SEQUENTIAL CIRCUITS BIST WITH STATUS BIT CONTROL

JAAN RAIK, ELMET ORASSON, RAIMUND UBAR
DEPARTMENT OF COMPUTER ENGINEERING, TALLINN UT, ESTONIA

KEYWORDS: Sequential Circuits, Built-In Self-Test, Finite State Machines, Design-for-Testability

ABSTRACT: The paper proposes a Design-for-Testability (DfT) technique of Built-In Self-Test (BIST) for sequential circuits. The technique is based on making the status signals entering the control part controllable during the test mode to force the device under test to traverse all the branches in the FSM state transition graph. Extra outputs are added to the circuit under test in order to observe the values of the status bits masked out. This type of architecture requires little device area overhead since a simple controller can be implemented to manipulate the control signals. Experiments were carried out on six sequential examples in order to compare different approaches to sequential BIST. In general, the experiments show that simple LFSR does not provide an acceptable fault coverage for sequential designs. However, no universally best test generating approach was identified and the optimal solution appears to be highly dependent on design's pseudo-random testability characteristics.

1. INTRODUCTION

Current paper proposes a technique known from software testing to be implemented in Built-In Self-Test (BIST) for synchronous sequential circuits. Here, path coverage metrics [8] is used to generate masks for controlling the FSM of the device under test in the test mode. In addition, general architecture to allow such application of the masks is proposed. It contains an LFSR and a simple controller to manipulate the masked out bits. Due to the small number of such signals in most of the circuits, very little area overhead is required.
The problem of Built-In Self-Test (BIST) for combinational and full-scan circuits has been thoroughly researched in the past. Implementing weighted random test patterns [1,2] and bit-flipping [3] have been among the most efficient solutions.
Nachman, Saluja et al. [4] propose an a method were the values are held at inputs and scan registers while a certain number of clock pulses are applied. This requires preliminary testability analysis of the circuit structure. Furthermore, the above approach is applicable for circuits containing scan-chains only.
However, rather limited amount of work is available on BIST for non-scan sequential designs. The main motivation for sequential BIST is that, unlike in scan-chain approach, there is no need to reconfigure the circuit flip-flops during the test mode. This allows testing of the circuit at its normal operating speed.
Pomeranz and Reddy present a solution for the general case of sequential circuits [5]. However, the main problem is an excessive hardware overhead since dedicated test pattern generators are to be tailored for each individual primary input. In [6], Chakrabarty proposes a method similar to the reseeding approach [7]. Here, deterministic patterns are embedded to a sequence generated on-chip by using twisted ring counters (or Johnson counters, as they are also refered to).
The paper is organized as follows. Chapter 2 explains the functional fault model of covering all the branches in an FSM state transition graph. Chapter 3 presents the proposed DfT-based BIST approach technique. In Chapter 4, experimental results are provided. Finally, main conclusions are given.

2. TEST COVERAGE METRICS FOR FSM

An FSM may be represented using a state transition table or a state diagram. A state diagram is a directed graph, where the nodes correspond to states and the branches correspond to transitions between the states. Marked on the branches are the conditions required to activate them. In a digital system, these conditions usually correspond to status bits originating from conditional operations in the datapath.
The approach proposed in this paper is based on all-branches coverage metrics [8], which is known to be more powerful than all-statement coverage. Let us consider an example in Figure 1, where covering all the branches in the state transition graph of the FSM is presented. In Figure 1a, we traverse a sequence s0 → s1 → s5 → s0 by setting the status signal A to be 1. Figure 1b shows traversing the next sequence s0 → s1 → s2 → s3 → s4 → s1 by assigning A := 0 and B := 0. Finally, the sequence s1 → s2 → s4 is covered by assigning 1 to the status bit B (Figure 1c). As it can be seen, all the branches of the state transition graph for the example FSM are covered by the paths in Figure1.
The main idea of current approach is to force the FSM to traverse all the brances in the state transition graph. This is implemented by controlling the status bits entering the control part and feeding pseudorandom data to the primary inputs of the circuit. The next Section explains this architeture more in detail.

