wjert, 2016, Vol. 2, Issue 5, 163-173. *Research Article* **ISSN 2454-695X**

World Journal of Engineering Research and Technology
WJERT
www.wjert.org **SJIF Impact Factor: 3.419**

FUNCTIONAL BROADSIDE TESTS USING A FIXED HARDWARE STRUCTURE WITH LFSR

D.Anjaneyulu*, D.Prasanth Varma and G.Rama Rao

Asst. Prof CMR Institute of Technology Kandlakoya (V), Medchal Road, Hyderabad-501401.

Article Received on 16/07/2016 **Article Revised on 05/08/2016** **Article Accepted on 26/08/2016**

***Corresponding Author**
D. Anjaneyulu
Asst. Prof CMR Institute of Technology Kandlakoya (V), Medchal Road, Hyderabad-501401.

ABSTRACT

Functional broadside tests are two-pattern scan based tests that avoid over testing by ensuring that a circuit traverses only reachable states during the functional clock cycles of a test. In addition, the power dissipation during the fast functional clock cycles of functional broadside tests does not exceed that possible during functional operation. On-chip test generation has the added advantage that it reduces test data volume and facilitates at-speed test application. This paper shows that on-chip generation of functional broadside tests can be done using a simple and fixed hardware structure, with a small number of parameters that need to be tailored to a given circuit, and can achieve high transition fault coverage for testable circuits. 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: Functional broadside tests, power dissipation.

I.INTRODUCTION

OVERTESTING due to the application of two-pattern scan-based tests was described in.[1][3] Overtesting 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 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 overtesting 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] and.[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 busilt-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] and[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-in states, which are allowed by themethods described in.[14]–[16] However, the tests that are needed for achieving this higher fault coverage are also ones that can cause overtesting. 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 is applied in functional mode starting from a reachable state, all the states traversed under 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 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. It should be noted that, for the detection of a set of faults, at most 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 does not need to take the circuit through all its reachable states, but only through a sufficiently large number relative to, in order to be effective for the detection of target faults. The hardware used in this paper for generating the primary input sequence consists of a linear-feedback shift-register (*LFSR*) as a random source[17], and of a small number of gates (atmost 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.

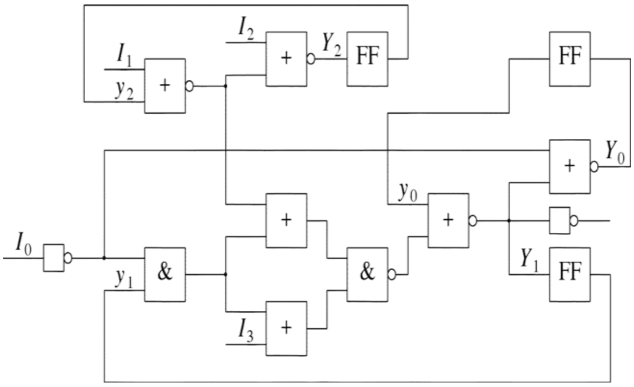
The on-chip test generation hardware is based on the one described in.[19] It differs from it in the following ways. The logic that produces the primary input sequence is designed in this paper to reduce the dependencies between the values assigned to the primary inputs, considering the following sources of dependency. In[19], for a circuit with primary inputs and a parameter *mod*, the *LFSR* used for producing has 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.

Model Design of Hardware

The discussion in this paper assumes that the circuit is initialized into a known state before functional operation starts. Initialization may be achieved by applying a synchronizing sequence, by asserting a reset input or by a combination of both. The initial state of the circuit is denoted by. The discussion also assumes that functional operation consists of the application of primary input sequences starting from state. In addition to producing reachable states, the primary input sequence can also be used as a source for the primary input vectors of functional broadside tests. In particular, every subsequence of length two of defines a functional broadside test. T(u)=(s(u),a(u),a(u+1)) By using a(u)and a(u+1) from, it is possible to avoid the need for a different source for these primary input vectors during on-chip test generation.

