Formal Modeling and Verification of Hardware in VHDL

William D. Young
Ken Albin
Warren A. Hunt, Jr.
David Russinoff

Computational Logic, Incorporated
1717 West Sixth Street, Suite 290
Austin, Texas 78703-4776
(512) 322-9951

CLI
COMPUTATIONAL
LOGIC
INCORPORATED

Hardware Verification: What Is It?

Industry-standard techniques for assuring the quality of hardware are:

- Simulation;
- Testing.

These are crucial tools, but we hope to *model* hardware mathematically and *prove* the correctness of our hardware modules.

We would like to work toward the point where industry standard documentation for a hardware system is a *formula manual*, a collection of fully formal specifications that provide a precise *and complete* basis for using the device.

CLI
COMPUTATIONAL
LOGIC
INCORPORATED

What Stands in the Way

To date, hardware verification has been hampered by:

1. industry reluctance to embrace an unproven technology;
2. the limited *coverage* of our formalisms;
3. the perception that formal techniques are only applicable to toy circuits;
4. the use of arcane, non-standard notations;
5. the inaccessibility of the supporting tools;
6. the significant mathematical sophistication required to practice formal verification.

CLI
COMPUTATIONAL
LOGIC
INCORPORATED

What is VHDL?

VHDL is the VHSIC Hardware Description Language, a widely used HDL suitable for describing circuits at various levels of abstraction. VHDL

- permits hardware descriptions in behavioral, structural, and dataflow styles;
- is an IEEE standard, rapidly becoming a *lingua franca* among the hardware design community;
- is a rich, expressive (read complicated) language;
- is well documented in a rapidly growing literature;
- is implemented with an event-driven simulator.

Consequently, any formal semantics for VHDL should be in terms of event-driven simulation, if it is to coincide with the intuitive semantics of the language.

CLI
COMPUTATIONAL
LOGIC
INCORPORATED

Some Key Lessons

From our work with the Brock/Hunt HDL and from studying others' attempts at modeling VHDL, we have arrived at the following key aspects that our formalism must have.

- Deep semantic embedding;
- Support hierarchical design;
- Event-driven simulator semantics;
- Points of Abstraction;
- Maximum automation.

CLI COMPUTATIONAL LOGIC INCORPORATED