International Journal of Advanced Science and Research
ISSN: 2455-4227, Impact Factor: RJIF 5.12
www.newresearchjournal.com/science
Volume 1; Issue 5; May 2016; Page No. 28-34

Implementation of functional & broadside testing using a fixed hardware

Manikumar P, Karthik T, Haritha Sree P, Leepika S, Sunesha Y
Department of Electronics & Communication Engineering, VITS College of Engineering, Sontyam, Visakhapatnam, Andhra Pradesh, India.

Abstract
In this proposed method we are test 4-bit input ISCAS- 27-channel interrupt controller. This paper described an on-chip test generation method for functional broadside tests. The hardware was based on the application of primary input sequences starting from a known reachable state, thus using the circuit to produce additional reachable states. Random primary input sequences were modified to avoid repeated synchronization and thus yield varied sets of reachable states. Two-pattern tests were obtained by using pairs of consecutive time units of the primary input sequences. The hardware structure was simple and fixed, and it was tailored to a given circuit only through the following parameters: the length of the LFSR used for producing a random primary input sequence; the length of the primary input sequence; the specific gates used for modifying the random primary input sequence; the specific gate used for selecting applied tests; and the seeds for the LFSR. With the proposed on-chip test generation method, the circuit is used for generating reachable states during test application. This alleviates the need to compute reachable states offline.

Keywords: hardware, Functional Broadside testing, D flip-flop

1. Introduction

Broadside and Functional Broadside testing

A deterministic broadside test generation procedure is proposed for transition path delay faults. Under this fault model, a path delay fault is detected if and only if all the individual transition faults along the path are detected by the same test. This is important for detecting both small and large delay defects. To handle the complexity of test generation, the procedure consists of five sub-procedures: a test generation procedure for transition faults, a pre-processing procedure that identifies undetectable transition path delay faults without performing test generation, a fault simulation procedure that identifies transition path delay faults that are detected by the tests for transition faults, a heuristic procedure similar to dynamic test compaction for transition faults that generates tests without backtracking on decisions made for previously detected faults, and a complete branch-and-bound procedure. Experimental results show that for most of the transition path delay faults in benchmark circuits either a test is found or the fault is identified as undetectable.

Over testing due to the application of two patterns can-based tests was described in [1, 3]. Over testing is related to the detection of delay faults under non-functional operation conditions. One of the reasons for these non-functional operation conditions is the following. When an arbitrary state is used as a scan-in state, a two-pattern test can take the circuit through state-transitions that cannot occur during functional operation. As a result, slow paths that cannot be sensitized during functional operation may cause the circuit to fail [1]. In addition, current demands that are higher than those possible during functional operation may cause voltage drops that will slow the circuit and cause it to fail [2, 3]. In both cases, the circuit will operate correctly during functional operation.

Functional broadside tests [4]. ensure that the scan-in state is a state that the circuit can enter during functional operation, or a reachable state. As broadside tests [5], they operate the circuit in functional mode for two clock cycles after an initial state is scanned in. This results in the application of a two-pattern test.

Since the scan-in state is a reachable state, the two-pattern test takes the circuit through state transitions that are guaranteed to be possible during functional operation. Delay faults that are detected by the test can also affect functional operation, and the current demands do not exceed those possible during functional operation. This alleviates the type of over testing described in [1, 3]. In addition, the power dissipation during fast functional clock cycles of functional broadside tests does not exceed that possible during functional operation.

Test generation procedures for functional and pseudo-functional scan-based tests were described in [4, 6, 13]. The procedures generate test sets offline for application from an external tester. Functional scan-based tests use only reachable states as scan-in states. Pseudo-functional scan-based tests use functional constraints to avoid unreachable states that are captured by the constraints.

This work considers the on-chip (or built-in) generation of functional broadside tests. On-chip test generation reduces the test data volume and facilitates at-speed test application. On-chip test generation methods for delay faults, such as the ones described in [14, 15]. Do not impose any constraints on the states used as scan-in states. Experimental results indicate that an arbitrary state used as a scan-in state is unlikely to be a reachable state [4]. The on-chip test generation method from [16]. applies pseudo-functional scan-based tests. Such tests are not sufficient for avoiding unreachable states as scan-in states. The on-chip test generation process described in this work guarantees that only reachable states will be used.

