

# FPGA Implementation Of ALU Using BIST

Navdeep Kaur<sup>1</sup>, Neeru Malhotra<sup>2</sup>

<sup>1,2</sup> DAV Institute of Engineering and Technology, Jalandhar, Punjab, India.

## Abstract

The increasing growth of sub-micron technology has resulted in the difficulty of VLSI testing. Test and design for testability are recognized today as critical to a successful design. Field Programmable Gate Arrays (FPGAs) have been used in many areas of digital design. Because FPGAs are reprogrammable, faults can be easily tolerated once fault sites are located. This paper presents a new effective Built-In Self-Test (BIST) scheme that achieves 100% fault coverage with low area overhead, and without any modification of the circuit under test (CUT), i.e., no test point insertion. Finally designed circuit is implemented on the FPGA Spartan-3e kit.

## Introduction

Field programmable Gate Arrays (FPGAs) are pre-fabricated silicon devices that can be electrically programmed in the field to become almost any kind of digital circuit or system. For low to medium volume productions, FPGAs provide cheaper solution and faster time to market as compared to Application Specific Integrated Circuits (ASIC) which normally require a lot of resources in terms of time and money to obtain first device. FPGAs on the other hand take less than a minute to configure and they cost anywhere around a few hundred dollars to a few thousand dollars. Also for varying requirements, a portion of FPGA can be partially reconfigured while the rest of an FPGA is still running.

Normally FPGAs comprise of:

- Programmable logic blocks which implement logic functions.
- Programmable routing that connects these logic functions.
- I/O blocks that are connected to logic blocks through routing interconnect and that make off-chip connections.

A generalized example of an FPGA is shown in Fig.1 where configurable logic blocks (CLBs) are arranged in a two dimensional grid and are interconnected by programmable routing resources. I/O blocks are arranged at the periphery of the grid and they are also connected to the programmable routing interconnect. The “programmable/reconfigurable” term in FPGAs indicates their ability to implement a new function on the chip after its fabrication is complete. The reconfigurability/ programmability of an FPGA is based on an underlying programming technology, which can cause a change in behavior of a pre-fabricated chip after its fabrication

Many methods have been proposed to test FPGAs. In some works, the circuits under consideration are programmed FPGAs, in which logic circuits have been implemented. Since an FPGA can be programmed in many different ways, this method is not applicable to manufacturing time testing, as we do not know the final configuration. Testing faults in general FPGAs has been proposed by many researchers. In these methods, the FPGA under test is not mapped to a specific logic function. As a result, multiple test sessions are usually required, with each session dealing with one configuration [1].

BIST for random logic is becoming an attractive alternative in IC testing. However, recent advances in IC process technology will certainly lead to more widespread use of logic BIST since external testing is becoming more and more difficult and costly. This is confirmed by the ITRS (International Technology Roadmap for Semiconductors) statement that by 2014 it may cost more to test a transistor than to manufacture a transistor unless techniques like logic BIST are employed. The use of logic BIST for the digital modules, analog BIST for the analog modules, and memory BIST for the memory modules, provides that a single, low-cost external tester is sufficient [2]. Figure 1 shows the generic FPGA architecture for VLSI design.

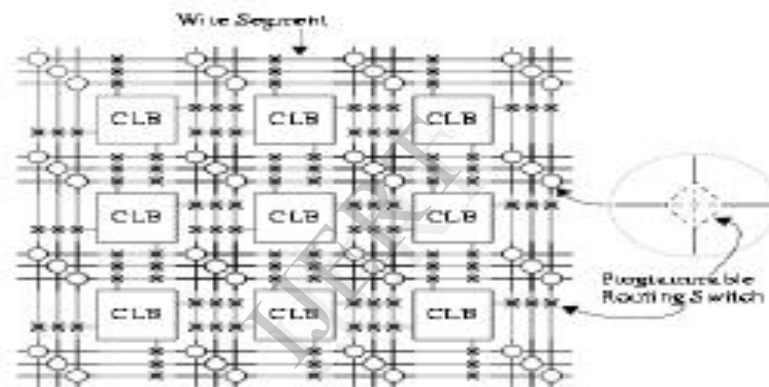


Figure 1: Generic FPGA

An arithmetic and logic unit (ALU) is a digital circuit which makes arithmetic and logical operations. The ALU is an important block of the core unit of a computer. The processors found inside modern CPUs and graphics processing units (GPUs) accommodate very powerful and very complex ALUs, a single component may contain a number of ALUs. An ALU loads data from input registers, an external Control Unit then tells the ALU what operation to perform on that data and then the ALU stores its result into an output register. The Control Unit is responsible for moving the processed data between these registers, ALU and memory.

