INTERNATIONAL JOURNAL OF RESEARCH
In Advanced Engineering Technologies

Volume 3, Issue 3 OCT 2014

A FIXED HARDWARE STRUCTURE USED FOR BUILT IN GENERATION OF FUNCTIONAL BROADSIDE TESTS

GOWRA KRISHNA [(1)], P.VENKATALAVANYA[(2)]

M.TECH SCHOLAR, VLSI SYSTEM DESIGN, TKR COLLEGE OF ENGINEERING AND SCIENCE [(1)]

M.TECH, ASSOC.PROFESSOR, TKR COLLEGE OF ENGINEERING AND SCIENCE [(2)]

ABSTRACT:
Functional broad side tests are the example of the 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.

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

I. INTRODUCTION

Over testing due to the application of two-pattern scan-based tests was described. 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. 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 . In both cases, the circuit will operate correctly during functional operation.

Functional broadside tests ensure that the scan-in state is a state that the circuit can enter during functional operation, or a reachable state. As broadside tests, 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 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. 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 to be 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 or pseudo-functional broadside tests. This is due to the fact that functional broadside tests avoid unreachable scan-in states, which are allowed by the methods described. 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

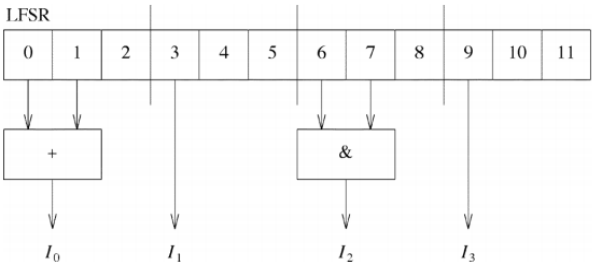
proposed on-chip test generation method, the circuit is used for generating reachable states during test application. This alleviates the need to compute reachable states or functional constraints by an offline processes. The underlying observation is related to one of the methods used in for offline test generation, and is the following.

If a primary input sequence 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 |F| 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 |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.

II.LFSR DESIGN

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, 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. In addition, the on-chip test generation hardware consists of a single gate that is used for determining which tests based on A will be applied to the

Circuit



.

FIG 1.On-chip generation of A

actually used for producing values for the input,

LFSR SEQUENCE TABLE:

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

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 A.
4) The specific gate used for selecting the functional broadside tests that will be applied to the circuit based on A.
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. It differs from it in the following ways.

1) The logic that produces the primary input sequence A is designed in this paper to reduce the dependencies between the values assigned to the primary inputs, considering the following sources of dependency. In a circuit with n primary inputs and a parametermod, the LFSR used for producing A has n + mod bits. The n 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 n primary inputs, where d is a parameter such that d>mod .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

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, higher fault coverage is achieved for several of the circuits considered. In addition, other parts of the test generation hardware can be simplified compared with the designed, as discussed next.

2) 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 uses a different gate for every sequence. Since the number of sequences is in 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 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) 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
initial state sr=000. For every time unit u, Table shows the accommodated in one of the following ways.

1) The logic used for producing the primary input sequence A can 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.

III. PROPOSED METHOD FOR ON-CHIP GENERATION OF FUNCTIONAL BROADSIDE TESTS

This section gives an overview of 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 s_r. The discussion also assumes that functional operation consists of the application of primary input sequences starting from state s_r. With s_r as the initial state for functional s_r operation, is a reachable state. In addition, the set of reachable states consists of every state such that there exists a primary input sequence that takes the circuit from s_r to s_i .Since can be entered during functional operation starting from s_r, s_i is a reachable state.

