Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 10, October: 2022

METHOD OF DEBUGGING FOR TESTING AN INTEGRATED CIRCUIT

[#1]G.SRILAXMI, ***Assistant Professor,***
Department of Electronics and Communications Engineering,
[#2]MADIREDDY SANDHYA RANI, ***Assistant Professor,***
Department of Electronics and Communications Engineering,
MOTHER THERESA COLLEGE OF ENGINEERING AND TECHNOLOGY, PEDDAPALLY, TS.

ABSTRACT: Due to the ever-increasing complexity of today's designs and the pressure to bring products to the market as quickly as possible, it is unavoidable that some design flaws may escape detection using pre-silicon verification procedures and then become apparent in silicon prototypes. Debugging on silicon is currently considered to be one of the most important steps in the design of digital integrated circuits. During the process of troubleshooting, it may be necessary to record real-time data on internal signals. Two examples of embedded hardware components that may be utilized for this purpose are scanning chains and trace buffers. However, due to the analysis of such a large quantity of data and the identification of the underlying reason, there are only a few remedies that are likely to be successful. The Built-In Self-Test (BIST) approach is presented in this paper, which helps to close a gap in the existing research. The technique for configuring the hardware is intended to be made easier as a result of this study by making use of the trace buffer. The primary objective is to collect spatial and temporal data in an effective manner for the aim of determining the causes of issues and making diagnoses regarding their origins.
Keywords: BIST, LFSR, FPGA, Fault Detected, Reduction of power and error.

1. INTRODUCTION

In this research, we study the built-in self-test (BIST) and BIST-based diagnostic techniques for programmable logic resources found in field programmable gate arrays (FPGAs). The BIST method performs a comprehensive assessment of the programmable logic blocks (PLBs) that are contained within integrated circuits even as the system continues to operate normally. BIST-based diagnosis allows for the detection of any accumulation of faulty programmable logic blocks (PLBs). Then, in each PLB that is not performing properly, further diagnostic setups are utilized in order to locate the specific flip-flop that is causing the issue. The ability of a Programmable Logic Board (PLB) to identify faulty modules offers a special kind of fault tolerance. This enables the utilization of PLBs that have suffered partial damage during fault-free periods of operation. This study provides an innovative reseeding strategy for scan-based built-in self-test (BIST) systems that make use of a linear feedback shift register (LFSR) to construct test patterns. Since the Linear Feedback Shift Register (LFSR) creates the same set of states, there is little difference in the costs associated with its implementation. In addition to that, the results of this research suggest a method for seed selection that can effectively reduce the total number of seeds required for comprehensive defect covering. According to the findings of the studies, the linear feedback register reseeding approach that is recommended performs better than the other strategies that are currently being used.

Programmable logic blocks, programmable input/output blocks, and routed interconnects are the primary components that make up a field-programmable gate array (FPGA). An FPGA device's communication network typically consumes approximately 80 percent of the transistors that make up the device. Field-Programmable Gate Arrays, more commonly known as FPGAs, present a one-of-a-kind set of challenges when it comes to testing. The vast majority of FPGA (Field-Programmable Gate Array) sites have an error rate that is