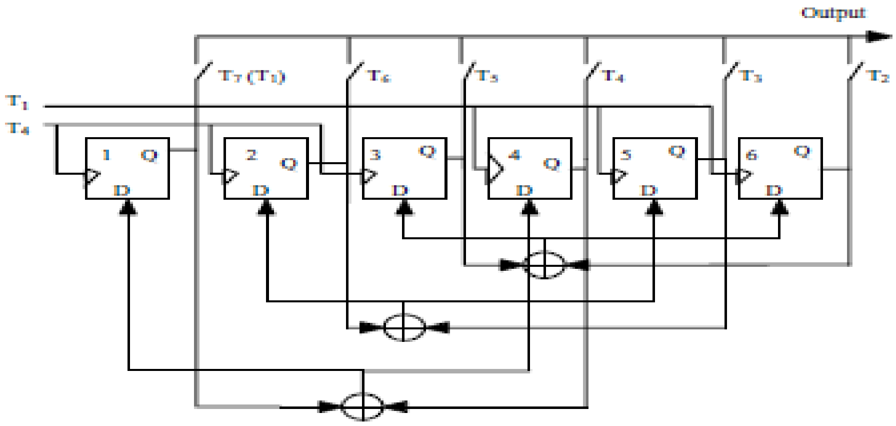
II.ON-CHIP GENERATION OF FUNCTIONAL BROADSIDE TESTS

The simplest way to generate a primary input sequence on-chip is to use a random source such as an LFSR. However, random sequence A may bring the circuit from the initial state into a limited set of reachable states sr. This can be explained by the effect observed in and referred to as repeated synchronization. According to, a primary input cube synchronizes a subset of state variable s(c) if the following conditions are satisfied. Let be applied to the primary inputs when the circuit is in the all-unspecified present-state. Suppose that this results in a next-state. The state variables whose values are specified in are included in.



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

Fig 1: S27 as an example.



The on-chip test generation hardware described so far has parameters l, d, mod and sel. These parameters determine the primary input sequence, and the tests that will be applied based on it. Keeping l, d, mod and sel constant in order to keep the hardware fixed, there is flexibility only in determining the seed of the LFSR. Different seeds yield different primary input sequences and different tests. Therefore, it is possible to increase the fault coverage by using several different seeds. To select seeds for a circuit it is possible to use an approach similar to the one used for test data compression. Using a symbolic seed, it is possible to compute a primary input sequence and the subset of tests based on it, and then solve equations based on functional broadside tests that are known to detect target faults. The approach used in this paper avoids deterministic test generation to identify effective functional broadside tests by considering random seeds. A set of seeds is selected using the following process.

The generalized implementation flow diagram of the project is represented as follows.

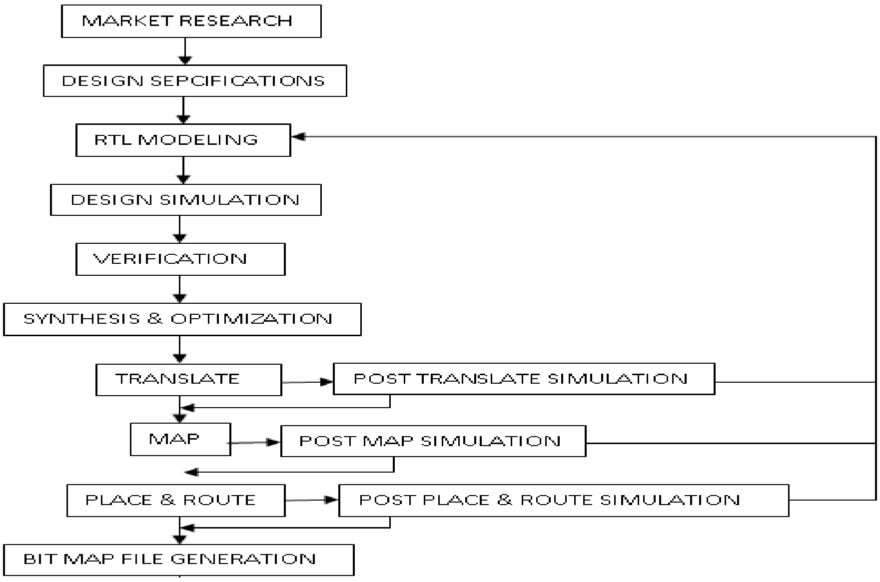


Fig 2: The generalized implementation flow diagram of the project.

