LIFTING: an Open-Source Logic Simulator

Alberto Bosio, Giorgio Di Natale

LIFTING: an Open-Source Logic Simulator

A. Bosio, G. Di Natale

Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier Université Montpellier II / CNRS

UMR 5506 161 rue Ada, 34392 Montpellier Cedex 5, France

{bosio, dinatale}@lirmm.fr ; http://www.lirmm.fr

Abstract

This work presents LIFTING, an open-source simulator able to perform both logic and fault simulation for stuck-at faults and single event upset (SEU) on digital circuits described in Verilog.

1. Introduction

From the test point of view, logic and fault simulations are very important steps. While logic simulation aims at simulate the behavior of a circuits, when some stimuli are applied, fault simulation is performed in order to evaluate the test quality by determining its fault coverage with respect to a given fault model.
This work presents LIFTING (LIRMM Fault Simulator), an open-source simulator able to perform both logic and fault simulation for single/multiple stuck-at faults and single event upset (SEU) on digital circuits described in Verilog. LIFTING is based on an event-driven logic simulation engine and performs a fault injection fault simulation. Compared to existing tools, LIFTING provides several features for the analysis of the fault simulation results, meaningful for research purposes. Moreover, as an open-source tool, it can be customized to meet any user requirements especially thanks to its objects oriented architecture [1].
Experimental results show how LIFTING has been exploited on field research. Eventually, execution time for large circuit simulations is comparable to the one of commercial tools.

2. Fault simulator architecture

Fig. 1 sketches the object oriented fault simulator architecture.

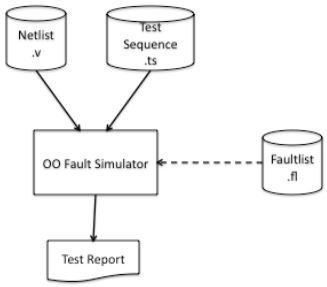

Figure 1: Fault simulator architecture

The Fault Simulator reads two mandatory files: the netlist of the circuit described in verilog (.v), and the input test sequence described in a proprietary format (.ts). An optional third input file is the list of faults that need to be simulated (.fl). In case this file is not specified, LIFTING creates the complete fault list w.r.t. the considered fault model (for instance, for the stuck-at fault model, it includes the entire possible faults of the circuit). Possible fault models are the single/multiple stuck-at and the Single Event Upset (i.e., bit flip in a storing cell, like flip flops, latches, memory elements). The results of the fault simulation are stored in the test report file.
We exploit the benefits of the object oriented programming in order to build a very flexible and easy to customize tool.

3. Experimental results

In this section we present some experimental results obtained by using the proposed LIFTING fault simulation tool. The aim of these experiments is to show the benefits carried out by using LIFTING w.r.t. commercial tool. Two cases are here presented. The first one aims at provide the feasibility of LIFITNG in terms of required CPU time when performing logic simulation only. The second one highlights the fault simulation facility and show how LIFITNG could be used to obtain more information w.r.t. commercial tool.
Let us introduce the two cases on which LIFTING has been used. As a first test bench we consider a complex SoC [2] composed of three cores:

- an 8-bit microcontroller
- a 64Kx8 bit SRAM memory
- a 16x16 parallel multiplier.

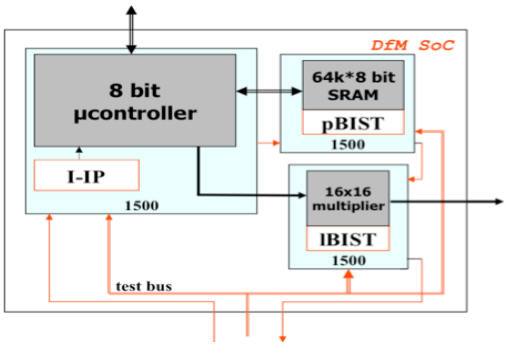

Figure 2: SoC architecture

The architecture of the considered SoC is shown in Fig. 2. The result of the process is actually a simple fault-free simulation, but the usage of the extended symbolic algebra

and the flexibility feature the tool provides enable accurate analysis for delay testing.
The evaluated test set includes 18 self-test programs, 1 of them targeting 16x16 multiplier coverage and 1 executing a memory test (approaches like this one are normally referred soft-BIST). In the past, these test programs were generated and evaluated by means of commercial tools for the SA fault model. Table I reports the test set characteristics and required logic simulation times.

	Patterns (#)	Simulation time (h:m:s.c)
Test program 01	6,275	1:54.87
Test program 02	6,671	2:08.55
Test program 03	7,639	1:48.94
Test program 04	8,387	2:07.11
Test program 05	8,519	2:26.07
Test program 06	10,543	2:47.46
Test program 07	14,151	4:25.23
Test program multiplier	21,939	4:37.70
Test program 08	22,379	6:45.50
Test program 09	12,787	4:31.44
Test program 10	25,811	7:40.68
Test program 11	44,423	10:16.64
Test program 12	44,907	8:07.90
Test program 13	54,059	18:02.25
Test program 14	55,775	33:11.87
Test program 15	73,903	26:27.00
Test program 16	96,299	20:48.89
Test RAM (soft-BIST)	279,559	1:18:42.14

Table 1: Logic simulation results

Experiments were run on an Intel Pentium M working at 1.73 GHz while primary memory occupation for the SoC memorization structure is about 200 MB.
The second test bench has been a circuit implementing the Advanced Encryption Standard (AES) [3]. For this work we propose the following self-test procedure:

1. Encrypt an initial message M_0 into M_1=Encryption(M_0)
2. Repeat n times : M_{i+1} = Encryption(M_i)
3. Compare the final cipher M_n with the expected one E(E(E(…..E(M)…)). If they differ, the circuit is faulty otherwise it is correct.

In other words, the result of an encryption is used as the next test vector. We simulated the circuit with a particular initial value and then we just let the circuit encrypts its own output for 2500 clock cycles.
The circuit counts up to 10000 logic cells. We used a target 130nm CMOS technology provided by ST [4]. The simulator needed 8 hours to fully simulate all the input patterns. Using LIFTING fault simulator provided us the possibilities to understand the minimum number of clock cycles required to fully test the circuit (with commercial tools we were not able to automate this process).

4. Conclusion

In this work we presented LIFTING (LIRMM Fault Simulator), an open-source simulator able to perform both logic and fault simulation for single/multiple stuck-at faults and single event upset (SEU) on digital circuits described in Verilog.
LIFTING provides several features for the analysis of the fault simulation results, meaningful for research purposes. Moreover, as an open-source tool, it can be customized to meet any user requirements especially thanks to its objects oriented architecture.
The source code of the tool can be downloaded by the following internet address:
http://www.lirmm.fr/~dinatale/LIFTING/

5. References

[1] A. Bosio, G. Di Natale, "LIFTING: a Flexible Open-Source Fault Simulator", IEEE Asian Test Symposium, 2008, pp. 35-40.
[2] D. Appello et al. “On the Automation of the Test Flow of Complex SoCs”, IEEE VLSI Test Symposium, 2006, pp. 166-171
[3] M. Doulcier, M.L. Flottes, B. Rouzeyre, “AES-Based BIST: Self-Test, Test Pattern Generation and Signature Analysis”, 4th IEEE International Symposium on Electronic Design, Test and Applications, 2008 (DELTA 2008), pp. 314-321.
[4] http://www.st.com

Acknowledgement

The authors would like to thank Paolo Bernardi from Politecnico di Torino (Italy) for his help during the development of the fault simulator.