International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 14
October2016

Traditional Scan Based Design For Atpg Of A Feedbach Shift Register Using Lbist

Doddi Sowjanya,Pg Student

Mrs.M.Anusha, Associate Professor

Dept Of Ece

Malla Reddy College of Engineering & Technology,Secunderabad

Abstract : Testing cost is one of the major contributors to the manufacturing cost of integrated circuits. Logic Built-In Self Test (LBIST) offers test cost reduction in terms of using smaller and cheaper ATE, test data volume reduction due to on-chip test pattern generation, test time reduction due to at-speed test pattern application. However, it is difficult to reach a sufficient test coverage with affordable area overhead using LBIST. Also, excessive power dissipation during test due to the random nature of LBIST patterns causes yield-decreasing problems such as IR-drop and overheating. In this dissertation, we present techniques and algorithms addressing these problems. In order to increase test coverage of LBIST, we propose to use onchip circuitry to store and generate the "top-off" deterministic test patterns. First, we study the synthesis of Registers with Non-Linear Update (RNLUs) as on-chip sequence generators. We present algorithms constructing RNLUs which generate completely and incompletely specified sequences. Then, we evaluate the effectiveness of RNLUs generating deterministic test patterns on-chip. Our experimental results show that we are able to achieve higher test coverage with less area overhead compared to test point insertion. Finally, we investigate the possibilities of integrating the presented on-chip deterministic test pattern generator with existing Design-For-Testability (DFT) techniques with a case study. The problem of excessive test power dissipation is addressed with a scan partitioning algorithm which reduces capture power for delayfault LBIST. The traditional S-graph model for scan partitioning does not quantify the dependency between scan cells. We present an algorithm using a novel weighted S-graph model in which the weights are scan cell dependencies determined by signal probability analysis. Our experimental results show that, on average, the presented method reduces average capture power by 50% and peak capture power by 39% with less than 2% drop in the transition fault coverage. By comparing the proposed algorithm to the original scan partitioning, we show that the proposed method is able to achieve higher capture power reduction with less fault coverage drop.

INTRODUCTION

1.1 Scan Design

Scan chain is a technique used in design for testing. The objective is to make testing easier by providing a simple way to set and observe every flip-flop in an IC. The basic structure of scan includes the following set of signals in order to control and observe the scan mechanism.

1. Scan_in and scan_out define the input and output of a scan chain. In a full scan mode usually each input drives only one chain and scan out observe one as well.
2. A scan enable pin is a special signal that is added to a design. When this signal is asserted, every flip-flop in the design is connected into a long shift register.
3. Clock signal which is used for controlling all the FFs in the chain during shift phase and the capture phase. An arbitrary pattern can be entered into the chain of flip-flops, and the state of every flip-flop can be read out.

In a full scan design, automatic test pattern generation (ATPG) is particularly simple. No sequential pattern generation is required - combinatorial tests, which are much easier to generate, will suffice. If you have a combinatorial test, it can be easily applied.

- Assert scan mode, and set up the desired inputs.
- De-assert scan mode, and apply one clock. Now the results of the test are captured in the target flip-flops.
- Re-assert scan mode, and see if the combinatorial test passed.

In a chip that does not have a full scan design -- i.e., the chip has sequential circuits, such as memory elements that are not part of the scan chain, sequential pattern generation is required. Test pattern generation for sequential circuits searches for a sequence of vectors to detect a particular fault through the space of all possible vector sequences.

Even a simple stuck-at fault requires a sequence of vectors for detection in a sequential circuit. Also, due to the presence of memory elements, the controllability and observability of the internal signals in a sequential circuit are in general much more difficult than those in a combinational logic circuit. These factors make the complexity of sequential ATPG much higher than that of combinational ATPG.

There are many variants:

- **Partial scan**: Only some of the flip-flops are connected into chains.
- **Multiple scan chains**: Two or more scan chains are built in parallel, to reduce the time to load and observe.

- Test compression: the input to the scan chain is provided by on-board logic

1.2 Scan Methods

In-circuit testing works fine, up to a point. It doesn't do much good for custom VLSI chips and ASICs, because the internal signals simply aren't accessible. Even in board-level circuits, high-density packaging technologies such as surface mounting greatly increase the difficulty of providing a test point for every signal on a PCB. As a result, an increasing number of designs are using "scan methods" to provide controllability and observability.

A scan method attempts to control and observe the internal signals of a circuit using only a small number of test points. A scan-path method considers any digital circuit to be a collection of flip-flops or other storage element interconnected by combinational logic, and is concerned with controlling and observing the state of the storage elements. It does this by providing two operating modes: a normal mode, and a scan mode in which all of the storage elements are reorganized into a giant shift register. In scan mode, the state of the circuit's n storage elements can be read out by n shifts (observability), and a new state can be loaded at the same time (controllability).

Figure 1.1 shows a circuit designed using a scan-path method. Each storage element in this circuit is a scan flip-flop that can be loaded from one of two sources. The test enable (TE) input selects the source—normal data (D) or test data (T). The T inputs are daisy-chained to create the scan path shown in colour. By asserting ENSCAN for 11 clock ticks, a tester can read out the current state of the flip-flops and load a new state. The test engineer is left with the job of deriving test sets for the individual combinational logic blocks, which can be fully controlled and observed using the scan path and the primary inputs and outputs.

Scan-path design is used most often in custom VLSI and ASIC design, because of the impossibility of providing a large number of conventional test points. However, the two-port flip-flops used in scan-path design do increase chip area. For example, in LSI Logic Corp.'s LCA500K series of CMOS gate arrays, an FD1QP D flip-flop macro cell uses seven "gate cells," while an FD1SQP D scan flip-flop macro cell uses nine gate cells, almost a 30% increase in silicon area. However, the overall increase in chip area is much less, since flip flops are only a fraction of the chip, and large "regular" memory structures (e.g., RAM) may be tested by other means. In any case, the improvement in testability may actually *reduce* the cost of the packaged chip when the cost of testing is considered. For large ASIC designs with rich, complicated control structures, scan-path design should be considered a requirement.

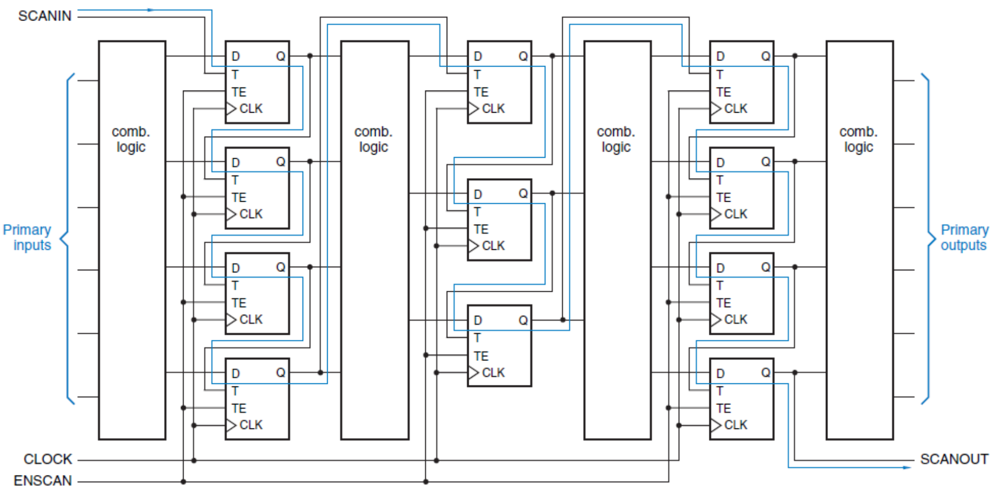


Figure 1.1: circuit containing a scan path, shown in colour.

