Built-in Self-Test with an Alternating Output

T. Bogue
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada, N2L 3G1
E-mail: tbogue@rabbit.uwaterloo.ca

M. Gössel
Arbeitsgruppe Fehlertolerantes Rechnen
Institut für Informatik, Universität Potsdam
Am Neuen Palais 10, D-14469 Potsdam, Germany
E-mail: mgoessel@mpag-inf.uni-potsdam.de

H. Jürgensen
Department of Computer Science
The University of Western Ontario
London, Ontario, Canada, N6A 5B7
and
Institut für Informatik, Universität Potsdam
Am Neuen Palais 10, D-14469 Potsdam, Germany
E-mail: helmut@uwo.ca

Y. Zorian
Logic Vision Inc.
101 Metro Drive
San Jose, CA, USA 95110
E-mail: zorian@lvision.com

Abstract

In this paper, a new compaction technique based on signature analysis is presented. Rather than comparing the final signature with the expected one after the test is completed, the binary output of the MISA is converted into an alternating binary signal by two simple cover circuits. An error is indicated whenever the alternation of the output signal is disturbed. This technique results in a higher fault coverage, improved fault diagnosis capability, a greater test autonomy in core-based designs, and early fault notification.

1. Introduction

In this paper, a new compacting technique for BIST based on signature analysis is presented. Instead of comparing the actual signature with the stored correct one after the test is completed, an alternating binary sequence is generated and monitored for alternation during the test. The main advantages of this method are as follows.

- It allows for early identification of an erroneous chip by the first deviation from the alternation of the monitored sequence; this may result in a considerable reduction of the necessary test time.
- For fault diagnosis, the deviations from the alternation of the sequence can be utilized.
- For a core based design, the cores do not need to be supplied with a specific signature; this allows for greater autonomy of the different cores.
- Checking the alternation of the sequence during the whole test time instead of comparing the final signature with the stored correct one results in better fault coverage.
- This method is applicable in any situation where a (pseudo) deterministic test sequence without repetition is applied to the circuit under test.

As a consequence of the last point, this approach works well with recently developed methods for (pseudo) deterministic testing [3, 7, 10].

In a typical built-in self-test (BIST) structure, a sequence $x(0), \dots, x(T-1)$ of inputs generated by a test input generator (TIG) is applied to the circuit under test (CUT). The corresponding output sequence $y(0), \dots, y(T-1)$ is applied to a multiple-input signature analyzer (MISA) with initial state $z(0)$. Let $z(T)$ be the final state vector of the MISA after the output sequence $y(0), \dots, y(T-1)$ has been processed. Then $z(T)$ is called the signature of the MISA. Error detection occurs by comparing the actual signature obtained from the test to the correct signature. A fault is indicated in the system if the two signatures are not equal.

If the CUT is erroneous but the actual signature is equal to the correct signature, aliasing has occurred. Several methods have been proposed which reduce the probability of aliasing by monitoring also the output of the MISA [11, 8, 4, 1].

In this paper, we propose to monitor only the one-dimensional output of the MISA during the test, rather than checking the signature after the test. The correct output sequence of the MISA is transformed into an alternating binary output by two simple cover circuits. A fault is indicated in the CUT if the output sequence created from the cover circuit

outputs is not alternating. The cover circuits are constructed by a method similar to that of [1], and are a modification of the original cover circuits introduced in [5].

This paper is structured as follows. In Section 2, the proposed method is introduced. A simple example is presented in Section 3. Section 4 contains the experimental results for ISCAS'85 and Berkeley benchmark circuits. Finally, Section 5 contains some concluding remarks.

2. Description of the method

In this section, we introduce the general principle of the proposed BIST method. Figure 1 shows the new testing scheme. The test input generator generates a test sequence; this sequence is applied to the CUT as well as the two combinational *cover circuits* C_0 and C_1. The corresponding output sequence of the CUT is applied to the multiple-input signature analyzer. The first component of the state of the MISA and the outputs of the cover circuits C_0 and C_1 are combined to form an alternating binary output signal $\varphi(t)$.

