the biggest downturn in the semiconductor industry in history. So you can see that from the microscopic level all the way to the macroeconomic, they all hit at the same time.

de Geus: Last, but not least, on top of that you have sort of an economic meltdown in 2001 and that is caused, of course, by the end of the great party of 2000, where not only did you have the end of the dot-com era, you had Y2K, which was enormous amount of spending, not necessarily for the right reasons. You had the change from it's not about revenue, it's about eyeballs. All that was completely over, resulting in the semiconductor industry essentially going down twice as much as any previous recession of about 46 percent. That is enormous for any industry. So the combination from the most minute physical change all the way to the most macroeconomic Gestalt of the entire industry all occurred at the same time and brought about a huge shift in the industry. It brought about the shift that many semiconductor companies went out of manufacturing. They couldn't afford it anymore. So we see the growth of the foundries come up massively. We see a lot of the design becoming more concentrated. So many shifts. Those are some of the big things that occurred in these 20 years, and there've been a number of these other earth shattering difficulties to overcome.

Hall: You're sitting there at the helm of Synopsys. These changes are going on. They are certainly affecting Synopsys at the same time. You're sensing needs evolve, needing to pick up thermal or power distribution, power management, especially I imagine as you get more consumer devices that are smaller, thinner. Grow from within? Go find someone that's doing it well that would match as a culture? How do you wrestle with the evolution of Synopsys looking at new capabilities you want to pick up in serving your customers?

de Geus: We're sitting in an environment that has arguably the highest rate of change in the history of mankind now for half a century, which is amazing every which way you look at that statement. We're sitting not only in the middle of that, I would argue we're one of the key enablers to that. We wrestle with what are the trends? Where are they going? What is practical? How do we execute? Oh, and by the way, please remain backward compatible to the last 10 years of everything you've done. Right there, that's the challenge.

Now the second comment to that is an innovation can occur in many places. On the one hand, we absolutely must invest heavily into internal R&D to keep that engine going and try to push all the frontiers forward that we know are going to be issues or will be key trends. We do that by spending literally about 30 percent of our revenue constantly on R&D. On the other hand, there are good ideas at other places. And, by the way, there are also other places that have grown some revenue and you need that economic engine to fund that R&D and moreover, and that is new in the last 10 years, it's increasingly important to have these streams of product actually be compatible. Otherwise, you have terrific product here, terrific product there and together lousy results. Not any different than in, let's say, a modern car you were to say, "I'm going to design the engine and the transmission and the wheel system independently." They may be terrific on their own; the complete system will be completely inefficient. In electronic design it's the same thing. So in our history, I don't recall the exact number, but we must have done over 50 acquisitions right now and they range from very small companies, two or three people with a great idea, all the way to the largest in our industry which was a company called Avanti, that brought us a wealth of what's called place and route which is really the programs that take the net list, the functional representation and map it physically as to where are the gates on the chip.

Again with the objective make it as small as possible. Make it as fast as possible. Make sure that the heat dissipates in all the right places. So these acquisitions have become for us absolutely core competence and a skill on how to do it, how to evaluate, how to judge the value, how to integrate, how to sell over time. But the R&D is just as important. So there's always this tug of war of, well, you could do the R&D now and the challenge there is you will spend part of your P&L to do it. P&L is always under pressure. Or you can say, "No, no. We'll buy it" and it's off the balance sheet. Nobody will complain. It's often not as efficient than if you did it yourself. That balance is never ending.

Hall: Leadership. As you look ahead, just sort of growing that culture within Synopsys. Anything you can speak to, just ways to help companies manage the leadership evolution, growth, nurturing, care and nurture of future leaders?

de Geus: I think there, too, one goes from the microscopic to the most macroscopic. The higher up the leaders are, the more they have to be able to work at many layers. Actually, I think one of the characteristics that makes high tech management, in my opinion, so different from a number of other things is that in high tech, (a) because of the evolution being so fast in technology; (b) because the evolution having been already so long in technology, it's almost impossible to be a leader at a high level without having depth and technology. Right there you encounter the first major dilemma which is if you spend most of your life learning about screws and so on, how can you even think about knowing anything about management? That's the opportunity. This is for people that have passion to do it all. Ideally, we would like to have renaissance type people that know enough about everything so that they can judge if technology's good or where it's going to go and yet also know everything about change management and, by the way, also know how to act in a global environment. Oh, and by the way, know enough about what the reaction of Wall Street is going to be.

