Fault Set Partition for Efficient Width Compression[1]

Emil Gizdarski*[2] and Hideo Fujiwara**

* Department of Computer Systems, University of Rousse, 7017 Rousse, Bulgaria

** Nara Institute of Science and Technology, Ikoma, Nara 630-0101, Japan

E-mail: egizdarski@ieee.org, fujiwara@is.aist-nara.ac.jp

Abstract

In this paper, we present a technique for reducing the test length of the counter-based pseudo-exhaustive built-in self-testing (BIST) using the width compression method and the divide-and-conquer strategy. More formally, the target faults are divide into K groups such that a binary counter can generate a test set for each group. By selecting the size of binary counter, this technique allows a trade-off between test application time and area overhead to be achieved. We study different options to apply this BIST technique based on the test-per-clock and test-per-scan model. The experimental results for the ISCAS'85 and ISCAS'89 benchmark circuits demonstrated the efficiency of the proposed technique. In all cases, this low-overhead BIST technique achieved complete fault coverage of the stuck-at faults in reasonable test application time.

1.Introduction

Logic BIST became increasingly popular in the VLSI industry because it reduces the cost of manufacturing test as well as improves test quality by providing at-speed test capability [1,4]. When BIST is used, the circuit is partitioned into a number of circuits under test (CUT). Each CUT has an associated test pattern generator (TPG) and output response analyzer usually implemented by a multiple-input signature register (MISR). The efficiency of a BIST technique strongly depends on the abilities to design low-overhead TPGs that achieve very high fault coverage at acceptable test lengths. Different design techniques have addressed this problem, e.g., pseudo-exhaustive testing [19], pseudo-random testing [3,9,18,21,22], and deterministic test set embedding [2,12,17,23]. Recently, the width compression method [7,8,11] has been introduced to further reduce the test length of the pseudo-exhaustive testing. In general, the width compression method involves an analysis of the CUT and finds compatibility between its inputs. Unlike pseudo-exhaustive testing, two inputs are considered as compatible if they can be connected to the same output of the TPG without any loss of fault coverage.

Figure 1 illustrates a test-per-clock model of the counter-based BIST. Accordingly, we suppose a TPG implemented using an N-bit binary counter or complete LFSR, called a de Bruijn counter [4]. This scheme also

[1] *This work was supported by JSPS under grant P99747*

2 Currently visiting at Nara Institute of Science and Technology

includes a mapping logic connected via multiplexers (MUXs) to the outputs of an M-input combinational CUT where M>>N. In this case, the width compression method is used to minimize the size of counter by finding as many compatibility relations as possible between the inputs of the CUT. Four compatibility relations have been introduced in the literature: direct and inverse compatibility [8], decoder compatibility [7] and combinational compatibility [11]. As a result, the mapping logic in [8] includes only direct connections and inverters, whereas the mapping logic in [7] and [11] includes also binary decoders and 2-input gates, respectively.

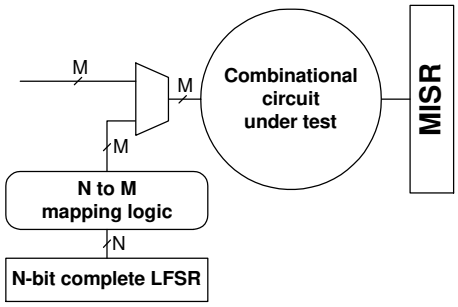

Figure 1: A test-per-clock model of the counter-based BIST

Figure 2 presents a test-per-scan model of the counter-based BIST. Accordingly, we suppose R scan chains each one having at most L flip-flops such that for $\forall i \in \{1,..,L\}$ the inputs of the CUT connected to the i-th flip-flop in all chains are directly compatible. This precondition can be achieved using pseudo-exhaustive technique [19], i.e., initially, two inputs are considered as compatible if they do not belong to a cone (primary output). This scheme includes a mapping logic that determines the compatibility class of all the flip-flops stored during the current load cycle.