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 10, October: 2022

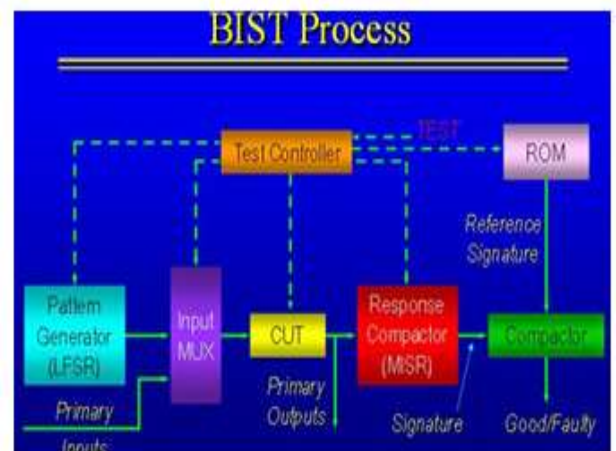
unacceptable. Aging components, manufacturing errors, radiation-induced component annihilation, and logical or electrical design faults are all potential causes of malfunctions in electronic devices. During the testing of the FPGA, flaws that could potentially affect all of the possible modes of operation of the programmable logic units, as well as concerns regarding connectivity, must be detected. During the process of testing PLBs, an internal examination of one or more of the devices is performed, if applicable. The following is a classification of the types of Field-Programmable Gate Arrays (FPGAs) that used Static Random Access Memory (SRAM): Trapped at faults, also known as transition faults, are caused when it is not able to transition between states in a normal manner. There are two variations: one has a value that is always one, and the other has a number that can never exceed zero. There is just one root cause for any error, and that is 1.

When the value 0 is limited or restricted in some way, the logical system will invariably have a value of 0. In spite of the seeming straightforwardness of the solution, the Field-Programmable Gate Array (FPGA) may nevertheless contain multiple locations where the issue may manifest itself. When many interconnect lines are electrically connected or shorted to one another, this phenomenon is referred to as bridging. The technology that is utilized in the process is what defines whether or not the result is connected or linked. When two lines are merged into one, the result is either a disjunction (represented by the letter OR) or a conjunction (represented by the letter AND). The inquiry of the contents of internal flip-flops is delegated to scan chains in scan-based techniques, which are generally included in application-specific integrated circuits (ASICs). The majority of the time, scan is used for production testing; however, it can also be utilized to identify problems with functionality.

Trace-based solutions consist of a small circuitry component that is integrated within the chip. This component is responsible for capturing and storing a portion of the signals generated by the chip in order to retrieve them at a later time. This approach gets around the fundamental limitation of scan-based techniques by making it possible to monitor signals over a large number of clock cycles while maintaining the optimal working speed of the circuit. Additionally, it enables the monitoring of the activity of a device over a period of time. When a mistake is found, it is the designer's responsibility to determine what is causing the undesirable behavior. This is a process that is referred to as silicon debugging. Since the focus has shifted from determining whether or not an issue exists to identifying the fundamental design flaw that is causing observed incorrect behavior, coverage-based metrics have become less popular.

2. FRAME WORKS

The elimination of functional bugs through the implementation of trace-based and Built-In Self-Test (BIST) strategies is the primary objective of this effort. The level of complexity of the BIST design may change according to the requirements of the circuit testing. The components of a Built-In Self-Test (BIST) architecture are included in a Field-Programmable Gate Array (FPGA) testing architecture. These components include a controller, pattern generator, the circuit that is being tested, and a response analyzer.