A more detailed description of the proposed method is as follows. For a test of length T, the TIG generates an input sequence $x(0), \ldots, x(T-1)$ of m-dimensional binary vectors without repetition. The corresponding output sequence $y(0), \ldots, y(T-1)$ of n-dimensional binary output vectors of the CUT is applied to the MISA, which has initial state $z(0)$. In the error-free case, the CUT implements the combinational function $f_c : X \rightarrow Y$ with $X = \{0,1\}^m$ and $Y = \{0,1\}^n$. For $i = 0, \ldots, T-1$, we have $y(i) = f_c(x(i))$. The function implemented by the CUT when a fault k is present is denoted by f^k.

The MISA is realized as a multiple-input linear feedback shift register with at least n stages. Let $z(1 \mid f_c), \ldots, z(T \mid f_c)$ be the error-free sequence of states of the MISA for the CUT. Then $z(T \mid f_c)$ is the state of the MISA after the correct output sequence $y(0), \ldots, y(T-1)$ of the CUT has been processed by the MISA. On the other hand, when a fault k is present, the final state is $z(T \mid f^k)$, which may or may not be different from $z(T \mid f_c)$.

Let f be the function actually implemented by the CUT. At each step t, the first component $z_1(t+1 \mid f)$ of the state $z(t+1 \mid f)$ of the MISA is combined with the outputs $C_0(x(t))$ and $C_1(x(t))$ of the cover circuits to form the binary output signal $\varphi(t)$. The circuitry is designed in such a way that the signal $\varphi(t)$ is alternating as long as no error occurs.

For the test inputs $x(0), \ldots, x(T-1)$, we define an m-ary Boolean function $g_{f_c}(x)$ using the following rules:

(1) $g_{f_c}(x)$ is defined for all $x \in \{x(0), \ldots, x(T-1)\}$.

(2) For $x = x(t)$, let $g_{f_c}(x) = z_1(t+1 \mid f)$.

The function g_{f_c} is total if and only if $T = 2^m$. It is well-defined because the test sequence $x(0), \ldots, x(T-1)$ for $t \leq 2^m$ contains no repetitions. The cover circuits C_0 and C_1 are partially defined for $t = 0, \ldots, T-1$ by

$$C_0(x(t)) = \begin{cases} t \bmod 2, & \text{if } g_{f_c}(x(t)) = 1, \\ \text{don't care}, & \text{otherwise}, \end{cases} \qquad (1)$$

and

$$C_1(x(t)) = \begin{cases} t \bmod 2, & \text{if } g_{f_c}(x(t)) = 0, \\ \text{don't care}, & \text{otherwise}. \end{cases} \qquad (2)$$

For all x, $x \notin \{x(0), \ldots, x(T-1)\}$, the covers $C_0(x)$ and $C_1(x)$ are not specified. The unspecified cover circuit outputs can be optimized by an optimization program such as *espresso* [2, 9]. During this process, all of the don't-care values are fixed to either 0 or 1. We then have the following theorem.

Theorem 1 *The output $\varphi(t)$ of the monitoring circuit M is alternating as long as no error occurs.*

Proof: By Figure 1,

$$\varphi(t) = C_0(x(t)) \cdot g_{f_c}(x(t)) \vee C_1(x(t)) \cdot \overline{g_{f_c}(x(t))}.$$

One verifies that $\varphi(t) = t \bmod 2$. □

The error detection capability of the monitoring circuitry M is described in the following theorem.

Theorem 2 *An erroneous output*

$$g_{f^k}(x(t)) = z_1(t+1 \mid f^k) \neq z_1(t+1 \mid f_c) = g_{f_c}(x(t))$$

of the MISA at time t results in an erroneous output $\varphi^k(t) = \overline{\varphi(t)}$ of the monitoring circuitry M if and only if $C_0(x(t)) \neq C_1(x(t))$.

Proof: One verifies the statement by computing $\varphi^k(t)$ and $\varphi(t)$ for the possible cases. □

A fault k in the CUT is detected in the following way. The test sequence $x(0), \ldots, x(T-1)$ is applied to the faulty CUT implementing f^k instead of f_c. The MISA produces the output sequence $z_1(1 \mid f^k), \ldots, z_1(T \mid f^k)$. If, for the first time, the output $z_1(t+1 \mid f^k)$ is different from the correct value $z_1(t+1 \mid f_c)$ and if $C_0(x(t)) \neq C_1(x(t))$, the correct output $\varphi(t)$ of the monitoring circuitry M will be changed for the first time into $\varphi^k(t) = \overline{\varphi(t)}$ which is equal to $\varphi(t-1)$. In this situation, the output φ of the monitoring circuit is not alternating and the fault will be detected.