It should be noted that the delay fault coverage achievable using functional broadside tests is, in general, lower than that achievable using arbitrary broadside tests as in [14, 15] or pseudo functional broadside tests as in [16]. This is due to the fact that functional broadside tests avoid unreachable scan-

instates, which are allowed by the methods described in [14, 16]. However, the tests that are needed for achieving this higher fault coverage are also ones that can cause over testing. They can also dissipate more power than possible during functional operation.

Only functional broadside tests are considered in this work. Under the proposed on-chip test generation method, the circuit is used for generating reachable states during test application.

This alleviates the need to compute reachable states or functional constraints by an offline process as in [4, 6, 13] and [16]. The underlying observation is related to one of the methods used in [4] for offline test generation, and is the following.

If a primary input sequence A is applied in functional mode starting from a reachable state, all the states traversed under A are reachable states. Any one of these states can be used as the initial state for the application of a functional broadside test. By generating A on-chip and ensuring that it takes the circuit through a varied set of reachable states, the on-chip test generation process is able to achieve high transition fault coverage using functional broadside tests based on A. It should be noted that, for the detection of a set of faults F, at most |F| different reachable states are required. This number is typically only a small fraction of the number of all the reachable states of the circuit. Thus, the primary input sequence A does not need to take the circuit through all its reachable states, but only through a sufficiently large number relative to |F|, in order to be effective for the detection of target faults.

The hardware used in this paper for generating the primary input sequence A consists of a linear-feedback shift-register (LFSR) as a random source [17] and of a small number of gates (at most six gates are needed for every one of the benchmark circuits considered). The gates are used for modifying the random sequence in order to avoid cases where the sequence takes the circuit into the same or similar reachable states repeatedly. This is referred to as repeated synchronization [18]. In addition, the on-chip test generation hardware consists of a single gate that is used for determining which tests based on will be applied to the circuit. The result is a simple and fixed hardware structure, which is tailored to a given circuit only through the following parameters.

1. The number of LFSR bits.
2. The length of the primary input sequence.
3. The specific gates used for modifying the LFSR sequence into the sequence.
4. The specific gate used for selecting the functional broadside tests that will be applied to the circuit based on.
5. Seeds for the LFSR in order to generate several primary input sequences and several subsets of tests.

The on-chip test generation hardware is based on the one described in [19]. It differs from it in the following ways.

1. The logic that produces the primary input sequence is designed in this paper to reduce the dependencies between the values assigned to the primary inputs, considering the following sources of dependency. In [19], for a circuit with n primary inputs and a parameter mod, the LFSR used for producing A has n + mode bits. The left-most bits are used for driving the primary inputs of the circuit, and the mod right-most bits are used for modifying the random sequence in order to avoid repeated synchronization. With this structure, all the primary input values are modified using the same function of the mod right-most bits of the LFSR. Thus, they are always modified together and to the same values. In addition, some primary inputs receive shifted values of the primary inputs immediately preceding them.

   The structure used in this paper reduces these dependencies between primary input values by using a (d.n)–bit LFSR for a circuit with primary inputs, where is a parameter such that d>mode. Every consecutive bits of the LFSR are used for producing the value of a different primary input. At most mod of the bits dedicated to a primary input are actually used for producing values for the input, including the modification of the input values in order to avoid repeated synchronization. Since the modification is done using different bits for every primary input, the dependencies between primary input values are reduced. In addition, the unused bits serve to reduce the dependencies between the values of different primary inputs further by avoiding cases where a primary input receives shifted values of the primary input immediately preceding it. With reduced dependencies, the primary input sequence A is more likely to take the circuit into a varied set of reachable states. As a result, a higher fault coverage is achieved for several of the circuits considered in [19]. In addition, other parts of the test generation hardware can be simplified compared with the design in [19] as discussed next.
2. Both [19] and this paper apply multiple primary input sequences in order to achieve the highest possible fault coverage. To select which tests will be applied to the circuit based on every sequence, the approach of [19] uses a different gate for every sequence. Since the number of sequences in [19] is significant, a large multiplexer and a significant number of gates are needed for this purpose. The approach in this paper fixes the gate used for test selection in advance, and ensures that all the primary input sequences used for the circuit fit with the preselected gate. In this way, a single gate is needed for test selection regardless of the number of sequences used, and there is no need for a multiplexer to distinguish between different sequences.
3. The lengths of the primary input sequences is varied in [19]. in order to control the number of tests applied to the circuit. In this paper, all the sequences have the same length. This makes the test application process uniform across different sequences.

