Oral History of Randal E. “Randy” Bryant

Interviewed by:
Douglas Fairbairn

Recorded: June 16, 2014
Mountain View, California

CHM Reference number: X7201.2014

Douglas Fairbairn: OK, we're here at the Computer History Museum. It's June 16, 2014. My name is Doug Fairbairn, and I'm here with Randy Bryant, who's come to share his experiences in a career focused on design automation and related topics. And we'll find out what else if there's other things to cover as well during the discussion. Welcome, Randy. Happy to have you here.

Randal E. “Randy” Bryant: Thanks. It's an honor to be here to be interviewed-- to be considered worthy of being interviewed, so--

Fairbairn: [CHUCKLES] Well, you've been on our list for a while, so I'm glad we finally made it work. What we'd like to do here is sort of find out where it all started, and how you got to be where you are today. So that involves where you were born, what kind of family you were born into, brothers, sisters-- especially in terms of how that may or may not have impacted and guided you towards your career choices. So let's start. When were you born? Where were you born?

Bryant: So, I was born in 1952, and I was born in New Jersey. My father was working for IT&T that time, which was a real electronics company. He was a microwave engineer. He'd gotten his PhD after World War II. And he'd met the department head's daughter and married her--

Fairbairn: [LAUGHS]

Bryant: So my mother was my grandfather's--

Fairbairn: And what school was that?

Bryant: University of Illinois.

Fairbairn: OK.

Bryant: So my mother's father was an electrical engineering professor, first at Ohio State and then at UIUC for many years-- in fact, he was the dean of engineering there.

Fairbairn: So you come from a long line of PhDs and professors.

Bryant: Yeah, well, his ancestors had all been Presbyterian ministers, so he's the one that really broke the trend and then set this other one of becoming engineers.

Fairbairn: They all spoke from a podium, though.

[LAUGHTER]

Bryant: Yeah, they all spoke from a podium. And so my grandfather clearly was a big influence on my life. As a kid, we used to go down and visit Urbana-Champaign, and I thought it was the coolest place on earth, with all the student union and ice rink and bowling alleys and stuff like that. And just the whole college life seemed great. He'd take us on tours of various labs, and I just was really enamored by that.

Fairbairn: So you really had it from the beginning.

Bryant: I had it from the beginning. My mother's bachelor's degree was in physics, so--

Fairbairn: Oh, boy.

[LAUGHTER]

So the daughter of a professor.

Bryant: Daughter of a professor, although she did it more to get into journalism, so. And she'd sort of, in the classic mode of her time, sort of out of college gotten a job, and then when she got married, she kind of set that aside to become a mother. But as soon as we kids-- and then, oh, by the way, my father moved to Michigan, Detroit area, when I was about three years old to become part of Bendix, which was aviation--

Fairbairn: Avionics.

Bryant: Avionics, electronics, at the time.

Fairbairn: But you were born in New Jersey.

Bryant: I was born in New Jersey, but--

Fairbairn: But then moved to Michigan.

Bryant: Moved to Michigan.

Fairbairn: OK.

Bryant: And I really spent-- for Michigan, I was there up through high school and then through college. And my mother still lives in Ann Arbor, Michigan, in fact.

Fairbairn: So brothers, sisters?

Bryant: Yeah, I have two sisters. One's older, and one's younger. Neither of them have gone the technical route, so I was the family nerd.

[LAUGHTER]

Fairbairn: Were you involved in doing electronic projects or anything?

Bryant: A little bit of electronics. I more did mechanical stuff. I went off to college thinking I'd become a mechanical engineer, in fact. I mean, I liked electronics well enough, but somehow, I liked more physical stuff. My father had a pretty good workshop, and I'd build a lot of stuff. And built go-karts and stuff like that. But anyways--

Fairbairn: So you had your path pretty well set.

Bryant: I had my path pretty well set.

Fairbairn: What about during high school or whatever? Were there any teachers that had a particular impact?

Bryant: So I think the classes I really liked the best were math and physics. And I was fortunate. I went to a very strong suburban high school where they had advanced math courses, and especially the math teachers were very encouraging of us getting involved in these math competitions that were held within the state of Michigan at the time. And sort of got a group of maybe half a dozen or so of us, all boys, who kind of clustered together and would study for these problems and stuff.

And one of the math teachers-- his name was Mr. Thumser-- was very encouraging of that kind of stuff. And so that sort of combination, which I look at still today-- the combination of sort of the theoretical math side of things and the practical building side of things-- still I like that. And that's what I've found through my career.

Fairbairn: Really served you well.

Bryant: Yeah.

Fairbairn: So it came time to go to college. What were your choices?

Bryant: Well, I really, really wanted to go to MIT, and I didn't get in.

[LAUGHTER]

And I kind of wanted to go to Stanford, and I didn't get in. I got into Michigan and a few other places, and at the time, I was at this big suburban high school. So I think 74 of us from my graduating class went to Michigan. And to me, that felt like, oh, geez. It's like--

Fairbairn: This is 68?

Bryant: 70. It just felt too obvious. But of course Michigan's a good school, so it was not an unreasonable choice. But I really wanted to be more at a nerd school, quite honestly.

Fairbairn: Was Caltech anywhere on your radar?

Bryant: You know, I didn't even-- I've thought of that often, that I would have done well to have applied to Caltech. And I don't think it even, for some reason, crossed the--

Fairbairn: Wasn't on your radar--

Bryant: On my list, I think partly being more on the east.

Fairbairn: Yeah. So you headed off to Michigan.

Bryant: So I headed off to Michigan.

Fairbairn: You stated a major in engineering, or what--

Bryant: So I was in the engineering college all along. And at the time, they didn't have computer engineering. They had computer science, and it was part of Arts and Sciences-- a bit quirky at Michigan at the time. And like I said, I actually came in thinking I'd be a mechanical engineer. And a couple things happened. One was we all-- there was a mandatory course in Fortran programming, and I remember my first program was to convert from Celsius to Fahrenheit.

Fairbairn: From what to what?

Bryant: Celsius to Fahrenheit--

Fairbairn: Celsius to Fahrenheit. It must be everybody's first program.

Bryant: Yeah.

Fairbairn: Now that you mention it, I think that was mine, too.

[LAUGHTER]

Bryant: Well, you literally were given the exact text of it. And the job was to type key punch it in, submit the deck with 10 different numbers you asked to do the conversion for. But I was hooked the moment I saw my cards get sucked into the card reader and then this roll of paper shooting out afterwards-- through the window of the computing center, because of course--

Fairbairn: Of course.

Bryant: No one could get near to the actual computer. I was hooked. And so that really triggered my interest. And I decided to major in applied math at Michigan, which at the time, was a route you could take that was within engineering. But it was a little bit self-defined, and it was a little bit let you go more toward computers.

Fairbairn: Was that sort of the closest thing to a computer science degree?

Bryant: At the time. I mean, maybe part way in, the electrical engineering department expanded and started providing a computer engineering degree. And for some reason, I didn't decide to switch over to that. But I ended up taking courses in math, and there are some courses I had to take in various parts of engineering. But I did gravitate more towards the electrical engineering side of things. And in fact, I remember, speaking of hardware, the first course I took in hardware design with TTL logic, of course, was by Dan Atkins, who's still alive and kicking and quite well-known at Michigan.

And again, I was hooked right from the beginning of that sort of hands-on-- we were using PDP-8 minicomputers. And the ability to write, like, interrupt handlers and see the thing responding to keyboard prompts and running-- that whole layer of the computer is running so fast that it can cycle through code and you hardly even notice it's happening. All that kind of stuff, I just really liked it. So again, that sort of feeling of both hands-on and some theoretical basis to it. I really liked that.

Fairbairn: So when you were-- as an undergraduate, did you have a clear view as to what you were going to do afterwards in terms of graduate school, or go to work?

Bryant: Yeah, I knew I was going to go to graduate school. That was pretty clear to me. And partly-- my mother got her PhD when I was in high school, so--

[LAUGHTER]

So there I was-- PhDs, both parents. Grandfather's a professor. It's not a very big stretch to think I should go to graduate school.

Fairbairn: Right, exactly.

Bryant: And so, somehow, let's see--

Fairbairn: So I don't want to jump ahead, but I just wanted to get an idea of what you thought you were preparing yourself for at that point.

Bryant: I don't know if I remember, but I did have an interesting job between my sophomore and junior year in college. So this would have been '72. I went to Munich, Germany and worked for Siemens in Munich-- basically, writing Fortran programs. And also programming a PDP-8 with a graphics display and all that stuff. And I had exclusive use of it. I got that little key that you got to turn on in the morning, and I could program--

Fairbairn: It was yours.

Bryant: It was mine, and I had a great time.

Fairbairn: How did you wind up with Siemens in Munich?

Bryant: Well, my father, kind of. I wanted to be in Germany. I'd been an exchange student in Germany in high school. And so I wanted a job in Europe, and Germany seemed obvious. And quite honestly, my father arranged that through his company's connections. My father started his own company in the early '60s-- a microwave electronics company-- at a time way before people were generally going off and starting their own companies. So he--

Fairbairn: Now that was a very interesting time. I've done some research in terms of what entrepreneurialism in the technical world at that point was. And it was pretty different than now.

Bryant: Yeah, so he borrowed the money from his friends and family, basically. Got bank loans.

Fairbairn: The venture capital world hadn't coalesced. There were people lending venture money, but it was not smart money. It was just people trying to find a way to leverage their dollars.

Bryant: Yeah. And this was in the Detroit area, which was heavily dominated by the car companies.

Fairbairn: Sure.

Bryant: And very much, therefore, a heavy-duty union-- white collars here, blue collars there, don't mix, all that kind of stuff. And it was in the era-- basically, the Apollo era. So he was mostly cranking out microwave electronics to be used in aerospace applications.

Fairbairn: Building stuff out of solid state, or is it all--?

Bryant: Well, his stuff-- he was more of the plumber. So building the actual cables and connectors all out of gold fixtures and stuff. I actually had worked there two summers doing assembly stuff-- just mindlessly tedious, awful machine shop kind of stuff.

Fairbairn: Made sure that you had no vision of doing that in the future.

