

Speeding up Fault Simulation using Parallel Fault Simulation

Jiahua Fan, Zhifeng Zhang a*

Department of Electronic Science and Technology, Tongji University, Shanghai 200092, China

Abstract

In this paper, a novel approach is introduced on accelerating the fault simulation speed on field programmable gate array (FPGA). The approach is based on parallel simulation methodology. More than one faulty circuit is handled in the fault simulation system, but the relative area overhead is low and it will accelerate the simulation process. A new metrics – Speedup relative to the Ratio of Hardware Overhead (SRHO) is introduced, by which the experimental results are evaluated. Experimental results in terms of simulation time, hardware overhead and SRHO for ISCAS-85 benchmark circuits are compared to a previous work to show its advantage.

© 2011 Published by Elsevier Ltd. Open access under [CC BY-NC-ND license](https://creativecommons.org/licenses/by-nc-nd/4.0/).
Selection and/or peer-review under responsibility of [CEIS 2011]

FPGA; SRHO; Fault Simulation; Parallel Simulation

1. Introduction

The results [1][2] show that the reliability of integrated circuit will be reduced with the shrink of CMOS size. The fault verification becomes more and more important.

All the methods on fault verification can be divided into two types. One is simulating on a PC or PCs while another is on actual hardware. For the first kind, there are a lot of parallel algorithm for fault simulation with software simulation [3][4]. It uses the multi-threading feature of a processor or multi-processors to achieve parallel simulation. It cannot perform full speed simulation, so it is time-consuming. For the second kind, the circuits are verified on the ATE [5] or FPGA [6]. The conversional method is

* Corresponding author. Tel.: +86-21-69589122; fax: +86-21-69589122.
E-mail address: zhangzf@tongji.edu.cn

testing the circuit on the ATE. The main usage of it is to find whether the IC works or not. It uses a machine to give the input vector to the IC and compare the output vector to detect a fault. It only can be used when the IC is produced. It can perform at full speed but the machine that it uses is very expensive. Nowadays, it becomes more and more popular to use FPGA to do the verification. It has several advantages. First, the circuit is easily modified. Second, it can perform at full speed. Third, the cost is very low because of the reusability of FPGA. The method which uses FPGA can be divided into two types. One is using partial reconfiguration [7] while another is using extra-circuit to insert fault [8]. The first method is area saving and time-consuming relative to the second one. In this paper, the second method will be used.

Two techniques, which are a new division method of circuit partitioning and a new type of the fault activation scan chain, are proposed to improve the simulation speed. The new division method of circuit partitioning makes the parallel simulation more efficient and saves a lot of hardware resource. The new type of fault activation scan chain is proposed to save the hardware resource. Experimental results for ISCAS-85 benchmark circuits show that our scheme is better than the previous work from the viewpoints of simulation time, hardware overhead and Speedup relative to the Ratio of Hardware Overhead (SRHO).

2. Parallel Fault Simulation

2.1. Sensitization Path

Path sensitization at the logic gate level of representation is currently the preferred ATPG method [10]. This is also used to analyze the circuit partition method in this paper. A sensitization path for a fault means a path through which you can control the input vectors and get the different value from the output for a fault.

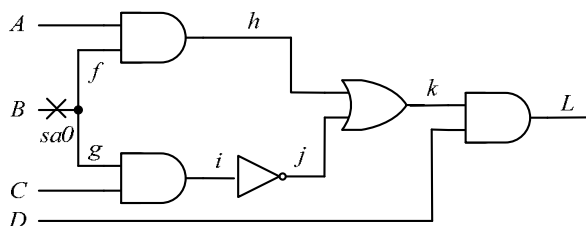


Fig. 1. A combinational circuit example for path sensitization.[10]

As the Fig. 1 shows, we can find two paths for B stuck-at-0. One is along the path $B \rightarrow f \rightarrow h \rightarrow k \rightarrow L$ while another is along the path $B \rightarrow g \rightarrow i \rightarrow j \rightarrow k \rightarrow L$. The only valid sensitization path is the path $B \rightarrow g \rightarrow i \rightarrow j \rightarrow k \rightarrow L$ for the value of j is more difficult to set it as 1.

2.2. Circuit Partition and the fault injection scan chain

In [9], there is some definition of the circuit partition. They are described as followed.

- Separate each primary output
- Trace the circuit from the primary outputs to the primary input
- Each sub-circuit should not include the fanout stem unless all the fanout branches are contained in the sub-circuit.

