International Journal of Research Studies in Science, Engineering and Technology
Volume 1, Issue 8, November 2014, PP 9-15
ISSN 2349-4751 (Print) & ISSN 2349-476X (Online)

Hardware Implementation of Functional Verification Using Signature Analysis

Ms. K. Bhagya Sree
M.Tech in VLSI Design, Dept. of E.C.E,
Srinivasa Ramanujan Institute of Technology,
Ananthapuramu, Andhrapradesh, India.
bagi.kommineni@yahoo.com

T. Sai Lokesh
Assistant Professor in E.C.E,
Srinivasa Ramanujan Institute of Technology,
Ananthapuramu,
Andhrapradesh, India

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: *Built-in test generation, functional broadside tests, reachable states, transition faults.*

1. INTRODUCTION

Test generation procedures for functional and pseudo-functional scan-based tests were described. 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, 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 tube a reachable state. The on-chip test generation method from 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 or pseudo-functional broadside tests as in . This is due to the fact that functional broadside tests avoid unreachable scan-in states, which are allowed by the methods described in. 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. The underlying observation is related to one of the methods used in 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. The structure used in this paper reduces these dependencies between primary input values by using a -bit

LFSR for a circuit with primary inputs, where is a parameter such that. 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 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 be extended to incorporate some functional constraints.

2) A separate logic block can be used for modifying 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.

The proposed method for on-chip generation of functional broadside tests. 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 m 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.

The proposed on-chip generation method of functional broadside tests is based on placing the circuit in the initial state sr , applying a primary input sequence A , and using several of the functional broadside tests that can be extracted from A in order to detect target faults.

2. 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.

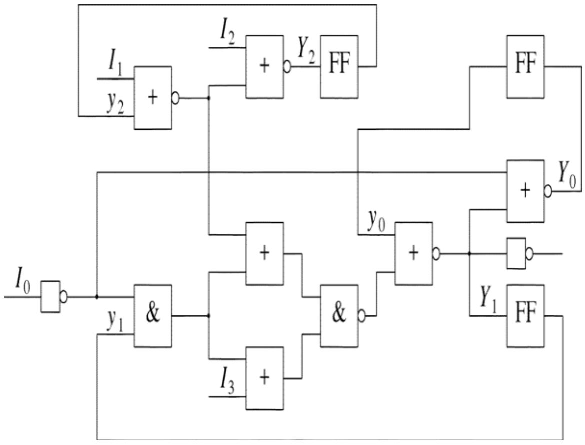
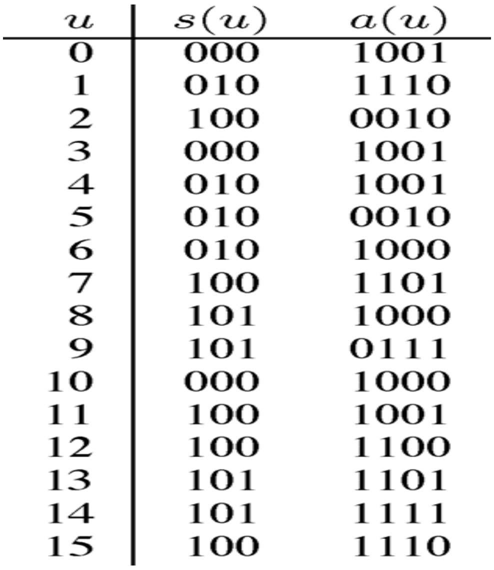


Fig1. *s27*

The primary input cube I0I1I2I3=0xxx applied in present-state y0y1y2=xxx results in the next-state Y0Y1Y2=0xx, synchronizing state variable. In addition, the primary input cube I0I1I2I3=xx1x applied in present-state y0y1y2=xxx results in the next-state Y0Y1Y2=xx0 synchronizing state variable.