The result is that the test generation hardware used in this paper has a simple and fixed structure, and it is independent of the number of sequences used. The sequences differ only in the seed used for the LFSR. The seeds can be stored on-chip, or a seed can be scanned in together with the initial state of the circuit before the application of every primary input sequence. The paper focuses on the generation of input test data, which is unique to functional broadside tests. For the output test data the paper assumes that an output compactor such as a multiple input shift-register (MISR) [17] will be used.

When the circuit-under-test is embedded in a larger design, its primary inputs may be driven by other logic blocks that

are part of the same design. In addition, the primary inputs of the circuit-under-test include any external inputs of the design that drive the circuit-under-test. The primary outputs of the circuit- under-test may drive other logic blocks, or they may be primary outputs of the complete design. For simplicity this paper assumes that primary inputs can be assigned any combination of values. Functional constraints on primary input sequences can be accommodated in one of the following ways.

1) The logic used for producing the primary input sequence can A be extended to incorporate some functional constraints.
2) A separate logic block can be used for modifying A so as to satisfy functional constraints.
3) Placing the on-chip test generation hardware for a logic block on the inputs of the logic blocks driving it can create some of the functional constraints for the block without requiring additional logic.

2. Overview

Block Diagram of Broadside and Functional Broadside Testing

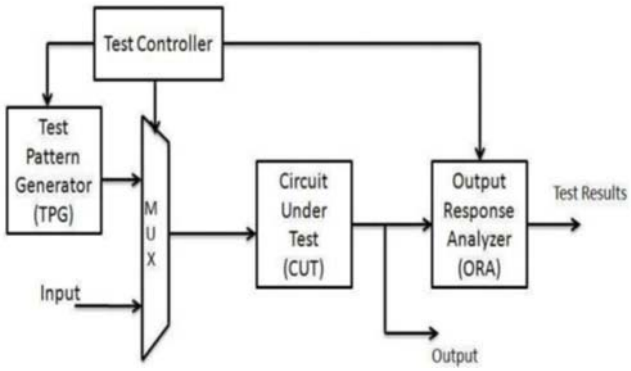


Fig 1: Simple BIST circuitry

Broadside Tests in Partial-Scan Circuits

In a full-scan circuit, a broadside test starts by scanning in a state denoted by two primary input vectors, denoted by and are then applied in functional mode. The final state reached at the end of the test is scanned out. The test can be partitioned into two patterns, applied in one functional clock cycle, and applied in a second functional clock cycle. The application of is done under a slow clock to allow signal transitions in the circuit to settle. The application of is done under a fast clock in order to capture delayed signal-transitions. Faults are detected by observing the primary output vector obtained in response to and when the final state is scanned out. In a full scan circuit, the scan-in state is a fully-specified state. After scanning in all the state variables of the circuit are assigned known values. In addition, the values of all the state variables are observed during the scan-out operation at the end of the test. For illustration, we consider a circuit with two primary inputs and five state variables, which are denoted by suppose that and are scanned, that and are UN scanned. A possible scan-in state is 000xx, where x stands for an unspecified (unknown) value. In a broadside test for this circuit, may be a partially specified state as well. For example, suppose that with and we obtain 1x01x. Let we obtain the two pattern test000xx 00, 1x01x 11. With partially- specified patterns, it may not be possible to activate certain faults. In addition, faults whose effects are propagated by the second pattern to or will not be detected by the scan-out operation at the end of the test. Therefore, it is necessary to consider broadside tests with more than two primary input vectors. The primary input vectors are then applied in functional mode. The state at time unit of the test, where is the next-state obtained when the present-state is and the primary input vector is There is one time unit where such that is applied under a fast clock in order to capture delayed signal-transitions. Application of such that is done under a slow clock to allow signal transitions in the circuit to settle. Under the slow clock the circuit operates as a fault free circuit.