In this paper, the circuit partition should meet these demands first. The faulty circuit must be separated into several parts to support the parallelism of the simulation system. There are two kinds of method to divide the circuit: parallel or vertical to the sensitization path.

In this paper, the second one is a good choice. One input vector can active all the related faults along the sensitization path. For example, as the dotted line with an arrow in Fig. 2 shows, we can find an input vector [0, 1, 1, 1] which can active single stuck-at faults along the sensitization path B sa0 → i sa0 → j sa1 → k sa1 → L sa1. Fig. 2(a) shows the first method while Fig. 2(b) shows the second method. If we use the scan chain along this sensitization path to active the faults one by one. For the circuit in Fig. 2(a), it takes 6 clocks to detect all the faults along the sensitization path while it takes only 3 clocks to detect all the faults along the sensitization path for the circuit in Fig. 2(b). The second method is 2 times faster than the first method.

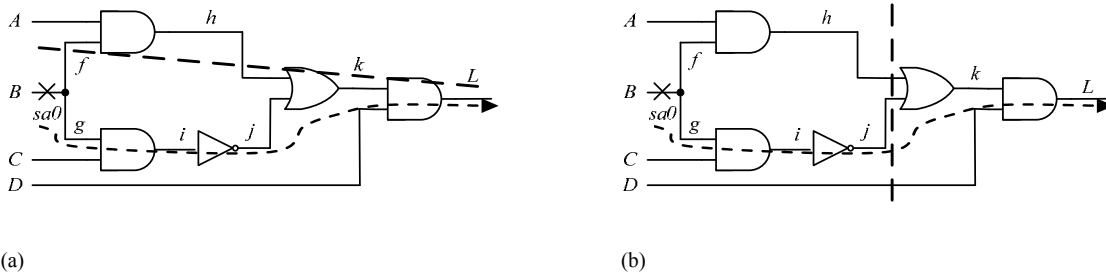


Fig. 2. (a) Parallel to the Sensitization Path; (b) Vertical to the Sensitization Path.

In this paper, the structure of the scan chain in [9] is rearranged. There is no need to keep the entire scan chain in each faulty circuit. The scan chain is divided to several parts according to the division of the faulty circuit. Then it can save hardware resources and support the parallelism of the simulation system.

3. Architecture of the System

In this paper, the XUPV5 Development Board, which contains an xc5vlx110t FPGA, is used as the main verification device. All the results are verified on this platform. Fig. 3 shows the architecture of the whole system used in this paper.

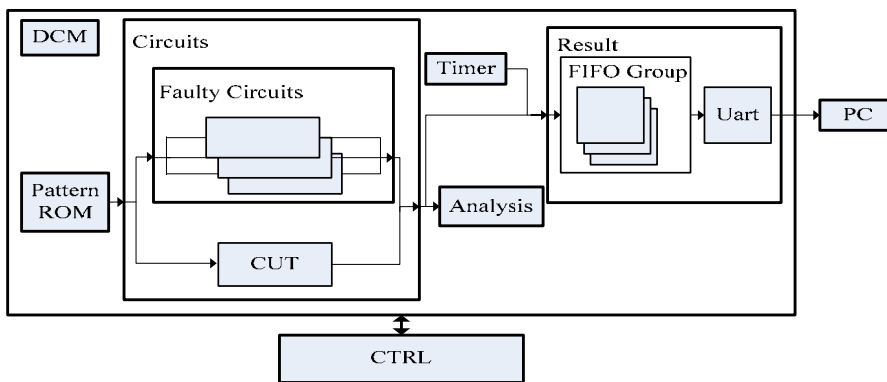


Fig. 3. Architecture of the System

The system is divided into 7 parts. The Timer module is used to generate the time that the simulation process spends. The DCM module is used to generate the operating clock. It will generate a 50MHz clock for other parts to use. The test vectors for CUT and faulty circuits are stored in the Pattern ROM module. The CTRL module provides the control signals for the system and the fault location for the Result module. The Circuit module contains the CUT and the faulty circuits. The faulty circuits are modified manually. The Analysis module is used to analyze the result that comes from the Circuit module and returns the result to the CTRL module. The Result module provides the location information of the detectable faults and the time that all the process spends for UART module to print them on the screen of the host PC. The number of FIFOs equals to the number of the faulty circuits.

4. Experimental Result

Two ISCAS-85 circuits are used in this paper and the faulty circuits are divided into two parts. Table 1 shows the speedup of the method in this paper over that in [9]. The speedup is about 1.65.