Bryant: Yeah, and I actually became-- I had to join the United Auto Workers, was the representative of them. So that's a little bit of a reverse.

Fairbairn: Interesting.

Bryant: So through connections through him, quite honestly, I got this job at Siemens in Germany. And I had a great time. I enjoyed the work, and I enjoyed being in Munich. It was the same summer as the Olympics, although I left before the Olympics began. And I was like a little whiz kid. I was 19 years old at the time, which in Germany, if you're 19, you're still finishing up high school, technically. And I looked maybe 15, so--

[LAUGHTER]

Fairbairn: And there weren't that many people programming computers at the time.

Bryant: No, I mean that was it an era when computers were still the sort of distant things you looked at behind windows. And this was a place I actually had the hands on of this PDP-8. It was pretty cool.

Fairbairn: Interesting.

Bryant: So sort of with that, and classes and summer jobs and stuff, it was pretty clear to me this is what I wanted to be working on. Computers. And somehow, the more hands-on parts of it I enjoyed as well. I took a really good class at Michigan using some logic design modules that Gordon Bell had designed specifically to help for teaching digital design, where they were kind of in these larger chunks that you could assemble together. I remember, in fact--

Fairbairn: Flip Chip modules?

Bryant: No, I think actually--

Fairbairn: Pre--

Bryant: I've got a manual for it back in my office. I've kept it.

Fairbairn: It's OK.

Bryant: I don't remember much about them. I think they might've used some of the asynchronous logic coming out of Wash U at the time. So I finished, because I'd gotten enough advanced placement and stuff that I could have gotten through college in three years. But I didn't want to get drafted, and so I kind of-- so it took me three a half years to get through college.

But my last semester there-- so that would have been the fall of '73-- I took an advanced digital design class where we got to program an [Intel] 8008 microprocessor shortly after it'd come out. And we could interface it with hardware. And I remember trying to develop, as a class project, a sort of a rasterizer-- graphics rasterized. Which didn't get it working. It was way too hard a project. But that whole business is kind of software, hardware combination, and thinking about, well you could put this in software. You could put this in hardware. I really like that.

And I applied-- actually, I returned to Germany for the winter after I finished. But I'd submitted all my graduate school applications before then.

Fairbairn: So the winter of-- what year is this, now? Winter of '74. 40 years ago.

Bryant: In the summer of '73, I'd gone on a road trip and visited Berkeley, Stanford, MIT. And I remember thinking Berkeley was sort of like a big Ann Arbor. It was a bunch of hippies-- the whole kind of hippie culture. And it seemed like a pretty good place, but somehow I thought, hmm, I don't know. Stanford, I really liked. I remember I hitchhiked from Berkeley to Stanford and met up with somebody who worked at the Stanford AI lab and lived in a commune. I think it was on the Stanford campus, and so I spent the night at this commune.

Fairbairn: Do you remember who that was?

Bryant: No, I've no idea. But I should have, because he talked about this club they had building home computers-- the Homebrew Computer Club--

Fairbairn: Oh, the Homebrew Computer Club.

Bryant: So he and these other people were talking about this, and I thought, well, why would you want to do that?

[LAUGHTER]

But I visited Stanford at the time computer science was part of Arts and Science. It was a graduate-only program. And I remember visiting there. And then the AI lab was up in an old--

Fairbairn: Oh, I know where it is. I worked there. I went to Stanford, and I worked there for part-time time job during the school year and during the summer.

Bryant: Yeah, so I went and hung out there some, with some pretty weird people.

Fairbairn: Yeah, it was a--

Bryant: I remember.

Fairbairn: Quite a good collection, though.

Bryant: Quite a collection of oddballs, which I really was pretty intrigued by.

Fairbairn: Yeah, I was working there, and Alan Kay was working there. And so Alan Kay went to Xerox and had recommended that I apply for a job at PARC, and so that's how I ended up with PARC, through--

Bryant: OK.

Fairbairn: AI lab connections.

Bryant: Yeah. So then I was off in Germany, and I got into all to graduate school I applied-- so I applied to Michigan and Illinois. Those were backup schools. And then Stanford and MIT. And I got into everything. So it really was Stanford vs. MIT And I remember my father going, so you're going to MIT, right?

[LAUGHTER]

Fairbairn: Of course.

Bryant: I don't know why he felt that way, and I thought, well-- to me, it was a little of this, and a little of that. OK, MIT. But I do remember that feeling like, well, I'd be-- I could go to Stanford. He had done a lot of business in the Bay Area as an electronics guy. In fact, he'd taken me to the Bay Area, and we'd stayed at Ricky's, which, at the time, was sort of the end of the developed world there. And south of Ricky's was all farm fields.

So anyways, it wasn't like he was anti-Stanford. But he had a bias toward MIT. And so that's how I ended up at MIT. Arrived there in fall of '74, and it was a little scary for me, because my background was not really computer science. I'd done some computer science courses, and some EE courses. And most of my programming had been in Fortran or Assembly Code.

Fairbairn: So you went in to get a degree in computer science? Was that the department?

Bryant: Yeah I went into that what MIT was all the EE department. But I was part of the CS part of that. It's all numbered-- course six, area two--

[LAUGHTER]

All MIT stuff. So I got in there. I was in computer science. And also, I felt there was a definite bias at the time at MIT of the people who'd been undergrads at MIT were sort of the people in charge. And even a lot of the faculty were MIT lifers. They'd been there undergraduates, graduates, and then-- so it had this culture that was, in some ways, pretty inbred and pretty exclusionary. And so I came in feeling weak in terms of my preparation and stuff like that, but also really excited. And really excited to sort of, being a nerd, to not have to do any more humanities courses.

[LAUGHTER]

Like focus totally. And obviously, going to MIT was a pretty exciting prospect as well.

Fairbairn: Yeah, yeah, for sure.

Bryant: So I jumped right in, and I took a few undergraduate courses as prep, but I did just fine in courses. But I do remember at MIT-- so I linked up with a fellow there, a faculty member named Jack Dennis, who I was attracted to because he did this balance between more theoretical things, and he was thinking about hardware. At the time, he was trying to devise a new class of computers he called data flow computers that were radically different from anything anyone was thinking of at the time. It turned out not to be a very good idea, but it was pretty exciting to think about being part of a change of reconceptualizing the whole idea of a computer.

But I did find it hard, first of all, coming up to a level of competence. And then I found the transition from being a course-taker to a researcher to be difficult. And I've since learned that most students do it because their adviser says, here, let's work on this project. You help me out, and we'll get you on your way. And that didn't happen for me.

Fairbairn: You say most professors now sort of help the students in that transition?

Bryant: Yeah they don't just expect them to come up with the research and deal--

Fairbairn: Blosssom into a research topic, right?

Bryant: And part of it, I was on an NSF graduate fellowship. So I was sort of a little bit of an independent operator.

Fairbairn: I see. You didn't have to get somebody else's coattails to get funding.

Bryant: Yeah, which was both a good thing and a not-so-good thing, in terms of having somebody for whom I felt beholden. And I'd get going and work on some things. So I had a little bit of a-- and this will be relevant as we talk later-- a sort of difficult time finding my way and feeling like I'd really engaged.

Fairbairn: How far into your time at MIT did you come up against this?

Bryant: Well, first of all with MIT at the time, you had to get a master's degree, which included a thesis, before you could get a PhD So first was a master's, and then between masters and PhD, I had the same problem. So I'll tell you first the master's.

Fairbairn: Yeah, please.

Bryant: Fortunately, one of Jack Dennis's more senior PhD students came up to me and said, we've been thinking about the following research problem and thought you might be interested. And he said, we're basically building these data flow systems you can think of as a very large distributed asynchronous computer system. And they were building some hardware-- microprocessor based-- to create sort of an emulation facility.

And he said, but we actually have no idea how to model these distributed systems on this hardware. Do you want to think about-- it might be a good topic for you. So I went off and started thinking about it and actually came up with some ideas that were pretty successful. So my master's thesis was basically how to simulate one distributed system on another.

Fairbairn: Sounds like a pretty aggressive topic for a master's thesis.

Bryant: Yeah, it was. In fact, the thesis is still cited. It's like in my Google Scholar account, it's like my tenth most cited work.

Fairbairn: Oh, wow.

Bryant: And it turns out-- I mean, I had this idea that anyone who thought about it very long would have figured out roughly the same thing. It's just that I was thinking about it before anyone else was thinking about it. And so at the time, my adviser didn't even suggest, and it didn't even cross my mind, to try and publish a paper about it. Probably at MIT, which I sort of over-exaggerate, there's such a attitude there that since MIT had really smart people there, in fact, all the smart people are at MIT.

[LAUGHTER]

So you really don't have to publish papers outside of MIT, because look it. All the people who really matter are right here now, so you--

Fairbairn: Everything is to know is being developed here.

Bryant: Everything you need to know is being developed here, and so somehow I'd drunk that Kool-Aid and applied it. And sort of not getting much direction to say, oh this is such an interesting idea. You should take it out. So lo and behold, so I turned in my master's thesis-- I think in '76, spring of '76 or so. And then a little bit later, my adviser was down at University of Texas talking to Mani Chandy and Jay Misra, who were all excited to be telling him this new idea they had for doing simulation. And he realized they'd basically rediscovered what I did in my master's thesis.

And fortunately, I had a tech report for my master's thesis. It was actually a physically existing thing, and so Jack pointed these two guys to this thesis, and they were pretty bowled over. And quite kindly, they had a paper that was about to be published, and they put a footnote. They got the publisher to add a little footnote to the paper that said, we just found out that very similar ideas had been developed by blah blah blah, and it gave a citation to the tech report. And so sometimes, this is called the Chandy-Misra-Bryant simulation method.

And like I said, they could have said, psh, it's a tech report. What's this? But they didn't, and they're very generous about it. Because I didn't feel like I'd really done much to--

Fairbairn: Did you ever have future collaboration with them or any of that?

Bryant: No, and I never worked on this area again.

Fairbairn: Hm.