1.3 Scan-Path Design and rules

Any sequential circuit may be modeled as:

Rules:

A designer needs to observe four rules during functional design:

- Only D-type master-slave FFs should be used.
- No JK, toggle FFs or other forms of asynchronous logic.
- At least on PI must be available for test.
- As shown in previous circuit, the Scan-in and Scan-out pins can be multiplexed (only One additional MUX is needed at Scan-out).

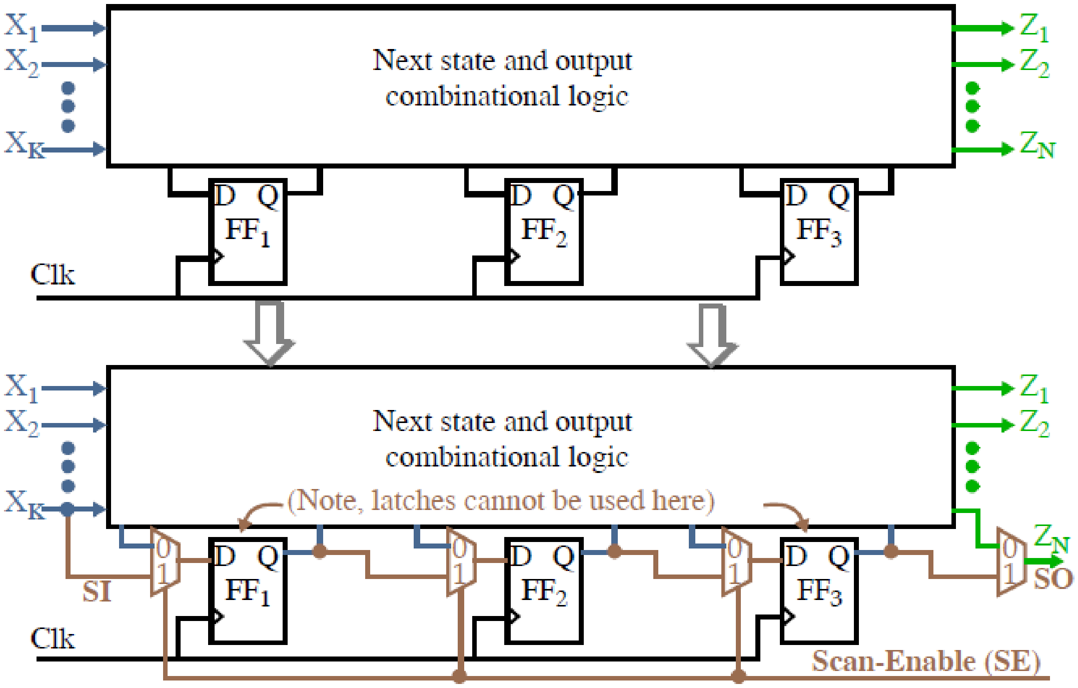

X_1
X_2
X_K
Next state and output combinational logic
Z_1
Z_2
Z_N
D Q
FF_1
FF_2
FF_3
Clk
(Note, latches cannot be used here)
0
1
SI
SO
Scan-Enable (SE)

Figure 1.2: scan path design

- Therefore, the only required extra pin is Scan-Enable, SE (or Test Control, TC).
- All FFs must be controlled from PIs.
- Simple circuit transformations can be used to change FFs whose Clk is "gated" by an Internal logic signal.

- Clocks must not feed data inputs of the FFs.
- A race condition can result in normal mode otherwise.
- This is generally considered good design practice anyway.

1.4 Tests for Scan Circuits

Two phases:

- Shift test
- Set TC= 0, and shift toggle sequence 00110011... using Clk.
- The length is n_{sff} + 4, where n_{sff} are the number of scan flops.
- This sequence produces all 4 transitions, 0->0, 0->1, 1->1 and 1->0, catches all/most SA faults.
- The Shift test can be used in either single-clock or two-clock designs.

- A Flush test is also possible in two-clock designs:

- φ1 (Master Clk) is held low while φ2 and φ3 are held high.
- This creates a continuous path between SI and SO for application of 0 and 1.
- Combinational logic test
- This phase allows the combination logic circuit to be tested for SA faults.
- An ATPG algorithm is used where outputs of Scan FFs are treated as pseudo-PIs (completely controllable) and inputs are treated as pseudo-POs.

AUTOMATIC TEST PATTERN GENERATION

ATPG (acronym for both Automatic Test Pattern Generation and Automatic Test Pattern Generator) is an electronic design automation method/technology used to find an input (or test) sequence that, when applied to a digital circuit, enables automatic test equipment to distinguish between the correct circuit behavior and the faulty circuit behavior caused by defects. The generated patterns are used to test semiconductor devices after manufacture, or to assist with determining the cause of failure (failure analysis) the effectiveness of ATPG is measured by the number of modeled defects, or fault models, detectable and by the number of generated patterns. These metrics generally indicate test quality (higher with more fault detections) and test application time (higher with more patterns). ATPG efficiency is another important consideration that is influenced by the fault model under consideration, the type of circuit under test (full scan, synchronous sequential, or asynchronous sequential), the level of abstraction used to represent the circuit under test (gate, register-transfer, switch), and the required test quality.

PROJECT DESCRIPTION

4.1 Existing Method:

Cryptographic methods are used to protect confidential information against unauthorized modification or disclosure. Cryptographic algorithms providing high assurance exist, e.g. AES. However, many open problems related to assuring security of a hardware implementation of a cryptographic algorithm remain. Security of a hardware implementation can be compromised by a random fault or a deliberate attack. The traditional testing methods are good at detecting random faults, but they do not provide adequate protection against malicious alterations of a circuit known as hardware Trojans. For example, a recent attack on Intel’s Ivy Bridge processor demonstrated that the traditional Logic Built-In Self-Test (LBIST) may fail even the simple case of stuck-at fault type of Trojans. In this paper, we present a novel LBIST method for Feedback Shift Register (FSR)- based cryptographic systems which can detect such Trojans. The specific properties of FSR-based cryptographic systems allow us to reach 100% single stuck-at fault coverage with a small set of deterministic tests. The test execution time of the proposed method is at least two orders of magnitude shorter than the one of the pseudo-random pattern-based LBIST. Our results enable an efficient protection of FSR-based cryptographic systems from random and malicious stuck-at faults.