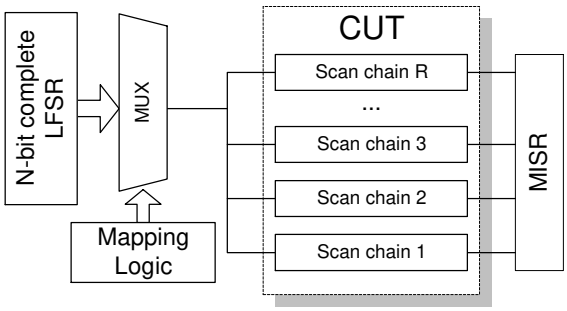

Figure 2: A test-per-scan model of the counter-based BIST

The advantage of the counter-based pseudo-exhaustive testing is a low-area overhead to achieve complete fault coverage of the target faults. Although some recent improvements, this BIST technique still requires relatively long test application time. This is especially valid for the test-per-scan model. In this paper, we deal with this problem and propose a new technique based on fault set partition to further reduce the test length of the counter-based BIST. More formally, we divide the target faults into K groups such that a binary counter can generate a test set for each group. In this way, the TPG is designed as an N-bit configurable LFSR that consists of two or more S_i-bit complete LFSRs able to generate all possible S_i-bit test patterns where i=(1,2,..,K) and S_i<N.

The rest of this paper is organized as follows. In section 2, the compatibility relations previously introduced in the literature are summarized as well as some new compatibility relations are presented that allow to further reduce the test length of the counter-based BIST. In Section 3, the synthesis procedure is provided. In Section 4, the experimental results are given and Section 5 concludes the paper.

2.Compatibility relations

In this section, we first summarize the compatibility relations previously introduced in the literature. Next, we show that a self-testing property of the TPG has to be achieved to guarantee the fault coverage of BIST solution based on the width compression method. Finally, we present new compatibility relations for reducing the test length of the counter-based exhaustive testing.

Two inputs of a combinational circuit are said to be *directly compatible* if they can be shorted together without introducing any redundant stuck-at fault in the circuit. Similarly, two inputs of a circuit are said to be *inversely compatible* if they can be shorted together via inverter without introducing any redundant stuck-at fault in the circuit [8].

A set of inputs of a circuit are said to form *a compatibility class* if all these inputs are directly or inversely compatible to one another, i.e. shortening together directly or via inverters do not introduce any redundant fault in the CUT. Since the same (or opposite) logic value is applied to all inputs in a compatibility class during testing, then the size of the binary counter used for the counter-based exhaustive testing is equal to the number of compatibility classes, i.e., the test length is 2^N where N is the number of compatibility classes.

Two or more inputs of a circuit are said to be *decoder-compatible (D-compatible)* if all detectable stuck-at faults in the circuit can be tested by a test set in which no test vector requires more than one of these inputs to be in value one [7]. This compatibility relation can be used to further reduce the size of the binary counter since the values of D-compatible inputs $x_1,x_2,..,x_k$ can be generated by counter connected to the inputs $x_1,x_2,..,x_k$ via decoder.

In [11], a compatibility relation called *combinational compatibility (C-compatibility)* has been introduced to further reduce the test length of the counter-based exhaustive testing. We use this definition because it covers all previous definitions presented in the literature.

Definition 1: Inputs $x_1,x_2,..,x_k$ of a circuit are said to be Ck-compatible with input xk+1 if a combinational circuit $z=f(x_1,x_2,..,x_k)$ exists whose output z can be connected to input x_{k+1} without introducing any redundant fault in the circuit.

For example, the direct and inverse compatibility relations introduced in [8] can be viewed as C1-compatibility and the D-compatibility relation introduced in [7] is a special case of Ck-compatibility where the combination circuit is a decoder.

Clearly, all previous definitions for width compression require a certain properties, no loss of fault coverage, for the *original* circuit. Herein, we adapt the network example presented in [15] to show that, in some cases, this is not enough to guarantee the fault coverage of BIST solution based on the width compression method.

