Self-Test Libraries Analysis for Pipelined Processors Transition Fault Coverage Improvement

Riccardo Cantoro, Patrick Girard, Riccardo Masante, Sandro Sartoni, Matteo Sonza Reorda, Arnaud Virazel

Self-Test Libraries Analysis for Pipelined Processors Transition Fault Coverage Improvement

Riccardo Cantoro[1], Patrick Girard[2], Riccardo Masante[1], Sandro Sartoni[1], Matteo Sonza Reorda[1], Arnaud Virazel[2]

[1]Politecnico di Torino
Turin, Italy

[2]LIRMM
University of Montpellier / CNRS
Montpellier, France

Keywords. software-based self-test, software test libraries, on-line test, transition delay test, safety, functional test

Abstract

Testing digital integrated circuits is generally done using Design-for-Testability (DfT) solutions. Such solutions, however, introduce non-negligible area and timing overheads that can be overcome by adopting functional solutions. In particular, functional test of integrated circuits plays a key role when guaranteeing the device's safety is required during the operative lifetime (*in-field test*), as required by standards like ISO26262. This can be achieved via the execution of a Self-Test Library (STL) by the device under test (DUT). Nevertheless, developing such test programs requires a significant manual effort, and can be non-trivial when dealing with complex modules. This paper moves the first step in defining a generic and systematic methodology to improve transition delay faults' observability of existing STLs. To do so, we analyze previously devised STLs in order to highlight specific points within test programs to be improved, leading to an increase in the final fault coverage.

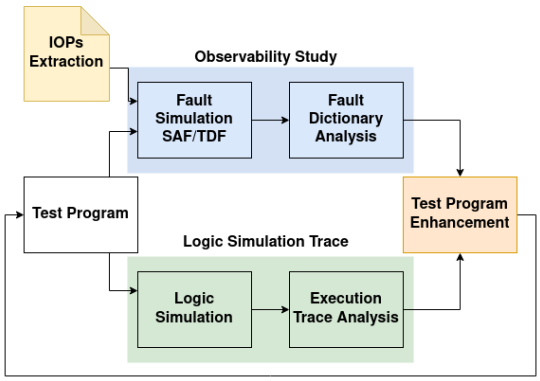

Figure 1: Proposed test flow

1.1 Proposed Approach

Fig. 1 shows the proposed methodology, which can be divided into two different processes:

1. **Observability study:** this process aims to give some insights on not observed (NO) faults: devising test strategies for such faults depends on where their effects propagated and stopped. For this reason, we define two groups of internal observation points (IOPs), namely *User Accessible Registers (UARs)*, registers directly accessible by the user through available instructions, and *Hidden Registers (HRs)*, hidden within sub-modules or glue logic and not directly accessible. In this process, we generate a fault dictionary, which includes timing information on fault detection. Moreover, we analyze and correlate SAF and TDF coverages, based on the implications existing between the two fault models, as testing a TDF implies testing the relative SAF.

2. **Logic Simulation Trace:** this process allows to map the execution time to the instructions currently executed by the processor core. Combining data from the fault dictionary and the execution trace, the proposed test flow allows to easily identify what portion of the code must be improved to cover NO faults. Which and how many instructions to use in general depends on the IOP reached by the fault and is not the main focus of the current work.

1.2 Experimental results

The approach presented in this work has been validated on the open-source SoC PULPino [1], which has been synthesized using a 45nm library and resulted in a total number 159, 326 transition delay faults. As for the test programs, we used three STLs based on different algorithms to test stuck-at faults. Data on these STLs, together with the experimental results, are reported in Table 1.

Table 1: STLs general information

Test Program	*#Clock cycles*	*Memory size [kB]*	*SAF coverage %*	*Initial TDF coverage %*	*Potential TDF coverage %*	*Potential gain [percentile units]*
STL1	17, 308	27.32	81.42	61.73	70.88	9.15
STL2	31, 158	27.86	81.86	44.19	53.15	8.96
STL3	80, 455	16.68	82.18	62.54	80.39	17.85

Results show that, assuming the test engineer can deploy strategies to recover all NO faults, a total number of 14, 580, 14, 275, and 28, 433 transition delay fault can be recovered for STL1, STL2, and STL3 respectively, leading to the potential gain increment reported in the results table.

Future works will focus on the development of strategies to observe effects of NO faults at the DUT's primary outputs.

Paper origin

This paper has been accepted and presented at the conference IEEE International Symposium on On-Line Testing and Robust System Design (IOLTS) 2021.

References

[1] ETH Zurich and Università di Bologna: PULPino microcontroller system, 2020, available online: *https://github.com/pulp-platform/pulpino*