DESIGN OF LFSR-BASED TEST PATTERN GENERATOR FOR ANALYSIS OF FAULT REDUCTION TECHNIQUES

KARRI CHAKRARAO[1],D VIJENDRA KUMAR[2].

[1]M.TECH (VLSI) IN DEPT OF ECE IN GODAVARI INSTITUTE OF ENGINEERING AND TECHNOLOGY IS AN ENGINEERING COLLEGENH-16, CHAITANYA KNOWLEDGE CITY, RAJAHMUNDRY, and ANDHRA PRADESH 533296.

[2]ASSISTANT PROFESSORIN DEPT OF ECE IN GODAVARI INSTITUTE OF ENGINEERING AND TECHNOLOGY IS AN ENGINEERING COLLEGENH-16, CHAITANYA KNOWLEDGE CITY, RAJAHMUNDRY, and ANDHRA PRADESH 533296.

ABSTRACT: In a random testing environment, a significant amount of energy is wasted in the LFSR and in the CUT by useless patterns that do not contribute to fault dropping. Another major source of energy drainage is the loss due to random switching activity in the CUT and in the scan path between applications of two successive vectors. In this work, a new built-in self-test (BIST) scheme for scan-based circuits is proposed for reducing such energy consumption. A mapping logic is designed which modifies the state transitions of the LFSR such that only the useful vectors are generated according to a desired sequence. Further, it reduces test application time without affecting fault coverage. Experimental results on benchmark circuits reveal a significant amount of energy savings in the LFSR during random testing.

KEY WORDS: Linear-feedback shift register (LFSR)-based test generation, multi-cycle tests, test compaction, test data compression.

INTRODUCTION

With the emergence of mobile computing and communication devices, design of low-energy VLSI systems has become a major concern in circuit synthesis. A significant component of the energy consumed in CMOS circuits is caused by the total amount of switching activity (SA) at various circuit nodes during operation. The energy dissipated at a circuit node is proportional to the total number of $0 \rightarrow 1$ and $1 \rightarrow 0$ transitions the logic signals undergo at that node multiplied

by its capacitance (which depends on its fan-out and its transistor implementation). Energy consumption in an IC may be significantly higher during testing due to increased SA than that needed during normal (system) mode, which can cause excessive heating and degrade circuit reliability. The average-power optimization help extend the battery life in mobile applications. The generation of multi cycle tests for test compaction becomes more complex when test data compression is used. In one of the commonly used test data compression methods, a test is compressed into a seed for a linear-feedback shift register (LFSR). The on-chip decompression logic uses the LFSR to apply the test to the circuit. A seed is typically computed based on an incompletely specified test cube by solving a set of linear equations that relate the bits of the seed with the specified values of the test cube. With this process, optimizing a multicycle test to increase the number of faults it detects requires a seed to be recomputed after every step that modifies the test, and some modifications of the test cannot be accepted because a seed does not exist for the modified test. Motivated by these observations, the goal of this paper is to develop a procedure for computing seeds for LFSR-based generation of multicycle tests that are effective for test compaction. To avoid sequential test generation, the procedure uses a single-cycle test set similar, and optimizes the multicycle tests to increase the numbers of faults they detect. In contrast, the procedure optimizes the compressed multicycle tests in order to avoid producing tests for which seeds do not exist. The linear feedback shift register (LFSR) is most frequently used as a test pattern generator (TPG) in low area overhead built in self-test. BIST procedure can adequately reduce the adversity and intricacy of VLSI testing, by considering this actuality that an LFSR can be made with slight area overhead and decreases switching activity. In typical BIST architecture, the LFSR is frequently used in test pattern generation and output analyser. The leading fault of this architecture is that pseudorandom Patterns generated by LFSR lead to extremely high switching activities in circuit-under-test (CUT) which can motive redundant power dissipation. They can also spoilage the circuit and reduce product yield and period. BIST is a design-for-testability approach that places the testing behaviour physically with the circuit under test (CUT). LFSR are the sequential logic circuits used to create pseudorandom binary sequences (PRBS) s. To achieve the goal of producing compressed multicycle tests that are effective for test compaction, the procedure described in this paper optimizes the seed si, the primary input vector vi, and the number of functional clock cycles li together to increase the number of faults that the test detects. By considering the seed si directly, the procedure optimizes