Feedback Shift Register (FSR) based cryptographic systems are the fastest and the most power-efficient cryptographic systems for hardware applications. The speed and the power are two crucial factors for future cryptographic systems, since they are expected to support very high data rates in 5G ultra-low power products and applications. A hardware fault can compromise the security of a cryptographic system. For example, suppose that the output of a pseudo-random pattern generator used in a stream cipher is stuck to the logic 0. A stream cipher encrypts a message by combining it with a pseudo-random pattern, typically by a bit-wise addition modulo 2. Therefore, if the pseudo-random pattern is all-0, the message is sent unencrypted. To make possible periodic fault detection in functional circuits during their lifetime, cryptographic systems often employ Logic Built-In Self-Test (LBIST). However, as shown by a recent attack on Intel's cryptographically secure Random Number Generator (RNG) used in the Ivy Bridge processors, traditional LBIST techniques have a limited use against malicious alterations of the original circuit known as hardware Trojans. This is not only due to the fact that a Trojan can be inserted into the LBIST itself, but also because the Trojan can be designed not to trigger the LBIST, since LBIST usually detects only a subset of all possible faults.

4.2 Proposed Method

We present a new method for LBIST which makes possible detecting stuck-at fault. The presented method specifically targets FSR-based cryptographic systems. First, we use a deterministic test set which covers 100% of single stuck-at faults in the circuit under test, Test Pattern Generator (TPG) and Test Response Analyzer (TRA). Second, we do not compact output responses into a Multiple Input Signature Register (MISR) signature. So, an attack based on selecting suitable values for the Trojan which generates the correct MISR signature for the inputs provided during the LBIST becomes impossible in our case. Furthermore, Trojans inserted into the LBIST circuitry itself (TPG and TRA) will be detected. The presented method is similar to the traditional scan design in that is provides a simple way of setting and observing each flip-flop in a circuit. However, unlike in the case of scan, we do not connect flip-flops in scan chains. Instead, to support a test mode, we multiplex the input of cells which serve as state variables for the feedback functions and put a switch at the output of cells which correspond to outputs to non-trivial feedback functions. This allows for loading and unloading of flip-flops' contents during the test mode.

The size and the number of Boolean functions used in cryptographic systems are typically considerably smaller than the size of an FSR. Therefore, the presented approach has small area overhead. Furthermore, our technique does not affect the propagation delay of the original circuit. In the traditional scan, the propagation delay is always increased by the delay of a multiplexer.

Boolean functions used in cryptographic systems are commonly represented in Algebraic Normal Form (ANF). It was shown by Reddy that a combinational logic circuit implementing an n-variable Boolean function represented in ANF can be tested for all single stuck-at faults using at most 3n+4 tests. We use Reddy's result as a base to derive a minimal complete test set for single stuck-at faults for combinational logic circuits implementing feedback functions of an FSR. These features make WSN very flexible and opened up a wide range of applications. An overview of the history of sensor networks can be found. Wireless sensor networks started out in the military with sensor nodes so big as to till the bed of a truck. The first reported deployments of small, pager sized nodes were for environmental monitoring and for habitat monitoring. Examples are monitoring birds on Great Duck Island on the coast of Maine, and collecting microclimate data in the James San Jacinto Mountains Reserve. There are also several military applications like target tracking, detecting radiation, biological, and chemical weapons, and localization of shooters in urban terrain. Most of these applications require the nodes to operate unattended for a long period of time which is limited by their energy source, usually a battery. In order to minimize energy consumption the nodes have a low duty cycle, i.e. most of the time they are turned off. The range of their radio transmitters for wireless data transfer is limited to conserve power; hence they can only communicate directly with nodes in close proximity. They establish a routing tree with the base station at its root. The base station collects the data from the sensors and communicates with the outside world. It is assumed to have sufficient power for all computations and communications with the nodes and the outside world. Pico net was an early general-purpose, low-power ad hoc radio network by the University of Cambridge.

The "Smart Dust" project at the University of California, Berkeley set out to develop sensor nodes of 1mm3 in size. Their early studies even showed designs for flying motes which probably inspired the book "Prey" by Michael Crichton. These early Smart Dust motes used a steered laser beam for communication, however battery powered nodes with wireless radio frequency transmitters have become standard. Current sensor platforms can be divided into four classes: Gateway, high bandwidth sensing, generic sensing, and specialized sensing. For example motes for each class and Linear feedback shift registers (LFSRs) over finite fields are widely exploited and play an important role in cryptography and coding theory, see Golomb, Lidl and Niederreiter etc.

Most LFSRs used in traditional stream cipher are based on binary field F2, which have good cryptographic properties but produce only one bit per step. It is well-known that hardware implementations of traditional binary LFSR are simple and efficient, but their software implementations are inefficient for modern processors with word operations. Generally, there are two principles to evaluate FSR sequences: 1). security; 2). efficiency and resource consumption. These two principles have same importance, in other words, if a FSR has extremely excellent cryptographic properties but its implementation efficiency is low and resource consumption is large, their application value is limited. In fact, modern computer processors provide many fundamental word operations:

a). logic operations such as Xor, And, Or, complementary operation, left shift, right shift, cycle shift etc;

b). Arithmetic operations such as addition, subtraction, multiplication, division etc.

So it is interesting to research on how to use the word operation above to design word-oriented feedback shift registers (FSRs) with good security, easy hardware and fast software implementations. In FSE of 1994, Preneel set forth the following problem: how to design fast and secure FSRs with the help of the word operations of modern processors and the techniques of parallelism. In the stream ciphers such as SOBER, SNOW, and Turing, word-oriented primitive LFSRs over finite field were used, which were carefully chosen so that the Hamming weights of the generating primitive polynomials of the component sequences are large and have fast software implementation. This paper is arranged as following. In section 2, we introduce the concept of σ−linear feedback shift register (σ−LFSR) based on logic operations on words, which is the generalization of TSR introduced by Tsaban and Vishne, and give basic properties of σ−LFSR. In Section 3, we discuss the 32-bit σ−LFRS and give an algorithm to search for the primitive σ−LFRS with few logic operations on words, as a result we give two examples: HHZ-1 and HHZ-2. In Section 4, we compare HHZ-1 and HHZ-2 with the LFSRs appeared in SNOW1.0, SNOW2.0, SOBERt-32 and Turing by the point of view from security and efficiency of software implementation

4.3 Block Diagram:

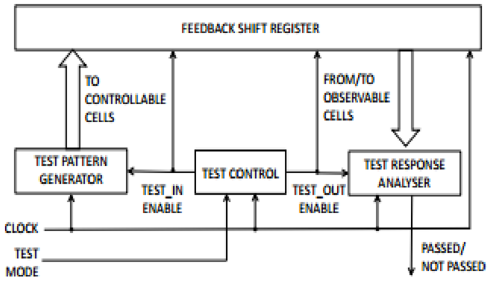


Fig. 4.1: A block diagram illustrating the presented method.