Bryant: You know, that it came time to get a PhD. Topic. And I could have taken the master's and turned into PhD., but I couldn't really see where that would take me. And this is just an idea. It was purely a theoretical, conceptual idea. I didn't implement anything. In fact, there's a weird thing there in my group, at least, at MIT. I think it was sort of an Edsger Dijkstra influence. It said, well, you know, real computer scientists don't write programs.

Fairbairn: [CHUCKLES]

Bryant: We think about and reason about these things, but programming is no, no.

Fairbairn: Develop algorithms or mathematical whatever.

Bryant: Yeah, it's all kind of this conceptual stuff, which is really weird. So I didn't write any programs at all in graduate school until I got further along and interested in design automation, and therefore had to write some programs.

Fairbairn: So I'm curious for other reasons as to, do you think that philosophy-- does that philosophy continue at MIT? Is that--?

Bryant: Hm-mm. Totally gone. And in fact, I think everyone, everywhere-- ah, no, most people everywhere-- really understand computers are this amazing tool that you can do all kinds of stuff. So even theoretical computer scientists write code, just-- there's no point in trying to prove a theorem until you've run enough cases through it to give you pretty good evidence if it's really true. So I think that flipped tremendously.

And that was just a kind of weird place. And I don't know how widespread-- I don't think it was. Like the AI lab at MIT, they're all hacking code like crazy. But my little niche of it was more the sort of yeah, we just think about it.

Fairbairn: OK. So you went off to do a PhD So what's the--

Bryant: So I went off to [work on my] PhD, and I was sort of running in circles and getting nowhere. Because I wanted-- this is part of my own ego problem-- I wanted it to be a big splash. Because I'd sort of had success as a masters student. By the way, at least the master’s thesis kind of gave me a feeling like OK, I belonged with this crowd now.

So I was grasping at different topics here and there, and trying to figure it out within the context of the data flow computing stuff my adviser was doing, and just not getting anywhere. And then, that's when I heard that there'd be a course taught by this woman visiting from Xerox on VLSI design. And so several of my office mates and I signed up for the course. I believe that was fall of '78. And I loved that stuff from the beginning. It just was perfect for me. We were using a draft version of the book, *Introduction to VLSI Systems.*

Fairbairn: Yeah, no, I remember. I did help put that together.

Bryant: Yeah. Which is a pretty ragged book, by the way. You know, just to-- the first chapter by Carver Mead, I didn't have a clue what he was talking about in that one. It was a book that was put out, and it was the right book for the right time. But it was a little bit of an uneven book.

Fairbairn: [CHUCKLES] That's OK. So who were the other office mates that joined you in--?

Bryant: OK, so they're all-- actually, there was quite a crowd. It was an interesting event, because it brought in-- at MIT at the time, the CS was in one building and the EE part was in another. And they were literally across railroad tracks from each other. And so I'd had very little contact in EE, more the EE side of things.

And this course brought together an interesting collection of people who were coming at it from very different perspectives. It was really pretty fun. And when, as you've probably seen, she continues to-- she has a whole website about this course and the students in it. All her projects, and she's carefully documented everything about that course. It's kind of funny.

Fairbairn: Oh yeah, she's totally anal about--

Bryant: Yeah. But that turned out to be very valuable thing that she did in terms of documenting all of her notes, because it became the mechanism by which it could be propagated.

Bryant: I think she knew she was creating a legacy at the time. And so what about the course? So yeah, my office mates, including a fellow named Dean Brock, who's now in Asheville, a guy named Clement

[Leung, who's somewhere in the Bay Area-- I've lost track of him. And various other students, that it was fun because we all kind of-- oh, and a guy named Andy Boughton, who's still in the Boston area.

Fairbairn: So I'm just looking at the Lambda Magazine, fourth quarter, where your article appeared on MOSSIM. And the other MIT contributors were Ron Rivest, Clark Baker, Chris Terman.

Bryant: Yeah, I'll talk about them in a minute.

Fairbairn: And yourself. OK. I wanted to see if it triggered any thoughts.

Bryant: Yeah. So I'm not quite to that point yet.

Fairbairn: Good. OK, go ahead.

Bryant: So, the course was sort of-- Lynn had taken everything and condensed it down, as I said, to red crosses green, and you've got a transistor. And all the design rules and all that stuff-- which, in a way, was an amazing experience. And the way I often say, it was all one of the greatest lies in history-- the whole Mead and Conway stuff. The lie being anyone who's not colorblind can do VLSI design, right?

Fairbairn: Mhm.

Bryant: You just need to know these rules, and you can do it. The lie was you don't really need to know all that electronic stuff, and it's pretty straightforward. And so it sort of got a lot of us into thinking we really knew what we were talking about when we didn't. And then naïveté actually helped me, and I'll talk a little about how MOSSIM could come to an existence only because I was sufficiently naive not to know what was really going on.

But Lynn had really turned it into this pretty nice curriculum that really only lasted about the first half of the term. And then the rest of the term, the lectures were just her friends from Xerox who came and gave talks. And then we are all working on our own projects, and so the four of us from my advisor's group worked together on a project, which I don't remember much about except that it was incredibly frustrating. Because we each were doing some part of this circuit, and we'd find ourselves in arguments-- I thought you were putting the pull-up transistor on that line! No, it's your job!

Fairbairn: [LAUGHTER]

Bryant: Oh crap, we better put one in here somewhere. So it's like, we're just completely-- it was basically just manual circuit design and checking by hand---is this a reasonable design? Design rule checking? We didn't have any tools at all. And so I remember in that course when Dick Lyon came in and gave a guest lecturer-- who was at Xerox, as you know, at the time. And he made this offhand comment. He said, “We don't even really have tools to simulate these things.” And I was thinking, you know, he's right. And that's what's driving me crazy, for our group-- if we had a simulator, some way to actually run the circuit through something, we could figure out that the pull-up transistor was missing.

[LAUGHTER]

And so that little sentence-- I think it was one phrase in this lecture that I don't even remember what the rest of the lecture was about-- but it got me thinking. And so after the term was over, I remember that winter break, I started thinking about, gee, I wonder if we could figure out a way to run a simulator. And I started to think about switch level simulation-- the idea of modeling a transistor as just a simple switch, on or off. It's perfectly bi-directional. So at the time, the only people thinking about logic simulations somehow had to get it into a logic gate form, which meant you were propagating signals through gates-- and not the idea of it just representing the transistors themselves.

[CELL PHONE RINGS]

Let me turn this off. But one of the things that I think was implicit in the way that Mead and Conway was taught, was it was all stick diagrams and all that stuff-- that it was pure transistor design. And I think Carver had probably absorbed some of the circuit design lore at the various companies he'd been part of-- Intel, I presume. To sort of pick up the idea that hey, a lot of this stuff, you really want to be down at the transistor level, rather than the normal thing you'd see in a textbook was that you'd start by building up logic gates. And then you’d be up there doing gate level design.

So he hadn't really codified that part in the presentation of the book, but I sort of noticed that. And I was very intrigued just by that stuff. The other thing I'll say is that I was sufficiently naive that I didn't really know much more about it than that. I remember thinking, there's a rule that you'd get a one threshold voltage drop through a pass transistor. And I remember thinking at the time, it's too bad they get those threshold drops. [CHUCKLES] I wonder if that's a problem they could fix or something-- not realizing that they designed it that way on purpose.

[LAUGHTER]

Because I didn't really know what was going on. You know, I sort of--

Fairbairn: Those switches.

Bryant: Those switches, and too bad they're not better switches than that. But the whole idea that you could sort of mix ideas that really came out of relay circuit design with logic gate design and do all that stuff was kind of fun. And somehow I thought about the idea of switches. And once you start thinking about it, it's not that hard to come up with some sort of an algorithm for it.

And I started to fiddle around, but getting back to-- I hadn't actually written any computer programs in four or five years. And the only programming language I actually knew was Fortran, and nobody seemed to be doing Fortran. So I actually had to learn a local programming language being developed by one of the research groups around called CLU, and learn it enough to start writing code to even test out some of these ideas.

And that was this huge change for me-- to go from having this idea and trying it out, writing code. I found that I could make it work. I tested out the circuit from my class project, and it kind of simulated OK, and went through the various examples in the textbook and got them coded up, and simulating them. And so I realized this was going to work.

And I remember Lynn Conway came back and had a dinner at the MIT Faculty Club for all the people in her class. And in fact, I ate prime rib. And I remember talking to her after the dinner was over and said, oh, I kind of had this idea coming out of class of trying to model these transistors like switches and do a simulator. She said something like “that sounds really exciting. Please keep me filled in on anything you do.” And that was enough of an encouragement to kind of keep me rolling. So that was the spring of '79. And I managed to ingratiate myself with Jon Allen, who was going to then take over in the fall of '79 and teach--

Fairbairn: Teach the course.

Bryant: The course. And I'd TAed for Jon before, so I knew him already and had some credibility with him. But I told him partly, I've got this new tool, and I'd like to try it out on the students, because I think it would really help in their class to have this. And he encouraged me, and so basically I started hacking all these design tools for this course that was taught in the fall of '79.

Fairbairn: So MOSSIM was the switch level simulator.

Bryant: MOSSIM was the switch level simulator that I came up with and wrote. So MOSSIM I, which is described in that article you've got there, was the simulator I devised after taking a course from Lynn and getting it ready for this course in the fall of '79.

Fairbairn: Next fall, yeah.

Bryant: And so that was my beginning of EDA. And it was really interesting, because then in the fall, there was a new collection of students. And we-- Jon and I-- made up assignments using MOSSIM, and the students started using it. And I started getting useful feedback from them. And it was my first experience of having created a tool that other people were using, and realized I had to work a lot harder on documentation, making sure the tool had some consistent, high level abstraction that you could understand what it was going to do without necessarily having to run the program.

And it really started pushing me-- in fact, realizing that I had to get ready for circuit-- this thing had to be able to run on circuits that I wouldn't have thought of myself and had never seen before. And that whole attitude of therefore, you have to have an algorithm behind it-- you know, a model and an algorithm. That it has to implement that algorithm, and all those kind of things that, of course, are pretty standard ideas really jelled in my mind at that point. It was really a lot of fun.