the scan-in state pi, and avoids modifications of pi for which a seed does not exist. Moreover, the singlecycle test set that the procedure uses as guidance does not need to be compressed. To accommodate this case, the procedure initializes the seed si randomly, and not based on the scan-in state qi of a singlecycle test. It is thus possible to use a compact single-cycle test set that is not constrained by the LFSR. The possibility of optimizing a seed si was used to modify seeds that produce fault detection tests into seeds that produce diagnostic tests. The modification of a seed si is implemented by complementing bits of si one by one, and recomposing the test ti that the LFSR produces. A bit complementation is accepted when ti satisfies certain objectives (in these objectives are related to the generation of diagnostic tests). In the procedure described in this paper, bits of si and vi, as well as the value of li, are modified together in order to produce an effective multicycle test. The target faults in this paper are single stuck-at faults. The procedure is developed assuming that an LFSR is given. This paper also describes a modified binary search process for selecting an LFSR out of a given set of available LFSRs.

RELATED WORK

As the complexity of modern chips increases, external testing with ATE becomes extremely expensive. Instead, built-in self-test (BIST) is becoming more common in the testing of digital VLSI circuits since it overcomes the problems of external testing using ATE. BIST test patterns are not generated externally as in case of ATE; instead they are generated internally using some parts of the circuit, also the responses are analysed using other parts of the circuit. When the circuit is in test mode, test patterns generators (TPGs) generate patterns that are applied to the CUT, while the signature analyser (SA) evaluates the CUT responses. One of the most common TPGs for exhaustive, pseudo-exhaustive, and pseudorandom TPG is the linear feedback shift register (LFSR). LFSRs are used as TPGs for BIST circuits because, with little overhead in hardware area, a normal register can be configured to work as a test generator, and with an appropriate choice of the location of the XOR gates, the LFSR can generate all possible output test vectors (with the exception of the 0svector, since this will lock the LFSR). The pseudorandom properties of LFSRs lead to high fault coverage when a set of test vectors is applied to the CUT compared with the fault coverage obtained using normal counters as TPGs. Also LFSRs can be configured to act as signature analyzers for the responses obtained from the

CUT. Despite their simple appearance, LFSRs are based on complex mathematical theory that helps explain their behavior as TPGs or SAs.

When the LFSR is used to generate test patterns for full scan-chain sequential circuits, one of its flip-flop outputs is connected with the scan-chain input. In this case the LFSR will be considered as a one-dimensional TPG. The main problem of this configuration is the long time needed to scan-in a test vector which is equivalent to the number of flip-flops in the scan-chain. In order to speed-up the scanning of test vectors (i.e. reducing test application time), the flip flops in the circuit can be divided into groups, and each group forms a separate scan-chain. This approach is called multiple scanchains. In this case a two- dimensional TPG should be used to scan-in test vectors in the multiple scan-chains in parallel. The LFSR can be used for this purpose, where different flip-flops outputs can be connected with the different scan-chain inputs and the outputs of the scan-chains are connected with a multiple input signature register (MISR).

EXISTING WORK

The scan-in and scan-out operations of a test, a single cycle test has a single functional clock cycle, while a multicycle test has one or more functional clock cycles. Multicycle tests were considered. Their effectiveness for test compaction was demonstrated and results from the following observations. During a functional clock cycle of a test, the combinational logic of the circuit receives an input pattern that can be used for detecting faults. A larger number of functional clock cycles allows more faults to be detected. As a result, a multicycle test may detect more faults than a single-cycle test. With more detected faults for every test, the number of tests is reduced.

Multicycle Test Scheme

Fig. 1 shows the proposed multi-cycle test scheme with partial observation of flip-flops. Unlike the case of [10], we use scan-based logic BIST instead of full-scan test. This is because logic BIST with reseeding will be more feasible for field test with small memory resource than compression-based deterministic full-scan test in the current technologies. In this scheme, we target to reduce the number of seeds that is needed for achieving the given fault coverage or to

improve the fault coverage for the given number of seeds. In the figure, input vectors to the combinational circuit under test (CUT) is provided to flip-flops (FFs) through scan chains from a test pattern generator (TPG), which can be a linear feedback signature register (LFSR) or a cellular Automata (CA) with reseeding capability. The output values of CUT are captured into FFs at each clock cycle during test mode (capture mode). They are scanned out to Compactor A, which consists of an XOR based space compactor and a MISR. At the same time, a part of FFs are connected directly (i.e. without scan-out) to additional Compactor B, which also consists of an XOR based space compactor and a MISR. Note that we refer to the FFs connected to Compactor B as partial FFs in this paper. Primary inputs and outputs are isolated from the CUT using boundary scan cells in case of at-speed test. The figure is simplified to a single scan chain, but it can easily be enhanced to multiple scan chains.