Primary sequences;



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

Fig2. *s27 test sequence*

A primary input cube with a small number of specified values is likely to appear repeatedly in a random primary input sequence. When this happens, the state variables in assume the same values repeatedly under. This may prevent the circuit from entering certain reachable states(c), and limit the ability of the functional broadside tests extracted from to detect target faults. Repeated synchronization can also be caused by a sequence of primary input cubes.

LFSR Architecture:

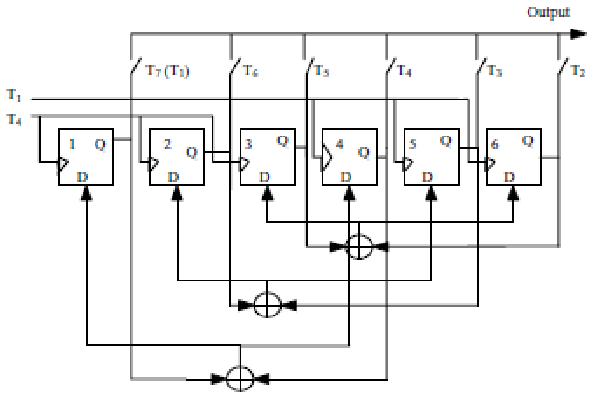


Fig 3. *LFSR*

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.

1) If the fault coverage does not exceed that of for any solution, we report on the following solutions.

a) The solution with the highest fault coverage and the lowest number of seeds.

b) The solution with the highest fault coverage and the lowest number of applied tests.

2) If the fault coverage of at least one solution exceeds that of, we report on the following solutions.

a) The solution with the highest fault coverage.

b) The solution with a fault coverage higher than that of and the lowest number of seeds.

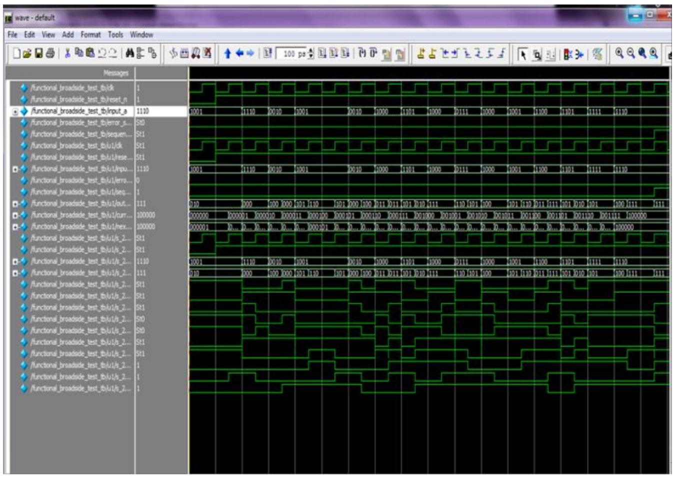
c) The solution with a fault coverage higher than that of and the lowest number of applied tests. One of these solutions is expected to be the most appropriate for the circuit. In addition to these solutions there are others with intermediate numbers of seeds and applied tests, which are not reported.

For the approach from, the modification of the primary input sequence based on requires a two-input gate for every specified bit of. In addition, each sequence requires a gate for selecting the tests that will be applied to the circuit. The multiplexer required for selecting the appropriate gate for each sequence has one AND gate for every sequence, and an OR gate to produce the output of the multiplexer. Overall, the number of gates is estimated as the number of specified bits in plus the number of sequences. The hardware structure. The modification of

the primary input sequence based on requires a gate for every specified bit of . In addition, a single gate is used for selecting the tests that will be applied to the circuit. The same gate is used for implementing the selection for all the sequences, and no multiplexers are needed to select among the different sequences. The number of gates is equal to the number of specified bits in plus one. Column *LFSR* shows the number of *LFSR* bits. In spite of the larger size of the *LFSR* compared with the approach from, the approach described in this paper has the advantage that it reduces the need for random logic, and it already accommodates the need to reduce dependencies between primary input values in order to increase the fault coverage. it can be seen that, the simplified hardware and the reduced dependencies between the values assigned to the primary inputs, allows the on-chip test generation approach described in this paper to achieve a higher fault coverage than that reported in for several circuits. The fault coverage using the proposed hardware structure is never lower than that reported in.