So two of the people in the course were Chris Terman and Clark Baker, who, then, they already were in a group that was doing a lot more computer programming than I was. And they could program circles around me. And I got a little bit frustrated at first, because basically, Chris read the manual I'd written for MOSSIM-- you know, it's a five-page manual-- and basically wrote his own version. I go, “how'd he do that so fast?”

[LAUGHTER]

Bryant: And then Clark came up with this design rule checker based on a sort of pixel level grid. Because of the Lambda design rules, you could assume he you had a pretty coarse raster. And so he just came up with this idea instead of a line-based DRC, to do these pixel-based or raster-based DRC. And he, again, was this smart kid that could program circles around everybody. And so he and Chris came up with these tools that became the basis for that article they wrote there.

And they got-- if you remember, that summer, some MIT guys went out to Xerox PARC to try and create a Lisp machine on a chip. It started with a fellow named Guy Steele--

Fairbairn: Guy Steele, I remember that. Yeah.

Bryant: Had been in the course with me-- Lynn's course-- and had designed the beginning of it. And the guy was an incredibly smart guy, and he'd written a bunch of his own tools. As I recall, though, he'd forgotten to put any metal over his contact cuts. Or there was something structurally, fundamentally wrong with his chip that was so far off the mark that it never-- it would not possibly work.

Fairbairn: Couldn't even do anything, yeah.

Bryant: Because at the time, DRC didn't exist. And so one of the tools Clark came up with was a circuit extractor, to go from the layout back to the circuit-- doing it on this pixel-based idea. And again, you think about it a bit, you can figure out an algorithm to do it. But it was the first-- as far as I know, he was the first person to ever do that, period. Right?

Fairbairn: Yeah, could very well be.

Bryant: And I remember-- because the VLSI book had this sort of correct by construction business that you shouldn't need to check back, because your designs are correct by construction. Right? There is sort of that attitude, again, that comes from Dijkstra, but it permeated at Caltech, because Dijkstra used to visit there. That said we're such smart people we don't need to check our work, because of course it's correct. We prove a theorem, and therefore, there it is.

So that idea [using a circuit extractor] was in some ways, I think, a pretty radical idea at the time among a certain set of people-- that you should have to go from the layout itself back to a circuit. But he kind of closed that loop. I'm jumping ahead. That was the fall of '79. In the summer of '79, Gerry Sussman and a various team had gone out to Xerox to try and design this chip.

Fairbairn: Mhm, Lisp machine.

Bryant: A Lisp machine chip using the tools that you know of well, the layout editor and stuff.

Fairbairn: Yeah, ICARUS.

Bryant: ICARUS. On the Alto. And they'd basically-- this was before Red Bull, but they'd just gone through the whole summer eating junk food and drinking coffee and stuff like that, and came back with this almost-complete design. And they ran it through these tools that Chris and Clark had developed and found there was a power ground short. [LAUGHS]

With ICARUS, there became a term people called “mouse droppings”, meaning that you could drop a rectangle on something by just clicking on a button.

Fairbairn: Clicking the mouse, sure.

Bryant: And so there, when they ran the circuit extractor and it said, you got a problem, buddy. And they printed out this giant wall-size picture of it and went through it looking for it. And sure enough, there was this big, fat rectangle that was connecting-- they're all in the same metal layer. And there it was connecting power and ground, so.

That really, I think, was the moment of revelation for a lot of people-- both collectively, MIT, Xerox, I don't know who else. It says, oh yeah, OK. We really need to get our act together on tools. And this idea of circuit extraction. And then the beauty of that is then, with a switch level simulator, you close the loop back to the logical behavior.

You could simulate the thing in terms of zeros and ones, which was-- at the time, in theory, we could have run SPICE on the thing, but SPICE just couldn't handle more than a 20 transistor circuit at the time. So this was the idea that really closed that loop of now we can go through the layout and then come back and make sure that we've actually got a working design. I remember Dick Lyon and other people at Xerox been pretty excited about that idea.

Fairbairn: Yeah. So got through that. You TAed the course. What was the next major-- did you take that further? And did you want to say something more about the term you coined-- the “big lie”?

Bryant: I only really appreciated the big lie later in life, so maybe I should come to the big lie later in this interview.

Fairbairn: OK, that's fine.

Bryant: So anyways, let's see. I'm trying to remember. Somehow, it didn't really strike me that this MOSSIM could be a PhD topic, even though I was pretty excited by it. And I remember there was a meeting. Again, there was a whole, as you know, a community building around VLSI that-- not just at Xerox and MIT, but obviously, at other places.

And there was a DARPA program that got started to fund research at universities in VLSI. And that, also, was building up all the time. And of course, I was just a graduate student, so I didn't have the very top level view of all the stuff going on. But I remember there was a meeting that Lynn organized here in Xerox, Coyote Hill Road, that was bringing together various university people working in this.

And I came to that meeting, and I remember just getting in a session where I was talking to Forest Baskett and various other people. And I described the algorithms behind MOSSIM, and just having this amazing exchange of-- I felt like I had these ideas, and they were listening to me and asking questions and pushing me and questioning me. And that was pretty cool.

And I also visited IBM. One of my office mates, a guy named Glen Miranker, had gotten a job at Yorktown heights. He was actually in the same course we were, but he was a little ahead of us in getting through his PhD So he'd gotten a job at Yorktown Heights. His father was also at Yorktown Heights, a guy named Will Miranker.

Fairbairn: I'm sorry, his father was what?

Bryant: His father was also at Yorktown Heights.

Fairbairn: Oh, OK.

Bryant: So Glen invited me to come give a talk at Yorktown Heights. I'm having trouble remembering exactly what sequence things occur. And it was sort of a job talk, except I said I'm really looking for academic jobs. But it was a pretty good crowd that collected there, and I gave some kind of talk. I don't-- probably not very good. Well, it was about the same time I was on the job circuit giving talks at other places, too.

But I'm getting into this because after my talk, I met with Gary Hachtel and Alberto Sangiovanni-Vincentelli. So Gary, at the time, was working at IBM. And Alberto was spending the term or the year there on sabbatical. And Gary and Alberto kind of pushed me. They said, you know, this is a pretty interesting idea. But everything you described is completely ad hoc. You know, what are you really trying to do there?

I said, well, there's no real model for this kind of thing, because it's not really circuits. It's all just ones and zeroes. They go, well, you could have a model with ones and zeros. You know, said to me something like that. Anyways, they kind of challenged me to come up with something that was more than, well, the program does what it does. And here's the algorithm that does it. Right? And have something that you could actually--

And so I went back, and I don't know the exact combination, but somehow, that became my PhD thesis-- was to come up with a formal model of switch level circuits and an algorithm that would arguably implement this model and kind of put it all together. And that became my PhD thesis. And in the course of it, I much generalized and improved the actually very ad hoc and inefficient algorithm I'd used in the first MOSSIM. But I didn't actually implement it. I just came up with this model in theory. But that was my PhD topic.

And meanwhile, I'd attracted the attention of Chuck Seitz at Caltech. And he recruited me heavily to come to Caltech. So I went on the interview circuit, and I got offers at some place--

Fairbairn: This is once you completed your PhD?

Bryant: No, I went on the interview circuit too soon. I was too optimistic about when I'd be getting out. So I went on the interview circuit in the spring of 1980, thinking that I'd be through by the end of 1980, which was completely unrealistic. And so I went out, and at the time, the whole VLSI design stuff existed at a very small handful of the elite schools-- MIT, Caltech, Berkeley--

Fairbairn: Stanford.

Bryant: Stanford. Maybe University of Utah. You know, just a very small collection of places had gotten-- so mostly I was giving talks to people who'd never thought that it was possible for normal people to design chips. And so I just showed them pictures of chips, and everyone got wowed and stuff. But I wasn't really getting offers from the top schools, except for Chuck really worked very hard to recruit me to Caltech. And I didn't really want to come to Pasadena, but I was, of course, very excited by Caltech and all that meant. And here it was, like the center the universe in terms of a lot of these ideas.

So it turned out I took longer than I expected, and I didn't get my Ph.D. done until the spring of '81. And I showed up at Caltech for the fall of '81. And that was the beginning of my time there. So it is pretty exciting in many ways, and a little bit disappointing in others, in that Caltech was an extremely small school. And I think there were seven people in computer science at the time. And Carver and Chuck where there, but both of them had various issues going on with their lives and were not as attentive as I'd thought.

But on the other hand, it was a good place to be. I made contacts. There was this industry-sponsored consortium called the Silicon Structures Project that had visitors from companies-- Digital, Intel, Fairchild, all these companies-- they would come through for a year. And I met a lot of people through that that became industry contacts that have continued for me.

And in fact, I got there, and I'd done this thesis of creating a new model, but not implemented. And so my first task I really focused on was OK, now let's implement it. And that became MOSSIM II. And I had one of these SSP member companies-- it was Burroughs. And so the visitor from Burroughs was a fellow named Mike Shuster, who read my thesis, got really excited by it, and decided to help me implement it. And so we and one of Carver's students kind of dug in and implemented MOSSIM II.

And we implemented it in a programming language called MAINSAIL.

Fairbairn: Oh, yeah.

Bryant: Which was a commercial derivative of a language called SAIL from the Stanford AI lab.

Fairbairn: Yeah, VLSI Technology--

Bryant: VLSI Technology did it. Somehow, within this SSP, there was a sort of corporate decision to use MAINSAIL as a basis. At the time, C was a little too low-level. And also, we were still running on DEC-20 machines—PDP-10 based machines. And there wasn't really a C implementation, so--

Fairbairn: And MAINSAIL was very portable.

Bryant: And MAINSAIL was very portable. That was its feature. And it had garbage collection and all this stuff. So yeah, so implemented MOSSIM II in MAINSAIL and got it working. And Mike-- maybe some of the other SSP member company people-- wrote a circuit extractor and basically a set of tools on it.

And the fact it was MAINSAIL became its greatest strength and its greatest weakness at the same time. So weakness meant there were all these people at universities who were dying for design tools, and the company doing MAINSAIL, which was called Xidak-- [CHUCKLES]

Fairbairn: Hadn't thought of that name for a while.

Bryant: Yeah, X-I-D-A-K, remember that?