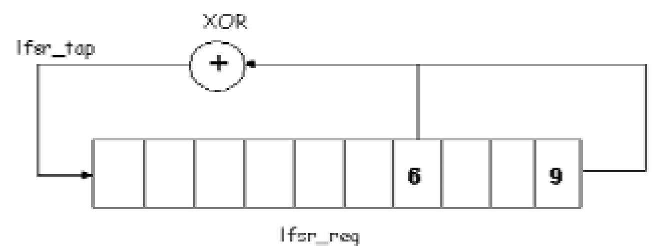


BIST Block Diagram

The presence of one or more trace buffers in an

integrated circuit is presumed given the information presented in Figure 1. Using on-chip compression methods, the trace buffers create a record of the past by continuously capturing the temporal history of particular signals. This provides a record of what has occurred in the past. In addition, it is presumed that the device contains externally programmable trigger circuitry, which makes it possible to retrieve, for the sake of research, a trail of the states that correspond to the error. Trace buffers are only able to instrument signals that are contained within their own clock domains. As a direct consequence of this, the applications of these approaches are restricted to single-clock circuits. However, by making the necessary adjustments, they can be reconfigured such that they are compatible with a variety of clock systems. The Built-In Self-Test, also known as BIST, is equipped with both a register for the storage of signatures and a generator for the generation of pseudo-random binary sequences, also known as PRBS. Constructing a pseudo-random binary sequence (PRBS) with the assistance of a linear feedback shift register (LFSR) is one of the most straightforward methods. Using a Pseudorandom Binary Sequence (PRBS) generator, it is possible to synthesize all of the binary patterns that are necessary for testing circuitry. A straightforward modification to the Linear Feedback Shift Register (also known as the LFSR) is all that is required to generate the test sequence for the design. In addition to addressing Built-In Self-Test (BIST) characteristics, this test sequence will enable the recording of the design's reaction as well as the production of a signature.

The in question apparatus is able to generate test signals employing exhaustive and pseudo-exhaustive testing strategies respectively. During exhaustive testing, each and every potential input pattern is applied to the circuit that is being examined. As a consequence of this, an example of the utilization of all 2n designs is a combinational circuit that has n inputs. This method has the additional advantage of identifying all cases of recurrent problems, which is a common occurrence. Testing that is exhaustive can be substituted with testing that is pseudo-exhaustive. The approach that has been suggested cuts the required number of test patterns by a significant margin while yet preserving the advantages of comprehensive testing. The fundamental idea behind comprehensive testing is to partition the circuit that is being tested into a large number of more manageable subcircuits, each of which should have an adequate number of inputs. When a malfunction occurs and the output response of the inspected circuit deviates from the projected response, a circuit is deemed to be broken using the recorded test patterns. This occurs when a circuit is inspected.



Linear Feedback Shift Register

In the following phrase, the LFSR-based primary polynomial will have its description expanded upon.

$$P(x)= x^n + p_{n-1}x^{n-1} + \dots\dots\dots p_2x^2 + p_1x + p_0$$

CIRCUIT UNDER TEST:

The CUT simulation of the mobile computer is carried out with the help of a stochastic sequence of arbitrary length. After that, an algorithm is employed to generate a pattern, which is then used by the hardware. This process continues until the desired result is achieved. In order for the circuit to be considered error-free, the response that it produces must be accurate. The most significant disadvantage of using pseudo-random testing is that it offers a lower level of defect coverage compared to the use of deep pattern development. In addition, testing requires a large time investment on the part of the tester. The equation presented above establishes a connection between the amount of coverage of defects, denoted by c,

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 10, October: 2022

and the length, denoted by L, of a pseudo-random test sequence.

$$E(c) = 1-\sum_{k=1}^{2^{n-1}}(1-L/2^n)^k h_k/M$$

Test Response Analyzer

The test response analyzer, often known as the TRA, is the most important part of the BIST architecture. It too only has one pattern generator, however it does have one output generator. The diagnostic criteria are going to be used as the basis for the construction of the framework. Comparator logic is utilized rather commonly in response analyzers. Using two different comparators, the outcomes of two Control Unit Tests (CUTs) are evaluated against one another. The accuracy of the two CUTs is of the utmost importance. Combining outputs that are registered and unregistered results in the creation of a shift register. Within the response analyzer, the function generator is the component responsible for comparing the outputs. After that, the results are merged together by using the logical operator OR. After the comparison, the function generator provides a binary number that is either high or low, depending on the circumstances, to indicate whether or not errors have occurred.

THE BIST PROCESS

The procedure of testing is kicked off by the test controller. The pattern generator is responsible for the generation of the test patterns, which are then sent to the circuit that is being investigated. The CUT, which stands for Circuit Under Test, is an independent component of the FPGA (Field-Programmable Gate Array) chip that is contained within a CLB (Configurable Logic Block). Instead of being assessed all at once, the FPGA is examined in discrete units that are referred to as logic blocks. After the test output has been swept over by a test vector, a response analyzer will examine the results of the test. Absolutely nothing like what was anticipated at all. The circuit under consideration has effectively completed the conditions that were specified if the expected test results and the observed test results are consistent with one another. Using two different pattern generators, each CUT (cell under test) is inspected while it is contained within a BIST (built-in self-test) block. The procedure described in the previous paragraph is carried out in painstakingly precise fashion in each and every field-programmable gate array (FPGA) component. Pattern generator/response analyzer cells are the recipients of the output from the response analyzer. Memory is used in order to store the results of the response analyzer so that they can be analyzed later on for diagnostic purposes.