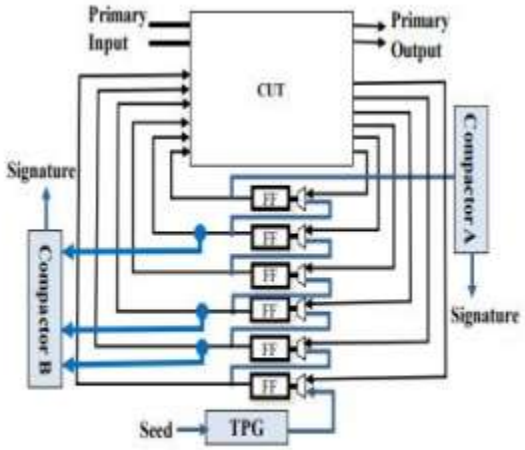


Fig.1 Block Diagram of Multicycle test Scheme

PROPOSED SYTEM

TRANSITION FAULT MODEL:The transition fault model captures delay defects on slow-to-rise transition or a slow-to-fall transition at a specific line in the circuit. In general transition faults are used for its simplicity in modeling scheme which can able to spot the defects and that will affect the delays at the input or output of the gate circuits. Under scan based test circuit,

these faults are combined with an extra delay which is very large enough to cause the delay of any path through the fault site to exceed the clock period. A slow-to-rise transition fault at line c is a 3 input circuit and Input b and d have constant values. The value of the input a change from 0 to 1 at time point t1, hence the rising transition occurs at line a. The transition propagates through the circuit, when the circuit is fault free, the value of the output e is to be 1 at the time point t2, here t2-t1 is the clock period and also due to the slow-to-rise transition fault at line c, the value of the output e remains o at the time period t2.

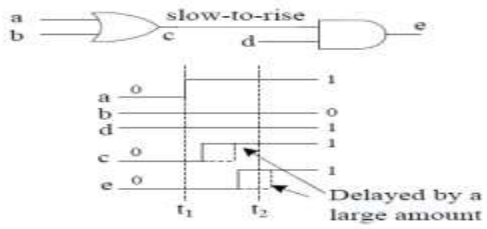


Figure 2. Transition Fault Model

STUCK-AT FAULT MODEL: Single stuck line is one of the fault model used in digital circuits and it is also used in post manufacturing of testing and not for design testing. Actually the model assumes one line or one node in the digital circuit is may stuck at logic 0 (Low) or logic 1 (High).when the line is stuck is called as fault. Digital Circuit can be divided into two types as explained below: 1. Gate level or combinational circuits which contain no storage memory. 2. Sequential circuits which contain storage memory. These faults model is applies to gate level circuits, or a block of sequential circuit which can be able to separate from the storage elements as mentioned above. A gate-level circuit would be completely tested by applying all possible inputs and checking which gave the right outputs, but it is completely impractical: For example an adder supposed to add two 32 bit numbers, require 264 = 1.8*1019 tests, taking 58 years at 0.1 ns/test.

The stuck-at fault model assumes that only one input will be faulty on one gate at a time, assumes that if more are faulty, a test which can detect any single fault, can easily find multiple faults in the circuit. Each of the faults is called a single stuck-at-0 (low) or a single stuck-at-1(high) fault, respectively.

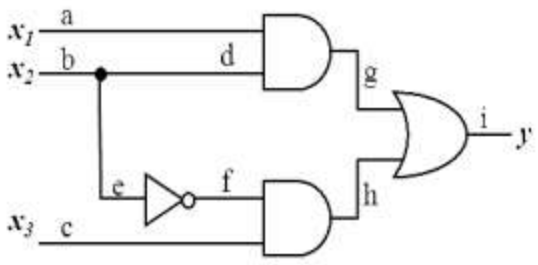


Figure 3. Stuck-At Fault Model

In the above circuit diagram contains 3 input lines are x1, x2, x3 and 1 output line y. Here Any line can be Stuck-at-0 (SA0) and Stuck-at-1(SA1) respectively. Consider the fault types are k=2. The fault sites n: 9 and the single faults are 2x9=18 faults can occurs.