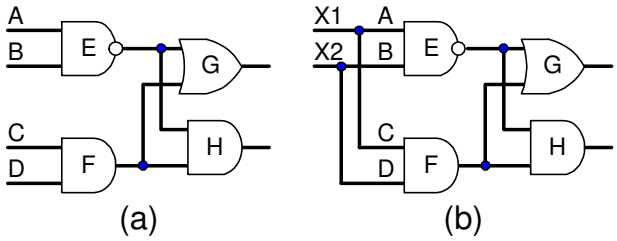

Figure 3: The original and modified circuits

Example 1: Let us consider the circuit shown in Figure 3(a). For this circuit, inputs A,C and B,D are directly compatible. Figure 3(b) shows the *modified circuit* representing the original circuit in test mode. In this case, the stuck-at faults in primary inputs x1 and x2 of the modified circuit are redundant. If one of these redundant faults exists, then an incomplete test set will be applied and some detectable stuck-at faults in the original circuit remain untested due to a fault in the TPG. For example, if stuck-at-1 fault in input x1 exists, then an incomplete test set (two instead of four test patterns) is applied to the original circuit and stuck-at-1 faults in lines A and C remain untested. If stuck-at-1 faults in both inputs x1 and x2 exist, then 8 of 12 stuck-at faults in the original circuit remain untested.

Example 1 shows that some extra properties are necessary to guarantee the fault coverage of BIST solution based on the width compression method. More formally, no lost of fault coverage in respect to the modified circuit instead of the original circuit has to achieve. The following definitions can be used to guarantee the fault coverage of the counter-based exhaustive testing using the width compression method.

Definition 2: Inputs $x_1,x_2,..,x_k$ of a circuit are said to be *strongly* Ck-compatible, denoted as SCk-compatible, with an input x_{k+1} if these inputs are Ck-compatible and all single stuck-at faults in the inputs of the modified circuit are detectable.

Definition 3: A BIST solution has a self-testing property if any single stuck-at fault in the added logic is also tested.

The following definitions can be used to further reduce the test length of the counter-based pseudo-exhaustive testing using the width compression method.

Definition 4: A compatibility relation is said to be *primary*, denoted by α, if this relation does not introduce any redundant stuck-at fault in the circuit. Otherwise, the compatibility relation is said to be *secondary*, denoted by β.

Definition 5: Let $\{\beta_1,\beta_2,..,\beta_k\}$ be a set of secondary compatibility relations for a circuit and $\delta(\beta_i)$ be a set of the redundant faults introduced by β_i. A set $\{\beta_1,\beta_2,..,\beta_k\}$ defines a composite *Pk-compatibility* relation iff $\delta(\beta_1)\cap\delta(\beta_2)\cap\ldots\cap\delta(\beta_k)=\varnothing$.

3.Synthesis procedure

In this section, we present the synthesis procedure for the counter-based pseudo-exhaustive testing using the width compression method and the divide-and-conquer strategy. First, we describe the procedure for mapping

logic synthesis using SC1-compatibility relation. Next, we describe the procedure for further reducing the test length using Pk-compatibility relation. Finally, we discuse on some application aspects of this BIST technique and propose effective solutions to overcome them.

3.1. Mapping logic synthesis

The procedure for mapping logic synthesis is similar to that presented in [7,8,11] and finds SC1-compatibility classes using a greedy strategy. More formally, the first input is assigned to the first compatibility class. Next, if the current input is compatible to all inputs in an existing compatibility class, then this input is added to that compatibility class, otherwise this input is assigned to a new compatibility class. To find SC1-compatibility classes, the synthesis procedure modifies the circuit and also includes the stuck-at faults in the inputs of the modified circuit not presented in the original circuit.

First, all isolated lines, i.e. the lines without associated logic that are both input and output of the original circuit, are removed. In this way, we initially set more restrictive conditions to achieve strong compatibility during width compression. Clearly, if each compatibility class has at least one isolated line, then all stuck-at faults in the inputs of the modified circuit are detectable. Thus, if the circuit has enough isolated lines, then strong compatibility can be easily achieved by adding one isolated line to each compatibility class.

