Functional Test Generation for Digital Integrated Circuits Using a Genetic Algorithm

Xiaoming Yu† Alessandro Fin‡ Franco Fummi‡ Elizabeth M. Rudnick†

† Center for Reliable & High-Performance Computing, University of Illinois, Urbana, IL
‡ Dipartimento di Informatica, Verona, ITALY

In order to manage the complexity and ensure the correctness of modern integrated circuit (IC) designs, such as microprocessors, designers usually adopt a top-down approach. Given the functional specification of a design, designers translate it into a description written in some hardware description language, such as VHDL, Verilog, or SystemC. This initial design description is typically entered at the register transfer level (RTL). However, a higher level behavioral description may be used instead before the design is further refined to the RTL. The RTL is subsequently synthesized into a network of transistors or logic gates, known as the gate-level description. Using this design process, designers can focus on the most important part of a design. Since a design may go through behavioral, register transfer, and gate levels, it is important to verify the initial description and also the functional equivalence of descriptions at different levels of abstraction. Insufficient verification can lead to product recalls when bugs are finally detected by customers, which can be extremely costly. Simulation-based approaches, which involve applying level-dependent test generation techniques, are typically used for verifying a design. Tests are generated after each level of the design is completed and then simulated at that level and the higher level. Results are compared across different levels to check if there is a mismatch. However, test generation is a difficult and time-consuming process.

Genetic algorithms (GA's) have been shown to be effective when solving state space search problems. In the area of test generation for manufacturing test of digital ICs, GA's have been effectively used to solve the gate-level test generation problem. In this work, GA's are proposed to generate functional tests for higher levels that can be reused at lower levels in a homogeneous design and test generation environment. The functional tests can be used for design verification and also for development of tests for manufactured parts. In the proposed method, a GA is first used to generate tests at the behavioral level for descriptions written in SystemC. As the design is translated into RTL, the generated tests are simulated first and another GA is used to generate additional tests to obtain better coverage using coverage metrics targeted at the more detailed design. Then at the gate level, vectors generated at the behavioral and register transfer levels are reused before a gate-level test generator is applied. By reusing functional tests from higher levels, the overall test generation time can be decreased, thus reducing the development period of the design. Furthermore, the functional tests generated at behavioral and register transfer levels are useful in identifying design errors during the design verification process.

A simple GA was used to repeatedly generate test sequences to target statement coverage, i.e., coverage of the statements in a SystemC or RTL description, and path coverage, i.e., coverage of all paths in the SystemC or RTL description. The fitness value of each individual of the GA is determined by the number of covered statements or paths. The GA tries to cover each statement or path several times so that each bit error, which reproduces the single stuck-at fault at behavioral and register transfer levels, gets a chance to be detected. Each sequence generated by the GA is fault simulated to remove the detected bit errors. The GA stops when no more bit errors can be detected. By targeting statement coverage or path coverage, only logic simulation, which is much faster than fault simulation, is involved in the GA, and it is performed only once for each candidate sequence. This is the first reported approach for targeting path coverage automatically.

Experiments were carried out on a sample of the ITC99 benchmarks to demonstrate the effectiveness of the proposed functional test generation method. The GA was used to generate test sequences at the RTL targeting 100% path coverage. The sequences were then simulated at the gate level to remove detected faults before a gate-level test generator was applied. Experimental results show that the fault coverages achieved by the proposed method are greater than or equal to those of gate level test generation alone for five out of six circuits in less time. For the two large circuits b07 and b11, significantly higher fault coverages are achieved in much less test generation time.