Figure 2.General Implementation Flow Diagram

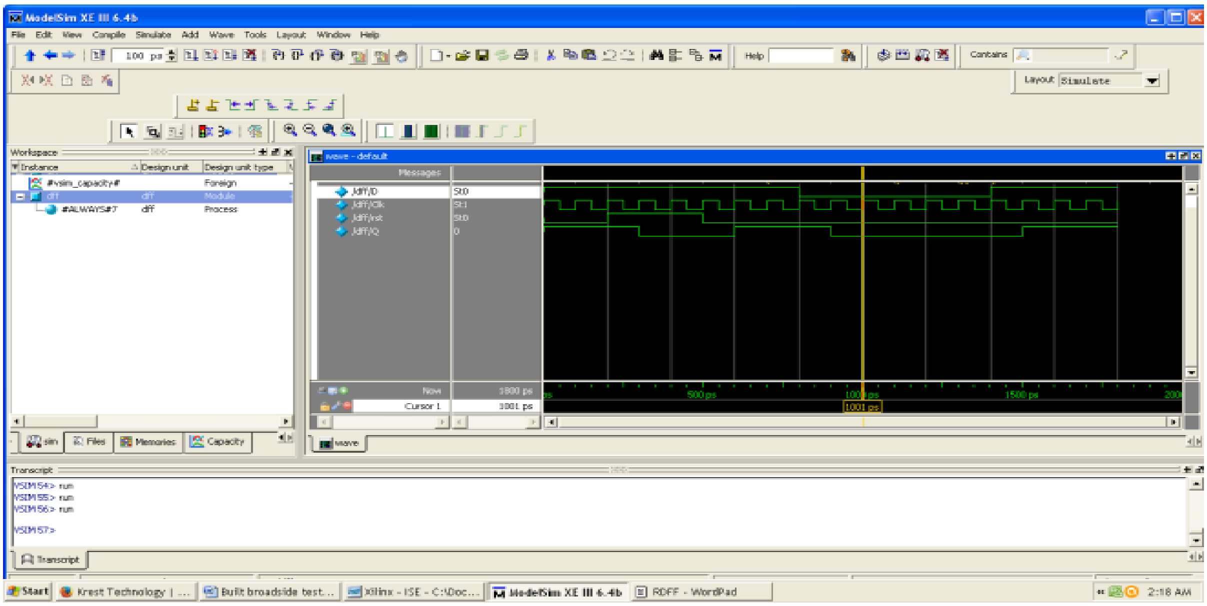
Once the functional verification is clear, the RTL model will be taken to the synthesis process. Three operations will be carried out in the synthesis process such as

- Translate
- Map
- Place and Route

III. SIMULATION AND SYNTHESIS REPORT.

Simulation waveforms.

D-FF



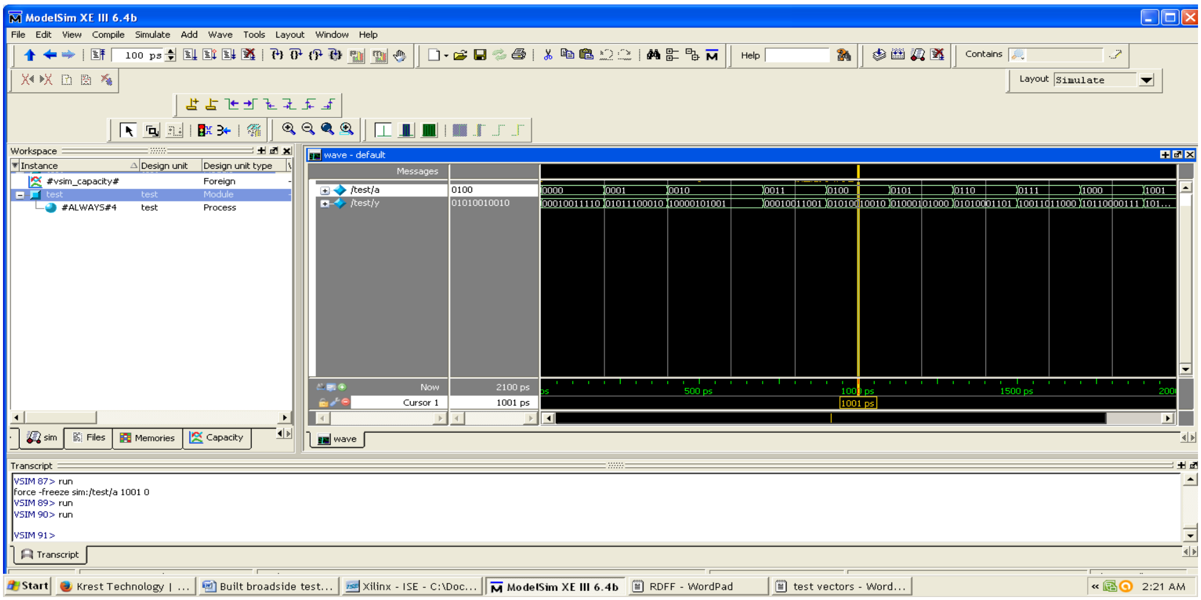
Instance	Design unit	Design unit type
#vsim_capacity#		Foreign
dff	dff	Module
#ALWAYS#7	dff	Process

