Verification of Circuit Generators

Warren A. Hunt, Jr. and Erik Reeber
{hunt,reeber}@cs.utexas.edu

The verification of an automatically generated circuit description usually involves verifying the netlist post-synthesis. We have developed an approach for verifying the correctness of the circuit generator software directly, thus obviating the need to verify the resultant circuit descriptions. Our approach is based on deeply embedding a HDL into the ACL2 logic; we then use the ACL2 theorem-proving system to verify the circuit generators. Our approach also permits the specification and verification of non-functional circuit properties.

This work builds on the FM9001 microprocessor circuit generator work [2] by adding both verification techniques and language annotation facilities. In our earlier work, we only employed theorem-proving techniques, but our current effort now also permits the use of SAT and BDD based techniques. In addition, our current approach to verifying circuit generators permits a generator to make choices based on non-functional criteria. For example a circuit generator may produce different structural circuit descriptions depending on wire lengths, circuit primitives, target technology, and circuit topology.

This work is similar in spirit to work by the functional language community to generate regular circuits using functional programs. For instance, the WIRED language has been used to improve performance of multipliers by incorporating layout information into the design of circuit generators [1].

To make it possible to verify circuit generators, we require our circuit generators to produce circuit descriptions in our formally defined **DE2** FSM language [3]. To prove the correctness of a circuit generator, we prove that all possible **DE2** circuits generated are functionally equivalent to an ACL2 model. Consider the verification of a circuit generator that uses 4-bit IP blocks to implement an n-bit ALU. We first define the 4-bit IP block within **DE2** and then produce its equivalent ACL2 *semantic* function; this process, usually automatic, proves that the semantic function correctly specifies the IP block. We write a parameterized circuit generator that selectively uses this IP block along with other primitive elements to generate a circuit. Finally, we mechanically verify the correctness of the circuit generator program with respect to the semantics of the **DE2** language.

SoC development strategies generally involve the use of specific configurations of parameterized IP blocks. Our approach facilitates SoC development by allowing the direct verification of the possibly nearly infinite number of intended chip configurations. Our approach is different from other work known to us in that we do mechanically verify our circuit generator programs.

References

[1] Emil Axelsson, Koen Claessen, and Mary Sheeran. Wired: Wire-Aware Circuit Design. In *Proceedings of the 13th Conference on Correct Hardware Design and Verification Methods (CARME 2005)*, pages 5–19. Springer, 2005.

[2] Warren A. Hunt, Jr. and Bishop C. Brock. A Formal HDL and Its Use in the FM9001 Verification. In C.A.R. Hoare and M.J.C. Gordon, editors, *Mechanized Reasoning and Hardware Design*, pages 35–48. Prentice-Hall, 1992.

[3] Warren A. Hunt, Jr. and Erik Reeber. Formalization of the DE2 Language. In *Proceedings of the 13th Conference on Correct Hardware Design and Verification Methods (CARME 2005)*, pages 20–34. Springer, 2005.