A Generation Framework for the Verification of Open-Source Processor Cores

Annachiara Ruospo
Politecnico di Torino
Corso Duca degli Abruzzi,24 10129 Torino (TO)

annachiara.ruospo@polito.it

Keywords. Verification, Embedded Systems, Evolutionary Algorithm

Abstract

Nowadays hardware devices are becoming even more complex and heterogeneous. Many Systems-on-a-Chip are composed by multi-core processors and several IP interfaces, different kind of memories and analog circuitry which communicate via many different interface protocols. On a single Integrated Circuit (IC) there are millions of logic gates shrunk on a smaller area. This means that delivering a bug-free design is more complicated than before. The interest for open-source hardware in real products is demanding for tools and advanced methodologies for verification to provide high reliability to open and free IPs. To address these technical challenges, a new effort and more importance must be invested on verification processes. In this work, an open-source evolutionary optimizer has been used to create functional test programs that improve the verification test set for an open-source microprocessor, enhancing in this way, the verification level of the device. A simulation-based verification framework based on an evolutionary algorithm (μGP) is developed to increase the verification level of a RISC-V core by exploiting an automatic test program generation scheme. μGP (microGP) is an open-source evolutionary-based tool devised since 2002 in the CAD group at Politecnico di Torino which is able to automatically generate syntactically correct assembly programs. Riscy is an open-source, 32bit, in-order, 4 pipeline stages RISC-V microprocessor based on the open-source RV32IMFC extensions of the RISC-V ISA and it is described in SystemVerilog. It has been developed by ETH Zurich and University of Bologna under the Parallel Ultra-Low Power Platform (PULP) project. The verification programs are generated by μGP to optimize code coverage metrics and are tested against a high-level model to find device incorrectnesses during the generation time. A perturbation mechanism has been included in the verification framework to cover parts of the device under verification not reachable with only software stimuli such as interrupts or memory stalls. The proposed methodology uncovered 10 bugs still present in the RTL description of the analyzed device and demonstrated the effectiveness of open-source verification tools for the next generation of open-source microprocessors. As already remarked, to experimentally demonstrate the effectiveness of the proposed method, the RTL description of the Riscy micro processor is targeted. The fitness of every verification program is evaluated by measuring its capacity to maximize the code coverage metrics on the processor model. For any verification program, a fitness value was computed by resorting to an equation that includes together the different metrics; in particular, the used metrics are: Statement Coverage, Branch Coverage, Condition Coverage, Expression Coverage, FSM State Coverage, and FSM Transition Coverage. At the beginning, It was observed that by running handwritten programs code coverage was always lower than 50%. Then, a rich Instruction library has been created to cover the open-source RV32IMFC extensions of the RISC-V ISA, but the results were

still unsatisfactory. To stress more complex units and corner cases, a second library has been designed to focus on FPU instructions without polluting the individuals with the generation of integer instructions. Many illegal instructions and corner cases (such as NaN, infinite, division by zero, etc.) have been added to increase the coverage. This approach has also been extended to other critical units, achieving good results. The union of all the best individuals results in a final code coverage average of about 90%. At the beginning, to check the correctness of RI5CY core, the set of programs with the 90% of final average code coverage has been picked and used to identify incorrect DUV behavior. Unfortunately, not even a bug has emerged during this process. This fact strengthens the idea that a high code coverage is not a measure of correctness. To reach a 90% of device's code coverage, the evolutionary algorithm can reject some low-coverage individuals that could contain a bug. In fact, having the generation and verification phases split means using only the best individual to test the correctness of the core. In our approach generation and verification have been merged. In this way, every individual is not only used to calculate the code coverage, but it has the possibility of being executed and compared against the golden model to find design bugs. This allows to explore a wider space of programs and increasing the probability of success. This approach shows that it is not only important to maximize the code coverage but also to leverage less important individuals during the generation phase. Thus, the final test set uncovered 10 design bugs during the verification process.

Acknowledgment

This project was born from a collaboration between Politecnico di Torino and ETH University intended to define a new open source verification framework to increase the verification level of the RISC-V embedded systems. The main supervisors were Ernesto Sanchez and Luca Benini. The paper "An Open-Source Verification Framework for Open-Source Cores: A RISC-V Case Study" has been submitted to the VLSI-SOC 2018. The authors of this paper are Pasquale Davide Schiavone, Ernesto Sanchez, Annachiara Ruospo, Francesco Minervini, Florian Zaruba, Germain Haugou and Luca Benini.