Messages	
/dff/D	St0
/dff/Clk	St1
/dff/rst	St0
/dff/Q	0
Now	1800 ps
Cursor 1	1001 ps

```
VSIM 54> run
VSIM 55> run
VSIM 56> run

VSIM 57>
```

TEST VECTOR



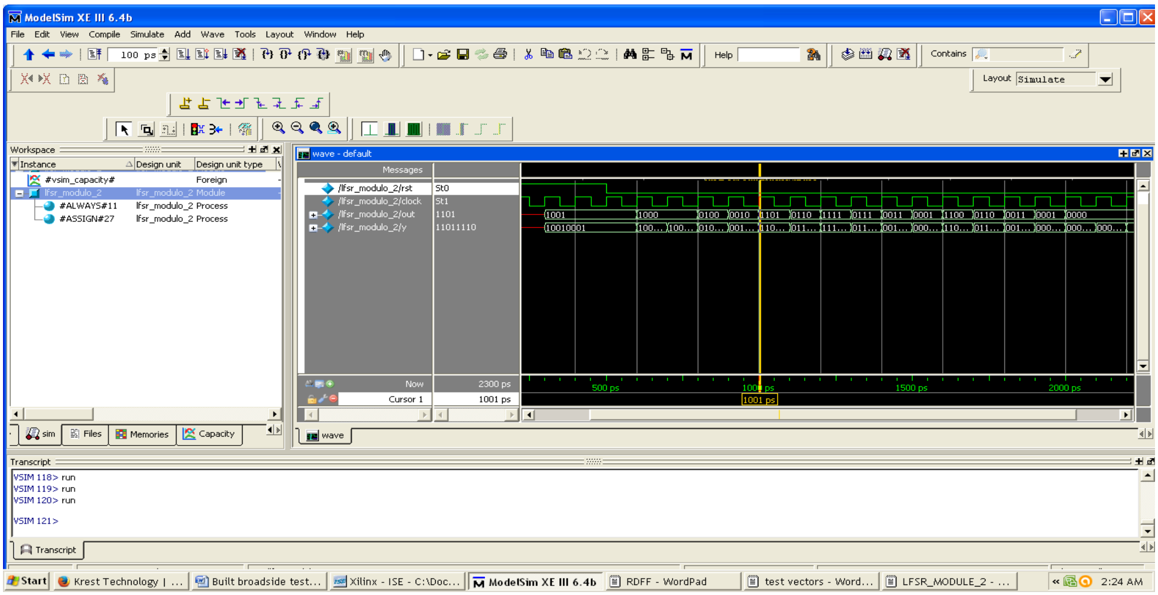
Instance	Design unit	Design unit type
#vsim_capacity#		Foreign
test	test	Module
#ALWAYS#4	test	Process

Messages	
/test/a	0100
/test/y	01010010010
Now	2100 ps
Cursor 1	1001 ps

```
VSIM 87> run
force -freeze sim:/test/a 1001 0
VSIM 89> run
VSIM 90> run

VSIM 91>
```