Table 1. Simulation Time

ISCAS-85 benchmark circuit	Simulation Time(s)		Speed-up
	This paper	[9]	
C17	0.0000014	0.0000024	1.71
C432	0.0002354	0.0003848	1.63

The circuits in this paper are larger than those in [9] for there are more faulty circuits in this paper than the original one. First, Table 2 shows that the circuits in this paper occupy more LUTs than those in [9]. But for the whole simulation system, C432 has less LUTs in this paper than that in [9]. The reason is that there is a 90-node sub-circuit. When the circuit is divided into two parts, the part which does not have the 90-node sub-circuit has far less logic in the Result module than the related part in the original circuit. Second, Table 2 shows the whole system in this paper has just a little bit more LUTs than that in [9]. The reason is that the Circuit module contains less than 10% LUTs of the whole system and the other parts of these two methods are similar.

Table 2. Simulation Time

ISCAS-85 benchmark circuit	Number of LUTs			
	C17		C432	
	Circuit Only	ALL	Circuit Only	ALL
[9]	12	319	246	4361
This paper	13	374	321	3214
Ratio of Hardware Overhead	1.08	1.17	1.30	0.74

To show the advantage of the method in this paper, a new metrics, called Speedup relative to the Ratio of Hardware Overhead (SRHO), is introduced. This metrics represents that the speedup can be obtained by per ratio unit of hardware overhead. The expression of this metrics is

$$SRHO = \frac{\text{Speedup of this paper over [9]}}{\text{Ratio of Hardware Overhead}} \quad (1)$$

Table 3 shows the result of SRHO. As Table 3 shows, all the SRHO is larger than 1.25.

Table 3. Simulation Time

ISCAS-85 benchmark circuit	SRHO	
	Circuit Only	ALL
C17	1.58	1.46
C432	1.25	2.20

5. Conclusions

In this paper, the parallel fault simulation is conducted on FPGA. Besides this, a new circuit partition method and a new division of fault injection scan chain are used to accelerate the simulation process. The hardware overhead is a little bit more than the related previous work but a speed up of 1.65 is achieved. The new metrics, SRHO, shows that a speedup of more than 1.25 can be obtained by per ratio unit of hardware overhead. In the future work, more circuits and more partitions will be tested.

Acknowledgements

The work was supported by the National Natural Science Foundation of China under Grants 60903033.

References

- [1] Srinivasan J, Adve SV., Bose P, Rivers JA. The impact of technology scaling on lifetime reliability. *2004 International Conference on Dependable Systems and Networks*; 2004, p. 177-86
- [2] Srinivasan J, Adve SV., Bose P, Rivers JA. Lifetime reliability: toward an architectural solution. *IEEE MICRO* 2005; **25**(3): 70-80.
- [3] Varshney AK, Vinnakota B, Skuldt E, Kelle B. High performance parallel fault simulation. *Proceedings 2001 IEEE International Conference on Computer Design: VLSI in Computers and Processors*; 2001, p. 308-13.
- [4] Narayanan V, Pitchumani V. A massively parallel algorithm for fault simulation on the connection machine. *26th ACM/IEEE Design Automation Conference*; 1989, p. 734-7
- [5] Hashempour H, Meyer FJ, Lombardi F. Analysis and measurement of fault coverage in a combined ATE and BIST environment. *IEEE Transactions on Instrumentation and Measurement* 2004; **53**(2): 300-7. .
- [6] Kafka L, Novak O. FPGA-based fault simulator. *2006 IEEE Design and Diagnostics of Electronic Circuits and Systems*; 2006, p. 272-6
- [7] Kubalik P, Kvasnicka J, Kubatova H. Fault injection and simulation for fault tolerant reconfigurable duplex system. *Proceedings of the 2007 IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems*; 2007, p. 357-60.
- [8] Civera P, Macchiarulo L, Rebaudengo M, Reorda MS, Violante M. Exploiting circuit emulation for fast hardness evaluation. *IEEE Transactions on Nuclear Science* 2001; **48**(6): 2210-16.
- [9] Lu S, Chen Y, Wu C, Huang S. Speeding-up Emulation-Based Diagnosis Techniques for Logic Cores. *IEEE Design & Test of Computers*; 2011
- [10] Bushnell ML, Agrawal VD. *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. New York, Boston, Dordrecht, London, Moscow: Kluwer Academic Publishers; 2002, p.162-3