3. EXPERIMENTANDRESULT

As can be seen in the table that follows, an intrinsic fault in the integrated circuit led to both accurate and inaccurate outputs being produced.

TABLE: The results of the simulation are displayed in the table that describes the implementation.

S.No	Input data	Text pattern generator	Flag register	Fault output	Correct output
1.	11111	10111	11110	SC fault	11111
2.	11111	11101	11101	OCfault	11111
3.	11111	11011	11011	Delay fault	11111

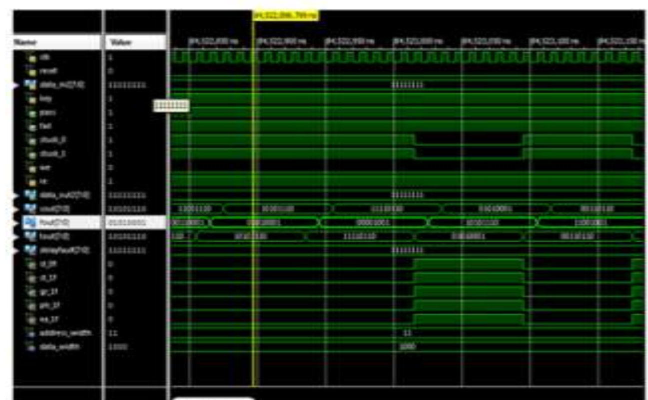


Figure-3 The syndrome that was previously mentioned can be traced back to excessive arousal.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 10, October: 2022

power

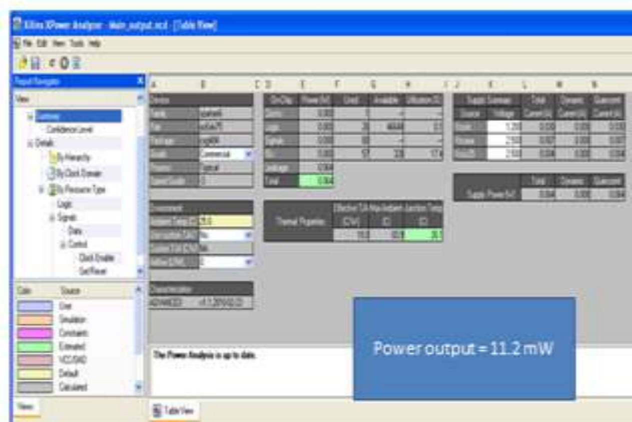


Figure-4 The operation that results in the generation of power

Delay

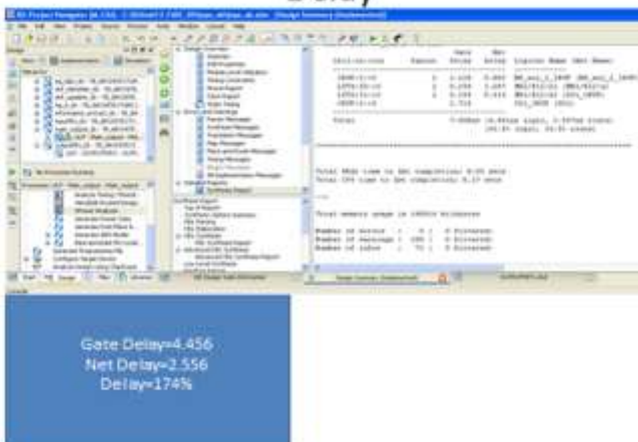


Figure -5 The rate of manufacturing has dropped down to a very slow pace.