Scanning Circuits

D flip-flop

The D flip-flop is the most common flip-flop in use today. It is better known as data or delay flip-flop (as its output Q looks like a delay of input D).The Q output takes on the state of the D input at the moment of a positive edge at the clock pin (or negative edge if the clock input is active low). It is called the D flip-flop for this reason, since the output takes the value of the D input or data input, and delays it by one clock cycle. The D flip-flop can be interpreted as a primitive memory cell, zero-order hold, and delay line. Whenever the clock pulses, the value of Q next is D and Q prev. otherwise.

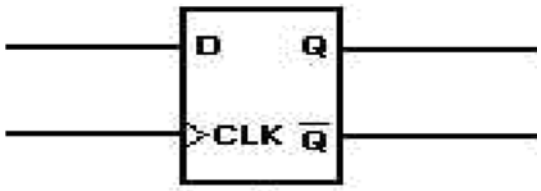


Fig 2. D flip-flop

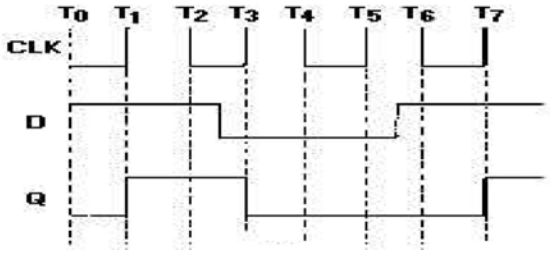


Fig 3: Timing diagram of D flip flop

S27 CIRCUIT

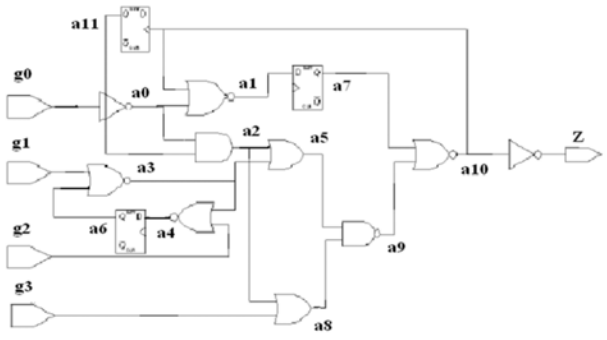


Fig 4: Circuit diagram s27

These notes constitute the publicly available information on the ISCAS'99 benchmarks. Note that several of the benchmarks come from companies that do not want their identities known. Where the benchmarks are publicly available, I give a link to the appropriate website. This is for informational purposes only - the benchmarks should be taken from the ITC'99 Benchmark website so that there is control of the versions used. Some of these measures are rough. The exact number of gates is not very important when dealing with large circuits. Since many of the designs are expressed in terms of cells, the gate count will always be approximate, and will depend on the library used In our project, we are using test bench circuit, which consists of 4 primary inputs,3 flip-flops 1 output.

Table 1: Primary Input Sequence For s27

u	$s(u)$	$a(u)$
0	000	1001
1	010	1110
2	100	0010
3	000	1001
4	010	1001
5	010	0010
6	010	1000
7	100	1101
8	101	1000
9	101	0111
10	000	1000
11	100	1001
12	100	1100
13	101	1101
14	101	1111
15	100	1110

3. Pseudorandom Test Generation

The three primary goals were:

a) to develop a battery of statistical tests to detect non randomness in binary sequences constructed using random number generators and pseudorandom number generators utilized in cryptographic applications,

b) To produce documentation and a software Implementation of these tests, and

c) To provide guidance in the use and application of these tests. Pseudorandom- generate patterns that appear to be random but are in fact deterministic (repeatable).Linear Feedback Shift Register (LFSR) Weighted pseudo-random test generation Adaptive pseudo-random test generation

Algorithmic Test Generation

List primary inputs controlling location where a fault should be detected.