$x_1x_2x_3$	000	001	010	011	100	101	110	111
y	0	1	0	0	0	1	1	1
a SA0	0	1	0	0	0	1	0	0
a SA1	0	1	1	1	0	1	1	1
b SA0	0	1	0	1	0	1	0	1
b SA1	0	0	0	0	1	1	1	1
c SA0	0	0	0	0	0	0	1	1
c SA1	1	1	0	0	1	1	1	1
d SA0	0	1	0	0	0	1	0	0
d SA1	0	1	0	0	1	1	1	1
e SA0	0	1	0	1	0	1	1	1
e SA1	0	0	0	0	0	0	1	1
f SA0	0	0	0	0	0	0	1	1
f SA1	0	1	0	1	0	1	1	1
g SA0	0	1	0	0	0	1	0	0
g SA1	1	1	1	1	1	1	1	1
h SA0	0	0	0	0	0	0	1	1
h SA1	1	1	1	1	1	1	1	1
i SA0	0	0	0	0	0	0	0	0
i SA1	1	1	1	1	1	1	1	1

Table 1. Truth Table for Fault free behavior and behavior of all possible stuck-at faults

BENCHMARK CIRCUIT: The ISCAS'89 benchmarks have a set of 31 digital sequential circuits. The mentioned benchmarks were distributed on tape to participants of the Special Session on Sequential Test Generation, Int. Symposium on Circuits and Systems, May 1989, and are partially characterized in F. Brglez, D. Bryan, K. Kozminski in "Combinational Profiles of Sequential Benchmark Circuits", Proc. IEEE Int. Symposium on Circuits and Systems, pp. 1929-1934, May 1989. Each of the circuit is described in to two files as mentioned below: (1) A generic gate-level netlist with a list of equivalence collapsed faults. (2) A simple translator is included to read and write the netlist. Here no schematic diagrams. ⟩ S27 implies Testing ⟩ S208 implies Fractional Multiplier ⟩ S298 implies Traffic Light Controller Based on the above S27 Testing circuit, we have generated simulation output for both the transition fault model and stuck-at fault model. The above S27 Testing circuit implies fault detection in the circuit based on the transition fault model and stuck-at fault model techniques.

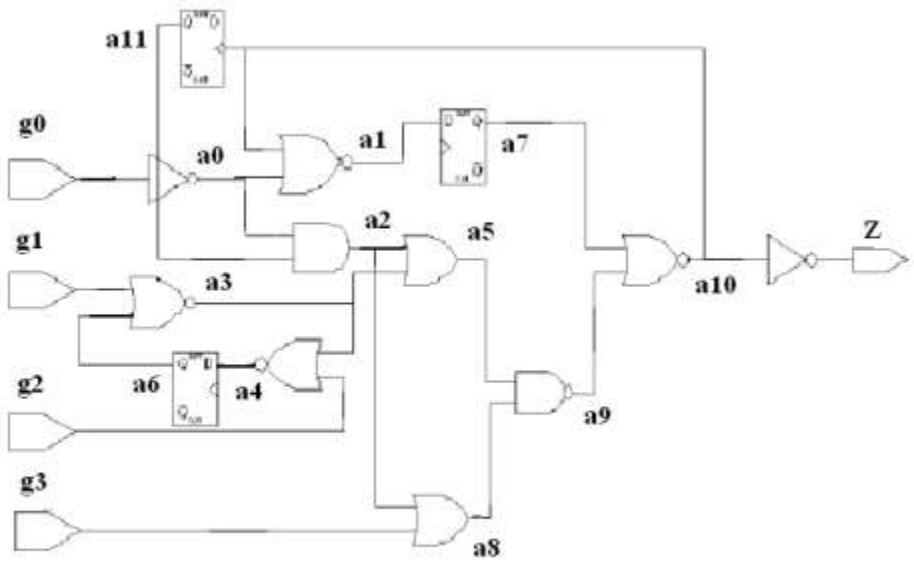


Figure 4. S27-Testing for Transition and Stuck-At Fault model

S27 –Testing: S27 benchmark circuit is the standard sequential circuit for testing. Here we are used S27 benchmark circuit for a testing of the circuit. We are applying test vectors as inputs to the S27 benchmark circuit. Here I0, I1, I2, I3 are input of this below specified circuit. Generally S27 circuit have three scan circuits and then its scan inputs are 2^3=8. Scan-in-state inputs are s0, s1, s2 and its scanned out denoted as s. LFSR register values initially we considered as "1010".We note that it is unspecified in the two un scanned state variables which implies can be