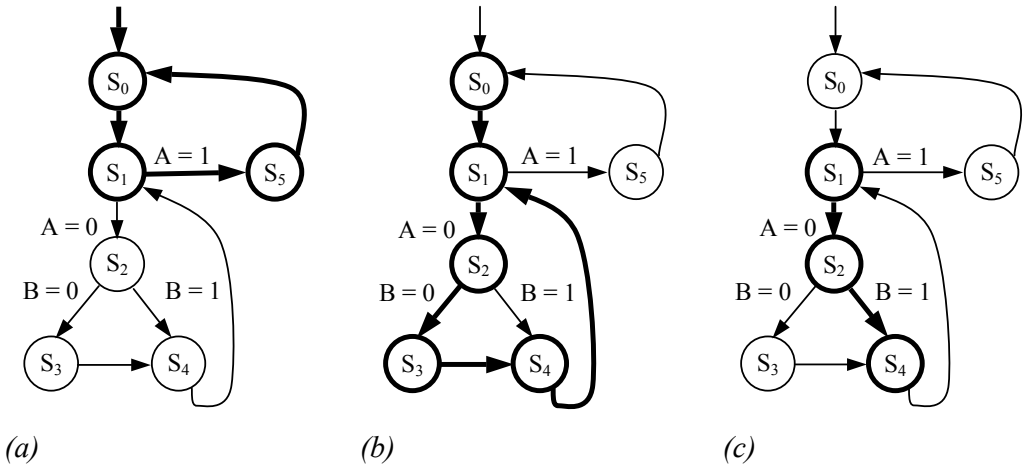

Fig. 1. Traversing all branches in the state transition graph

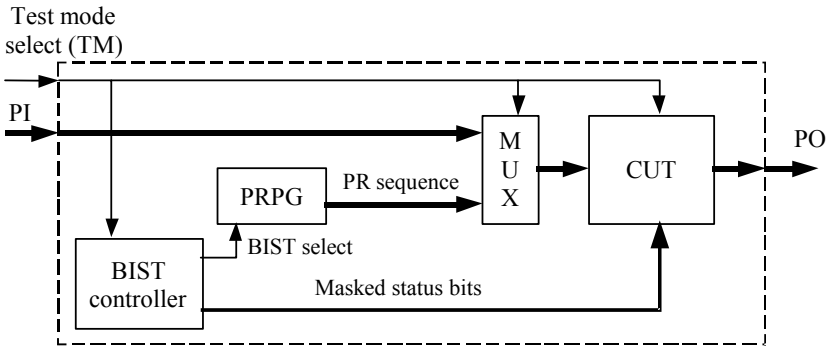

Fig. 2. General BIST architecture for status bit masking

3. GENERAL ARCHITECTURE

Figure 2 presents the general architecture of the DfT enhanced BIST. It contains a Pseudo Random Pattern Generator (PRPG), a BIST controller, Circuit Under Test (CUT) and a MUX to select between the normal inputs and the pseudorandom test. Linear Feedback Shift-Register (LFSR) has been implemented as the PRPG. The task of the BIST controller is to activate the pseudo-random pattern generator and control the values of the status bits. The pseudorandom test generation in the experiments were carried out by CAD tools belonging to Turbo-Tester [10]. Output response (signature) analysis is out of the scope of current paper.

Figure 3 shows the structure of a digital system modified according to the DfT approach. The circuit under test is divided into an FSM and a datapath. The DfT architecture implements multiplexers to mask out the status signals of the datapath entering the FSM. Normal status bit values are selected during the working mode (TM=0) and controller-generated masked values during the test mode (TM=1). The muxed-out signals are made observable by adding dedicated observation points.

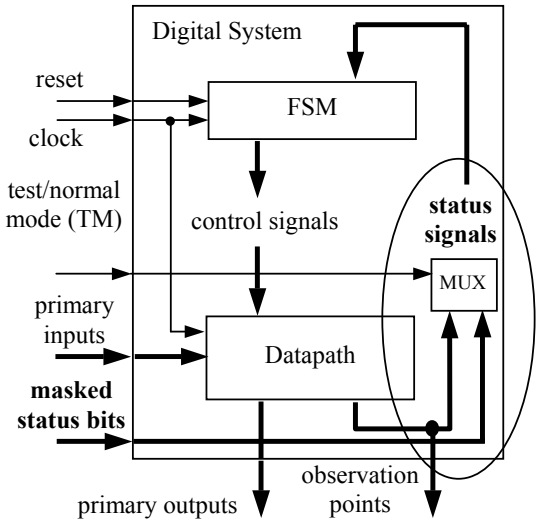

Fig. 3. Digital system modified for DfT

TABLE 1. Characteristics of the benchmark circuits

circuit	FSM states	PI bits	PO bits	registers	MUXes	F.U.s	faults	mask bits	area overhead
DIFFEQ	6	81	48	7	9	5	10360	-	-
DIFFEQ_MOD	6	83	48	7	10	5	10372	1	0.1 %
ELLIPF	28	130	113	17	7	3	5674	-	-
GCD	8	9	4	3	4	3	452	-	-
GCD_MOD	8	12	4	3	6	3	474	2	4.4 %
MULT8x8	8	17	16	7	4	9	2064	-	-
MULT8x8_MOD	8	20	16	7	6	9	2098	2	1.7 %
RISC	4	26	16	8	4	4	6418	-	-
SOSQ	5	9	32	7	2	6	1952	-	-
SOSQ_MOD	5	11	32	7	3	6	1964	1	0.5 %

As we can see in the experiments presented in the following Section, the number of such signals is usually very low (from zero to two in the considered benchmarks). Thus the area overhead required by the controller and the MUXes is low.

