ISSN No:-2456-2165

Fault Injection and Fault Detection Technique for Sram Based FPGA

Kajal Ambhore
PG Student,
Department of Electronics and Communication
Tulsiramji Gaikwad-Patil
College of Engineering & Technology,
Nagpur, India

Rahul Dhuture
Professor, Department of Electronics and Communication,
Tulsiramji Gaikwad-Patil
College of Engineering & Technology,
Nagpur, India

Rohini Pochhi
Professor, Department of Electronics and Communication,
Tulsiramji Gaikwad-Patil
College of Engineering & Technology,
Nagpur, India

Abstract:- Commercial FPGAs provide high computational power in extreme conditions such as space applications. Transient errors, such as singular upsets, which can be minimized with the well methods, are not as extreme as permanent faults, since they can place the application at risk if the FPGA fails. We provide hardware fault detection circuit and provide response whether the fault is occurred. The circuit is designed for fault injection and tolerance.

Keywords:- *Fpga, Fault Detection.*

I. INTRODUCTION

Computerized frameworks are electronic framework that are utilized in each mechanical space from rail line, car, interaction, and mechanization to correspondence innovations. Hardware descriptive language (HDL) are utilized to portray the design and conduct of computerized frameworks for complex installed frameworks, for example, application-explicit incorporated circuits, chip, and programmable rationale gadgets. Testing of these advanced circuits is utilized to check if the entire framework or a sub-segment strays from its plan detail [1]. Testing of circuits planned utilizing HDLs has been a functioning. research region for the most recent few decades. A deficiency is said to have happened in an advanced framework if the conduct of the circuit digresses from its ostensible conduct [2]. A computerized framework with a shortcoming will thusly not show comparative usefulness as a circuit without a flaw. Issue models [1,3] are utilized at various degrees of plan deliberations for displaying the intelligent correspondence to a physical defect(e.g., social, practical, underlying deficiency models). For instance, at the social plan level, an if build can bomb when one bunch of articulations is constantly executed and this sensible conduct identifies with a real deformity in the real equipment. Certain test sources of info can be created to recognize these legitimate contrasts between the right and the broken circuit. The headways of VLSI (Very Large-Scale Integrated Circuit) [4] innovation has empowered the assembling of complex computerized rationale circuits on a solitary chip. This postures numerous difficulties as far as both utilitarian and non-practical perspectives that should be viewed as when testing such frameworks utilizing HDLs. Creating a computerized framework starts with the detail of its plan utilizing a HDL and closures with assembling and transportation the general framework. This cycle includes reenactment, combination, testing, and support of a particularly advanced framework. A considerable lot of the methodologies proposed in scholarly world for test age have been received in various Electronic Design Automation (EDA) devices [5] during the most recent few decades.

II. BACKGROUND

In this part, we present advanced circuit testing an equipment depiction dialects.

A. Computerized Circuits

Computerized circuits can be planned utilizing interconnected legitimate components. A couple of instances of consistent components are AND/OR entryways, registers, inverters, flip failures, and others [6]. An advanced circuit should have the option to deal with limited esteemed discrete signs. A notable computerized circuit is a microchip utilized in advanced PCs. Computerized frameworks are essential for a more extensive classification of discrete frameworks. Stefan [7] characterized computerized frameworks in a scientific categorization of orders dependent on criticism circles, as follows:, zero-request frameworks containing the combinational circuits, 1-request frameworks containing the memory circuits, 2-request frameworks containing the automata, 3-request frameworks containing the processors, and n-request frameworks containing self-arranging circuit conduct. In Figure 1, we show an illustration of a choice circuit, a rudimentary multiplexer for two contributions to various structures from the sensible schematic to the underlying depiction in an equipment portrayal language.

ISSN No:-2456-2165

There are numerous different circuits that can be formed for programmable circuits. For instance, locks are memory components that are set off by a level delicate empower signal. Planning a computerized framework implies the utilization of such portrayals and determinations to design an unpredictable framework that considers the inside state and the development of yields in the entire framework.

HDLs [8] are dialects used to depict the equipment capacity of an advanced framework. HDLs can be utilized to execute and portray the capacity of an equipment at different degree of reflections (as demonstrated in Figure 1). HDLs are generally utilized dialects to oversee complex advanced frameworks and can be utilized to plan and reproduce a circuit before execution. Recreation helps in finding plan blunders. VHDL, Verilog, and System Verilog are three instances of equipment depiction dialects and are considered in this planning concentrate because of their modern materialness and prevalence [9]. The figure 1 depicts the primary portrayal that indicates every one of the subtleties at the degree of rudimentary entryways.

