FM08 Tutorial

Getting Formal Verification into Design Flows

Arvind
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology

There are dozens of papers and theses on successful verification of cache-coherence protocols and out-of-order processors. Yet this formal verification has little impact on the design of actual chips embodying these functionalities. Formal verification does not seem to reduce the need for ad hoc verification and testing of designs before fabrication. We will give reasons for this persistent "gap" and offer a possible solution.

Lecture-1 Why formal verification remains on the fringes of commercial development

The goal of formal verification has to be to help designers produce better (more efficient, cost-effective, faster, reliable, reusable, etc.) designs. We will show:

- Different applications have vastly different correctness criteria
- No single formal technique can be used for all of these applications
- For maximal impact, formal methods should be aimed at the design process and not on post-design verification
- Design languages with proper semantics are a prerequisite for integrating formal verification into the design flow

Lecture-2 Bluespec: Theoretical underpinnings

After brief motivation for the usefulness of Guarded Atomic Actions in describing behavior of concurrent digital systems, we will describe the Bluespec language and give its operational semantics.

Lecture-3 Gaining confidence in your designs: Applications with fuzzy correctness criteria

Using the implementations of an 802.11a (WiFi) transmitter and an H.264 video decoder as examples, we will discuss:

- The need for executable specifications and how they can be leveraged in the design process
- How formal verification simplify the task of modular refinement

Lecture-4 Gaining confidence in your designs: Applications with total correctness requirements

We will discuss, using the examples of out-of-order processors and cache-coherence protocols, the great value of formal methods from model-checking to mechanical theorem proving provided:

- Formal techniques apply directly to the design descriptions from which the final circuits are mechanically synthesized;
- Formal techniques allow designers to reason in the semantics of their design language as opposed to an abstract mathematical domain.

Biography: Arvind is the Johnson Professor of Computer Science and Engineering at MIT where in the late eighties his group, in collaboration with Motorola, built the Monsoon dataflow machines and its associated software. In 2000, Arvind started Sandburst which was sold to Broadcom in 2006. In 2003, Arvind co-founded Bluespec Inc., an EDA company to produce a set of tools for high-level synthesis. In 2001, Dr. R. S. Nikhil and Arvind published the book "Implicit parallel programming in pH". Arvind's current research focus is on enabling rapid development of embedded systems. Arvind also manages the CSAIL-Nokia collaboration and recently was inducted in the NAE. http://www.csg.csail.mit.edu/Users/arvind/