The don't-care conditions of the cover circuits which are partially defined by Equations (1) and (2) are fixed during the optimization process. For $t = 0, \ldots, T-1$, we expect on average half of the non-specified values to be set to 0 and half to be set to 1. Thus, we expect that $C_0(x(t)) = C_1(x(t))$ for half of the values $t \in \{0, \ldots, T-1\}$, and $C_0(x(t)) \neq C_1(x(t))$ for half of the values. Therefore, the expected probability p that the covers actively monitor the MISA output sequence for faulty values is $p = \frac{1}{2}$. Of course, a fault can only be detected by the monitoring circuit M if it produces an erroneous CUT output for at least one of the test vectors $x(0), \ldots, x(T-1)$.

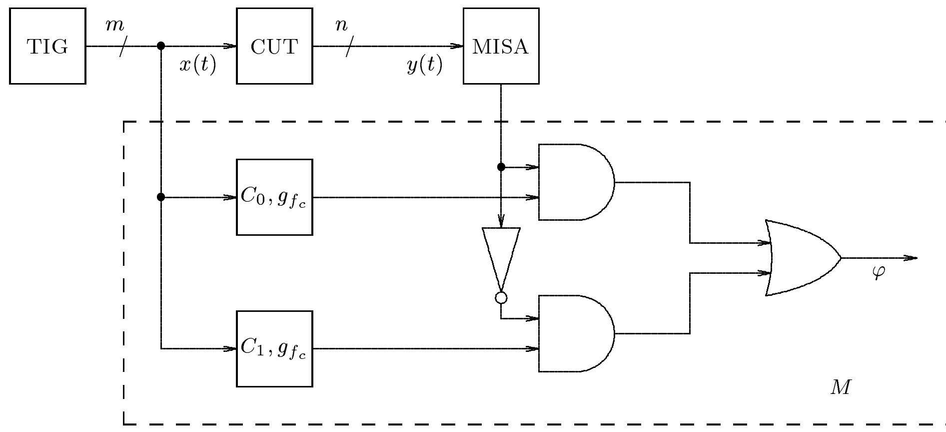

Figure 1. Fault Detection Circuitry Combining BIST and Covers

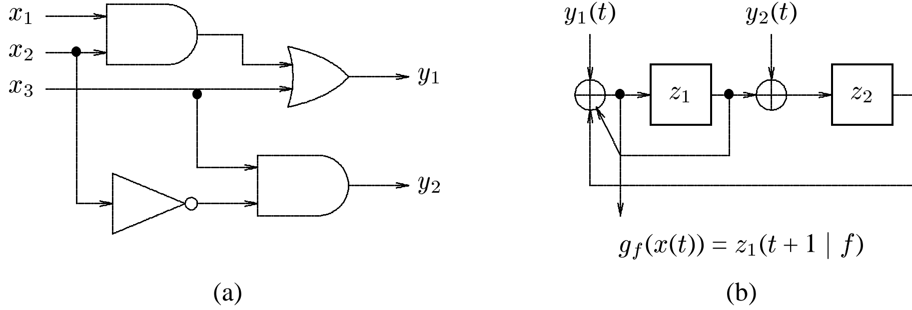

Figure 2. (a) Example circuit; (b) LFSR for example circuit.

3. Example

In Figure 3(a), a small combinational circuit C is shown. The circuit is unrealistically small; it was chosen only to illustrate the application of the proposed method.

The circuit C has three inputs and two outputs; it computes the function

$$f_c(x_1, x_2, x_3) = (f_1(x_1, x_2, x_3), f_2(x_1, x_2, x_3))$$

where $f_1(x_1, x_2, x_3) = ((x_1 \wedge x_2) \vee x_3)$ and $f_2(x_1, x_2, x_3) = (\overline{x_2} \wedge x_3)$. The TIG is a mod-8 counter with initial state 000. It generates the test sequence $x(0), x(1), x(2), x(3), x(4) = 000, 100, 010, 110, 001$. The MISA used is shown in Figure 3(b).

Table 1shows the cover construction for the example circuit. Columns 1–4 show the time t, the test sequence, the CUT output sequence, and the MISA state sequence for the correct circuit C, respectively. The values of the function $g_{f_c}(x(t))$ are given in Column 5. The partially-defined cover circuits are given in Columns 6 and 7. For all x with x not in $\{x(0), \ldots, x(T-1)\}$, the values of C_0 and C_1 are not defined. Possible solutions for the optimized covers, as shown in Columns 8 and 9, are $C_0^{\text{opt}}(x) = x_2$ and $C_1^{\text{opt}}(x) = x_1$. Column 10 shows the output φ of the monitoring circuitry M.