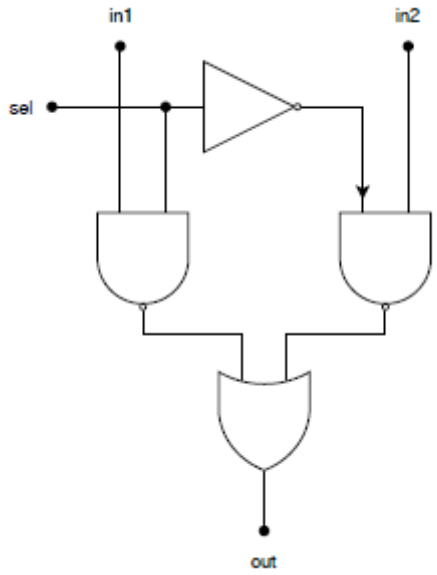

Figure 1. An example of a selection circuit showing the logic schematic for the elementary selector

Testing of Digital Systems

Building up an advanced circuit begins with a plan particular in a significant level plan language (e.g., Verilog) and finishes with assembling the framework. Diverse testing measures are engaged with every improvement stage. Testing and troubleshooting of the plan utilizing the HDL model as contribution to a test system can guarantee the architect that the portrayal of the plan meets the determination. The essential objective of this progression is identified with plan usefulness, and definite planning and actual flaws are not tended to. Subsequent to getting a mimicked and tried format, producing testing is done on the actual circuit against assembling abandons (e.g., semiconductor surrenders, broken wires). In this paper, we center around testing advanced frameworks addressed in a HDL (i.e., HDL-based Testing).To check a computerized framework plan, a test module should be intended to produce upgrades for the contribution of the System Under Test (SUT), execute the experiments, and screen the sources of info and the yields of the circuit. Wang [6] characterizes testing of computerized circuits as the way toward recognizing flaws in a framework and finding such blames to encourage their maintenance. A bunch of test upgrades called test vectors or examples are applied to the contributions of the SUT. The yields are dissected by gathering the reactions and contrasting it with the normal reactions. Assuming the real conduct goes amiss from the normal conduct, a shortcoming has been recognized. In Figure 2, we show the overall standards behind test age for computerized frameworks. Testing of computerized circuits can be grouped dependent on the sort of gadget being tried, the testing gear utilized, and the motivation behind testing. Diverse testing errands can be performed utilizing the HDL testing builds and abilities. An illustration of a test age technique is the arbitrary age of test vectors [10], applying them to the SUT, and choosing these dependent on their viability in distinguishing flaws. At times, a specific test gadget can be utilized to apply test vectors on the real test interface [11]. Virtual test stages are typically executed as HDL test seats that are equipped for controlling the SUT regarding perceptibility. A primary quality of these test age approaches is the HDL model inclusion standards used to manage the quest for experiments. Most inclusion measurements for HDLs depend on syntactic properties of a utilitarian plan. For instance, branch and way inclusion are utilized for discovering test arrangements covering all branches and ways in a HDL plan. HDL inclusion rules is additionally a proportion of register-move level ampleness that can likewise be applied at door level for estimating test sufficiency at the execution level.

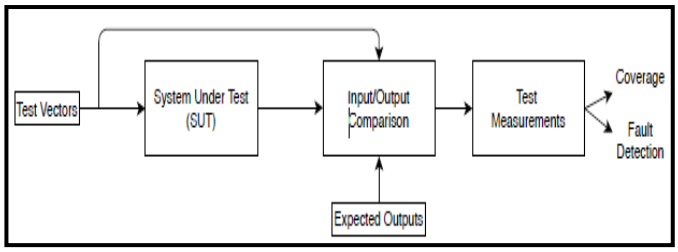

FIG 2:- AN OVERVIEW OF THE TEST GENERATION PROCESS FOR A SUT.

Many different test generation approaches for HDLs have been proposed at different levels of design abstractions: system, behavioral, register transfer, logical and physical levels.

III. ERROR DETECTION

The number of transistors in a single chip is increasing as CMOS technology continues to scale. Chip multiprocessors (CMPs) are a cost-effective way to make use of a chip's large number of transistors. According to many reports, high density integration makes modern processors vulnerable to temporary or permanent faults. However, as the temperature rises and the decrease in threshold voltages.

ISSN No:-2456-2165

The rate at which transient errors occur is expected to increase exponentially, making it one of the most critical issues in the design of future generation high-performance microprocessors.

IV. PROPOSED FAULT DETECTION CIRCUIT

The design of circuit which can detect the fault and test the circuit operation. Figure 3 shows the fault detection circuit. The circuit us designed which can detect the fault and provide the response to the output. The input is given by standard test pattern generator. The circuit goes into faulty condition. The fault is arising in multiplier circuit. The other output response which is correct in nature are compared. The output result provides the solution whether the circuit is in good or bad condition, The detection of fault and provide the circuit response.