4.3.1 Feedback Shift Register:

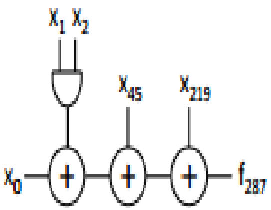


Fig. 4.2: The logic circuit implementing ANF of the feedback function f287 of Trivium.

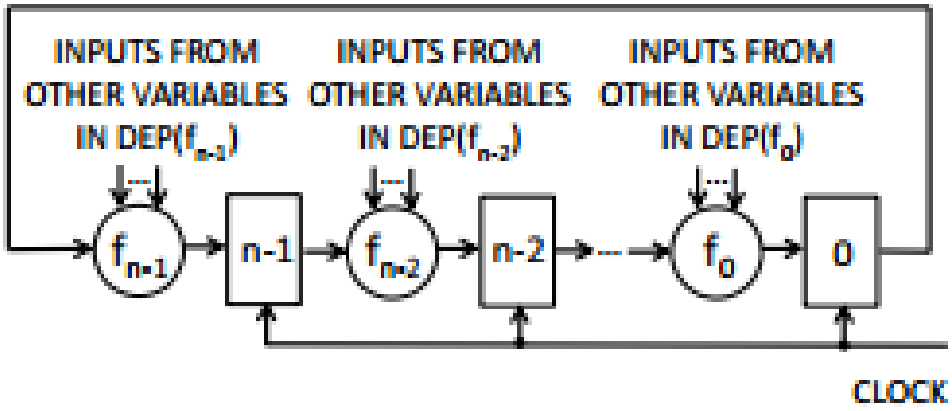
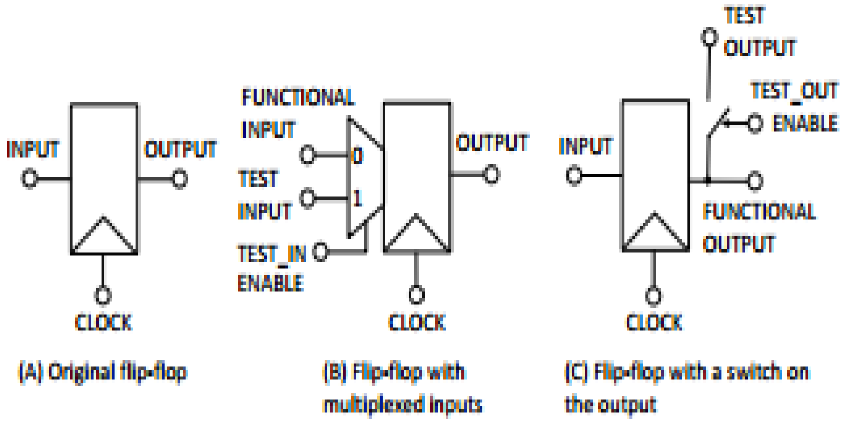

INPUTS FROM OTHER VARIABLES IN DEP(f_{n-1})
INPUTS FROM OTHER VARIABLES IN DEP(f_{n-2})
INPUTS FROM OTHER VARIABLES IN DEP(f_0)
f_{n-1}
n-1
f_{n-2}
n-2
f_0
0
CLOCK

Fig. 4.3: The general structure of an n-bit FSR.

We can see that functions f287, f194 and f110 use only 15 out of 287 possible state variables in their ANFs in total. As another example, consider the 128-bit FSR used in the stream cipher Grain-128 . All its feedback functions except the function f127 are of type fi = xi+1. The function f127 is given by: f127 = x0 ⊕x26 ⊕x56 ⊕x91 ⊕x96 ⊕x3x67 ⊕x11x13 ⊕x17x18 ⊕x27x59 ⊕ x40x48 ⊕x61x65 ⊕x68x84 This function uses 19 out of 128 possible state variables. From the examples above, the reader may see that none of ANFs uses the same variable twice. Furthermore, the same variable does not occur in more than one ANF. In addition, the same index is not used as input and output, i.e. if fi is non-trivial, then the state variable xi is not used. These typical features of ANFs used in cryptographic systems follow from the requirements for the cryptographic security of Boolean functions. Any n-variable Boolean function represented in ANF can be implemented by a logic circuit consisting of a linear cascade of two-input XOR gates fed by AND gates, one corresponding to each product-term of the expression with a non-zero constant ci , i ∈ {1,...,2 n −1}. For example, the function f287 of Trivium can be implemented by a circuit shown in Fig. 4.2.

4.3.4 Logic Built-In Self-Test:

The traditional LBIST typically employs a Linear FSR (LFSR) to generate pseudo-random test patterns that are applied to the circuit under test and a Multiple Input Signature Register (MISR) for obtaining the compacted response of the circuit to these test patterns. An incorrect MISR output indicates a fault in the circuit. Various techniques can be used to complement pseudo-random test patterns. A problem with the traditional LBIST is that many pseudorandom pattern (several thousands or more) need to be applied to reach satisfactory fault coverage. This implies that test execution time can be too long for some applications. The method presented in this paper is similar to the traditional scan design in that is provides a simple way of setting and observing each flip-flop in a circuit. However, unlike in scan, we do not connect flip-flops in scan chains. Instead, to support the test mode, we modify the original FSR as follows (see Fig. 4.4) we multiplex the input of each controllable cell as shown in Fig. 4.4. The input of the original flip- flop becomes the functional input of the multiplexer (MUX). The test input of MUX is connected to the Test Pattern Generator (TPG). 2) We duplicate the output of each observable cell as shown in Fig. 4.5. The duplicated output is connected to the Test Response Analyzer (TRA) through a switch 2 .When the test mode is selected, the flip-flops with multiplexed inputs become inputs to the combinational logic. The flip-flops which have a switch on the output become outputs of the combinational logic. As in a scan design, this increases controllability and observability, making possible testing a sequential circuit with tests for combinational logic.



(A) Original flip-flop

(B) Flip-flop with multiplexed inputs

(C) Flip-flop with a switch on the output

Fig. 4.4: Modifications of FSR flip-flops to support test mode.

Note that such a technique does not affect the propagation delay of the original circuit. In the traditional scan, the propagation delay is always increased by the delay of a MUX.

We add MUXes only to the controllable cells, whose feedback functions are trivial. Therefore, the propagation delay is still determined by the observable cells, whose feedback functions are non-trivial.

4.3.5 Test Response Analyzer:

The following signals are added to the FSR to control and observe its cells:

1) Test in enable signal controls the application of test vectors. When it is asserted, controllable cells are connected to the TPG and TPG is connected to the clock. Otherwise, controllable cells are connected to their predecessor cells and TPG is not connected to the clock.

2) Test out enable signal controls output response analysis. When it is asserted, observable cells are connected to both the TRA and their successor cells and TRA is connected to the clock.

