Friday, October, 26th 2007

Academic Insights

Berkeley ABC project reshapes logic synthesis

By Richard Goering **10/24/07**

A research project at the University of California at Berkeley is quietly working to redesign logic synthesis so that it's scalable, fast, and verifiable. Called ABC, the project has already broken new ground in combinational and sequential logic synthesis, as well as sequential logic verification.

The project is directed by logic synthesis pioneer Robert Brayton, professor of electrical engineering and computer science at U.C. Berkeley, and Alan Mishchenko, associate research engineer. The two have been working with graduate students since 2005 on ABC, described as "a system for sequential synthesis and verification." The project started with a mission to develop fast and scalable logic synthesis, and grew to encompass sequential logic synthesis that can be immediately verified through equivalence checking. It provides free public access to code and research results through the project's web site.

There's no lack of ambition behind the project. "The overall goal is to come up with logic synthesis for the 21st century," said Mishchenko.

"The higher-level motivation was to make things scalable so we can handle very, very large problems," said Brayton. "That meant keeping things simple. And we wanted to make things fast. So our key motivations were to make things fast and scalable, and we have succeeded in doing this way beyond expectations."

The ABC project has published several papers and publicly released code that's been picked up for internal use by several companies, Brayton and Mishchenko say. The project receives funding from the Semiconductor Research Corp. (SRC), National Science Foundation, and the State of California MICRO foundation. Industrial sponsors include Actel, Altera, Calypto, Magma, Synplicity, and Synopsys.

According to the project's web site, the current publicly-available version of ABC can optimize, map and retime industrial gate-level designs with 100,000 gates and 10,000 sequential elements for optimal delay and minimized area in about one minute of CPU time. The runtime of combinational synthesis, mapping and verification is typically faster.

ABC is receiving notice in the industrial world. Xilinx is looking into adopting some of ABC's algorithms for current and future use, said Stephen Jang, project lead for synthesis and mapping at Xilinx. "ABC is scalable and fast," he said. "The synergy of logic synthesis and verification adopted in ABC makes it possible to check the equivalence of the netlist in a timely manner. For a typical design, ABC can verify functional equivalence several orders of magnitude faster than other approaches."

Steven Burns, technical leader of the convergent design flow group in Intel's Strategic CAD Labs (SCL), noted that SCL has been funding ABC through matching grants with MICRO or with SRC funding. He said that ABC's SAT (satisfiability) sweeping, an algorithm used for Boolean reasoning in functional verification and logic synthesis, "is so fast it changes your mind about what you would consider trying during synthesis." Moreover, he said, ABC's technology mapping produces "very good results in both delay and area."

Jason Baumgartner, technical lead of IBM's SixthSense formal verification project, said his team frequently interacts with the ABC team to share ideas. "My team has consistently found the technology being pioneered by the ABC team to be of the highest caliber," he said. "This work is of central importance to all those who focus on synthesis or verification. The focus of the ABC team on 'verification after synthesis' furthermore promises a scalable approach to the increasingly important topic of validating the correctness of logic synthesis through equivalence checking."

Improving synthesis algorithms

Brayton, winner of this year's IEEE Council on EDA (CEDA) and EDA Consortium Phil Kaufman award, has already made significant contributions to logic synthesis. He played a key role in the development of fundamental synthesis algorithms such as Espresso, MIS, SIS, and the MVSIS multi-valued logic program.

Mishchenko received his PhD from the Glushov Institute of Cybernetics in Kiev, Ukraine in 1997, and was an Intel-sponsored visiting scientist at Portland State University in Oregon from 1998 to 2002. He joined the electrical engineering and computer science department at U.C. Berkeley in 2002, where he is currently a research scientist in Brayton's group.

The ABC project started out as an attempt to "architect MVSIS for the 21st century," as Mishchenko put it, by going back to binary synthesis and looking for better ways of doing things.

Robert Brayton (left) and Alan Mishchenko (right) are seeking to revitalize logic synthesis through the University of California ABC project.

For lack of a better name, Mishchenko said, ABC came to mind; it reminded him of the title of a book he had used to study English in his native Ukraine. "We wanted to keep things simple, so it kind of stuck," Brayton said. "It doesn't mean anything."

The ABC project started out with combinational synthesis. In casting about for ideas on how to improve MVSIS, Mishchenko came across a paper that had been given at the International Conference on Computer-Aided Design (ICCAD) 2004 called "DAG-aware circuit compression for formal verification" by Per Bjesse of Synopsys and Arne Boralv of Prover Technology. A DAG is a data structure called a directed acyclic graph, and the paper described a particular type of DAG, an and-inverter graph (AIG), in the context of formal verification. An AIG consists of a two-input AND gate and an inverter. Rewriting is an algorithm that minimizes AIG size.