So that then begs the question: How do you grow or groom such an organization? I think the first part of the recipe is to actually share a lot of the learning as broadly as possible. That's only possible if one has an atmosphere of actually being quite open about challenges, about what's happening and so on. But secondly, it's to try to attract people that fundamentally believe in the n+1 [redundancy] version first of themselves then of their group and then of their business unit and then of the company and then of the state and then of the world. I'm not surprised that many of the Synopsys top people also engage in the community, maybe active in politics or at least trying to help in all the challenges that the nation and the world face right now. As a matter of fact, one of the things we try to teach in our course is to paraphrase Gertrude Stein, "A leader is a leader is a leader." That means you want to be a leader at Synopsys? I assume you're also a leader in your family, in your community, in the environment that you have an impact on, because it's a state of mind more so than actually a skill set. The skill set follows the state of mind, not the other way around.

Hall: You mentioned change management.

de Geus: Yes.

Hall: I think I just heard a statement saying, “If you want things to stay the same, they have to change.” It’s one of those conundrums. But change management is almost, by what you’ve been saying, at the heart of the evolution. You have to be able to stay on top of your game to change and evolve. I just wonder if there’s some things you look at that would be cautionary to you, either in a company you’re acquiring or even things, warning signs within Synopsys that you would say, “These are things that set off warning bells for me. Change management. How can I encourage it? What do I worry about if I see these things happening that might signal ossification?

de Geus: I think none other than one of the pillars of the high tech industry, Andy Grove, said. “Only the paranoid survive.” I think paranoia is this sixth sense of knowing what may change on you before you change. Just because you’re not a paranoiac doesn’t mean they’re not after you to get you. They are. The reason one is always in the crosshairs of other forces is because the very nature of being successful is to create value. And the very nature of having value is to want to protect it. And the very nature of wanting to protect it is to try to make it not change. As you said, the more you want things to be the same, the more you have to change. That’s because what you want to be the same is you want to be your standard of living or your feeling good to stay the same. But underneath, you’re constantly building on sand. So there, too, I think there’s a piece of the personalities one needs to put together. Where there are some people that are constantly defending the fort and solidifying it, and there are others that constantly keep saying, “Well, let’s put in some windows here, because I know it’s dangerous to have windows if there are attackers, but at least then we can see outside” and be exposed to the change factors. What better way to be exposed than to be close to a customer? If there’s one thing, at least within the domain that one is in, that is a pretty good indicator. If the customers have more trouble or are changing course and you’re not there, oh, you’re going to miss that bend something ferocious. So customer interaction is actually one of the mechanisms for change management.

But the other mechanism is, simply put, courage. I find that often people know that it’s necessary to make change and the reason they don’t move is because of the second premise, execution excellence, which is if I’m changing, I don’t know how to do it. Therefore, I won’t be executing all that well. You know what? I grew up as an A+ student. I always do the exams well. I always execute well. Well, A+ students are notoriously bad artists, because artists need to always question things. They break it and so on. So one needs to have leaders that innately have a little bit of this tendency of on one hand having A+ aspirations, on the other hand, wanting to have their feet in the mud and break things and try out wacko new ideas. It’s interesting that the higher up you get in organizations, as much as people think that it’s fun if somebody’s really creative, most of Wall Street wants you to be more and more professional and buttoned up and not fall outside of the norm, because they don’t know how to read the outside of the norm. Yet, it’s being outside of the norm that’s the creativity. So there’s a tension pair here on expectations and necessity.