Now suppose that input x_3 in Figure 3(a) is stuck-at-1. Then the erroneous circuit C^k implements the functions $f_1^k(x_1, x_2, x_3) = 1$ and $f_2^k(x_1, x_2, x_3) = \overline{x_2}$. The behaviour of C^k is shown in Table 2. In this table, Columns 1–4 show the time t, the test sequence, the CUT output sequence, and the MISA state sequence for the faulty circuit, respectively. The faulty function $g_{f^k}(x(t))$ is shown in Column 5, while Columns 6 and 7 show the optimized cover circuit values. Column 8 shows the output φ^k of the monitoring circuitry M for the faulty circuit. The fault x_3 stuck-at-1 is detected by input $x(1) = 100$, as we have $g_{f^k}(100) \neq g_{f_c}(100)$ and $C_0(100) \neq C_1(100)$. We see that at time $t = 1$, $\varphi^k(1) = 0 = \varphi^k(0)$; the alternation of the output is interrupted, and the fault is detected.

4. Simulation results

In the simulation experiments, the single stuck-at fault model was used. However, the modified BIST method presented in Section 2 can be used with any fault model. To determine

1	2	3	4	5	6	7	8	9	10
t	$x(t)$	f_c	$z(t+1 \mid f_c)$	g_{f_c}	C_0	C_1	C_0^{opt}	C_1^{opt}	φ
0	000	00	00	0	-	0	0	0	0
1	100	00	00	0	-	1	0	1	1
2	010	00	00	0	-	0	1	0	0
3	110	10	10	1	1	-	1	1	1
4	001	11	00	0	-	0	0	0	0

Table 1. Cover circuit construction for the example circuit C.

1	2	3	4	5	6	7	8
t	$x(t)$	f^k	$z(t+1 \mid f^k)$	g_{f^k}	C_0^{opt}	C_1^{opt}	φ^k
0	000	11	11	1	0	0	0
1	100	11	10	1	0	1	0
2	010	10	01	0	1	0	0
3	110	10	00	0	1	1	1
4	001	11	11	1	0	0	0

Table 2. Behaviour of erroneous example circuit C^1.

the overhead of the alternating cover circuitry, simulation constructions were performed with several ISCAS'85 and Berkeley benchmark circuits. The circuits C432 to C3540 are ISCAS'85 benchmark circuits, while the circuits IN5 to X9DN are Berkeley circuits [6]. For each circuit, a pseudorandom test sequence of length $T = 1000$ was generated using a linear feedback shift register. The simulation of the alternating cover construction was accomplished as follows.

1. For each test input $x(t)$, $t = 0, \ldots, T-1$, check the value $g_{f_c}(x(t))$.
   - If $g_{f_c}(x(t)) = 1$, set $C_{0,g_{f_c}}(x(t)) = t \bmod 2$.
   - If $g_{f_c}(x(t)) = 0$, set $C_{1,g_{f_c}}(x(t)) = t \bmod 2$.
2. Optimize the partially specified covers using a circuit optimization program such as *espresso*.

Table 3 reports the simulation results for BIST monitored by alternating cover circuits. Columns 1–4 show the circuit name, number of gates, number of inputs, and number of outputs, respectively. In Column 5, the total overhead for the cover circuits is shown. The values given are total gates required; the first is the overhead of the best covers assuming that inverted values for all input lines are available, and the second value includes the cost of inverting the required input lines. The value of p obtained in the optimization is reported in Column 6. Finally, Column 7 reports the percentage of faults tested by the test sequence that were detected by the cover circuitry. As an example, suppose a CUT has 100 faults, 80 of which produce faulty CUT outputs for a given test sequence. If 60 faults are detected by the covers, then the value appearing in Column 7 would be $(60/80) * 100 = 75\%$.

The alternating cover circuit constructions yielded very consistent simulation results. Column 5 shows that the covers required between 59 and 83 gates without inversion cost for circuits ranging from 117 to 1719 gates. Thus, the size of the covers does not seem to depend on the size of the circuit under test. In fact, the smallest covers were obtained for the circuits with the most inputs, while the largest covers were required for those circuits with smaller numbers of inputs. For a fixed test length, the covers seem to depend only on the number of inputs of the circuit under test; an approximate indicator of the relative sizes of the covers is the ratio of test length to number of inputs.