LFSR



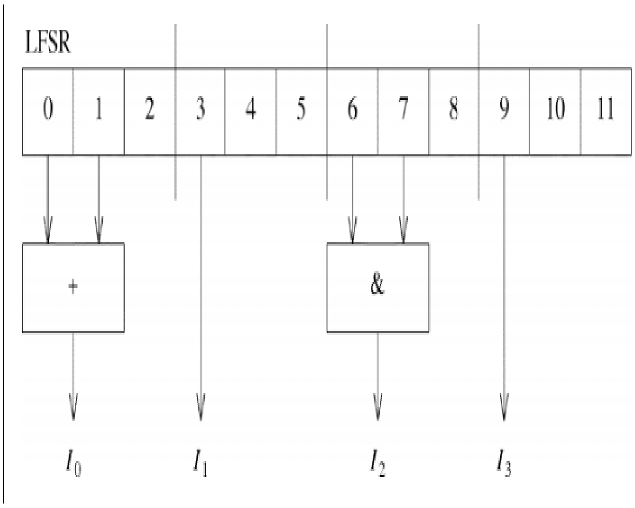
Instance	Design unit	Design unit type
#vsim_capacity#		Foreign
lfsr_modulo_2	lfsr_modulo_2	Module
#ALWAYS#11	lfsr_modulo_2	Process
#ASSIGN#27	lfsr_modulo_2	Process

```
/lfsr_modulo_2/rst     St0
/lfsr_modulo_2/clock   St1
/lfsr_modulo_2/out     1101      1001 1000 0100 0010 1101 0110 1111 0111 0011 0001 1100 0110 0011 0001 0000
/lfsr_modulo_2/y       11011110  10010001 100... 100... 010... 001... 110... 011... 111... 011... 001... 000... 110... 011... 001... 000... 000... 000...
Now       2300 ps
Cursor 1  1001 ps
```

```
VSIM 118> run
VSIM 119> run
VSIM 120> run

VSIM 121>
```

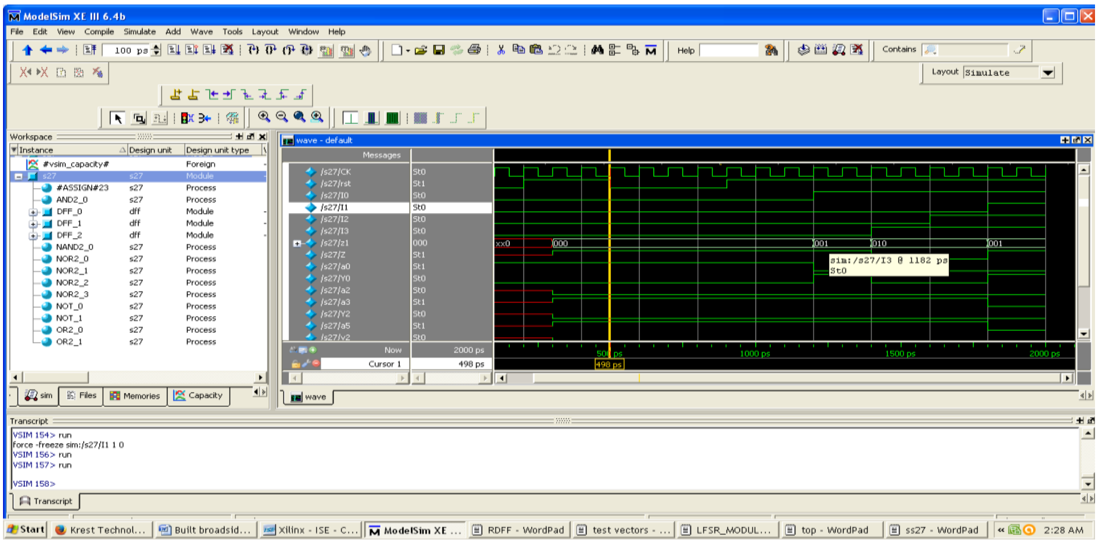
The hardware used for generating a primary input sequence consists of a Linear Feedback Shift Register and a small number of gates. Gates are used to modify the random sequence in order to avoid repeated synchronization that is the sequence takes the circuit to repeat the same or similar states. In addition to this, a single gate is used for determining the tests to be applied based on primary input sequence.



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

FIG 3: On-chip generation of A.

