

---

# An Online Detection and Diagnosis BIST approach for Testing SRAM-based FPGAs using JBits

**Dr. Shilpa Shrigiri**

Professor (E&CE), SKSVMACET, Laxmeshwar,

## ABSTRACT

*This paper explores the Built-In Self Test (BIST) concepts to test the configurable logic blocks (CLBs) of Static RAM (SRAM) based FPGAs using Java Bits (JBits). The proposed technique detects and diagnoses single and multiple stuck-at faults in the CLBs while significantly reducing the time taken to perform the testing. Previous BIST approaches for testing FPGAs use traditional CAD tools which lack control over configurable resources, resulting in the design being placed on the hardware in a different way than intended by the designer. In this paper, the design of the logic BIST architecture is done using JBits 2.8 software for Xilinx Virtex family of devices. The test requires seven configurations and two test sessions to test the CLBs. The time taken to generate the entire BIST logic in both the sessions is approximately 77 seconds as compared with several minutes to hours in traditional design flow.*

**Keywords:** Configurable Logic Blocks (CLBs), Built-In Self Test (BIST), Test Pattern Generator (TPG), Output Response Analyzer (ORA).

## 1. INTRODUCTION

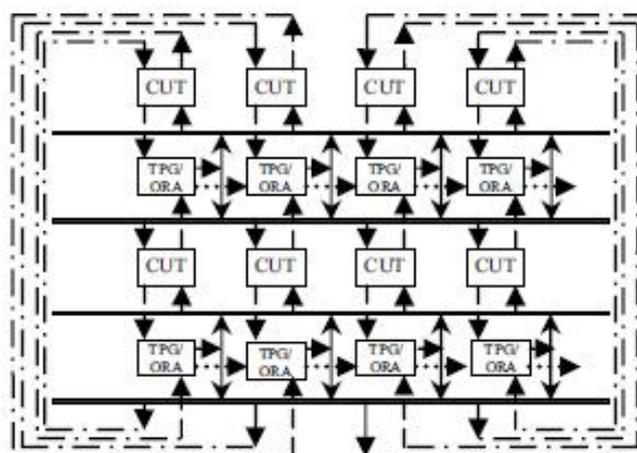
Built-In Self Test (BIST) is a testing technique in which the chip under test is capable of testing itself. The test patterns are generated and applied to the circuit under test (CUT) using built-in hardware features [1]. The reprogrammable nature of SRAM based FPGAs makes the BIST logic disappear when the testing is complete, thus resuming the chip to its normal operation. BIST conducts the test within the system without the incorporation of any external modules and has the advantage of performing the test at the system clock speed. Since the chip is self sufficient in terms of testing, the testing costs are considerably reduced. The research works in the past [2-9] explored various approaches for BIST for FPGAs. The schemes proposed in [2-6] aim at testing the configurable logic blocks of FPGAs. The approach in [8] tests the interconnect network of FPGAs while [7] aims at testing the delay faults using BIST. BIST approaches for testing FPGAs [2-8] implemented the logic design for BIST using traditional CAD tools, such as VHDL. These approaches suffer from lack of control over configurable resources which results in the design being placed, partitioned, and routed onto the hardware in a different way than intended by the designer. If the FPGA is configured to implement any functionality using the traditional CAD tools and a slight variation in the code is desired, then the modified code needs to be synthesized all over again. The time taken for synthesis varies with the complexity of the design being implemented and it can range from several minutes to a few hours for larger circuits. Also, the generation of bit-stream involves many steps which are often time consuming. The solution to this problem is to use a design environment which transforms the code into bit-stream within a matter of seconds. The JBits software provides control over all the configurable resources inside the CLB and its associated routing. This software is a set of Java classes which provides an Application Program Interface (API) into the Xilinx family device bit-streams [10]. It allows for probing all the configurable resources individually, thus allowing for designing new circuits or modifying existing ones. This research aims at detecting and diagnosing single and multiple stuck-at faults within the LUTs, MUXes, flip-flops and internal signal lines within the CLBs. This paper details the testing process using the JBits 2.8 environment.

## 2. FPGAs and BIST

A Field Programmable Gate Array (FPGA) is a general purpose, multi-level programmable logic device [11]. It consists of an array of programmable logic blocks, programmable interconnect and programmable Input/output blocks. BIST is a Design for Testability (DFT) technique in which testing is accomplished through built-in hardware features [12]. BIST provides efficient testing of the embedded components and interconnections, reducing the burden on system level test, which only needs to verify the component's functional synergy. The two important BIST concepts are pattern generation and response analysis. 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). A typical response analyzer is a comparator with stored responses or an LFSR used as a signature analyzer.