Fairbairn: No, I remember it. Yeah, I had a lot of business with Xidak.

Bryant: Yeah.

Fairbairn: Xidak.

Bryant: The world's a small place, isn't it? So Xidak offered free licenses to universities. But, in fact, Chris Terman and Clark Baker had bundled up their tools, which they'd written in C at a time when people weren't generally writing in C that much, and packaged it up and were distributing it widely to universities. I was in a little bit of a "frenemy" status with them, that Chris seemed to be getting this credit, because his tools were out there and being used. And I thought I was working this more abstract, better algorithms and stuff, but I was writing it in MAINSAIL.

But the biggest plus on it was Intel also was using MAINSAIL for all their tool suite. And so for them, the fact it was written in MAINSAIL was a big plus. And they basically made a corporate decision to use MOSSIM II as their mainline simulator-- logic simulator. Before that, they'd been using some hacked-up gate level simulator that was trying to fake how you'd model pass transistors. But they knew that wasn't working, and they very quickly picked up MOSSIM II and put several people on the software engineering-- taking that code and making it ready for corporate, and building their whole design methodology of how they'd-- their whole tool flow, essentially.

Fairbairn: Around MOSSIM II?

Bryant: MOSSIM II, running it, yeah. And that was their main simulator for microprocessors. So that would have started in about '83, and it continued up through the 486, I believe. So I think the 386 and the 286, at least, were run using MOSSIM II as their main simulator. And so that got me on the road.

Fairbairn: OK, so you were a Caltech. You completed MOSSIM II in what time frame?

Bryant: '83.

Fairbairn: '83. So what picks up then? What's the next thing on your--?

Bryant: Well, a couple interesting things happened. I'm still at Caltech, so because of this Intel connection, I really got-- there's a guy named Rob Willoner at Intel who left sometime in the '80s. But he was my main connection to Intel. And he kept saying, well, we'd really like you to do this, and we'd really like you to do that, and stuff like that.

And they were very good customers, let's say. And they were SSP members, so they were actually providing funds to Caltech. And since half my research salary was being covered out of SSP and the other half out of this DARPA program-- so I felt pretty willing to work with Intel. And Rob got me into a fault simulation. It was something they needed.

So fault simulation is-- you imagine that you have some defects in your chip. And you want to be able to devise a set of test patterns that you'd run through your chips to detect whether there are faults. And fault simulation is just grading those vectors, running tests through and seeing what particular faults would it pick up. And so there's an idea out there called concurrent fault simulation that had been developed by a fellow named Ernst Ulrich at DEC---Digital Equipment Corporation---that was a clever way of-- you could basically simulate the good, the nominal circuit, and all the variations created by as many faults as you wanted all at once. And that's why it was concurrent. You're running one simulation, and doing it.

And so this fellow Mike Shuster, who became my graduate student-- he left Burroughs and became a Caltech graduate student-- and I developed a fault simulator for MOSSIM we called F MOSSIM. And that was a pretty frustratingly difficult-- it was some of the hardest code I think I'd ever worked on, still to this day. But it was an interesting experience. We did actually get a paper out of it.

Fairbairn: So at some point you left Caltech.

Bryant: Yeah, that's coming in, but I'm still not ready to leave Caltech yet.

Fairbairn: OK.

Bryant: But actually, there was again, one of these little asides. Somebody, somewhere-- I don't even remember who, in this case-- said, well, thanks for this simulator. It's really good. But you know, we actually have no clue what to simulate.

[LAUGHTER]

Fairbairn: You mean what sort of vectors?

Bryant: Yeah, what kind of vectors. You know, how do we actually know the circuit's correct if we can run this simulation until the cows come home. But there still could be some lurking bugs. And that got me thinking about formal verification. And similarly-- oh, I remember. This is a funny story.

In the winter-- in November, December of '83, I got invited to this workshop in Germany that was held in this conference center in the middle of the Black Forest that was called Oberwolfach, where they ran sort of mathematical symposia where they'd have people from around the world come. And you'd hang out in this beautiful, middle-of-the-woods kind of place, and just have extended talks and discussions for several days about a topic.

And so I got there and said, oh, man. I need to sort of up my game here-- sound more mathematical than I really am. So I presented this switch level model in the most abstract way I possibly could, and then I got invited to a German university after that-- you know, as part of the trip, to a university called Saarbrucken. And they asked me to give a colloquium.

And I gave a talk about F MOSSIM. And I thought, oh, man. This code-- I mean, it was really grungy and grimy code. If you really got into it, you could get excited by it. But it was really hard to get-- it was one of these things. I've got to do something in this talk that's going to at least give a glimpse of this as a future possibility.

And I came up with an idea there-- excuse me, I've got to watch my crossing hands. That somehow I could take this idea of concurrent fault simulation and then use it-- rather than simulating a bunch of faulty circuits, use it to simulate a bunch of different inputs to a single circuit at the same time. Using-- I had this vague idea of a tree representation of all the possible values.

So it was sort of a hybrid of this idea of fault simulation, but morphing over toward a symbolic simulation-- wanting to run symbolic values through a circuit and seeing how it would behave over many collections of possible actual values. And this idea of a tree quickly morphed into a graph, and basically became the basis of thinking about binary decision diagrams [BDDs].

I'd already heard about this idea of BDDs. And there's this paper written by a guy named Sheldon Akers, where he'd posited these as an interesting way of representing Boolean functions-- a graph representation-- without any real algorithms behind it. But somehow, that was in a different part of my brain. But as I was working out these ideas from fault simulation concurrent, they kind of all merged together.

And that's how this idea of using a BDD as a representation of these symbolic values, and running a simulation where you're pushing BDDs through a circuit. And the core algorithm is, then, given the BDDs representing the possible values-- two inputs-- I was thinking gate level terms at the time. Logic gate-- what, then, is the BDD representation of its output?

And that, through some weird collection of thinking about it, got me going in that direction. I was still at Caltech at the time. But for various personal reasons, I decided I didn't want to stay at Caltech. I wasn't really enjoying Pasadena. Caltech had a lot of positives, but I felt it was too small at the time. I felt like I needed to be in a place where there was a wider range of people and things.

And so I went on the job market. And I interviewed at several places, but Carnegie Mellon became the place I chose to go to next. And at the time, I knew a fellow named H.T. Kung who'd been-- you know, the VLSI design crowd at the time was a pretty small bunch, and so we all kind of knew each other. So I had a connection into CMU that way. And then when I was early on it CMU--

Fairbairn: So you went to CMU in what year?

Bryant: So I went to CMU in the fall of '84.

Fairbairn: OK. And had you written a definitive paper on BDDs, or--?

Bryant: No. I'd written a paper about MOSSIM. I wrote a [IEEE] Transactions on Computers paper about it with the whole theory and algorithms and stuff like that. At the time, to get a paper through a journal took multiple years of refereeing and pipelines and stuff, so-- and I wasn't-- one of the mistakes I hadn't really gotten good enough guidance on was the sort of hustle of publication. I was a little too laid back about it, quite honestly.

But I had a few papers. I had a couple of papers at the Design Automation Conference. So I had this Transactions paper. There's a series of conferences on VLSI design held both nationally and internationally. And I generally went to those and was giving papers on various different parts for that. So I was sort of in the crowd there, and people generally came to know the switch level stuff.

But the BDD stuff-- so actually, I arrived in Pittsburgh in the fall of '84, and kind of holed myself up in my office and started writing a paper about BDDs. And that's a paper that was ultimately published in '86, and that's my highest-cited paper. And for a long time, it was one of the highest-cited papers in all of computer science. And still, a lot of citations-- several hundred per year, easily-- citing it as the sort of foundational paper on BDDs.

Fairbairn: So can you sort of summarize the most critical breakthrough or insider value that the BDDs provide?

Bryant: Well, I guess what I'd say is I took it from what Sheldon Akers-- and he built on work by Lee at Bell Labs-- which was just a sort of abstract representation of Boolean functions. And I built it into being a data structure with algorithms associated with it, that you had an API-- you know, an abstract interface-- that you could just say, OK, let's define some variables. A, B, C. OK, I want to take the XOR of A and B, and end that with C. And now, tell me if that's equal to some other thing.

And so basically, just thinking of these without even having to know what the data structure is-- the ability to just assume you had some magic representation of functions, so you could build them up based on low-level logic primitives. And then you could do testing. You could say, are these the same? Is this output of this? One of the examples I did was take the bit level ALUs out of the TTL catalog. Were those 2901?

Fairbairn: 2901 is bit slice.

Bryant: Yeah, bit slice, where you can put them together and build a-- and then there was-- 2902 was the carry logic for it?

Fairbairn: Could be.

Bryant: Right? And so you could basically build up. And so I showed that you could sort of take the circuit design-- the gate level circuit in the catalog. And then, on the other hand, they had a table that says, if this function is this, it will add. And I showed that, indeed, the catalogue was correct. I built up to 64 bit ALU out of AMD parts, you know. Or I don't think it was AMD-- TTL parts, using the gate network.

Fairbairn: Well, AMD had the 2901, yeah.

Bryant: Yep. Using the gate level parts on one hand and the documentation on the other, and proved that they were correct. And so, but that's really the main idea of BDDs-- was its data structure. You can just use it, and it works well for certain classes of functions that are common enough to be useful.

And so that was-- the paper, the reason why it still gets cited is-- it's not like a least publishable units kind of thing. It was a paper that had everything-- the basic representation, the set of algorithms, proofs and properties, had some experimental results, showed the whole idea that you're very sensitive to the order in which you declare the variables. You can get either big BDDs or small ones, depending on how you do that. All that's in that paper, along with sort of pseudo-code to implement the algorithms and stuff.

Fairbairn: So one of the important outcomes of that is progress and the whole world of formal verification.

Bryant: Yeah.

Fairbairn: It is basically a building block. Is that it?

Bryant: Yeah. Well, now they get used-- that's jumping ahead a little, I'm sorry. This is going kind of slow, isn't it? [CHUCKLES] I didn't really think it would take this long to talk, but--

Fairbairn: [LAUGHS]

Bryant: So then--

Fairbairn: James, do you have any deadlines, or--?

