New techniques for functional test of processor-based systems

Riccardo Cantoro

Introduction
Electronic devices may be affected by faults as an effect of physical defects. These defects may be introduced:
• during the manufacturing process
• while the device is operative (due to aging).
Safety-critical applications do not tolerate errors due to faults: this is the reason why testing such devices is needed so to guarantee a correct behavior at any time. Testing is performed with different approaches.

Design for Testability – the original design is equipped with special hardware devoted to testing
• Scan-chains
• Logic/memory built-in-self-test (LBIST, MBIST).
Software-Based Self-Test – a suite of test programs is stored in a memory accessible by the processor
• The processor executes the test program
• Results are gathered and compared with the expected ones.

New systematic SBST algorithms
In my research, the fault coverage of several processor sub-modules have been increased by means of systematic SBST algorithms
• Register forwarding & pipeline interlock [MTV’13]
• Decode units [DFT’14]
• ALU modules [EST’15]
• Cache coherency logic [LASCAS’15]
• Embedded floating-point units [DFT’16]

The test environment is represented with formal constraints written in hardware-description language (HDL). A satisfiability (SAT) solver is used to generate test programs able to cover all detectable faults and to respect the constraints.

Results
• The framework is able to prove that certain faults are untestable by means of functional programs.
• The testable fault coverage is superior to manual generation techniques (up to 98% coverage on a MIPS-like processor).

Part of my research focuses on quantitative evaluation of the drop in fault coverage coming from the adoption of the alternative approaches.

Effective industrial development flow
Industrial processors are typically too large to test as a unique module. In my research, a general SBST development flow has been implemented. The overall flow is based on the following principles:
Modularity – the fault list is split in sub-modules by taking into account structural and functional aspects.
Parallelization – test programs are developed in parallel on orthogonal fault lists.
Positive side-effects – previously generated test programs are evaluated on a larger set of faults before starting a new generation phase.

Additional aspects are considered when dealing with industrial test programs development:
• Time and memory constraints
• Coexistence with the final application (mission) and the Operating System.
• Robust execution.

Results refer to a SoC employed in safety-critical automotive embedded systems, such as airbag, ABS, and EPS controllers. SoC is currently manufactured by STMicrowoltrons.
• 73 test programs written in assembly
• Full execution time: 0.8 ms (@150 MHz).