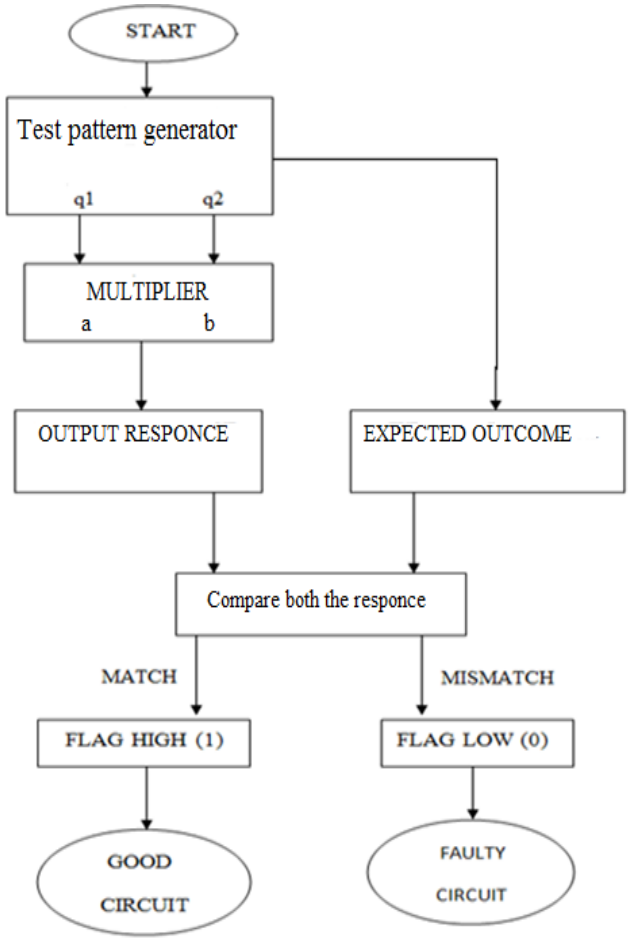

Figure 3 Fault detection circuit

V. CONCLUSION

The detection of fault is very important concept, There are various factors occurred which may causes fault. The design of proposed circuit is discussed, Some common fault occour are stuck at 0 or stuck at 1 fault. The detection of fault is plays a very important role. The technique can be implemented in any HDL language.

REFEARNCES

[1]. GNavabi, Z. Digital System Test and Testable Design; Springer: Berlin, Germany, 2011; pp. 97814419–97875485.
[2]. Lala, P.K. An introduction to logic circuit testing. Synth. Lect. Digit. Circuits Syst. 2008, 3, 1–100. [CrossRef]
[3]. Jha, N.K.; Gupta, S. Testing of Digital Systems; Cambridge University Press: Cambridge, UK, 2003.
[4]. Pla, V.; Santucci, J.F.; Giambiasi, N. On the modeling and testing of VHDL behavioral descriptions of sequential circuits. In Proceedings of the EURO-DAC 93 and EURO-VHDL 93-European Design Automation Conference, Cch Hamburg, Germany, 20–24 September 1993; pp. 440–445.
[5]. MacMillen, D.; Camposano, R.; Hill, D.;Williams, T.W. An industrial view of electronic design automation. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 2000, 19, 1428–1448. [CrossRef]
[6]. Wang, F.C. Digital Circuit Testing; Academic Press: Cambridge, MA, USA, 1991.
[7]. Stefan, G. Loops & Complexity in Digital Systems; Lecture Notes on Digital Design; 2016. Available online :http :// users .dcae. pub.ro /~gstefan/2ndLevel/teachingMaterials/0BOOK.pdf (accessed on 30 May 2020).
[8]. Ghosh, S.K. Hardware Description Languages: Concepts and Principles; Wiley-IEEE Press: Hoboken, NJ, USA, 1999.
[9]. Golson, S.; Clark, L. Language Warsin the 21st Century: VerilogversusVHDL–Revisited; Synopsys Users Group (SNUG): Beijing, China, 2016.ection of fault
[10]. Haghbayan, M.; Karamati, S.; Javaheri, F.; Navabi, Z. Test pattern selection and compaction for sequential circuits in an HDL environment. In Proceedings of the 2010 19th IEEE Asian Test Symposium, Shanghai,China, 1–4 December 2010; pp. 53–56.
[11]. Ahmad, A.A.S.; Brereton, P.; Andras, P. A systematic mapping study of empirical studies on software cloud testing methods. In Proceedings of the 2017 IEEE International Conference on Software Quality, Reliability–29 July 2017; pp. 555–562.