Mishchenko decided to apply AIG rewriting to synthesis, and the results were "quite amazing," he said. AIGs improve logic synthesis, he said, "because the AIG is so simple and easy to manipulate. All the nodes have two inputs and all the nodes are exactly the same." With AIGs, Mishchenko said, you can have very simple cost functions that are targeted by multiple algorithms. Synthesis is less effort because the intermediate representations are "very simple and homogeneous."

As a result, Mishchenko said, AIG-based synthesis speed is linear with respect to circuit size, which is not the case with traditional logic synthesis. Mishchenko and Brayton summarized their early ABC project research in a Design Automation Conference (DAC) 2006 paper titled "DAG-aware AIG rewriting: a fresh look at combinational logic synthesis." According to the paper, experiments on large industrial benchmarks show that the AIG-based methodology scales to very large designs, and is several orders of magnitude faster than SIS or MVSIS while offering comparable or better quality of results.

Sequential synthesis

More recently, the ABC project has focused heavily on sequential synthesis, which seeks to manipulate registers as well as the combinational logic between the registers. Register retiming is a sequential synthesis technique that's sometimes used today. Brayton noted that sequential synthesis will provide additional improvements in area, delay and power.

"And sometimes it's a surprising amount," he said. "We've shown in the past that if you do retiming, you can bring delay down substantially."

Brayton noted, however, that sequential synthesis has not been broadly accepted in commercial tools because engineers have generally been unable to verify the results through equivalence checking. And that's the problem the ABC project is trying to solve. The mandate, as Brayton put it, is that "we don't want to synthesize something for which we can't check equivalency."

It's a difficult problem. While combinational verification looks at one clock period and one "snapshot" of the circuit, sequential verification must look at sequences of executions that go for many iterations, Mishchenko noted. "You have to ask whether the design is behaving equivalently for many long runs across the system," he said. "Instead of just looking at logic correctness, you have to verify over a long period of time execution."

Mishchenko said the project made some good progress this past summer, conducting research in which sequential synthesis and verification "really clicked together in a big way for the first time." The trick, he said, is a flow that takes multiple synthesis "snapshots" without committing to a final network. It then does retiming and finds an optimal delay mapping for all the possible structures.

The ABC project is pushing the idea of taking intermediate "snapshots" of the synthesis process as a way of providing hints for sequential equivalency checking. "We have a way to do this so data storage doesn't go up a lot," Brayton said. He noted, however, that commercial synthesis tools today don't record this type of history. And Mishchenko acknowledged that commercial vendors may be hesitant to reveal what's going on inside their synthesis tools.

A better mapping

Meanwhile, the ABC project has extended the AIG concept to sequential logic synthesis. In a paper to be presented at next month's ICCAD 2007 conference, ABC researchers will present AIG-based sequential synthesis that integrates retiming with technology mapping, resulting in delay reductions with marginal area degradation. The paper describes the use of a mapping algorithm that uses what the authors call "priority cuts" to overcome memory and runtime problems.

Register retiming isn't the only type of sequential synthesis optimization. In a research paper prepared for the SRC, Brayton and Mishchenko use sequential synthesis to identify "sequentially equivalent" nodes, or nodes that have the same or opposite values in all reachable states. These nodes can be merged without changing the sequential behavior of the circuit, leading to a reduction in circuit size.

According to the paper, this flow was applied to a set of 50 industrial benchmarks, and it showed an average reduction of 32 percent in register count and 15 percent in area, while preserving the delay. Further, the results can be verified through equivalence checking. As with all ABC research, the implementation is publicly available.

The project's future work, Mishchenko said, will focus on several directions. One will continue to explore the synergy between sequential synthesis and

verification. Another is to build flexible synthesis and mapping flows that can be customized for target architectures such as ASICs, FPGAs, macrocells, hardware emulators, and nanotechnology or biology. A third is to provide elements of ABC within a modular "toolbox" rather than the all-in-one tool that's available today.

What started out as an effort to improve MVSIS seems to have gone much further. "We're completely revamping how synthesis is done," Brayton said. "If you look at the end result, there's very little of the classical techniques left. They're being replaced by a more modern, efficient, simple way of doing things."

Related Articles

Logic synthesis pioneer eyes new horizons

Back to Home Page...