The proposed hardware also provides the flexibility needed to produce solutions with low numbers of different seeds or low numbers of applied tests. The numbers of applied tests are typically higher than in due to the simpler hardware structure. However, even with higher numbers of applied tests, the numbers remain manageable.

3. Simulation Results



```
/functional_broadside_test_tb/clk          1
/functional_broadside_test_tb/reset_n      1
/functional_broadside_test_tb/input_a      1110
/functional_broadside_test_tb/error_s...   St0
/functional_broadside_test_tb/sequen...    St1
/functional_broadside_test_tb/u1/clk       St1
/functional_broadside_test_tb/u1/rese...   St1
/functional_broadside_test_tb/u1/inpu...   1110
/functional_broadside_test_tb/u1/erro...   0
/functional_broadside_test_tb/u1/seq...    1
/functional_broadside_test_tb/u1/out...    111
/functional_broadside_test_tb/u1/curr...   100000
/functional_broadside_test_tb/u1/nex...    100000
/functional_broadside_test_tb/u1/s_2...    St1
/functional_broadside_test_tb/u1/s_2...    St1
/functional_broadside_test_tb/u1/s_2...    1110
/functional_broadside_test_tb/u1/s_2...    111
/functional_broadside_test_tb/u1/s_2...    St1
/functional_broadside_test_tb/u1/s_2...    St1
/functional_broadside_test_tb/u1/s_2...    St1
/functional_broadside_test_tb/u1/s_2...    St0
/functional_broadside_test_tb/u1/s_2...    St0
/functional_broadside_test_tb/u1/s_2...    St1
/functional_broadside_test_tb/u1/s_2...    St1
/functional_broadside_test_tb/u1/s_2...    1
/functional_broadside_test_tb/u1/s_2...    1
/functional_broadside_test_tb/u1/s_2...    1
```

4. Conclusion

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: 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, pp. 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, pp. 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, pp.1–9, Paper 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.*, vol. 25, no. 10, pp. 2207–2218, Oct. 2006.

[5] J. Savir and S. Patil, “Broad-side delay test,” *IEEE Trans. Comput.- Aided Design Integr. Circuits Syst.*, vol. 13, no. 8, pp. 1057–1064, Aug. 1994.

[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, pp. 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, pp. 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, pp. 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, pp. 1–6.

[10] M. Syal *et al.*, “Astudy of implication based pseudo functional testing,” in *Proc. Int. Test Conf.*, 2006, pp. 1–10.

[11] A. Jas, Y.-S. Chan, andY.-S. Chang, “An approach tominimizing functional constraints,” in *Proc. Defect Fault Toler. VLSI Syst.*, 2006, pp. 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.*, pp. 583–587, 2008.

[13] I. Pomeranz and S. M. Reddy, “On reset based functional broadside tests,” in *Proc. Design Autom. Test Euro. Conf.*, 2010, pp. 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, pp. 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, pp. 123–130.

[16] Y.-C. Lin, F. Lu, and K.-T. Cheng, “Pseudofunctional testing,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, pp. 1535–1546, 2006.

[17] M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design*. Piscataway, NJ: IEEE Press, 1995.

AUTHORS' BIOGRAPHY

K. Bhagya Sree received the B.Tech degree in Electronics and Instrumentation Engineering in the year 2012 and pursuing M.Tech degree in VLSI Design from Srinivasa Ramanujan Institute of Technology. Her area of interests includes VLSI Design and FPGA.

T.Sai Lokesh received his M.Tech degree in VLSI Design from Manipal University, Manipal in 2012, and B.Tech degree in Electronics & Communication Engineering from JNTU, Anantapur (AP), in the year 2010. He is currently working as an Assistant Professor in Srinivasa Ramanujan Institute of Technology, Anantapur. His areas of intrest include embedded desigining.