This paper focuses on VHDL implementation of ALU with BIST capability using LFSR techniques on Field Programmable Gate Array (FPGA) technology.

## BIST Architecture

The BIST architecture requires the addition of three hardware blocks to a digital circuit: a pattern generator, a response analyzer and a test controller. Examples of pattern generators are a ROM with stored patterns, a counter and a linear feedback shift register. LFSR is constructed using flip-flops connected as a shift register with feedback paths that are linearly related using XOR gates. An LFSR can be used for generation of pseudo-random patterns, polynomial division, response compaction etc. [2]. LFSR is more popular for implementation of both TPG and ORA due its compact and simple structure. A typical response analyzer is a comparator with stored responses or an LFSR used as a signature analyzer. Traditional chip testing deals with fault detection only, while fault diagnosis is often conducted at the system level. This is because components in the chip cannot be repaired. However, faults in FPGAs can be easily tolerated by not including faulty elements in the final circuit. Therefore, FPGA chips with faults can still be used if we can identify the fault sites. Our method is also based on the BIST technique, which means that the testing process is conducted by the chip itself, and the requirement for external ATE support is limited. Figure 2 shows the blocks of BIST.

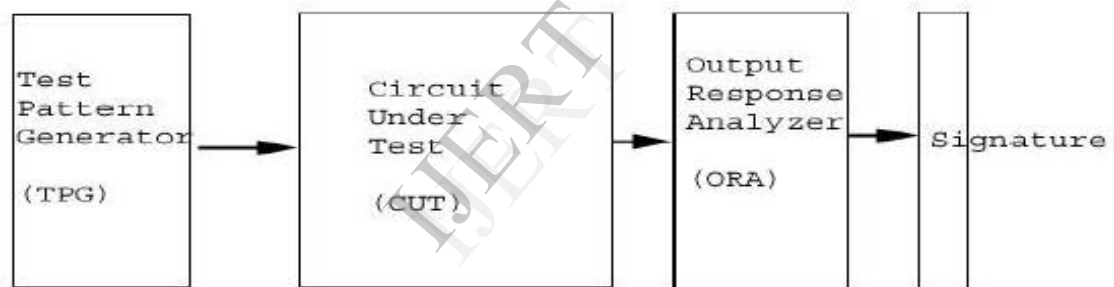


Figure 2 Blocks of BIST

## RELATED WORK

Many researchers have been working on the BIST implementation and FPGA related research some of them are given below:

Noorbasha et al [1] had discussed about fault detection and fault diagnosis techniques for FPGA CLBs. The most of the discussion will be made using Configurable Logic Block (CLB) instead of whole FPGA for simplicity. Nur A. Toubia [2], had concluded about Synthesis Techniques for Pseudo-Random BUILT-IN SELF-TEST. Crouch et al. [7] had discussed the main point is Built-In-Self Test (BIST) architecture for sequential circuits based on cellular Automata (CA). Yamani et al. [4], the BIST technique incorporated into the UART design before the overall design is synthesized by means of reconfiguring the existing design to match testability requirements. Hegde Suma T.et.al [5] emphasized on Design and Implementation of ALU using Redundant Binary Signed. FAGOT et al. [3], had demonstrating the effectiveness of the proposed approach in terms of area overhead, fault coverage and test sequence length.

## ALU IMPLEMENTATION WITH BIST FEATURE

A software tool is used which automatically generates built-in self-test blocks into VHDL models of ita circuits by giving the suitable values of initial seed and primitive polynomial in TPG block. The code is generated for BIST, when we insert the code of ALU into CUT; a generated code is synthesized in the Xilinx web pack 12.4 for the Spartan 3e devices. The hardware summary is obtained for each method implementation log file of Xilinx 12.4 project navigator .The RTL view of the ALU with BIST capability is shown in Figure3.