Determine primary input conditions to activate a fault and to sensitize the primary outputs such that the fault can be observed.

Pseudo-Random Test Generation

- Large set of patterns is generated by simple HW or SW pseudo-random generator
- The set is used to stimulate a system with fault simulator
- Fault coverage is analyzed and algorithmic approach is used to cove r remain faults

Pseudo-Random Test Generator generates complex pseudorandom (or random) sequences of test patterns. Its output is a set of variables which controls Input Simulator. Each variable represents different parameter of Input Simulator (like delay between messages, op code, fields, etc.)

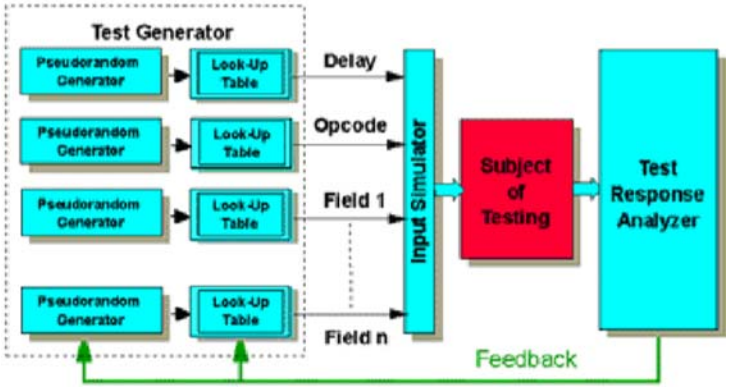


Fig 5: Block of pattern generator

Maximal Length Linear Feedback Shift Register (LFSR) as Pseudo-Random Test Generator (Generates a sequence of length (2n - 1)).

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
- Two types of LFSRs External Feedback, Internal Feedback
- Higher clock frequency

An LFSR generates periodic sequence must start in a non-zero state, The maximum length of an LFSR sequence is 2n - 1 does not generate all 0s pattern (gets stuck in that state)The characteristic polynomial of an LFSR generating maximum-length sequence is a primitive polynomial A maximum-length sequence is pseudo-random: number of 1s =number of 0s + 1 same number of runs of consecutive 0s and 1s 1/2 of the runs have length1 1/4 of the runs have length (as long as fractions result in integral numbers of runs).