4. EXPERIMENTAL RESULTS

Table 1 presents characteristics of the benchmark circuits that have been chosen from the HLSynth92 [11] and VILAB benchmark families [12]. The circuits with _MOD extension are the modified designs, where the test mode multiplexer has been inserted and the status bits have been made controllable.
The last column of the Table shows the area overhead imposed by the status-bit multiplexers. As we can see, the number of mask bits and therefore, the number of additional multiplexers, is very low. Thus, the required overhead of the multiplexers is neglectable ranging from 0.1 % to 4.4 %.
Table 2 shows the average and maximum fault coverages for all the benchmarks both, for 1000 and 10000 pseudorandom vectors. Five different test configurations were considered:

1) LFSR and original circuit

The original circuit was tested with pseudorandom patterns generated by an LFSR.

2) LFSR and modified circuit (_MOD)

The modified circuit (i.e. the circuit, where status bits have been made controllable) was tested with an LFSR.

3) LFSR and test masks in test mode (TM=1)

The modified circuit was tested with an LFSR, the test masks were applied in the test mode (TM signal was active).

4) LFSR and test masks in normal mode (TM=0)

The modified circuit was tested with an LFSR, but the test masks were applied in the normal working mode (TM signal was deactivated).

5) LFSR with reset handling (Reset)

The modified circuit was tested with an LFSR and the global reset was kept deactivated during each test sequence.

Figure 4 gives a clearer view of the results presenting the performance of the above-mentioned techniques on the six circuits. The data is presented for the maximal results obtained with 1000 clock-cycles. We can distinguish between several types of circuits with different characteristics. There are two circuits, which are well random-testable: GCD and DIFFEQ. As it can be seen from the Figure, all the bars for these circuits are of nearly similar height and reach nearly 100 %. This means that for these circuits any BIST scheme will do, including the pure pseudo-random approach. Another circuit that can be easily tested by pseudorandom data is RISC. However, here the main reason is most likely the very small sequential depth (4 clock-cycles).
The rest of the circuits can not be efficiently tested by pseudo-random vectors. While ELLIPF and MULT8x8 could be well tested by simple reset handling, for the SOSQ benchmark, signal masking should be prefered. Thus, depending on the pseudo-random testability characteristics, an appropriate approach can be selected for each individual case.