It is possible to obtain reachable states on-chip by placing the circuit in state s_r and applying a primary input sequence $A = a(0)\,a(1)\,\ldots\,a(L-1)$ of length L in functional mode. The circuit can be brought into state s_r by using a scan-in operation, or by using its initializing sequence. Let s(u) be the state that the circuit reaches at time unit u under A ,for $0 \leq u \leq L$.We have that s(0)=s_r. In addition s(u) , is a reachable state for $0 \leq u \leq L$. Therefore, every state s(u) can be used as the initial state for a functional broadside test $\langle s(u), a_1, a_2 \rangle$,where s(u) plays the role of a scan-in state. As in a broadside test a_1 and a_2 are primary input vectors that are applied in two consecutive functional clock cycles starting s(u) from using a slow and a fast clock, respectively. In addition to producing reachable states, the primary input sequence A can also be used as a source for the primary input vectors of functional broadside tests. In particular, every subsequence of length two of A defines a functional broadside test $t(u) = \langle s(u), a(u), a(u+1) \rangle$. By a(u) using a(u+1) and from , it is possible to avoid the need for a different source for these primary input vectors during on-chip test generation.

For illustration we consider ISCAS-89 s27 benchmark with

state s(u) and the primary input vector a(u).

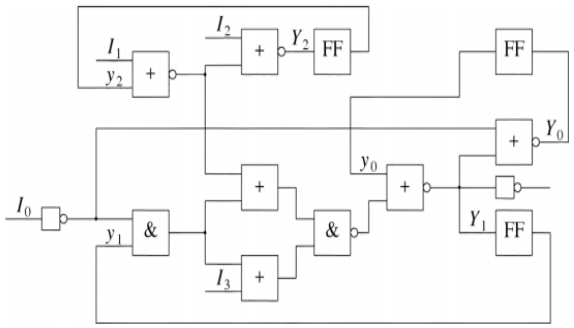


Fig 2.: s27

TABLE
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

Table yields the functional broadside tests $t(0) = \langle 000, 1001, 1110\rangle$ $t(1) = \langle 010, 1110, 0010\rangle, \ldots, t(14) = \langle 101, 1111, 1110\rangle$. The proposed on-chip generation method of functional broadside tests is based on placing the circuit in the initial state s_r, 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. Next, we discuss how the application of A is affected by the need to observe fault effects created by a test $t(u) = \langle s(u), a(u), a(u+1)\rangle$.

At time u unit the circuit is in state s(u) . Applying a(u) and a(u+1) in functional mode will result in the application of t(u) . A fault can be detected in one of the following two ways.

1) Based on the primary output vector z(u+1) obtained in response to a(u+1) , if this vector is different from the expected fault free primary output vector.

2) Based on the final state s(u+2) of the test, if this state is different from the expected fault free state.

In the context of built-in self-test, z(u+2) and s(u+2) need to be captured by an output response compactor such as a MISR. In the case of s(u+2) , the state needs to be shifted into the output response compactor over a number of clock cycles equal to the length of the longest scan chain. The circuit then needs to be brought back to state s(u+2) in order to continue the test application process under A.

In the example of shown in Fig s27, the primary input cube $I_0I_1I_2I_3$=0XXX applied in present-state $y_0y_1y_2y_3$= XXX results in the next-state $Y_0Y_1Y_2Y_3$=0XX, synchronizing state variable. In addition, the primary input cube $I_0I_1I_2I_3$=XX1X applied in present-state $y_0y_1y_2y_3$= XXX results in the next-state $Y_0Y_1Y_2Y_3$=XX0, synchronizing state variable y2.

TABLE
PARAMETERS

parameter	meaning
L	Length of primary input sequences
d	Number of $LFSR$ bits per primary input
mod	Number of $LFSR$ bits for modifying the value of a primary input. The probability of changing a primary input value is $1 - 1/2^{mod}$
sel	Tests starting at time units that are divisible by sel are applied to the circuit. $sel = 2^m$

IV. 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,“Too much 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, and M. Hachinger, “A case studyof IR-drop in structured at-speed testing,” inProc.

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,” inProc. 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,” inProc. 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,” inProc. Asia South Pacific DesignAutom.Conf., 2005, pp. 166–171.

[8]Z.Zhang,S.M.Reddy,andI.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,” inProc. Design, Autom. Test Euro. Conf., 2006, pp. 1–6.

[10] M. Syalet al., “A study of implication based pseudo functional testing,” inProc. Int. Test Conf., 2006, pp. 1–10.