James Fortier [Videographer]: [LAUGHS] Do I have deadlines?

Fairbairn: Yeah.

Fortier: No, no.

Fairbairn: OK, I just wanted to make sure we aren't running up against anything.

Fortier: No, like I mentioned.

Bryant: Well, I have some deadlines, but--

Fairbairn: No, keep going. Keep going.

Bryant: We'll move along.

Fairbairn: Just wanted to make sure. Yeah.

Bryant: Meanwhile, at Caltech, the student of mine-- Mike Shuster-- and I had tried to formulate essentially, a more symbolic way of thinking about switch level circuits-- so that instead of just being able to run a simulator, you could algorithmically take a circuit and crunch on it and come up with something that would look more like a logic or a gate level representation. Not really logic gates, but something that-- essentially, an executable model that you could just feed forward, take the state of the circuit, the input values, run it forward, and tell you what that circuit would do.

And we'd basically spun our wheels thinking about it, and came up with ideas. But at the core, we didn't really have an engine-- the sort of symbolic reasoning ability to make it work. But I actually had-- so this is all kind of converging here, because I also met up with a guy at IBM named John Cocke, who's the father of RISC machines.

Fairbairn: Father of RISC, yep.

Bryant: He was a character. I don't know. I hope that at some point, they were able to interview him before he died, because--

Fairbairn: Yeah, we do have some of him. In fact, we're trying to put together a team interview of the people that created the first RISC processor.

Bryant: OK.

Fairbairn: And some of them are still alive, and we're trying to work the logistics of doing that.

Bryant: John Cocke was a remarkable person-- brilliant beyond brilliant, and the goofiest guy you ever met.

Fairbairn: [CHUCKLES]

Bryant: You'd have a conversation, and he'd be talking about six things, and he'd sort of multiplex among them. And you just had to keep thinking, oh, now he's talking about error correcting codes. Oh, now he's talking about RISC machines. Now he's talking about compilers. He was an unbelievable person.

Anyways, I guess people at IBM had been trying to figure out-- well, he was the person behind a machine that they developed at IBM called the Yorktown Simulation Engine, which was an early version of a hardware simulation accelerator that worked-- it was an incredibly naive, straightforward, and beautiful design of running just-- the idea of it was just to propagate signals through a network of Boolean operators much faster than you could even imagine. And they actually built this thing.

And so John had been trying to-- I guess some people at IBM and John had known about this-- were trying to figure out, how could we map a transistor simulation onto this YSE? And so somehow, I got introduced to him. And he read my thesis. He read it in like four hours and came back and asked me more questions about the thesis than anyone else had. And he brought me in to talk to people-- so I spent like a week or two weeks at IBM in the summer, the summer of-- must have been '83.

And again, I was trying to take this idea of coming up from a circuit up to something that would look more like a gate network-- something that you could map onto the YSE and just propagate values through it and get it there. And that all kind of started merging together. So the BDDs were kind of one thing, and then this switch level stuff on another. And that all kind of merged together in fall of '84, when I was new at CMU.

And I realized I could put this all together, and came up with a program I called MOSSYM, M-O-S-S-Y-M, which I then presented at a conference. One of the VLSI conferences was in North Carolina the next spring. And I also presented a DARPA program meeting, which was where all of us VLSI people got together---that was the in-crowd. And if you could present something that would get them excited, that was a big thing.

And that was this huge breakthrough for me-- was sort of merging the idea of taking a transistor circuit, and being able to give it just pure Boolean symbols at the inputs, and have it come out and say, yep, that's an adder. Or yep, that's an ALU. It could actually run simulations through multiple clock cycles and say, yeah, you've implemented this logic and stuff. It was my first entree into formal verification, and it was a really exciting step forward. And I remember just that feeling like I'd really cracked a big nut in doing that.

So that's spring '85. Let's see. Let me try and move things along a little. [CHUCKLES]

Fairbairn: Well, I was wondering. There's a thread here in the formal verification. I don't know what the-- is that sort of the major thread here through the next--

Bryant: And that really became my major thrust. I realized, this is what I want to do.

Fairbairn: That's who you were. So I'd be curious as to what your view of it, then, was, in terms of how quickly it could advance or what the impact might be, or when the impact might be felt in sort of the real world designs. Did you have a feeling about that at that time? Or was it too--? And then how did your view, whatever it was, compare to how things actually developed? Because the world of formal verification has been something that has seemed to have more promise, or be--

Bryant: Yeah.

Fairbairn: Be closer to reality than turned out to be the case. And I'd like to culminate in terms of--

Bryant: Well, so I think I always knew that this-- so my version of this was-- first of all, it was simulation based, because I'd come up-- I wasn't trained in formal logic or any of that stuff, right? So I came at it from this more simulation view of it. And I was mostly trying to think of this both in terms of how would we actually use these tools, and are they going to work? Because the basic problem you get into with BDDs is, if you can get the right variable ordering, they can be super. But if you get it wrong, you're in terrible shape.

Fairbairn: In terms of the complexity?

Bryant: The complex-- they'd blow up. They'd have exponential blow up, and they'd just fail totally. And computer memory wasn't nearly what it is today, so it would become completely useless. So I don't remember all, but various things kind of move things along. So one was-- I wrote a paper about BDDs. Well, I wrote this journal article, but it didn't get published until '86.

But in the spring '85, I had DAC paper about the BDDs that I used this 2901 based ALU as my benchmark to show that I could do it. And that got picked up by a few people, including a fellow named Fujita at Fujitsu, who then implemented this and came up with his own sort of heuristic way of getting variable ordering automatically, based on just-- given a combinational circuit, trying to order the primary inputs in a way that you'd have a decent BDD. And he came up with some fairly straightforward heuristic for it.

And then he went and talked about it to Gary Hachtel, who at the time was then at University of Colorado. And I'd met them before, but we weren't like buddies or anything. And then he came to visit CMU, probably to visit somebody else. And he came to my office, and goes, um, I had this visitor from Japan who was telling me about this stuff you've been doing. And I'm pretty intrigued, because I've been trying to solve this problem for years. And he started really asking a lot of questions.

And then there was a fellow at Berkeley named Sharad Malik, who's now at Princeton-- very well known-- who also came up with his own-- at the time, the Berkeley-Colorado crowd were all kind of in cahoots with each other. And he came up with his own heuristic method for doing it. And they both published papers at ICCAD in '88. So I guess it took a-- this is a while afterwards. And that sort of broke open the whole thing about the BDDs in the EDA crowd. That's when they went from being a-- they sort of went mainstream, because that opened up. They showed they could take a lot of circuits that were out there at the time, a lot of benchmarks, and do BDD based stuff on them.

There's another parallel thread that my colleague at CMU-- a fellow named Ed Clarke-- had come up with this idea of model checking before he had arrived at CMU. He came to CMU in '82. And he had some students working for him who I started talking to-- one named David Dill, who's now a professor at Stanford. And even though Ed and I were not working directly with each other, we kind of knew each other's work a little bit.

And then he had a student come in named Ken McMillan, who is now at Microsoft. And within his first two weeks of being a Ph.D. student at CMU, he came up with the idea of symbolic model checking-- in other words, combining BDDs as a symbolic representation of, basically, sets and relations. And then casting model checking algorithms in terms of these computations over sets and relations that would let you do things like-- given a finite state system, but very non deterministic. You know, one where there's many possible behaviors. Can you ever reach a state where there's something gone wrong?

The classic example-- a traffic light controller. Can you ever reach a state where you've got green lights in orthogonal directions, right? Or for a cache memory system, can you ever get it where you access an invalid cache line-- a stale cache line, and you don't recognize it. Things like that. So Ken was an amazing guy-- still is an amazing guy.

And he was the one that really realized that this BDD stuff I'd been working on and this model checking his adviser, Ed, was working on could be married together. And that's where the things really took off. And model checking is still, today, considered one of the real key parts of formal verification. So that's sort of how I really-- at that point, I really realized yeah, this formal verification is stuff I want to do.

And I realized the advantage of working on hardware-- there'd been work, historically, on more software related stuff, based on proving theorems-- that with hardware, especially since I was part of this crowd that worked directly from transistor level design. To be able to go up to any level of abstraction and prove there was something useful coming out of that was a pretty big step forward.

And so I don't-- you mentioned, it's been a technology that's been a little bit tough sell from an EDA vendor perspective. But within companies like Intel and IBM, where they have a larger CAD group and it's more captive to the company and they're thinking things, actually some of these ideas got picked up pretty early. And so I spent more of my time sort of connected to these companies and their CAD groups, rather than to EDA tool vendors.

Fairbairn: So did you take on a role of advising, or help start an independent company in formal verification?

Bryant: No, hm-mm. Well, I mean, further down, yeah. By the way, I also, then, decided to do a redo of a switch level simulator I called COSMOS, which took stood for “COmpiled Simulator for MOS” circuits. So it could take this idea of a switch network and run it through this algorithmic way and derive a logic network that was functionally equivalent. And then you could run simulations of it that way, or you could run BDDs through it and do formal verification.

And so I created this whole suite of tools I had. This was my early days at CMU. I had four or five graduate students. We all did parts of it and created this whole major collection of tools-- which I really, at the time, wanted to get it out to universities as part of my credibility. And that was my first time writing in C. I decided I'd had enough of these niche languages-- time to go mainstream. So I went with C at the time.

And that tool also, then, got picked up by a variety of companies, including Intel, but also Digital Equipment Corporation, Hewlett Packard, and various others. And we licensed some of the technology in it to what was then Gateway Design Automation, which was acquired by Cadence.

Fairbairn: Cadence, yeah.

Bryant: And as you know, Gateway was the originators of Verilog. That's where Chris Moorby-- is that his name? Yeah.

Fairbairn: Phil Moorby.

Bryant: Phil Moorby was working. And they wanted to use these-- there was a switch level model built into Verilog, and they wanted to be able to do a better job running simulation and other things than they could. So we licensed parts of COSMOS to Cadence-- or to Gateway, that then got incorporated into the simulator. So I was pretty-- at that point, I was part of the EDA community, pretty solid. Very regular attendee at DAC, on program committees, editorials, journal editing, IEEE transaction type of stuff. So I was very solidly in the EDA community at that point.