Hall: Let’s look out a bit within synthesis, if that’s the right frame, or a larger the EDA set. I’m tempted to build on the puller metaphor used earlier. I don’t want to get into any secret sauce for Synopsys, but maybe look enough beyond that that we're talking about, that evolution in general. Where does the EDA industry need to evolve that it keeps pace with and in sync with where the scaling and the--

de Geus: It's interesting being at the Computer Museum, because a museum sort of preaches that if you understand history you will learn from it instead of repeating it. In our domain, learning and repeating are, in some ways identical. Let me explain. If you look at the history of EDA or the history of modern design, it's somewhat overly simplified. It's building blocks that you connect. What has changed are the building blocks. The very early building blocks were essentially a few rectangles that you connected them and now, "Here you go. A transistor, wow!" Then the building block became a transistor and you connect those and, "Wow, here you have a gate." Then you connect some gates and now you have a flip flop and then it became a little macro, a little adder or so. Then the macro became a much larger IP block, building block, maybe a processor core or now a complete USB core. Then on top of that came some embedded software. So you can see my initial principle which is the Lego blocks have become more complex. By the way, I love the Lego analogy, because growing up with them, maybe that's the only thing I understand, the 2 X 4 was the Lego blocks. That's the transistor of semiconductors. Then after a while we come some better Lego blocks, like some wheels and then came that magic day, it must have been mid 1960s or so, here comes the Lego engine. Now you have a processor core inside of your chip. Then came complete rail systems--USB package and so on. Then came this moment even Lego started to say, "You know, we have some software to go with it." We've got some software to go with it, too, in the chip. As a matter of fact, most chips have a lot of embedded software.

So if one fast forwards, one can say what is going to continue to happen is that the subsystems are going to continue to grow in sophistication. The interaction between the subsystems is going to continue to grow in sophistication. That's the argument of systemic thinking, all the way to a software subsystem interacting the power consumption of a hardware system. I'll give an example. A number of the systems around the arm core right now, actually this was enabled by Synopsys, was to, as the data stream slows down, let's say you speak, it's high and then you don't say something for a tenth of a second, there's no data. Let's slow down the processor. If you slow down the processor when there's no words, you save energy. As soon as words come in, you make this thing crank again. That's a good example of systemic complexity between some of the software, some of the hardware. We're gong to see that just evolve to the next level up. The interesting thing is every time one moves up one level, one also sees a wave of new tools and new evolution in the EDA industry. So we started with the spice program, which is really EDA system for transistors. Then at some point in time we got to basic gates and with that came simulation and then later synthesis and then came automatic place and route to connect those things. Now we're talking about 70, 80, 90 different programs, all crucially important to make a chip. These waves of productivity coincide with waves of abstraction. Going forward we're going to see that in entire subsystems and the very fact that we're talking about a system on a chip. The system is really a complete compute system or communication system on a chip today.

Hall: Two last questions. One is that next generation of Aart de Geus' that are out there, maybe they're back in Switzerland, maybe they're in a high school here, maybe she's somewhere thinking that she has an interest in science. But what would you say to them, the next generation coming? Any words of encouragement or advice or anything you'd pass on to the scientists and engineers of tomorrow?

de Geus: I have one, maybe two at home of those. I've got two daughters and one, for sure, is heading in towards the science engineering direction. I certainly don't want to influence her or any of the other people you're talking about in a specific direction. But I am absolutely convinced that we have a number of decades ahead of us of equally high impact technology evolution. You can see it in a number of domains where fundamental semiconductors leading to design, leading to computers is now leading to dramatic changes of what life is all about. Let me take just a couple of examples. I consider the breaking down, the segmenting of the DNA to be on par with what happened around 1500, when it's only the map of the world became coherent as a globe and every piece of data became 100 times more valuable, because it had a place and it interacted with others. The same is happening with DNA today. Compute technology and semiconductor technology will allow us, in a matter of a couple of years, to crack a complete individual human DNA for less than $1,000. That is changing the way we understand human life and the way we will impact human life with many more discoveries.

Earlier we were talking about the fact that there's some people building a complete compute model of the human brain. That will change a whole bunch of things of how we think about thinking, to understand what is happening there. Take the mother of all systems, the Internet. Internet is the ultimate connection machine between us and most of what we, us, all together know and do. That is of such a magnitude in terms of its potential that there are many, many, many opportunities. So when talking to the next generation, I would say you have all these opportunities. That's not the point. The point is to find your connection to those that really connects to your passion. Because more important than having impact on all of this is to be happy doing it. In other words, the most precious thing you have in your life is your life and what you do with it, if you can leverage it by this plethora of opportunity, that's great. I think you'll leverage it most by finding your passion in whichever area it is. Don't do stuff that's boring. Don't do stuff that you don't like. If you have the luxury, of course, economically to be able to choose.