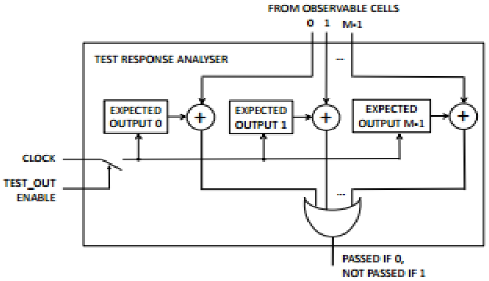


Fig. 4.5: The Structure of the TRA

LOGC BUILT- IN SELF - TEST

LBIST stands for Logic Built-In Self Test. As VLSI marches to deep sub-micron technologies, LBIST is gaining importance due to the unique advantages it provides. LBIST refers to a self-test mechanism for testing random logic. The logic can be tested with no intervention from the outside world. In other words, a piece of hardware and/or software is inbuilt into an integrated circuit to test itself. By random logic, is meant any form of hardware (logic gates, memories etc.) that can form a part or whole of the chip. A generic LBIST system is implemented using STUMPS (Self-Test Using MISR and PRPG) architecture. A typical LBIST system is as shown in the figure below:

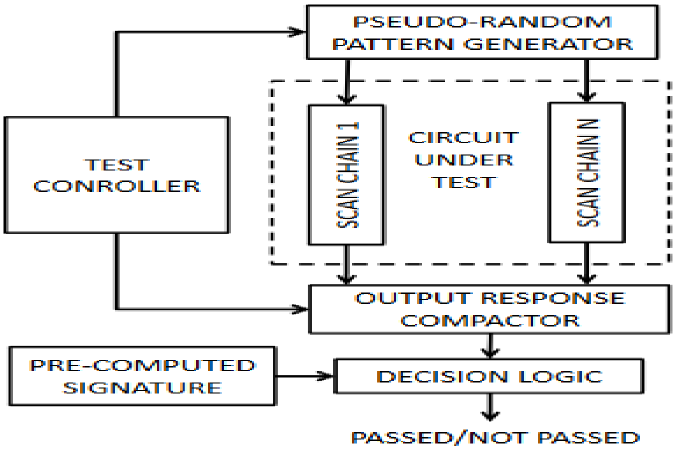


Figure 5.1: Structure of LBIST system

5.3 Linear Feedback Shift Registers (LFSRs)

Efficient design for Test Pattern Generators & Output Response Analyzers (also used in CRC)

- FFs plus a few XOR gates
- Better than counter
- Fewer gates
- Higher clock frequency

Characteristic polynomial

- defined by XOR positions
- P(x) = x4 + x3 + x + 1 in both examples

Two types of LFSRs

1. External Feedback
2. Internal Feedback
    - Higher clock frequency

5.4 Internal and External LFSR:

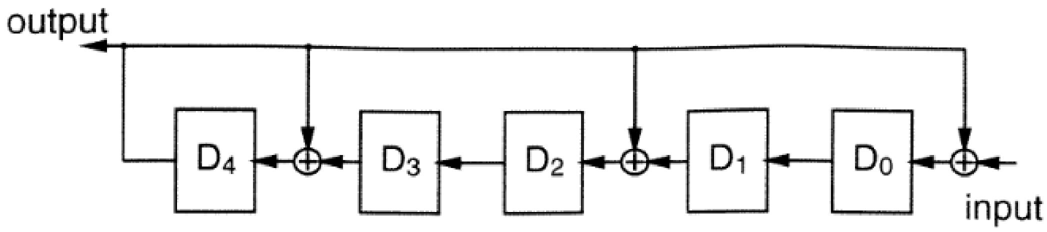


Fig. 5.2 Internal LFSR

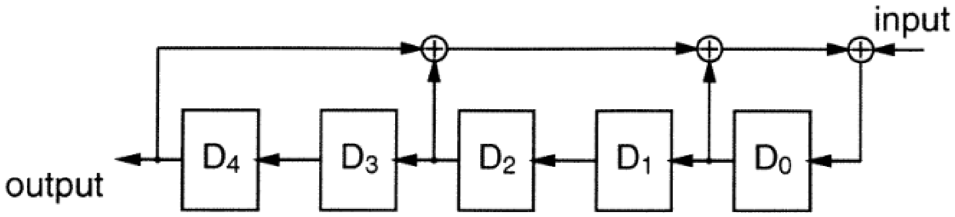
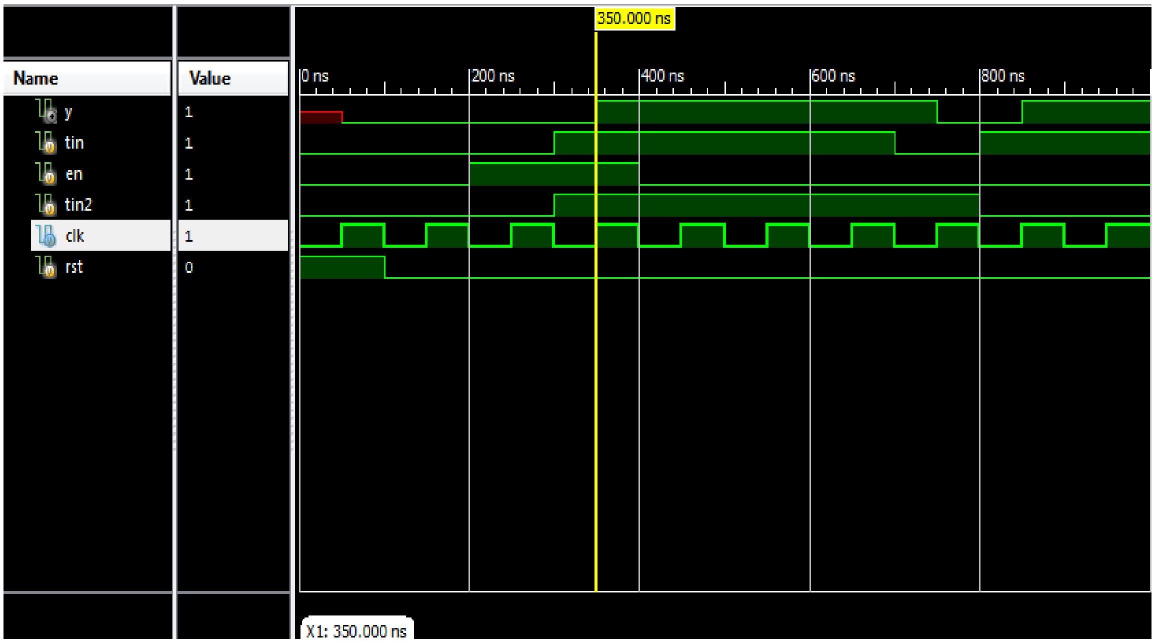