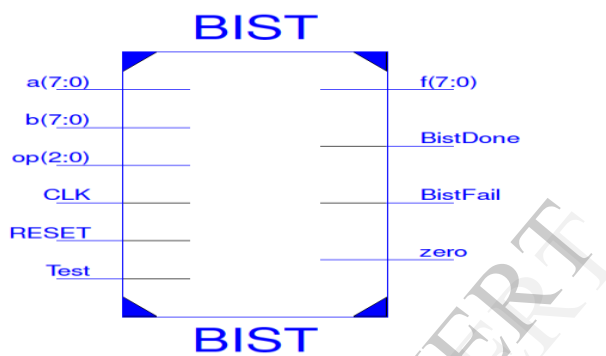


Figure 3: RTL view of the BIST

It has the following parts: Signal Register, Comparator, Controller, MISR, MUX, ALU and Test Pattern Generator. All these parts form the BIST. In this MUX is used to select operation, then operation is selected and corresponding operation is performed by the ALU. After this time Test Pattern is also selected by the MUX. Test Pattern output and ALU output is compared by the comparator after passing through the MISR. All the functions are controlled by the controller. Controller will select weather to test the ALU depending upon TEST signal. Controller also provides BIST Fail or BIST Pass depending upon the testing is successful or not. RTL view of the BIST with proper placing and routing. It has following pins: - inputs for input signals, operation input for selecting the oration like AND, OR, addition or subtraction etc. RESET signal is used to reset the bist operations. TEST signal will start the testing. If TEST =1, then there will be the testing of ALU or if BIST=0 then BIST will not work for testing. IF the testing is done then bistdone=1 otherwise bistfail=1.

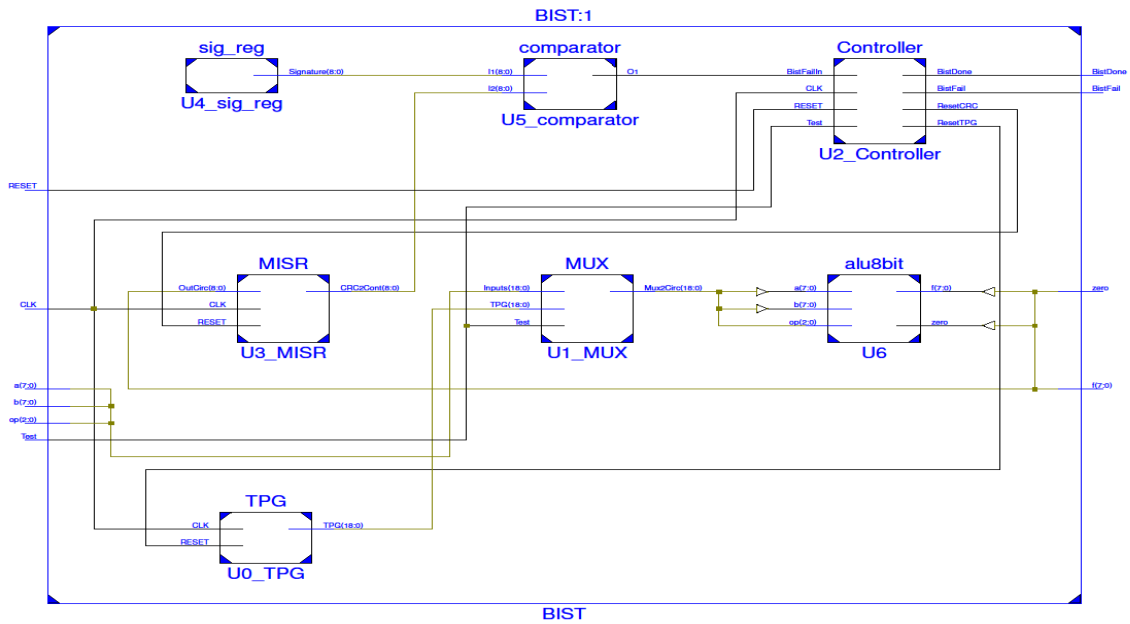


Figure 4: RTL view of the ALU with BIST

### Simulation of Complete BIST for ALU

In figure 5 simulation of complete BIST for ALU are shown, which is the combination all above four sections that are mux, ALU, TPG and controller section. "bist/a", "bist/b", "op", "bist/test", clk and reset are the inputs. Signal f, "bistdone", "bistfail and "zero" are the outputs.