On average, 498 of the 1000 cover circuit outputs were made active for the circuits during the construction process. Thus, the covers actively checked for an erroneous CUT output almost 50% of the time. As a result, the probability of detecting an erroneous CUT output sequence should be very high. This supposition is confirmed by the fault simulation results. All single stuck-at faults were considered in the fault model. As the table shows, *all* of the faults which could be detected by faulty CUT outputs with the pseudorandom tests of length 1000 were also detected by the alternating covers for every circuit except C1908. For C1908, 1763 of the 1764 faults in the fault model were detected by the cover circuits constructed. Thus, faults in the stuck-at fault model were detected with very high probability.

5. Concluding remarks

A new BIST method, which is a modification of ordinary signature analysis, has been introduced in this paper. During the test, the one-dimensional output of the MISA is monitored. This output is transformed into an alternating output signal

1	2	3	4	5	6	7
Circuit	Gates	Inputs	Outputs	Total Cover Size (Gates)	p	Tested Faults Detected (%)
C432	196	36	7	69 (105)	0.512	100
C499	243	41	32	67 (108)	0.475	100
C880	443	60	26	59 (116)	0.490	100
C1355	587	41	32	66 (106)	0.505	100
C1908	913	33	25	72 (105)	0.519	99.999
C3540	1719	50	22	65 (114)	0.495	100
IN5	192	24	14	79 (103)	0.465	100
IN7	117	26	10	80 (106)	0.507	100
VG2	189	25	8	83 (108)	0.505	100
X1DN	207	27	6	74 (101)	0.490	100
X9DN	215	27	7	79 (106)	0.516	100

Table 3. Simulation results for BIST with alternating covers.

by two simple cover circuits. An error is detected whenever the alternation of this output signal is disturbed. The additional area overhead for the cover circuits is between 59 and 83 gates for ISCAS'85 and Berkeley benchmark circuits. In the simulation experiments, the proposed method detected 10908 of the 10909 faults tested for by the given test sequences.

References

1. T. Bogue, M. Gössel, H. Jürgensen: BIST with negligible aliasing through random cover circuits. In *Proceedings of Asia and South Pacific Design Automation Conference, Chiba, Japan*. 223–228, 1995.
2. R. K. Brayton, R. Rudell, A. Sangiovanni-Vincentelli, A. R. Wang: MIS, a multi-level interactive optimization system. *IEEE Trans. CAD* **6** (1987), 1062–1081.
3. C. Dufaza, Y. Zorian: On some theoretical properties of LFSRs for BIST applications. In *Proceedings of 3rd IEEE International On-line Workshop*. 184–189, 1997.
4. M. Gössel, H. Jürgensen: Monitoring BIST by covers. In *Proceedings of European Design Automation Conference*. 208–213. IEEE Computer Society Press, 1993.
5. S. J. Hong, D. S. Jones, D. L. Ostapko: Error checking circuit. *United States Patent* **3,764,788** (October 9, 1973).
6. M. Lempel, S. K. Gupta, M. A. Breuer: Test embedding with discrete logarithms. *Proceedings of IEEE VLSI Test Symposium* (1994), 74–80.
7. S. K. Mukund, E. J. McCluskey, T. R. N. Rao: An apparatus for pseudo-deterministic testing. In *Proceedings of VLSI Test Symposium*. 125–131, 1995.
8. D. K. Pradhan, S. K. Gupta: Designing and analyzing BIST techniques and zero aliasing compression. *IEEE Transactions on Computers* **C-40** (1991), 743–763.
9. E. M. Sentovich, K. J. Singh, C. Moon, H. Savoj, R. K. Brayton, A. Sangovianni-Vincentelli: Sequential circuit design using synthesis and optimization. In *Proceedings of the International Conference on Computer Design, Cambride, MA, 1992*. 328–333.
10. H. J. Wunderlich, G. Kiefer: Bit-flipping BIST. In *Proceedings of International Conference on Computer-Aided Design*. 337–343, 1996.
11. Y. Zorian, V. K. Agarwal: Optimizing error masking in BIST by output data modification. *The Journal of Electronic Testing: Theory and Applications* **1** (1990), 59–71.