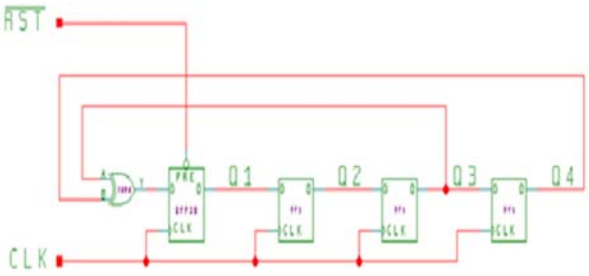


Fig 6: LFSR 4bit circuit

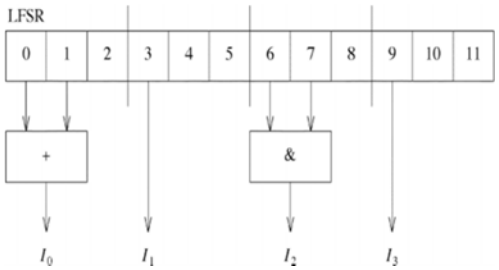


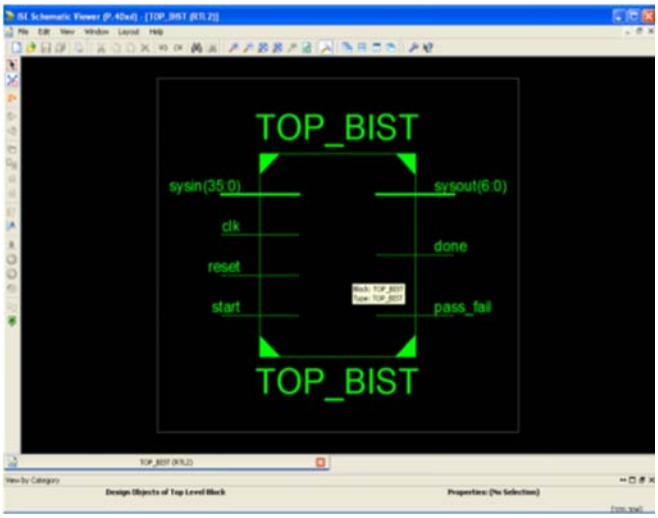
Fig 7: On-chip generation of A

Table 2: LFSR sequence for s27

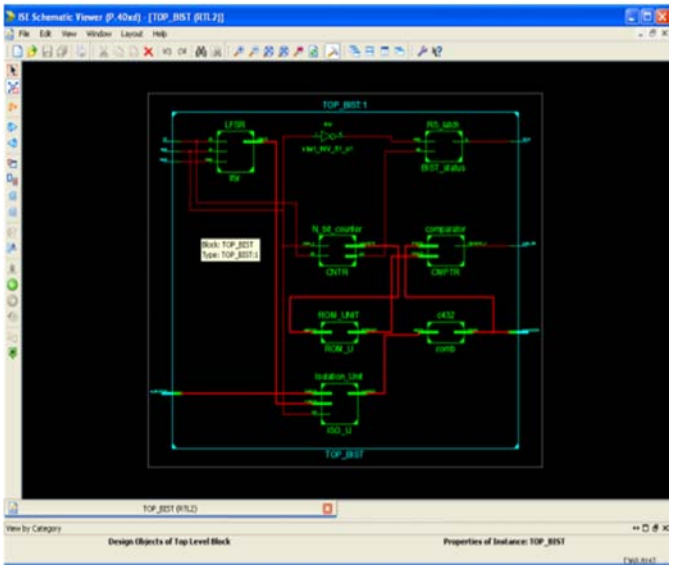
u	$lfsr(u)$			
0	101	011	100	100
1	010	101	110	010
2	001	010	111	001
3	100	011	001	100
4	010	001	100	110
5	001	000	110	011
6	100	010	001	001
7	110	111	010	100
8	011	011	101	010
9	001	101	110	101
10	100	000	101	010
11	010	000	010	101
12	101	110	011	010
13	010	111	001	101
14	101	101	110	110
15	010	110	111	011