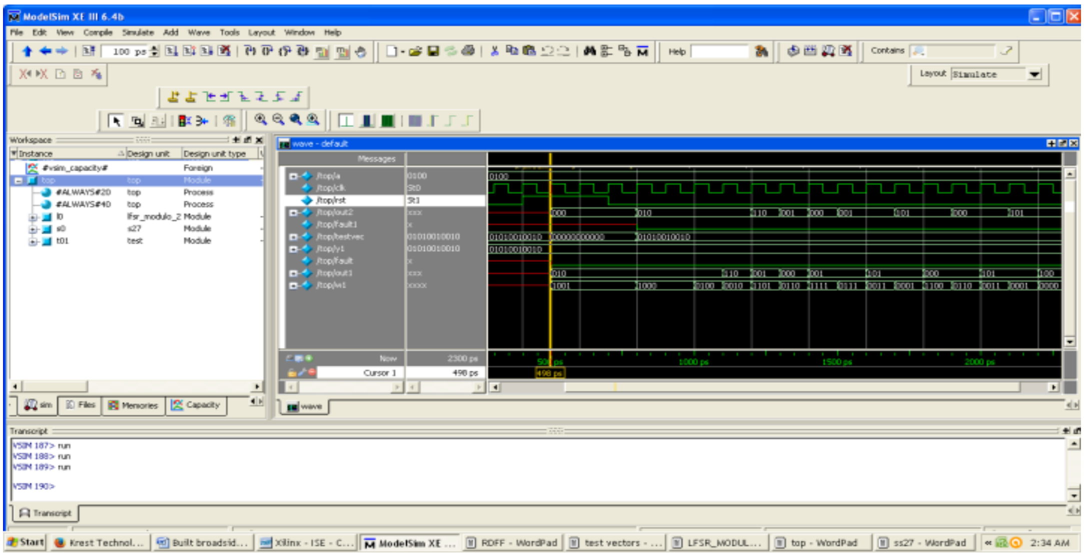
S27



```
VSIM 154> run
force -freeze sim:/s27/I1 1 0
VSIM 156> run
VSIM 157> run

VSIM 158>
```

