October 3, 2007 Newsletter

Logic synthesis pioneer eyes new horizons

Richard Goering
SCDsource.com, Latest News

EDA pioneer Robert Brayton, winner of this year's EDA Consortium Phil Kaufman award, has made fundamental contributions to circuit simulation, logic synthesis and formal verification – but he's not stopping yet. Brayton is hard at work today on new research that promises to integrate logic synthesis and formal verification.

Brayton, professor of electrical engineering and computer science at the University of California at Berkeley, will receive the EDA industry's highest award Nov. 1 at the 14th annual Kaufman award dinner in Santa Clara, Calif. Given annually to an individual who has had a demonstrable impact on EDA, the Kaufman award is sponsored by the EDA Consortium and the IEEE Council on EDA (CEDA).

Later in November, Brayton will give a keynote speech at the Formal Methods in Computer-Aided Design (FMCAD 2007) conference in Austin, Texas entitled "The synergy between logic synthesis and equivalence checking." It reflects his ongoing work with a U.C. Berkeley research project called ABC, "A simple system for sequential synthesis and verification."

Few people have been as deeply involved in the history of logic synthesis as Brayton. He managed the development of the pioneering Yorktown Silicon Compiler at IBM Research in the 1980s, and he did fundamental research in multi-level logic synthesis at both IBM and U.C. Berkeley. Brayton later applied some of the work he'd done in logic synthesis to formal verification.

Brayton's fascination with computers began when he was an electrical engineer in 1956 and 1957, at a time, as he said, "when there weren't many around." He was working for Sperry Rand Univac at the time. In 1961, he earned his PhD in mathematics from the Massachusetts Institute of Technology (MIT). At that time he developed the first Lisp compiler. That was "just for fun," he said – his PhD thesis was on differential equations.

From 1961 to 1987, Brayton worked at the IBM Thomas J. Watson research center in Yorktown Heights, N.Y. His first involvement with design automation came when he developed the "sparse tableau" approach, a way of formulating circuit equations and then solving them using efficient sparse matrix techniques. This approach was employed in the Advanced Statistical Analysis Program (ASTAP), a circuit simulator that IBM developed in the late 1960s, before deployment of the U.C. Berkeley Spice circuit simulator.

"In sparse matrix solving techniques, we were ahead of Spice," Brayton said. "I think we influenced what they did. They saw what was working well, but they had their own way of formulating equations. What we did stayed within IBM so it didn't get as much publicity."

Learning to synthesize

In the early 1980s, Brayton turned his attention to digital design. "At the time, nobody thought you could automatically synthesize logic very well, so we thought the next thing you needed to do was to automate that," he said. "The time was ripe, but the question was how well could you do it

compared to what you could do by hand?" Some of Brayton's research led to the development of Espresso, a two-level logic optimization algorithm still in use today.

Initially, Brayton recalled, some people at IBM didn't even want to use two-level logic synthesis, but two-level synthesis became superior to what engineers could do by hand. Multi-level logic synthesis soon became the next challenge. Brayton recalled a "very productive summer" in 1982 in which he and several other researchers developed some of the first techniques for multi-level logic synthesis. One of those researchers was Alberto Sangiovanni-Vincentelli, today a fellow professor of electrical engineering and computer science at U.C. Berkeley.

"I went to Berkeley in 1985 on a one-year sabbatical from IBM, so at that time we said let's take these [multi-level logic synthesis] ideas and develop them further and make sure they run fast," Brayton said. "We reprogrammed things in C and put it all together with wonderful grad students." The first result was a program called MIS, and the subsequent result was the publicly-available Sequential Interactive Synthesis (SIS). Ideas from MIS and SIS are "underneath the covers" of today's commercial synthesis tools, Brayton said.

The Yorktown compiler, Brayton noted, started up in the late 1980's and involved a combination of physical placement and routing with RTL synthesis. Ralph Otten, professor at Delft University in The Netherlands, was a driving force behind it, and Raul Camposano, who later became Synopsys CTO, was one of the developers.

"We developed something that went all the way from a behavioral description to the chip level," Brayton said. "It wasn't commercially developed, it just demonstrated what you can do. We had quite a nice group and we had fun developing this. I was the manager, but my way of managing was to stay out of the way of the people."

Going formal

Brayton went to U.C. Berkeley as a full-time professor in 1987. One new area he got involved in there is formal verification. That involvement followed what Brayton called an "eye-opening talk" by Bob Kurshan of Bell Labs in 1990. "It showed how you could formally verify things," Brayton recalled. "We had expertise in manipulating logic that had not been applied to verification, so we started a major project to develop an engine for formal verification using BDDs [binary decision diagrams] since we had expertise in those from SIS."

The result was Verification Interacting with Synthesis (VIS), a program used today in many formal verification tools. VIS is a system for formal verification, synthesis, and simulation of finite state systems.

This involvement with formal verification is serving Brayton well today with the ABC project, which is aiming to make logic synthesis "scalable" to extremely large circuits. Beyond scalability, the project's other goal is "sequential" logic synthesis. This kind of synthesis seeks to manipulate registers as well as the combinational logic between register boundaries. In addition to retiming, sequential transformations could potentially modify both the logic structure and the positions of latches.

Brayton and Alan Mishchenko, associate research engineer, have set forth a requirement: everything that's synthesized has to be verified with equivalence checking. "Most people don't accept sequential logic synthesis, and one reason is that you can't prove equivalency," Brayton said. "So we're co-developing techniques that can do both. We don't want to synthesize something for which we can't check equivalency."

Brayton is concerned that academic research for design automation is focusing on "hyped" areas such as power and high-level design, and bypassing more traditional areas. "Floorplanning, layout, and routing all still need to be there, and research support is kind of drying up," he said. "There will be a crunch on well-trained graduate students in these areas if funding doesn't turn around. There's not enough money to support traditional areas that still need to be supported. This could be something to worry about in the future."

Return to October 3, 2007 SCDsource Newsletter

Read September 26, 2007 SCDsource Newsletter

To register yourself or others to receive future newsletters and to enjoy the privileges of SCDsource Community Membership, please visit:
http://www.scdsource.com/registration_page.html