used as scan-in state. In that more specified values under, more faults are to be detected. Actually, to obtain a shorter test from the given test of length, we assume search for the highest time unit which has unspecified values on all the unscanned since it is selected from the unspecified on the un scanned state variables. For a full-scan circuit, all the tests will be a length of two, due to the fact of all the states are fully-specified.

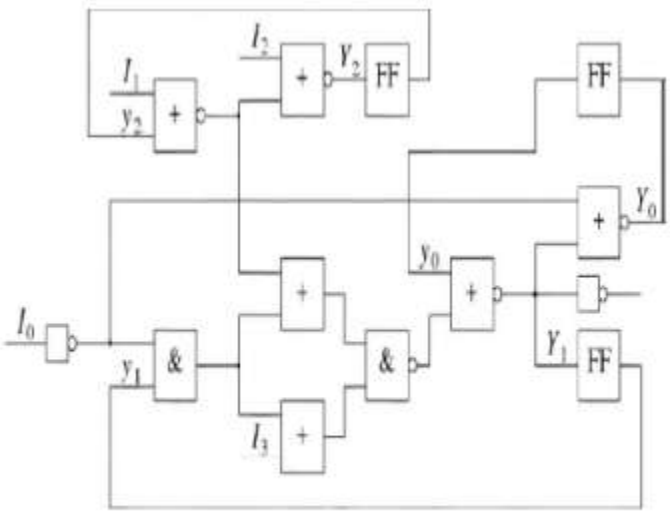


Figure 5. S27 Benchmark Circuit

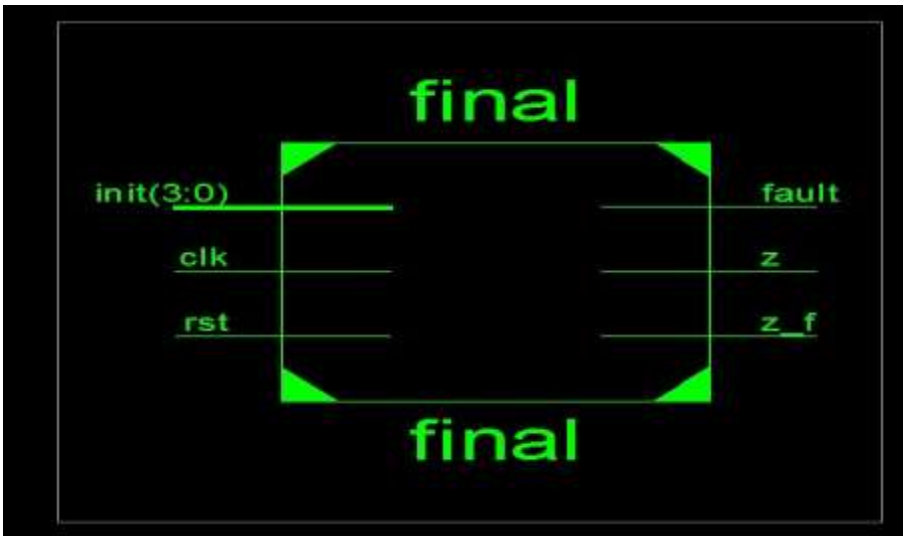
u	Lfsr(u)	I value	o/p of s27
0	101 011 100 100	1 0 0 1	X
1	010 101 110 010	1 1 1 0	X
2	001 010 111 001	0 0 1 0	X
3	100 101 011 100	1 1 0 1	1
4	010 010 101 110	1 0 0 1	1
5	001 001 010 111	0 0 0 1	1
6	100 100 101 011	1 1 0 0	0
7	110 010 010 101	1 0 0 1	0
8	111 001 001 010	1 0 0 0	0
9	011 100 100 101	1 1 0 1	0
10	101 110 010 010	1 1 0 0	0
11	010 111 001 001	1 1 0 0	0
12	101 011 100 100	1 0 0 1	0
13	010 101 110 010	1 1 1 0	0
14	001 010 111 001	0 0 1 0	0
15	100 101 011 100	1 1 0 1	1