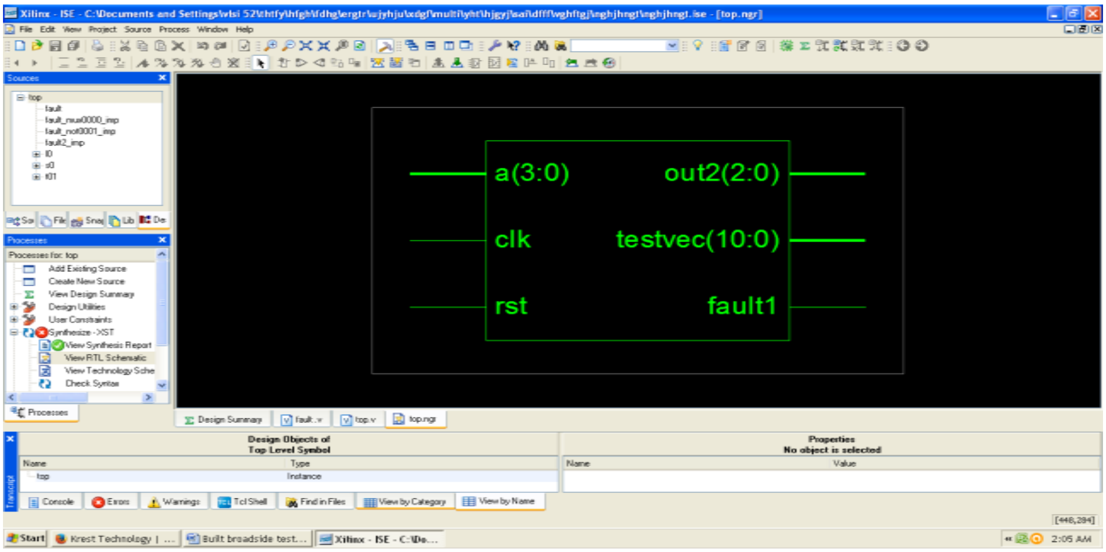
TOP

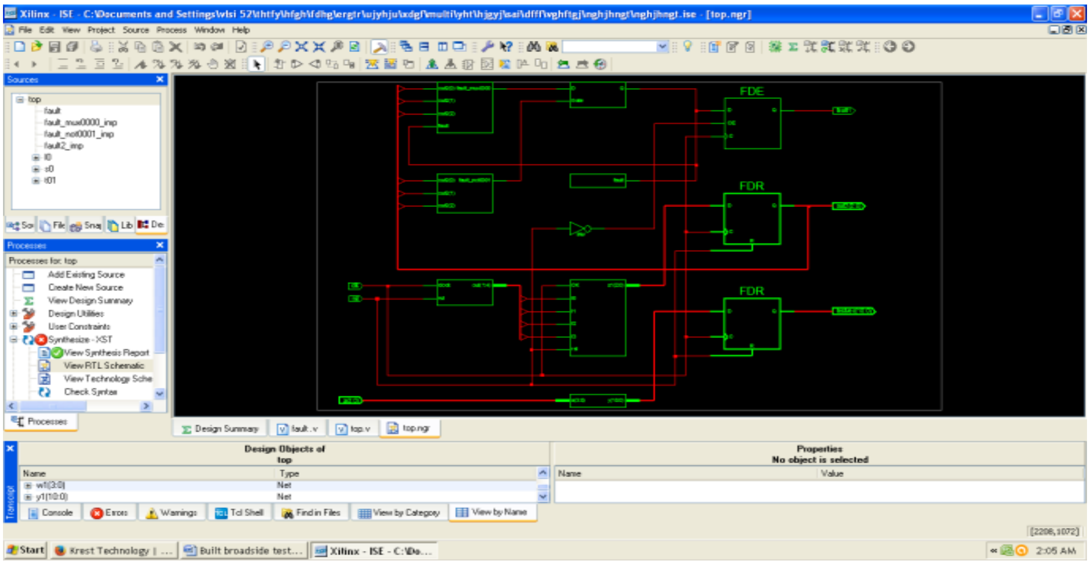


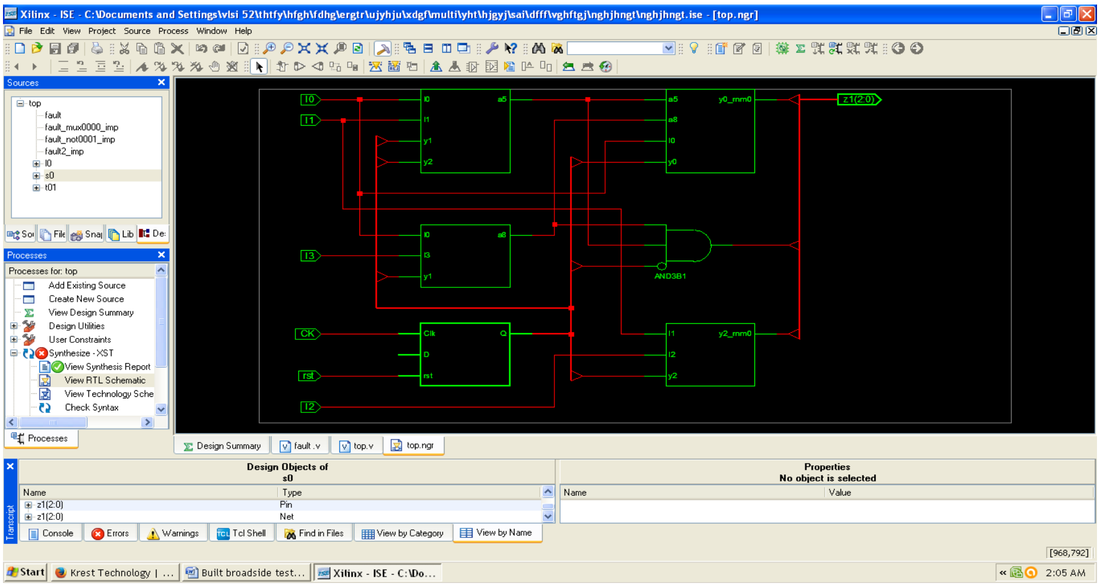
```
VSIM 187> run
VSIM 188> run
VSIM 189> run

VSIM 190>
```

Synthesis report.