Fairbairn: So this is late '80s by the time these things evolved?

Bryant: This is '90s.

Fairbairn: 90s.

Bryant: Well, yeah you're right. No, it's roughly late '80s, early '90s.

Fairbairn: So what was the next major evolution? Yeah, we'd like to speed it up a bit, but I don't want to miss--

Bryant: Yeah, [CHUCKLES] I know. So there I was. I was a successful professor. I had graduate students. We were getting papers in DAC. We were getting papers at ICCAD. One thing I'd learned at CMU was to sort of step up my--

Fairbairn: Paper publishing?

Bryant: Paper publishing. They had a sort of rule that you didn't go to conference unless you had a paper there, so that--

Fairbairn: Good motivator, huh?

Bryant: That motivated me. And being pretty successful. And that really kept me going for quite awhile. The COSMOS was the main simulator used at Intel for a long, long time. And I always was very interested in being able to map onto hardware accelerators-- there were fairly many varieties of those at the time. So I dabbled in that a little bit.

I was interested in testing-- the fault simulation had morphed over to test generation, and used COSMOS as a basis for doing some of that work. So that became a sort of center, a platform on which I could build quite a few projects and have quite a few graduate students working.

Fairbairn: So relative to my earlier question, whenever we talked about the whole start-up evolution of Silicon Valley and so forth, did it occur to you that hey, this is some stuff that maybe we ought to make a commercial product out of? And could we start a company? And was that a conversation you had?

Bryant: Not really. So part of it was being in Pittsburgh. Part of it, having watched how hard my father had worked and not really wanting to follow in those footsteps.

[LAUGHTER]

Fairbairn: It's a good life, being a professor, huh?

Bryant: I always said that sort of entrepreneurship-- I claimed that we'd sort of jumped generations--

Fairbairn: [LAUGHS]

Bryant: So that I was picking up my grandfather, which was really my mother's father. And my father was the entrepreneur. And I didn't. I said, I'm not an entrepreneur. Also, as you mentioned, it was clear that formal verification was extremely powerful in the hands of the right person, but completely useless for the ordinary folk. And if you looked at the EDA tools, especially in the early days, had to stay pretty generic in what they did and pretty predictable.

And so they were mostly simulators, and then, of course with Synopsys-- synthesis became a really big deal. But I could see that it would be really hard to get a formal verification tool that really connected. And the first commercially successful verification tools were equivalence checkers, right? That would compare-- basically, it was going from a Verilog to a transistor circuit and seeing if they were a faithful match to each other.

Because at the time, people didn't trust Synopsys. They'd run it through Synopsys, and Synopsys might introduce some bugs. And furthermore, they'd take the output of Synopsys, and they'd tweak it, because they thought they could improve the circuit. And so they just wanted to make sure that--

Fairbairn: Still had what they started with, right?

Bryant: Yeah.

Fairbairn: Well, what about a sort of different tack here? In terms of your career at Carnegie Mellon, you were a professor, but then rose in the management ranks.

Bryant: Yes, but I was still-- all this, I was pretty much your standard professor at this point. And then I became department head at CMU in '99--

Fairbairn: Of the computer science--?

Bryant: Of the computer science department. CMU has a computer science department inside of a school of computer science.

Fairbairn: Oh, OK.

Bryant: But I was sort of a senior and successful faculty member. I'd had some pretty major awards I got for the work. One of the papers behind COSMOS got what's called the WRG Baker Award, which was like a best paper award for all of IEEE for the whole year, across everything they did.

Fairbairn: Wow, that is a big one.

Bryant: Yeah, that was pretty exciting. I was a fellow of the IEEE at a pretty young age for that. And I got an award-- Ed Clarke and I and Ken McMillan and one of Ed Clarke's former students shared an award called the Kanellakis Prize from the ACM, which is sort of a theory and practice award that was because of symbolic model checking. So Ken, having connected Ed and me, and then fellow named Allen Emerson who'd worked for them earlier.

So I was pretty well-recognized. The work was definitely known at that point. So I became department head in '99. And I'd sort of steered more toward wanting to raise the level of abstraction and verification up, building on techniques that came more out of the world of theorem proving than came out of the world of low-level tools. And I'd have to describe that work as a mixed success.

So the problem then, and still now, to some degree, is that for most hardware design, the sort of Verilog model is the gold standard. It's the thing that people make sure that's the reliable model. And anything you try and get above that, if you try and have a separate model, then people start getting very uncomfortable, because they're worrying about are these consistent with each other? How do I construct it? How do I maintain it? And things like that.

So I did work that was very interesting from a theoretical point of view, and has had some practical impact-- but not as much direct industry impact as this earlier work. Of trying to sort of abstract more away from low-level level Boolean signals and thinking of more high-level level symbolic values. And that continued on, and I had a collection of graduate students who worked on that up through about 2004, 2005.

And one thing that happened is I'd sort of split between the ECE and CS at CMU. My appointment was in CS, but I spend a lot of time-- the EDA crowd was mostly in ECE. And most of my ECE, and most of my--

Fairbairn: ECE stands for?

Bryant: Electrical and computer engineering, which is a separate department, separate-- it's in the engineering college. Most of my PhD students were in ECE. See A few in CS. We also came up with some interesting work on trying to verify arithmetic circuits using a variant on BDDs we called binary moment diagrams, that seemed pretty exciting at the time. And there's some attempt to use them at Intel for verifying their floating point units.

And about the time they started having the Pentium-- if you recall, the first Pentium chips had a problem with the dividers, right?

Fairbairn: [INAUDIBLE] floating point.

Bryant: And basically, we did a proof of concept showing you could take-- we didn't have Intel's exact circuit design, but ones of that general style-- general algorithm that was believed to be in there-- and show that we could verify that a correct design. And we could also show what was known about their five missing table entries in this table look-up algorithm they had-- that it could have picked up those defects as well.

And Intel in some combination took this idea and munched it around and came up with their own variation on a set of tools that to this day, they use to verify their floating point circuitry. They rely on it completely. They don't even run simulations anymore of their circuits. They have all formal verification, so. That was a little bit of a side thing, but also pretty exciting.

And so there's a little bit of an interesting thing that happened in the formal verification world of the model checking more bottom up, purely algorithmic, on one hand. And then the more classical ideas, which were from theorem proving, where you'd basically try and prove a theorem that this circuit is correct. And you'd have a computer help you with that theorem, but it was mostly coming from the heads of the prover.

But more and more, they were realizing there's sort of middle ground between those two ideas. But this is getting further away from EDA, I feel, so--

Fairbairn: Yeah, well I don't specifically want to focus on EDA, but on your specific career. So why don't we just-- it sounds like things were mainly focused on the formal verification and related things through the '90s. And did that continue into 2000s, or--?

Bryant: Really, from roughly '85 on, I realized formal verification was the thing that I was the most excited about. And I did various steps off of that that were a little bit more-- more pure simulation type of stuff. But the formal verification was what I was real excited by.

Fairbairn: So where has that led you today? What's your-- what's been your activities and focus over the last few years?

Bryant: Let me just mention a few little things. You asked about companies. So there's been a few companies along the road that have started up themselves and asked me to be advisers to them. And the one that kind of connected the most directly was one called Innologic. It was started by some people, I think, from Silicon Graphics, that basically created a symbolic simulator. It would've been-- if I'd done a start-up, this would have been the company.

Fairbairn: This would have been it, huh?

[LAUGHTER]

Bryant: A BDD based symbolic simulator that could take transistor circuits and run them and prove various properties of that. And so now, that company ultimately found a niche in verifying circuits like FPGA based circuits with embedded memories, and being able to model the-- because at the time, there were no good tools that could cross that boundary between the RAMs of the embedded memory and their memory interfaces. And then the gate level circuits of the FPGA-- the logic part of the FPGAs. And they could handle all of this, because they had switch level ability and BDDs working together.

And so it was actually a modestly successful commercial product and was acquired by Synopsys, and is still available. There's a Synopsys product that uses this today. Similarly, Cadence, I said, licensed various parts of the technology along the way. Several of my graduate students have gone to work there, and built up-- and they, again, have formal verification tools that they've built up from various different pieces of stuff merged together. But there's little chunks of my DNA lurking in some of those tools. So I'd say they've had impact on the EDA world, even if I've never been directly doing my own company.

Fairbairn: So Cadence just acquired Jasper. Was Jasper-- have any of that?

Bryant: Jasper has-- I don't know much about Jasper. As you know, once you get going with one company, you're kind of excluded from others.

Fairbairn: Right. [CHUCKLES]

Bryant: And actually, I have some students who've gone off to work for Jasper. I've known various people there, but I've never been directly involved.

Fairbairn: OK. So where do you find yourself today?

Bryant: Well, since 2004, I've been the dean at CMU. And that's really taken me away from being down in the trenches doing hardcore research, quite honestly. When I first started-- 2004, so 10 years-- I'm just finishing up, by the way. End of this month, I'm no longer Dean. July 1, I'm no longer dean.

Fairbairn: Is that good?

Bryant: Yeah, it's going to be good to change. 10 years, I think, is long enough to do any one thing. So let's see. Into the beginning of dean, I still had some graduate students working away, doing more high level kind of symbolic reasoning-- not just hardware, but these ideas actually apply to software, too. So for example, one of my students works at Microsoft doing software verification based on these.

Another one of my former students is a fellow named-- his name is Shuvendu Lahiri. Another one of my former students is at Berkeley. His name is Sanjit Seshia. And he's a successful, tenured faculty member there, doing work that's built on—some related to work that we did early on. So those ideas live on, but I haven't directly done research in that area for eight years or something.

Fairbairn: So let's talk about deanship. You came in as dean. What were challenges? What was your vision of how it--

Bryant: Well, I had the luxury at CMU of becoming dean of a very strong place that was very well established before I began. So computer science really started at CMU in the mid '50s, when it was just a sort of backwater, rust belt-- before there was a rust belt--

Fairbairn: Before it was rusty.