Table 2. Output of S27 Benchmark Circuit

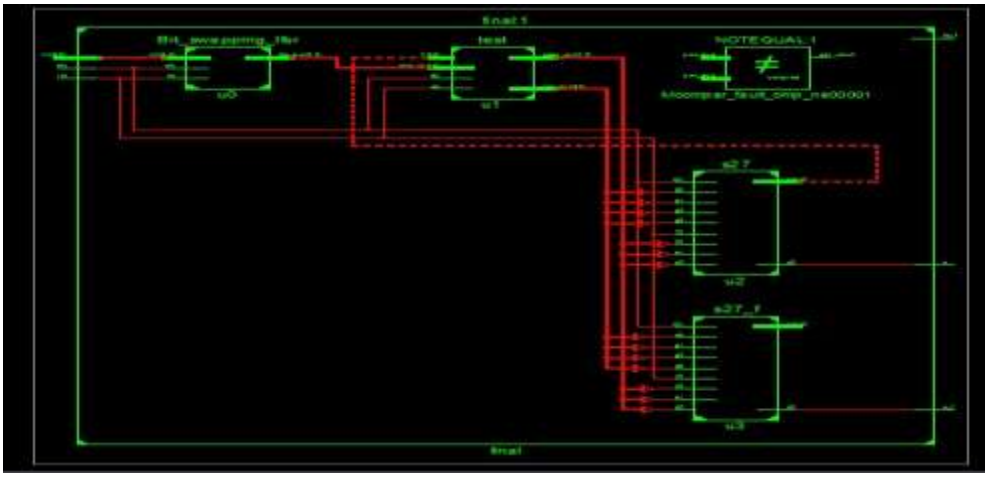
SIMULATION RESULTS

The following result for the Transition and Stuck-At Fault Model are successfully simulated by using Xilinx Simulator. The result shown in Transition Fault Model by giving variable parameters such as Clk,reset,fa,fa1,fa2,z,z1,z2, etc and the Output is identified in tmp_ram Parameter and the final value is 11111.The result shown in Stuck-At Fault Model by giving variable parameters such as Clk,reset,fa,fa1,fa2,z,z1,z2, etc and the Output is simulated using Fault Simulator in tmp_ram Parameter and the final value is 11111.