CONCLUDING REMARKS

This paper described an on-chip test generation method for functional broadside tests. The hardwarewas 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: 1) the length of the *LFSR* used for producing a random primary input sequence; 2) the length of the primary input sequence; 3) the specific gates used for modifying the random primary input sequence; 4) the specific gate used for selecting applied tests; and 5) the seeds for the *LFSR*. The on-chip generation of functional broadside tests achieved high transition fault coverage for testable circuits.

REFERENCES

1. J. Rearick, “Toomuch delay fault coverage is a bad thing,” in *Proc. Int.Test Conf.*, 2001; 624–633.
2. J. Saxena, K. M. Butler, V. B. Jayaram, S. Kundu, N. V. Arvind, P. Sreeprakash, andM. Hachinger, “A case study of IR-drop in structured at-speed testing,” in *Proc. Int. Test Conf.*, 2003; 1098–1104.
3. S. Sde-Paz and E. Salomon, “Frequency and power correlation between at-speed scan and functional tests,” in *Proc. Int. Test Conf.*, 2008; 1–9, 13.3.
4. I. Pomeranz and S. M. Reddy, “Generation of functional broadside tests for transition faults,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, Oct. 2006; 25(10): 2207–2218.
5. J. Savir and S. Patil, “Broad-side delay test,” *IEEE Trans. Comput.- Aided Design Integr. Circuits Syst.*, Aug.1994; 13(8): 1057–1064.
6. I. Pomeranz, “On the generation of scan-based test sets with reachable states for testing under functional operation conditions,” in *Proc. Design Autom. Conf.*, 2004; 928–933.
7. Y.-C. Lin, F. Lu, K. Yang, and K.-T. Cheng, “Constraint extraction for pseudo-functional scan-based delay testing,” in *Proc. Asia South Pacific Design Autom. Conf.*, 2005; 166–171.
8. Z. Zhang, S.M. Reddy, and I. Pomeranz, “On generating pseudo-functional delay fault tests for scan designs,” in *Proc. Int. Symp. Defect Fault Toler. VLSI Syst.*, 2005; 398–405.
9. I. Polian and F. Fujiwara, “Functional constraints vs. test compression in scan-based delay testing,” in *Proc. Design, Autom. Test Euro. Conf.*, 2006; 1–6.
10. M. Syal *et al.*, “Astudy of implication based pseudo functional testing,” in *Proc. Int. Test Conf.*, 2006; 1–10.
11. A. Jas, Y.-S. Chan, andY.-S. Chang, “An approach tominimizing functional constraints,” in *Proc. Defect Fault Toler. VLSI Syst.*, 2006; 215–226.
12. H. Lee, I. Pomeranz, and S. M. Reddy, “On complete functional broadside tests for transition faults,” *IEEE Trans. Comput.-Aided Design Integr.Circuits Syst.*, 2008; 583–587.

13. I. Pomeranz and S. M. Reddy, “On reset based functional broadside tests,” in *Proc. Design Autom. Test Euro. Conf.*, 2010; 1438–1443.
14. H. Lee, I. Pomeranz, and S.M. Reddy, “Scan BIST targeting transition faults using a Markov source,” in *Proc. Int. Symp. Quality Electron. Design*, 2004; 497–502.
15. V. Gherman, H.-J. Wunderlich, J. Schloeffel, and M. Garbers, “Deterministic logic BIST for transition fault testing,” in *Proc. Euro. Test Symp.*, 2006; 123–130.
16. Y.-C. Lin, F. Lu, and K.-T. Cheng, “Pseudofunctional testing,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 2006; 1535–1546.
17. M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design*. Piscataway, NJ: IEEE Press, 1995.
18. I. Pomeranz and S. M. Reddy, “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. I. Pomeranz, “Built-in generation of functional broadside tests,” presented at the Design Autom. Test Euro. Conf., Grenoble, France, 2011.
20. P. H. Bardell, W. H. McAnney, and J. Savir, *Built-In Test for VLSI*. New York: Wiley, 1987.
21. B. Konemann, “LFSR-coded test patterns for scan designs,” in *Proc. Euro. Test Conf.*, 1991; 237–242.