Hall: Let me toss one more question before I get to the final one. Blues and EDA. I imagine that you've been able to marry your passion in a way that actually indulging in both rewards the other. Is that right?

de Geus: Yes, in some ways. What blues is, it is maybe the simple form of jazz, which is really improvisational music mostly, where you have certain basic chord patterns and then as a group you improvise around those and if somebody takes a solo, they're a little bit more in the limelight at that point in time. But you essentially play together by listening a lot to each other and then building off each other. Actually, one of the most fun things, and I could use the same description for a cool staff meeting or a cool jam session, is to pick up some ideas and build on it and the other person realize that one heard the idea and did something with it. In other words, that's team play. I think what is interesting about the analogy with jazz as a broader form or blues which is a little bit more narrow and therefore, easier for me to play, is that it contains both elements of structure via chords and certain rhythmic expectations. But improvisation as breaking out of the mold and doing something unexpected. Then it brings it together by it being team work, which to me is the most fun way of learning, of interacting. So maybe moving only the paranoid survive which tends to be a little negative to only the creative create is creation when built on structure has a lot of potential for many people. I think that if you look at any modern organization or modern high tech company, the top notch high tech companies such as Awesome Team Play engines. An analogy I use sometimes in management is actually to soccer, which is just another form of this, which says you look at the little kids when they go to soccer class at first or soccer the first time. It's like a beehive. They all follow the ball. First rule, play your position. Then comes the second rule, completely opposite to that--be where the ball is. How can you play your position and be where the ball is? That's what good teams do. You're the CFO; I'm the VP of R&D. You're responsible for your thing, but if there's an issue, still rush to where the ball is and we play together. Then that comes to the third rule which is a good team always wins over great individuals. Team play is actually more powerful in what we do. That is far away from the Middle Ages, by the way. As much as we like to have renaissance people, we really want a renaissance group. Then the last group is the hungrier ones still one, which is the harder I work, the luckier I get. So you can say exactly the same about music, about management, about discoveries. It's team play and there's skills and there's attitude.

Hall: Of necessity, we had to jump over a lot of time and circumstance. Is there something we missed? Anything that would be good, an anecdote, either a myth that's out there that needs to be corrected, any story that would be worth capturing that would be great to insert at this time? If nothing of those, anything you'd like to ask your fellow participants in this journey to have them think?

de Geus: I think that rather than a question, I think there's a statement of thanks that is actually the normal closure. Notwithstanding whatever one thinks of our or my contributions, they are so small compared to the sum total of just all the people you can interview. You're interviewing a number of people that have had fabulous contributions, many of which I've had the privilege of building on. We're building on the fact that people have built fabulous computers that are driving the state of the art that make this possible. We're building on the fact that there are a bunch of people that have moved the state of the art from just doing net lists to inventing languages. Some of the languages were great; some were not. But over time it evolved. So much is standing on the shoulders of giants. What is, I think, so extraordinary about the time that we live in is the fact that all these giants are within our own lifetime. Earlier we were discussing why these tapes may be relevant 500 years from now is because when you look at history, history, too, has n+1 waves. I'd say some histories are continuing as well. But still you can see the waves. Specifically working backwards in high tech, you can say, "Well, one big wave is now. One big wave is the whole industrialization wave. One big wave way back when was when they introduced wheels." There are not that many. There are not many. We have the privilege of being in the middle of one of these and what is different, I think, than many of the others, it is so fast that within our own lifetime we have seen things that today we would consider the high tech middle ages. You wouldn't dream working on the computer we started with today. As a matter of fact, you wouldn't want to dream working on the thing from two years ago. So that is what is so unique about this time. Therefore, one needs to be so conscious of the interdependency of all this fabric of great brains that have worked together.

Hall: Well Art, thank you very much.

de Geus: Thank you for the opportunity.

END OF INTERVIEW