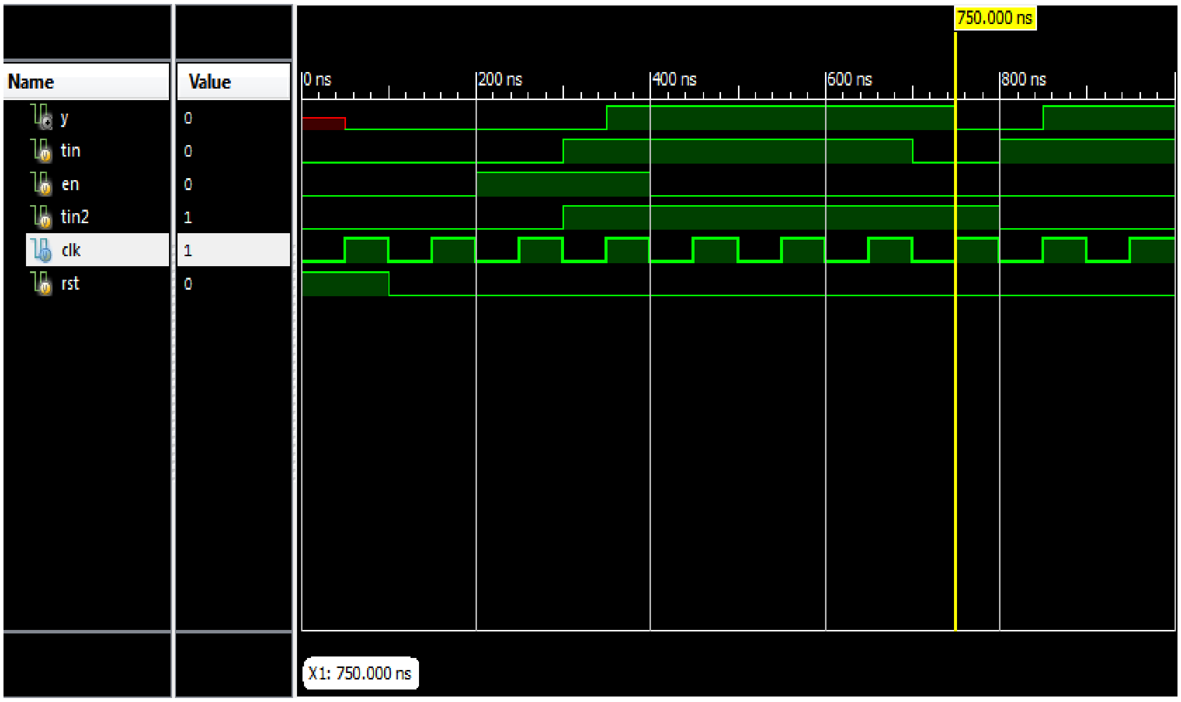
Fig. 5.3 External LFSR

SIMULATION RESULTS

Waveforms for Stuck at Faults:

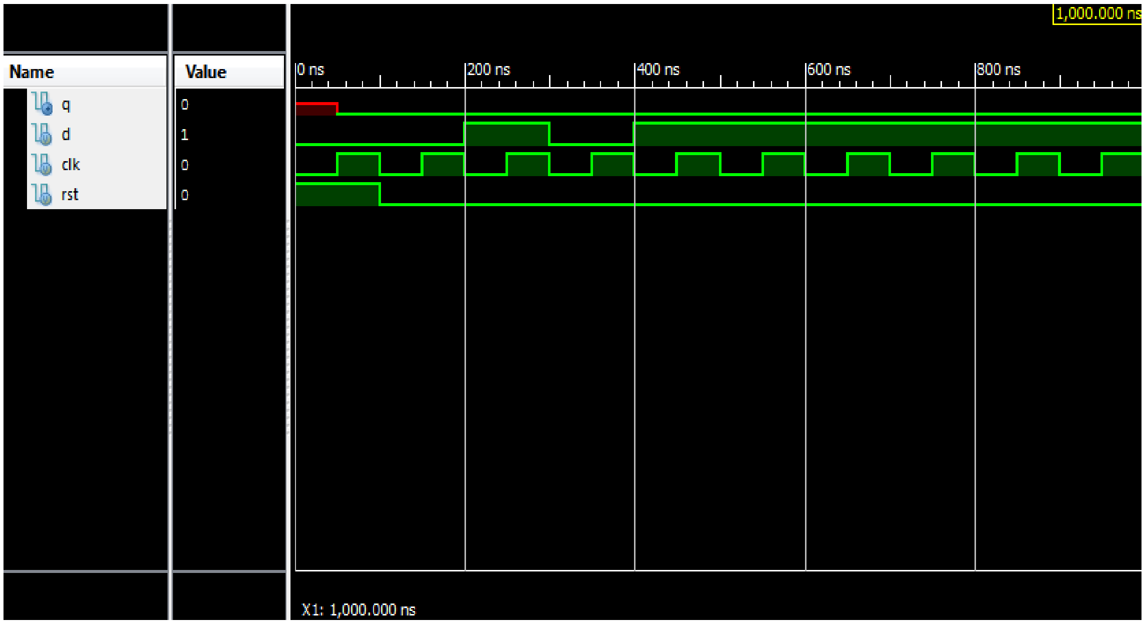


Name	Value
y	1
tin	1
en	1
tin2	1
clk	1
rst	0

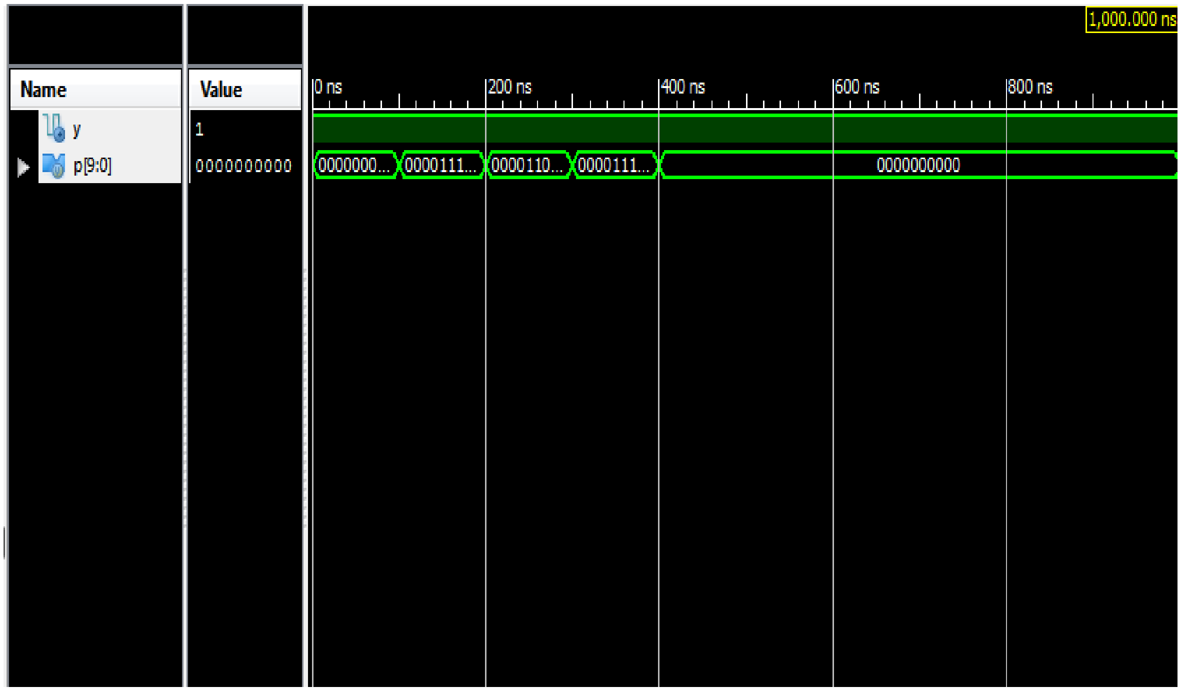


Name	Value
y	0
tin	0
en	0
tin2	1
clk	1
rst	0

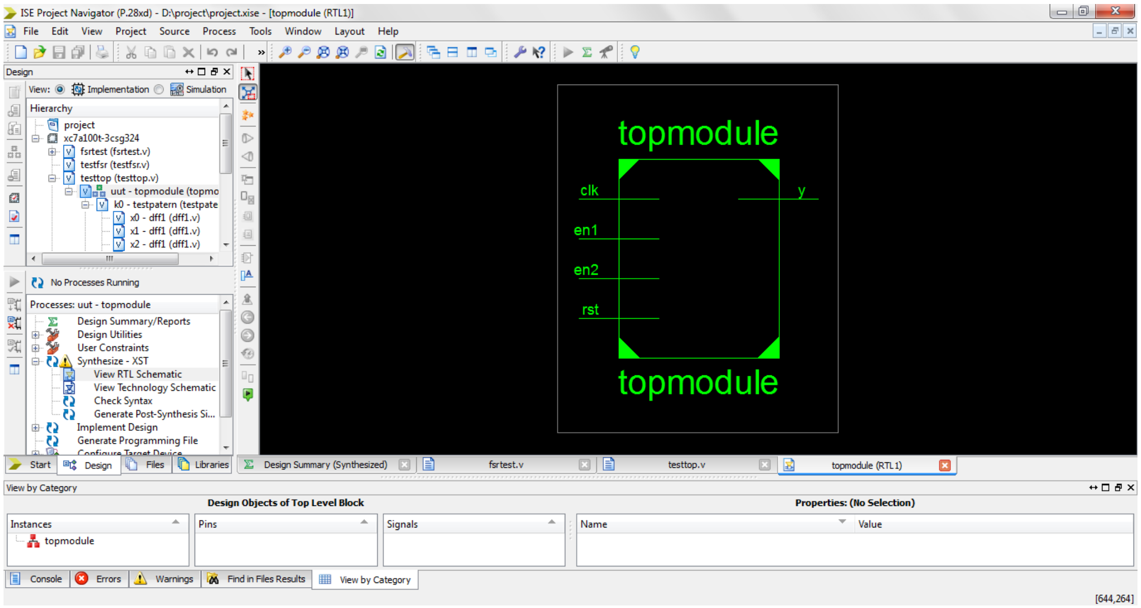
Test Pattern Generator:



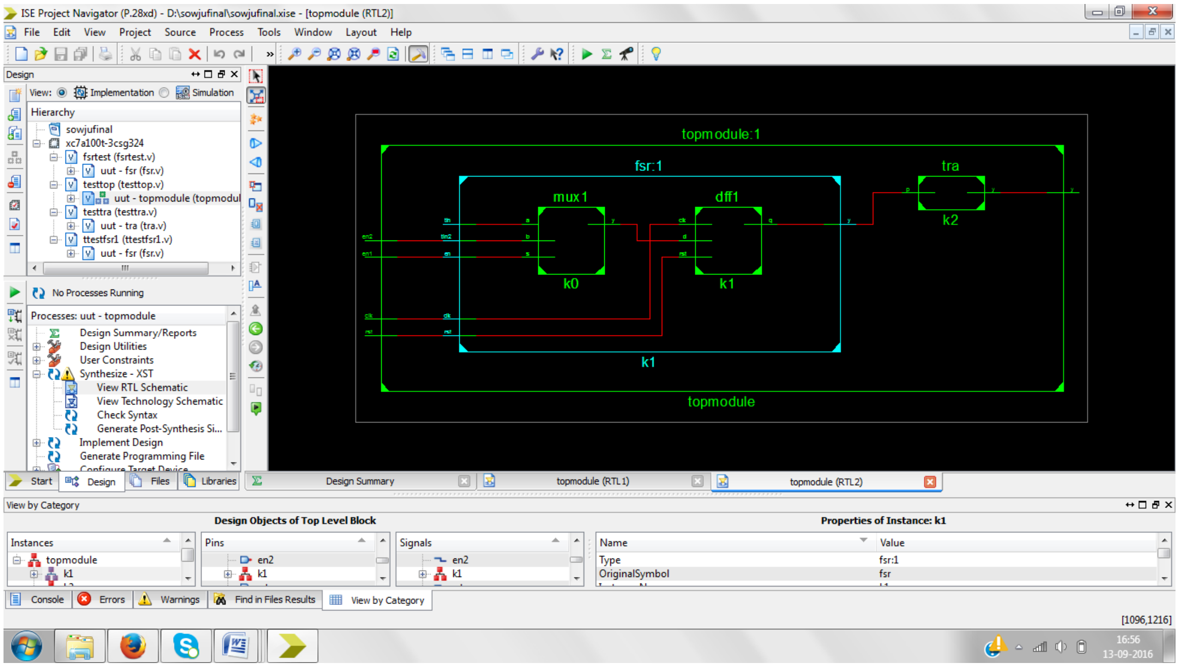
Test Response Analyzer:



Top Module:

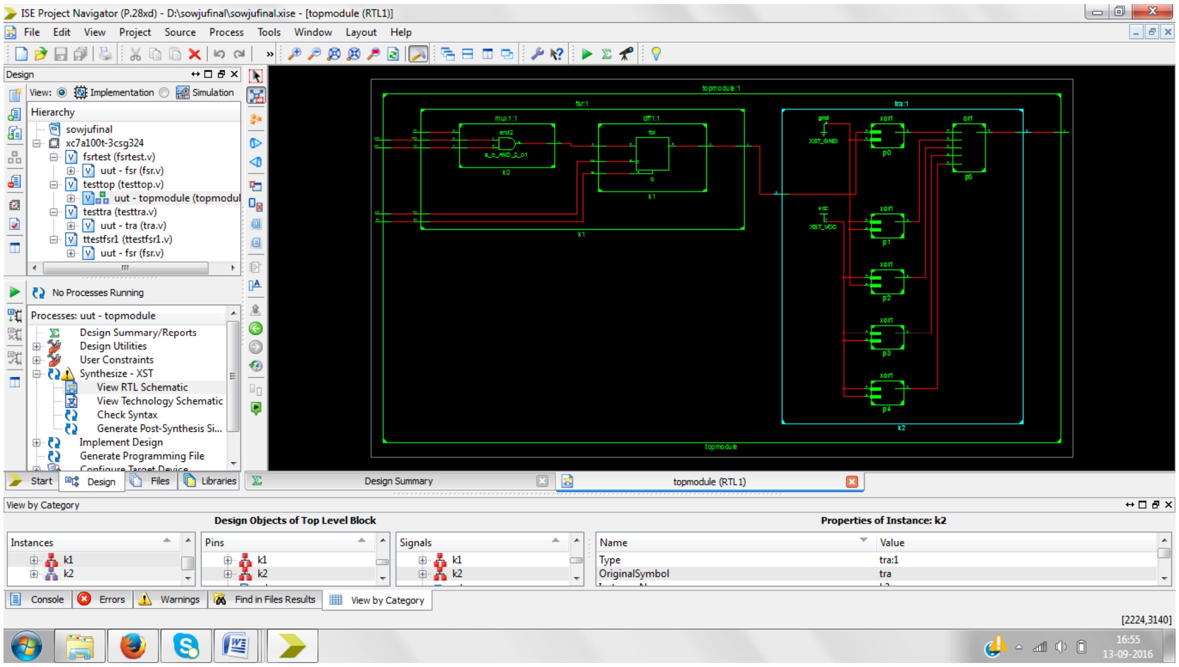

ISE Project Navigator (P.28xd) - D:\project\project.xise - [topmodule (RTL1)]
File Edit View Project Source Process Tools Window Layout Help
Design
View: Implementation Simulation
Hierarchy
project
xc7a100t-3csg324
fsrtest (fsrtest.v)
testfsr (testfsr.v)
testtop (testtop.v)
uut - topmodule (topmo
k0 - testpatern (testpate
x0 - dff1 (dff1.v)
x1 - dff1 (dff1.v)
x2 - dff1 (dff1.v)
No Processes Running
Processes: uut - topmodule
Design Summary/Reports
Design Utilities
User Constraints
Synthesize - XST
View RTL Schematic
View Technology Schematic
Check Syntax
Generate Post-Synthesis Si...
Implement Design
Generate Programming File
Configure Target Device
topmodule
clk
en1
en2
rst
y
topmodule
Start Design Files Libraries
Design Summary (Synthesized)
fsrtest.v
testtop.v
topmodule (RTL1)
View by Category
Design Objects of Top Level Block
Properties: (No Selection)
Instances
topmodule
Pins
Signals
Name
Value
Console Errors Warnings Find in Files Results View by Category
[644,264]

Internal Structure of Top Module:



Name	Value
Type	fsr:1
OriginalSymbol	fsr

RTL Schematic:



ISE Project Navigator (P.28xd) - D:\sowjufinal\sowjufinal.xise - [topmodule (RTL1)]

View: ● Implementation ○ Simulation

Hierarchy
- sowjufinal
- xc7a100t-3csg324
  - fsrtest (fsrtest.v)
    - uut - fsr (fsr.v)
  - testtop (testtop.v)
    - uut - topmodule (topmodul
  - testtra (testtra.v)
    - uut - tra (tra.v)
  - ttestfsr1 (ttestfsr1.v)
    - uut - fsr (fsr.v)

No Processes Running

Processes: uut - topmodule
- Design Summary/Reports
- Design Utilities
- User Constraints
- Synthesize - XST
  - View RTL Schematic
  - View Technology Schematic
  - Check Syntax
  - Generate Post-Synthesis Si...
- Implement Design
- Generate Programming File
- Configure Target Device

View by Category

Design Objects of Top Level Block

Instances	Pins	Signals
k1	k1	k1
k2	k2	k2

Properties of Instance: k2

Name	Value
Type	tra:1
OriginalSymbol	tra

[2224,3140]

16:55
13-09-2016

ADVANTAGES & APPLICATIONS

Advantages:

- It causes no performance degradation.
- It requires a small set of deterministic tests to cover 100% of single stuck-at faults. Thus, the test execution time is much shorter.
- It has a higher resistance against stuck-at fault**.**

Applications:

- Cryptography
- Games

CONCLUSION & FUTURESCOPE

Conclusion:

To summarize, the presented method has the following advantages compared to the traditional pseudo-random pattern-based LBIST using scan:

1) It causes no performance degradation.

2) It requires a small set of deterministic tests to cover 100% of single stuck-at faults. Thus, the test execution time is much shorter (at least two orders of magnitude).

3) It has a higher resistance against stuck-at fault.

Future scope:

In Future we are replacing the enable in and enable out both are connected to the multiplexer due to that logic performance will be improved.

REFERENCES:

[1] T. Good and M. Benaissa, "ASIC hardware performance," New Stream Cipher Designs: The eSTREAM Finalists, LNCS 4986, pp. 267–293, 2008.

[2] G. Becker, F. Regazzoni, C. Paar, and W. P. Burleson, "Stealthy dopantlevel hardware Trojans," Proceedings of Cryptographic Hardware and Embedded Systems (CHES'2013), LNCS 8086, pp. 197–214, 2013.

[3] T. W. Cusick and P. Stanic ˇ a,ˇ Cryptographic Boolean functions and applications. San Diego, CA, USA: Academic Press, 2009.

[4] S. Reddy, "Easily testable realizations for logic functions," IEEE Transactions on Computers, vol. 21, no. 11, pp. 1183–1188, 1972.

[5] R. K. Brayton, C. McMullen, G. Hatchel, and A. SangiovanniVincentelli, Logic Minimization Algorithms For VLSI Synthesis. Kluwer Academic Publishers, 1984.

[6] M. Abramovici, M. A. Breuer, and A. D. Friedman, Digital Systems Testing and Testable Design. Jon Willey and Sons, New Jersey, 1994.

[7] D. H. Green, "Families of Reed-Muller canonical forms," International Journal of Electronics, vol. 70, pp. 259–280, 1991.

[8] C. Canniere and B. Preneel, "Trivium," ` New Stream Cipher Designs: The

eSTREAM Finalists, LNCS 4986, pp. 244–266, 2008.

[9] M. Hell, T. Johansson, A. Maximov, and W. Meier, “The Grain family of stream ciphers,” New Stream Cipher Designs: The eSTREAM Finalists, LNCS 4986, pp. 179–190, 2008.

[10] S. Golomb, Shift Register Sequences. Aegean Park Press, 1982.

[11] E. McCluskey, “Built-in self-test techniques,” IEEE Design and Test of Computers, vol. 2, pp. 21–28, 1985.

[12] H.-J. Wunderlich, “BIST for systems-on-a-chip,” Integration, the VLSI Journal, vol. 26, no. 1-2, pp. 55 – 78, 1998.

[13] K. Chakrabarty, “Modular testing and built-in self-test of embedded cores in system-on-chip integrated circuits,” in The Embedded Systems Handbook (R. Zurawski, ed.), pp. 27–2–27–27, CRC Press, 2006.

[14] G. Hetherington, T. Fryars, N. Tamarapalli, M. Kassab, A. Hassan, and J. Rajski, “Logic BIST for large industrial designs: real issues and case studies,” in Proceedings of International Test Conference (ITC’1999), pp. 358 – 367, 1999.

[15] M. Tehranipoor and F. Koushanfar, “A survey of hardware Trojan taxonomy and detection,” IEEE Design Test of Computers, vol. 27, no. 1, pp. 10–25, 2010.