4. Experimental results

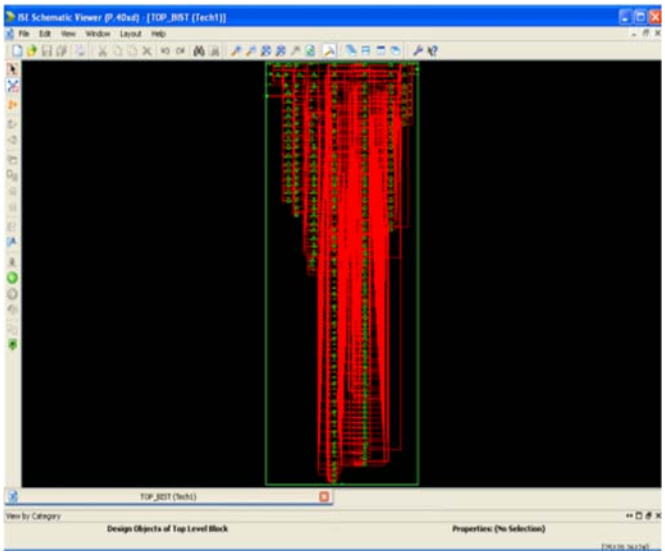
A) Block diagram



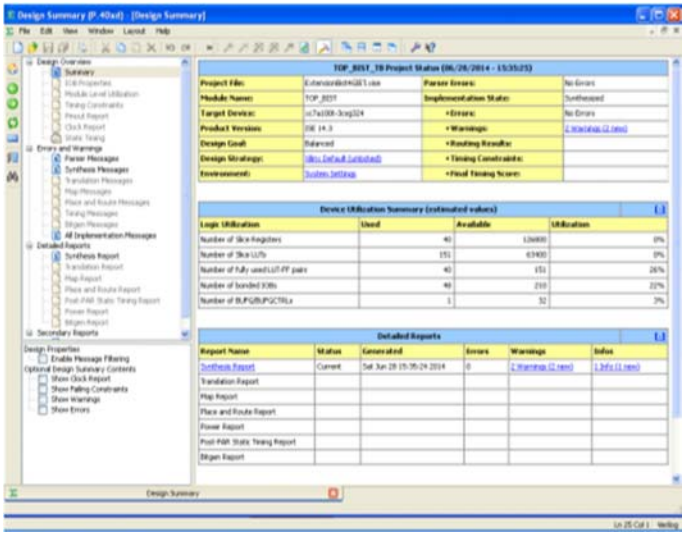
B) RTL schematic



C) Technology schematic diagram

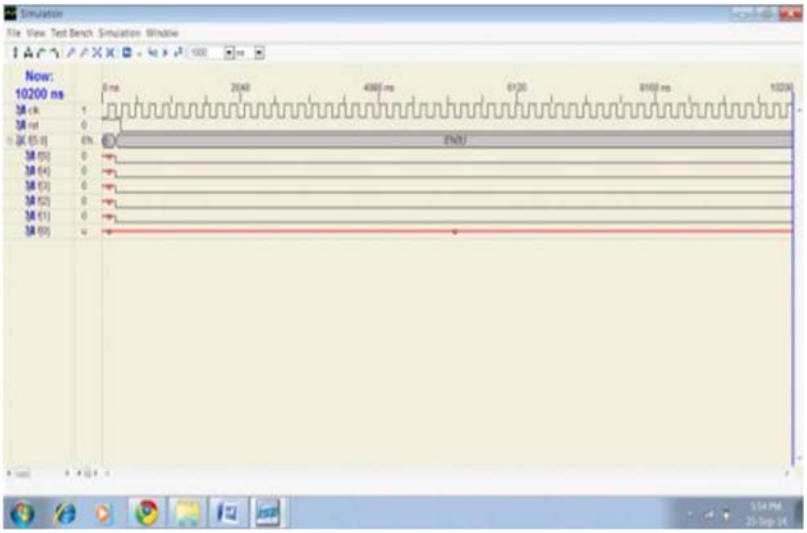


D) Design summary

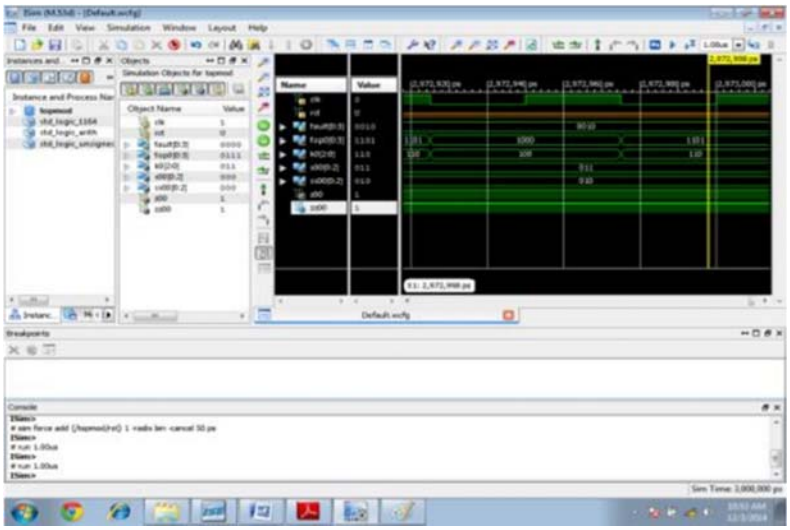


E) Output waveforms

a) Without fault



B) With fault



5. Conclusion

The presence of delay-inducing defects is causing increasing concern in the semiconductor industry today. To test for such delay-inducing defects, scan based transition fault testing techniques are being implemented. To Full scanning Process will Generated and then Fault coverage for Broadside testing is 80%, functional broadside testing is 40% and pseudorandom testing is 80%. Maximum length of the testing is 362. Full scanning percentage is 97.5%. Fault coverage for the Broadside testing with s_t_0 Fault will generate in above circuit is 85%. Maximum length of testing full scan circuit is 402. Scanning percentage is 97%. Testing time for partial scan process is reduces Maximum testing length is reduced at 284. Fault coverage is maximum one of partial scan process.