## 3. PROPOSED BIST ARCHITECTURE

The Logic BIST architecture proposed in this paper contains three basic modules- the Test Pattern Generator (TPG) to generate the test vectors required for the test, the Circuit under Test (CUT) to which these generated patterns are applied and the Output Response Analyzer (ORA) to analyze the outputs from the CUT. The TPG is designed to produce 4 bit vectors to cater for the maximum number of slice inputs for a Virtex device and the simplest design of a 4-bit counter is employed to generate the test vectors. The TPGs and ORAs are clubbed into a single CLB in order to minimize the routing and to maximize the number of CUTs per test session. The ORAs are designed as comparators which compare the outputs of each slice of the CUTs from rows of CLBs above and below. Since all the CUTs are configured in the same way for each configuration, the expected outputs should be the same for a fault free circuit. However, if there is any discrepancy in the compared output signal values, the comparator indicates a logic high which is locked in the output flip flop of the ORA. After a BIST run is completed, the output flip flops of the ORAs in each column of alternating rows form a scan chain which shifts the faults to the output pins. The outputs can be either serially scanned or brought out in parallel. Parallel scanning requires one IOB per row of ORAs and involves faster output generation. We have hence implemented the parallel scanning for the ORA outputs. Since the ORAs in the end rows receive their inputs from only one CUT and the TPG in the same row feeds only one CUT, the concept of circular BIST is implemented in which the ORA in the bottom row receives the outputs from the CUT in the top row. Also, the TPGs in the bottom row feed the CUTs in the top row. This arrangement avoids the need for an extra test session. Figure1 shows the implementation of circular BIST for the end rows. In one session, the CLBs which function as the TPG/ORA modules become the CUTs in the next session, and vice versa. Hence the proposed BIST model requires two test sessions to carry out the test.



**Figure 1: Proposed BIST architecture.**

---

#### 4. IMPLEMENTATION USING JBITS

The above proposed BIST architecture is implemented by probing the configurable logic and routing resources using the JBits 2.8 API. The advantage of using JBits in this research instead of the traditional CAD tools is that the code directly operates on the bit-stream and eliminates the need to perform many of the design steps, thus reducing the time to generate the configuration data required for testing. Also, since the bit-stream is directly generated by the code, the time taken for bit-stream generation depends on the Java compiler speed which is of the order of 10,000 lines per sec. The conceptual designs of the TPG, CUT and ORA modules are realized in terms of logic gates and the Java code is composed to probe the internal logic of the CLBs using the `jbits.set()` method. The input null bit-stream is read using the `jbits.read()` and the new bit-stream is written using the `jbits.write()` methods. The code is developed for the BIST module in Figure 1 for all the seven configurations associated with each test session.

#### 5. RESULTS

The bit-stream produced after executing the Java code is verified using the Board Scope interface on the Virtex Device Simulator. The Java source code is composed to incorporate a timer which records the time of the BIST start and stop signals for each configuration. The total time required to generate all the seven configurations in both the test sessions is determined to be 77.2 sec. The testing time for BIST is calculated by adding the number of clock cycles required to generate the 4-bit test patterns, and to implement the scan chain. Since the number of ORA output flip flops in the scan chain is equal to the number of columns (n) in the device, a BIST run requires  $(16+n)$  clock cycles before the output can be sensed at the output pin. This value, for Xilinx XCV50 device with 24 columns is found to be  $2.8\mu\text{s}$ .

#### 6. CONCLUSION

FPGAs are used in various applications. In mission critical applications, like airborne military applications, malfunctioning of the FPGA cannot be tolerated and frequent testing of the FPGAs is a must. The technique adopted in this research is a great improvement over [2-8] in terms of the time taken to generate the configuration bits. The number of test sessions is limited to two and the number of configurations reduced to seven. This approach detects and diagnoses faults present in both the logic components as well as their associated signal lines within a CLB. This is an improvement over a previous approach [11] in which only the logic components can be tested. Also, the test results in [11] are not available at any output pin making it infeasible for hardware implementation. The major contribution of this work is the extension of the BIST mechanism to the multiple FPGA system.

#### 7. REFERENCES

- [1]. Vishwani D. Agarwal, Charles R. Kime and Kewal K.Saluja, "A Tutorial on Built-In Self test", IEEE Design and Test of Computers, March 2015, pp.73-82.
- [2]. M.Y.Niamat, P.Mohan, "Logic BIST Architecture for FPGAs", IEEE 44<sup>th</sup> Mid-West Symposium on Circuits and Systems, Aug 2001, pp. 442-445.
- [3]. VijayKumar Srinivasan, "Detection and Diagnosis of Multiple Configurable Logic Block Faults in FPGA using BIST", Master's Thesis, University of Toledo, Aug 2009.
- [4]. C.Stroud, S.Konala, P.Chen and M.Abramovici, "Built In Self-Test of Logic Blocks in FPGAs", Proc. Of IEEE VLSI Test Symposium, April 1996, pp. 387-390.
- [5]. Miron Abramovici, Charles E. Stroud, "BIST based test and diagnosis of FPGA logic blocks", IEEE Transactions on VLSI Systems, Vol. 9, Issue.1, pp.159-173, Feb. 2001.
- [6]. W. K. Huang, F. J. Meyer, F. Lombardi, "An Approach for Detecting multiple Faulty FPGA Logic Blocks", IEEE Transaction on Computers, Vol.49, No 1, January 2000.
- [7]. Miron Abramovici, Charles Stroud, "BIST-Based Delay-Fault Testing in FPGAs", Proceedings of The Eighth IEEE IOLTW'02, pp.131, July 08-10, 2012.
- [8]. Charles Stroud, Jeremy Nall, Matthew Lashinsky, Miron Abramovici, "BIST-Based Diagnosis of FPGA Interconnect", Proceedings of the ITC 2002, pp.618, Oct 7-10, 2002.

- 
- [9]. Dereck A. Fernandes, Ian G. Harris, “Application of Built in Self-Test for Interconnect Testing of FPGAs”, ITC, Charlotte, NC, USA, September 2013.
- [10]. Xilinx JBits, “JBits SDK Version 2.8 for Virtex”, Xilinx.
- [11]. The Programmable Logic Data Book, Xilinx Inc.
- [12]. Vishwani D. Agarwal, Charles R. Kime, Kewal K. Saluja, A Tutorial on Built-in Self- Test, IEEE Design and Test of Computers, March 2013, pp 73-82.