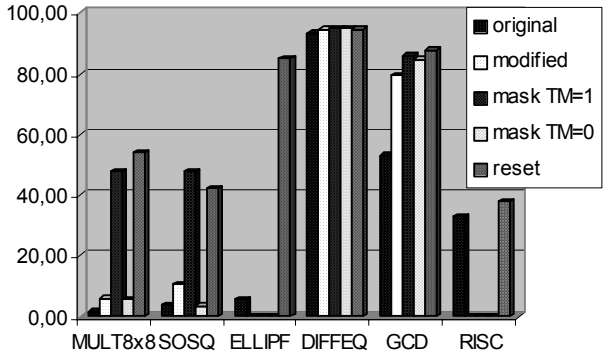

Fig. 4. Effect of DfT on different example circuits

TABLE 2. Comparison of sequential BIST solutions

circuit	average cov., %		maximal cov., %	
	1000	10000	1000	10000
MULT8x8	1,55	1,55	1,55	1,55
MULT8x8_MOD	5,90	5,90	5,90	5,90
MULT8x8_MOD (TM=1)	47,63	47,98	47,95	48,05
MULT8x8_MOD (TM=0)	5,67	5,67	5,67	5,67
MULT8x8_MOD (reset)	**49,71**	**58,29**	**54,10**	**58,52**
SOSQ	3,63	3,27	3,63	3,63
SOSQ_MOD	10,49	9,62	10,58	10,58
SOSQ_MOD (TM=1)	**46,63**	**47,71**	**47,71**	**47,71**
SOSQ_MOD (TM=0)	3,60	3,60	3,61	3,61
SOSQ_MOD (reset)	38,49	36,70	42,22	42,27
ELLIPF	4,96	5,26	5,59	5,80
ELLIPF (reset)	85,00	85,01	85,02	85,02
DIFFEQ	93,03	94,27	93,35	94,55
DIFFEQ_MOD	94,15	95,53	94,56	**95,96**
DIFFEQ_MOD (TM=1)	94,63	95,39	**95,06**	95,88
DIFFEQ_MOD (TM=0)	**94,75**	**95,40**	94,92	95,74
DIFFEQ_MOD (reset)	94,63	95,32	94,78	95,44
GCD	39,56	50,44	53,08	68,72
GCD_MOD	56,13	71,26	79,41	84,66
GCD_MOD (TM=1)	**85,95**	85,95	86,13	86,13
GCD_MOD (TM=0)	81,76	84,73	84,87	84,87
GCD_MOD (reset)	84,39	**87,29**	**88,03**	**88,03**
RISC	29,00	**40,03**	33,05	**42,43**
RISC (reset)	**36,87**	39,50	**37,91**	39,55

5. CONCLUSIONS

The paper proposed a technique known from software testing to be implemented in Built-In Self-Test (BIST) for synchronous sequential circuits. Path coverage metrics was used to generate masks for controlling the FSM of the device under test. In addition, general architecture to allow such application of the masks was proposed.

Experiments carried out on six sequential benchmarks showed that most of the circuits could not be tested by pseudorandom data. Controlling the FSM of the circuit under test considerably improved the results. However, the experiments also showed that there was no universally better solution among the compared architectures. Depending on the pseudo-random testability characteristics, an appropriate approach has to be selected for each individual test case.

Acknowledgements – The research has been partly funded by European projects REASON (IST-2000-30193), E-vikings II (IST-2001-37592) and ESF grants G5637, G5649 and G5910.

THE AUTHORS

Jaan Raik, Elmet Orasson and Raimund Ubar are with the Computer Engineering Department of Tallinn University of Technology, Raja 15, 12618 Tallinn, ESTONIA.
E-mail: {jaan | elmet | raiub}@pld.ttu.ee.

REFERENCES

[1] F. Brglez, et al., "Hardware-Based Weighted Random Pattern Generation for Boundary-Scan", Proc. IEEE Int. Test Conf., pp. 264-274, 1989.

[2] H.-J. Wunderlich. „Multiple distributions for biased random test patterns", Proc. ITC, 1988, pp. 236-244

[3] H.-J. Wunderlich, G. Kiefer. „Bit-Flipping BIST", Proc. ICCAD, November 1996, pp. 337-343

[4] L. Nachman, K.K. Saluja, S. Upadhyaya, R. Reuse, „Random pattern testing for sequential circuits revisited", Proc. of FTCS-26, 1996.

[5] I. Pomeranz, S.M. Reddy, „Built-in test generation for synchronous sequential circuits", ICCAD 1997: pp. 421-426.

[6] K. Chakrabarty, B.T. Murray, V. Iyengar, "Deterministic built-in test pattern generation for high-performance circuits using twisted-ring counters", IEEE Trans. VLSI Systems, Vol.8., No.5, Oct. 2000.

[7] S. Hellebrand, S. Tarnick, J. Rajski, and B. Courtois, "Generation of Vector Patterns through Reseeding of Multiple-Polynomial Linear Feedback Shift Registers", Proc. IEEE Int. Test Conf., pp. 120-129, 1992.

[8] M. Roper, "Software Testing", London, McGraw-Hill Book Company, 1994

[9] E. M. Rudnick, R. Vietti, A. Ellis, F. Corno, P.Prinetto, M. Sonza Reorda, "Fast sequential circuit test generation using high-level and gate-level techniques", Proc. of DATE, 1998.

[10] M.Aarna, E.Ivask, A.Jutman, E.Orasson, J.Raik, R.Ubar, V.Vislogubov, H.D.Wuttke, "Turbo Tester - Diagnostic Package for Research and Training", Proc. EWDTC'03, Scientific-Technical Journal Radioelectronics and Informatics, No. 3 (24), pp. 69-73, July-Sept. 2003.

[11] URL: http://www.cbl.ncsu.edu/pub/Benchmark_dirs/HLSynth92/

[12] URL: http://vilab.dcs.elf.stuba.sk/

[13] J.Raik, R.Raidma, R.Ubar, "Explorations in Low Area Overhead DfT Techniques for Sequential BIST", Proc. of the IEEE Norchip Conference, Nov. 11-12, 2003.