Bryant: Before as rusty. Smokey. Technical school. But this fellow named Herb Simon-- well-known person-- came in and recruited a protégé named Allen Newell. And they basically started-- a lot of artificial intelligence work came out of them. There's a fellow named Alan Perlis, who was early on in programming languages-- the first winner of the Turing award. And the three of them started the computer science department at CMU at a time when CMU-- it was actually called Carnegie Tech at the time-- was not a world-recognized place.

But these men-- they were men-- had this vision of computer science. They wrote an essay in Science Magazine basically arguing-- making the case why there should be a discipline called computer science. And one of the things they did that I think was unique was really project computer science in these very broad terms. So they said computer science is the theory and design of computers, as well as all the associated phenomena.

So you can think of theory and design of computers-- hardware, software, you know. That's kind of the normal mix you think about. But the "phenomena" kind of gave them-- provided a charter to think about, not just being machines that you build and program, but also robotics. Allen Newell did a lot of the early work in human computer interaction, a lot of work in artificial intelligence. And it sort of blossomed into this very expansive organization.

So I really came at it much more from the hard core computer systems, hardware viewpoint. But most of CMU is-- where I think it strikes its biggest impact is more on the AI and variations of AI. Computer vision, speech recognition, self-driving cars-- all those things that have really blossomed. I mean, just the whole field of this has.

Fairbairn: Sure.

Bryant: And I think they also, if you think about how pervasive computers have become for us, and how they're involved in everything we do. And they mediate every interaction we have with other people. Those are all the phenomena that have made it possible. So in a way, I just was lucky that I had landed in a place that had this vision.

And it became one of the first universities to have its own separate school of computer science-- separate from an engineering college or an arts and science college. That started-- actually, we're in our 25th year of that right now. And I was the fourth dean. So it wasn't like I had to do a lot of new stuff. And I feel like that 10 years, it's just a time when the whole field has expanded and exploded.

In the depth of the dot-com bust, we only were getting 1,700 applicants for our undergraduate program, where we wanted-- I mean, we want only about 120 students coming in, so that was OK. But now this year, we have over 6,000 applicants for that program.

Fairbairn: For how many slots? Is it still--?

Bryant: 142.

Fairbairn: 142. [CHUCKLES] Amazing.

Bryant: It's crazy. And so it's been pretty exciting. I'd describe it more as a ride-- just being there at school. I'd say I had some impact around various margins. One of the things that I got very excited about, starting in 2007, was big data. What we now call big data. It wasn't called that quite then.

But partly, having visited people at Google and seeing how they ran their data centers and sort of the systems end of it, and then all the-- what you could do with more and more information-- our language technologies people came back from a competition they'd had organized by NIST in language translation. Translating between newspaper articles, from Arabic to English and Chinese to English. And they'd just gotten their asses kicked by Google.

And the difference was Google wasn't particularly clever algorithmically, but they'd trained their translator on a trillion words of text and dedicated a 2000 node cluster to it at a time when everybody else was several orders of magnitude less than that. And it just showed that there's some problems that you can try and be clever or you can try and be big. And sometimes--

Fairbairn: Big is better.

Bryant: Big is better.

Fairbairn: [CHUCKLES]

Bryant: And Google has an interesting presentation they could give, where every time they doubled the size of their training data set, they got a little delta in the quality of the speech translation. And that was their key. So these people came back and say, umm, you need to get us a 2000 node cluster so we can compete against Google.

And so that kind of got me thinking about this sort of bigger picture of wow, this is amazing-- mostly being driven by industry. And the academia seemed kind of behind the game of really thinking about this. So I was able to organize both within CMU and a little bit at a national level-- sort of thinking about what it would mean to pull this into the universities.

And the National Science Foundation set up a funding program. And we formed a connection with Yahoo that let us get access to a 2000 node cluster they'd set up for university use. We were the first university to be able to do it. It was literally a shipping container on a parking lot of one of their Sunnyvale facilities, filled with machines. And it had been something they'd acquired as an experiment and decided not to use as a company. But they said hey, we could kind of get this going and let some university people get in and start using this, so--

Fairbairn: Interesting.

Bryant: So I will say I had that impact at CMU of trying to rally the troops, because we have everyone from low level computer systems people, to machine learning, to language technologies, computer vision, to the sort of people that want to use the stuff and the people developing it, and then more algorithms, programming languages, machine learning in between. So it's been a big part of my focus.

Fairbairn: Why don't we try to wrap things up then? One question we like to ask is if you put yourself back as an undergraduate and trying to figure out what you were going to do, what would you pursue today? Would you follow this path you're on? Where do you think the opportunities--?

Bryant: Well, the thing-- I purposefully got away from EDA, thinking that-- I think there are more bugs in software these days than hardware, so--

Fairbairn: Clearly.

[LAUGHTER]

By many orders of magnitude.

Bryant: And it's a lot harder problem, software verification. So a little, I morphed myself a little bit more toward the software side of things. And I still think that's a huge research area to work on--

Fairbairn: Especially with self-driving cars and all this other stuff out there, right?

Bryant: Yeah, all this other stuff that we really--

Fairbairn: You'd like to have some confidence that it actually--

Bryant: Actually works. Incredibly hard problems, because how do you even formulate a model of what a self-driving car might encounter, and therefore that it should deal with? It's really a hard set of problems.

So if I were to go through it all, obviously, I felt like I've sort of stumbled and had a lot of lucky accidents along the way. But I've mostly felt fairly fortunate about how that all progressed. And I will say that there's a lot of people on the road who've been-- you know, like I've mentioned, just a small comment here or there, a little bit of encouragement. I still remember those. And they had a pretty big impact.

Fairbairn: A pretty big impact.

Bryant: And I think people have been very generous with their time and attention on me, and I appreciate that. And I like to think I've been able to do that for some other people as well.

Fairbairn: All right, well that's a good note to wrap things up on. Thank you very much for spending a couple hours with us.

Bryant: Oh, we didn't talk about the big lie. Should we talk about the big lie?

Fairbairn: Yeah, let's talk about the big lie. Sorry, we definitely need to go back to that.

Bryant: OK.

Fairbairn: Please do.

Bryant: So I joke about this, but this is just my way of doing things. The whole Mead-Conway business was a big lie, meaning that this idea that, as I said, as long as you're not colorblind, you can do transistor circuit design.

So for example, I remember this project by Ron Rivest. RS&A-- Rivest, Shamir, and Adleman-- tried to design their own chip to do RSA encryption and decryption. And they really worked their butts off on the chip, but it had no prayer of ever working, because they didn't know circuit design. So for example, it used a DRAM circuit called a 3-transistor DRAM that I later learned basically never worked for anybody. And they didn't know that. I was only distant-- I wasn't really directly involved in their project.

But on the way, Ron thought of some interesting ideas for routing that became an algorithm he did. And so there is a whole bunch of people doing chip designs that didn't really have a clue what they were doing. But it brought in this whole collection of people, from very theoretical to people like Hennessy and Patterson and Randy Katz and that whole crowd there-- Berkeley and Stanford.

And then the whole EDA business, in many ways-- parts of that-- I think would have happened already. I think the Berkeley crowd-- the Newton, Sangiovanni-Vincentelli crowd-- they were already pretty well connected to the electronics industry. And that might have gone on its own, but other people-- I never would have gotten involved with it if it hadn't been for the Mead and Conway stuff.

And I think it did have this effect of really pulling people up out of the mire and thinking much more abstractly about what system design meant, and what it meant to design systems that had to map to the two dimensional structure of chips, and all that.

Fairbairn: Yeah, I think, as I said before, I have a very similar take on it. I was in the middle of it, and had some of the-- having an electrical engineering background, realized that there were some compromises and shortcuts taken and so forth. But on the other hand, the impact-- through using the notes that Lynn created at MIT, along with the book, along with some videotapes that we did and so forth.

To me, the most amazing thing was watching this idea, revolution, whatever, spread to hundreds of universities within a very small number of years. And whatever the missing pieces or little lies or big lies or whatever that were being told, the actual impact was the education of thousands of students in terms of high level view of an incredibly important area-- who then came to populate the semiconductor, EDA, computer science, companies, universities-- each of whom developed the additional knowledge to do things completely and make real working chips or real working tools.

Bryant: That's right, yep.

Fairbairn: But it provided a step that everybody could get excited about and get involved in, and then it was time to learn more. But it was an incredibly important launching pad--

Bryant: And it was done in a way-- I mean, there was some number of old-school people who said, [SCOFFS] it's just a bunch of bologna.

Fairbairn: Oh yeah, a very large number.

[LAUGHTER]

Bryant: But there were enough people-- you know, like Carver, who were, I'd say, more enlightened. They realized it was way harder than this, but they could communicate effectively. And you look at the places like Bell Labs that really picked up on these things. There were-- groups of people got together and educated each other in many different ways.

And the whole tool business-- I think the idea of a much more tool-oriented approach to it was part of what came out of that, too. So that was pretty exciting. So it was a--

Fairbairn: Yeah, absolutely.

Bryant: And it's amazing to think about how all that converged. What a perfect timing it was. It was at a time when there was a relatively stable NMOS process called HMOS-- remember?-- that was fairly standard across the industry. The chips were big enough you could do interesting things with them, but small enough that it was hard. And everything was just right on the verge, so the idea that this simulator developed with a student then became the mainline simulator for Intel.

[LAUGHTER]

Fairbairn: But that was an indication of the importance of getting people with that kind of background, that kind of perspective, involved-- even if you didn't know all the details, there was a huge gap in terms of who was participating in the world of chip design. There were very skilled, but very narrowly focused circuit designers. And they weren't going to get where we needed to go without the help of computer scientists such as yourself.

Bryant: Yeah. No, it was really a-- I call it the big lie, but it was an amazingly successful time. And then DARPA built up a big program and provided-- and NSF built up a big program. So there was real money behind it, and that's critical in a university to really make people put down whatever they're doing and start something new. There has to be funding, or it just doesn't happen.

And as I mentioned, the DARPA program meetings were-- to me, that was the members of the fraternity. That was where you really made your mark on who had the best ideas at any given time.

Fairbairn: Anything else we've left out?

Bryant: That's the main one.

Fairbairn: All right. That's a wrap. Thank you.

END OF INTERVIEW