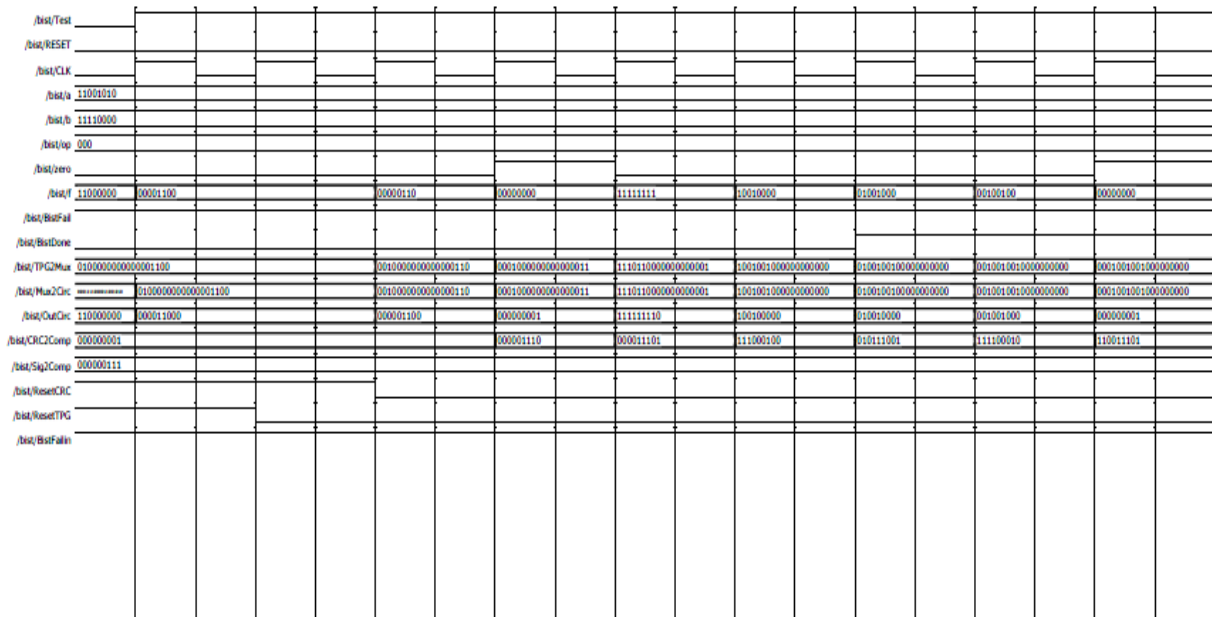


Figure 5: Simulation of complete BIST for ALU

## 6. REFERENCES

- [1] Noorbasha. F, Harikishore K., Hemanth Ch., Sivasairam A., Vijaya Raju,” LFSR Test Pattern for Fault Detection and Diagnosis for FPGA CLB CELLS” International Journal of Advances in Engineering & Technology, Vol. 3, Issue 1, pp. 240-246 March 2012.
- [2] Fagot C., Gascuel O., Girard P. and Landrault C.,” A Ring Architecture Strategy for BIST Test Pattern Generation” c 2003 Kluwer Academic Publishers .Manufactured in The Netherlands. Received December 6, 2000; Revised December 3, 2002.
- [3] Mohd Yamani Idna Idris “A VHDL Implementation of UART design with BIST capability” Malaysian Journal of Computer Science, Vol. 19 (1), 2006.
- [4] Nur A. Touba,” Synthesis Techniques for Pseudo-Random BUILT-IN SELF-TEST” a Dissertation in partial fulfillment of the requirements for the Ph.d,1996.
- [5] Bushnell M. L. and Agrawal V. D. (2000) Essentials of Electronic Testing. Kluwer Academic Publishers.
- [6] Suma T.Hegde ,Dr.Siva Yellampalli And Nandeesh R.,” Design and Implementation of ALU using Redundant Binary Signed Digit”

International Conference on VLSI, Communication & Instrumentation (ICVCI) 2011 Proceedings published by International Journal of Computer Applications® (IJCA) 30.

- [7] C. H. Roth “Digital System Design Using VHDL” PWS Publishing Company, 1998
- [8] Crouch, A. (1999), “Design-for-test for Digital IC’s and Embedded Core Systems”, Prentice Hall, 347 pp.

IJERT