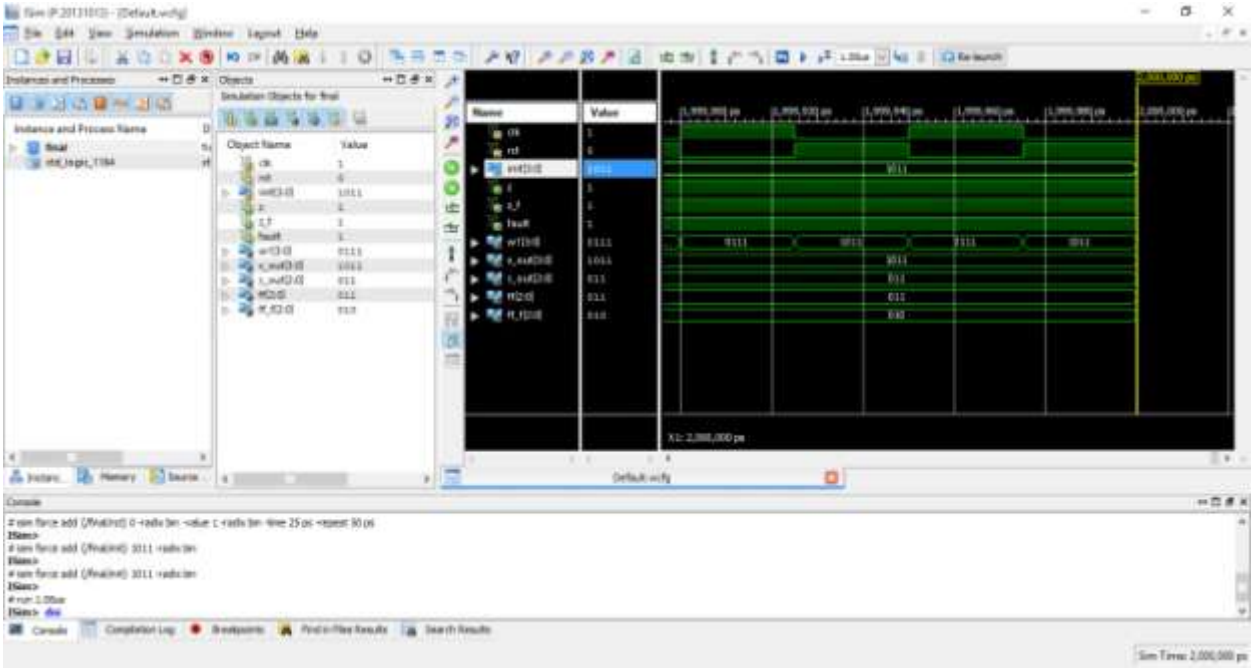
RTL



INTERNAL BLOCK DIAGRAM



Simulation Output for Stuck-At Fault mode



Area

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	13	9,312	1%	
Number of 4 input LUTs	26	9,312	1%	
Number of occupied Slices	18	4,656	1%	
Number of Slices containing only related logic	18	18	100%	
Number of Slices containing unrelated logic	0	18	0%	
Total Number of 4 input LUTs	26	9,312	1%	
Number of bonded IOBs	9	232	3%	
Number of BUFGMUXs	1	24	4%	
Average Fanout of Non-Clock Nets	2.64			

Delay

```
    ----------------------------------------
    Total                      6.312ns (4.907ns logic, 1.405ns route)
                                       (77.7% logic, 22.3% route)

=========================================================================


Total REAL time to Xst completion: 11.00 secs
Total CPU time to Xst completion: 11.25 secs
```

Power

CONCLUSION

LFSR Based Test generation for Transition fault model and stuck-at fault model is successfully tested in S27 Benchmark Circuit. In our project we have used Sequential Circuit to Rising and falling transitions; based upon the transitions may result in changes in the delay faults in Transition Fault Model. Basically, it captures delay defects on slow-to-rise transition or a slow-to-fall transition at a specific line in the circuit and we have also used Stuck-At Fault model which is used for fault simulators and ATPG tools. Here the results show that faults are detected as well as simulated efficiently for the test pattern generation.

FUTURE WORK In future, we will be use Low Power LFSR for low power consumption. Techniques like test point insertion can also be tried to improve the fault coverage for undetectable faults

REFERENCES

[1] M. Pallavi, “Low Power Testing Using Re-Configurable Johnson Counter And Scalable SIC Counter”, IEEE ICCSP conference 2015.

[2] R.C. Ravishankar Reddy and V.Sumalatha,, “ Implementation Of Area And Power Efficient Built In Self-test Pattern Generator”, IEEE WISPNET conference 2016.

[3] B. Michael L. and A. Vishwani D., Essentials of Electronic Testing for digital, Memory and Mixed Signal VLSI Circuits, Kluwer Academic Publishers (2000).

[4] N. Lubna and C. Tarana A., “Design of Low Transition Pseudo-Random Pattern Generator for BIST Applications”, International Journals of Computer Applications, Volume 87, number 15, February 2014.

[5] K. Arun,” Design of Low Power Test Pattern Generator For Built In Self- Test (BIST) Circuits”, International Journal of Technical Research and Applications, Volume 4, PP.210-214, June 2016.

[6] S. Abdallaitf, Abu-Issa, T. Iyad K. and G. Wasel T., “ SR-TPG: A Low Transition Test Pattern Generator for Test-per-Clock and Test-per-Scan BIST”, IEEE 10th International Design & Test Symposium, 2015.

[7] M. Puezko, V.N. Yarmolik, “Two-pattern test generation with low power consumption based on LFSR”, Information processing and security systems, Springer-Verlag, 2005, pp. 159-166.

[8]K.Jayasheeli and S.Maria Antony, “Low Power Test Pattern Generation of Logic Built in Self-Test”, International Journal of Latest Trends in Engineering and Technology, Volume 6, November 2015