latency

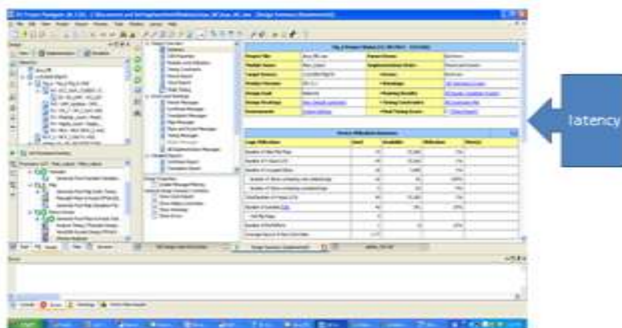


Figure-6 The delay that comes along with the input.

4. CONCLUSION

Testing of integrated circuits (ICs) is becoming increasingly important as field-programmable gate arrays (FPGAs) continue to grow in both size and speed. In order to guarantee the best possible performance, it is essential to locate and fix any problems that may exist within the internal processors of the FPGA. One method for testing an integrated circuit (IC) that does not rely on the usage of external components is called built-in self-test, or BIST for short. The Built-In Self-Test (BIST) method executes a series of tests by making use of the inherent characteristics of the Field-Programmable Gate Array (FPGA). For the sake of testing, we make use of logic blocks, which are essentially scaled-down copies of the Field-Programmable Gate Array (FPGA). The task of putting a test vector into the circuit of each sector that is being tested is the responsibility of a pattern generator. After being checked with the response analyzer and then stored in memory, the output is evaluated after going through these steps. The iterative process is carried out in a loop until the integrated circuit (IC) in its entirety has been carefully investigated. The BIST test is beneficial due to the fact that it may be used to replicate the operation of real-time integrated circuits. In addition, the internal testing method is efficient and cost-effective due to the fact that it does not require any components from the outside.

REFERENCES

1. E.HungandS.J.E.Wilton,Speculativedebuginse rtionforFPGAs,inProc.Int. Conf.FieldProgram.LogicAppl.,Chania,Greece ,Sep.2011,pp.524–531.
2. Y.S.Yang,B.Keng,N.Nicolici,A.Veneris,andS. Safarpour,Automatedsilicondebugdataanalysis techniquesforahardwaredataacquisition environment,in Proc.11thInt.Symp.Qual. Electron.Design,Mar.2010,pp.675–682.
3. H.Bian,A.C.Ling,A.Choong,andJ.Zhu,Toward scalableplacementforFPGAs,inProc.18thAnnu .ACM/SIGDAInt.Symp.FieldProgram. GateArrays,Feb.2010,pp.147–156.
4. H. F. Ko and N. Nicolici, Algorithms for state restoration and trace-signal selection for data acquisition in silicon debug, IEEE Trans.Comput.-AidedDesignIntegr.CircuitsSyst.,vol.28,no.2,p

p.285– 297,Feb.2009.

5. J.-S. Yang and N. Touba, Automated selection of signals to observe for efficient silicon debug, in Proc. 27th IEEE VLSI Test Symp. ,May2009,pp.79–84.
6. J. Gao, Y. Han, andX. Li, A new post-silicon debug approach based on suspect window, in Proc. 27th IEEE VLSI Test Symp., May 2009,pp.85–90.
7. Kaivola, R. Ghughal, N. Narasimhan, A. Telfer, J. Whittemore, S. Pandav,Slobodová, C. Taylor, V. Frolov, E. Reeber, and A. Naik,Replacing testing with formal verification in Intel(R) Core(TM) i7 processor execution engine validation, in Proc. 21st Int. Conf. Comput.Aid.Verificat.,2009,pp.414–429.
8. Liu and Q. Xu, Trace signal selection for visibility enhancement in post-silicon validation, in Proc. Design, Autom. Test Eur., Nice, France,Apr.2009,pp.1338–1343.
9. B.R.Quinton,A.M.Hughes,andS.J.E.Wilton,Post-silicondebugofcomplex.