6. Future implementation

In Scan Based testing will be extended to pseudorandom test Produced Various Fault generated and then Finding the Fault Coverage Scanning Percentage, Maximum Length of Testing are Generated in Project Phase two work. To Scan Process will allowed Second Phase work and then Pseudorandom testing Process is terminated.

7. References

1. Rearick J. Too much delay fault coverage is a bad thing, in Proc. Int. Test Conf., 2001, 624-633.
2. Saxena J, Butler KM, Jayaram VB, Kundu S, Arvind NV, Sreeprakash P, *et al*. A case study of IR-drop in structuredat-speed testing, in Proc. Int. Test Conf., 2003, 1098-1104.
3. Pomeranz. On the generation of scan based test sets with reachable states for testing under functional operation conditions, in Proc. Des. Autom. Conf., 2004, 928-933.
4. Y-C Lin, F Lu, K Yang, K-T Cheng. Constraint extraction for pseudo functional scan based delay testing, in Proc. Asia South Pacific Des. Autom. Conf., 2005, 166-171.
5. Zhang Z, Reddy SM, Pomeranz I. On generating pseudo-functional delay fault tests for scan designs, in Proc. Int. Symp. Defect Fault Toler. VLSI Syst., 2005, 398-405.

6. Pomeranz, Reddy SM. Generation of functional broadside tests for transition faults, IEEE Trans. Comput.-AidedDes. Integr. Circuits Syst., 25(10), 2207-2218.
7. Y-C Lin, F Lu, K Yang, K-T Cheng. Constraint extraction for pseudo-functional scan based delay testing, in Proc. Asia South Pacific Design Autom. Conf., 2005, 166-171.
8. Zhang Z, Reddy SM, Pomeranz I. On generating pseudo-functional delay fault tests for scan designs, in Proc. Int. Symp. Defect Fault Toler. VLSI Syst., 2005, 398-405.
9. Polian I, Fujiwara F. Functional constraints vs. test compression in scan-based delay testing, in Proc. Design, Autom. Test Euro. Conf., 2006, 1-6.
10. Syal M, *et al*. A study of implication based pseudo functional testing, in Proc. Int. Test Conf., 2006, 1-10.
11. Jas A, Y-S Chan, Y-S Chang. An approach to minimizing functional constraints, in Proc. Defect Fault Toler. VLSI Syst., 2006, 215-226.
12. Lee H, Pomeranz I, Reddy SM. On complete functional broadside tests for transition faults, IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., 2008, 583-587.
13. Pomeranz I, Reddy SM. On reset based functional broadside tests, in Proc. Design Autom. Test Euro. Conf., 2010, 1438-1443.
14. Lee H, Pomeranz I, Reddy SM. Scan BIST targeting transition faults using a Markov source, in Proc. Int. Symp. Quality Electron. Design, 2004, 497-502.
15. Gherman V, H-J Wunderlich, Schloeffel J, Garbers M. Deterministic logic BIST for transition fault testing, in Proc. Euro. Test Symp., 2006, 123-130.
16. Y-C Lin, Lu F, K-T Cheng. Pseudo functional testing, IEEE Trans. Comput. Aided Design Integr. Circuits Syst., 2006, 1535-1546.
17. Abramovici M, Breuer MA, Friedman AD. Digital Systems Testing and Testable Design. Piscataway, NJ: IEEE Press, 1995.
18. Pomeranz I, Reddy SM. Primary input vectors to avoid in random test sequences for synchronous sequential circuits, IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., 2008, 193-197.
19. Pomeranz I. Built-in generation of functional broadside tests, presented at the Design Autom. Test Euro. Conf., Grenoble, France, 2011.
20. Bardell PH, McAnney WH, Savir J. Built-In Test for VLSI. New York: Wiley, 1987.
21. Konemann B. LFSR-coded test patterns for scan designs, in Proc. Euro. Test Conf., 1991, 237-242.