Deriving the compatibility classes requires checking the compatibility of a set of inputs. The synthesis procedure uses similar techniques to those presented in [7,8,11]. These techniques are based on dependency matrix [19], incompatibility filters [8] and fault set reduction [11]. After quick identification of a large number of compatible inputs using dependency matrix, the synthesis procedure tries to reduce the number of compatibility classes using test generation. This technique checks weather shorting two inputs together introduces any redundant stuck-at fault in the modified circuit. The search space in this phase can be considerably reduced using incompatibility filters. More formally, the synthesis procedure calculates the incompatibility filters using both structural information and necessary input assignments of each fault in the target set. Accordingly, two inputs of a circuit are incompatible if they are inputs of one logic gate [8]. Also, two inputs of a circuit are be inversely (directly) incompatible, if a necessary input assignment to detect a fault requires both inputs to be in the same (different) value [11].

This phase is speeded up by fault set reduction. More formally, a test set is dynamically updated during width compression. When checking the compatibility of inputs, the synthesis procedure easily identifies the faults having incompatible values for these inputs. For example, if the synthesis procedure checks for direct compatibility of inputs x_a and x_b, and a test pattern in test set requires different values for these inputs, then all faults detected by this test pattern potentially become redundant, i.e. they should be included in the fault list. After shortening inputs x_a and x_b together (via wire or inverter), if all faults in the fault list are proven to be detectable then these inputs are compatible, otherwise these inputs are incompatible.

3.2. Fault set partition

In general, two approaches for width compression based on test generation [8,11] and precomputed test set [7] exist. Respectively, the divide-and-conquer strategy for width compression will result in fault set and test set partition.

From practical point of view, two different tasks for fault test partition are possible: (1) to minimize the number of groups, K, when the size of counter, S, is fixed and (2) to minimize test length when K=2. Both tasks assume a similar approach so that we present only the first procedure here.

Procedure 1: The input data are the modified circuit, parameter S, and the target faults, $\delta(\beta 0)$, including all detectable single stuck-at faults in the modified circuit. The output data are the Pk-compatibility classes defined by secondary compatibility relations $\beta 1,\beta 2,..,\beta k$ such that the intersection of redundant faults introduced by $\beta 1,\beta 2,..,\beta k$ is empty (see Definition 5).

K=0; while $\delta(\beta k)\neq\varnothing$ do the following:

1) K=K+1; derive βk using a greedy strategy to minimize $\#\delta(\beta k)$ on $\delta(\beta k-1)$. This step continues until the number of Pk-compatibility classes of βk becomes smaller than S.

2) Optimize βk by checking whether each input of the modified circuit can be connected to another Pk-compatibility class so that $\#\delta(\beta k)$ is minimized.

Procedure 1 is based on a dynamic calculation and minimization of $\#\delta(\beta i)$ - the number of redundant faults introduced by secondary compatibility relations βi. This analysis is speeded up by filtering the input pairs of the modified circuit introducing too many redundant faults. Also, Procedure 1 utilizes the following assumption:

Assumption 1: The number of redundant faults introduced in the current step, $\Delta\delta(\beta k)$, can be estimated as a sum of the number of redundant faults introduced by all new pairs of compatible inputs of βk, if each compatibility class of βk contains at most 1 new input.

Assumption 1 is very useful when N, the number of inputs of the modified circuit, is bigger than 30. In this case, for example, to reduce twice the number of compatibility classes, just 2-3 iterations are necessary.

3.3. Designing a self-testing BIST

An important aspect of BIST is to achieve a self-testing property for the added logic. That means, BIST is able to detect not only the target faults in the CUT but also any single stuck-at fault in the TPG. Otherwise, many detectable faults in the CUT may remain untested because of a fault in the TPG. We showed an example for this in the previous section. In fact, the case presented in Example 1 may happen quite often when the width compression method is used because this method allows a very high reduction in the number of the inputs [7,8,11].

First, let us consider the test-per-clock model of the counter-based BIST, shown in Figure 1, where N-bit complete LFSR is replaced by N-bit configurable LFSR. Clearly, some single stuck-at faults in the N-bit

configurable LFSR can be mapped as multiple stuck-at faults in inputs of the modified circuit. To avoid these cases, we propose the following rules for designing the configurable LFSR:

Rule 1: The data input of each flip-flop is feted by either the feedback network or different flip-flop than any other flip-flop in the configurable LFSR.

Rule 2: Each chain in the configurable LFSR longer than 2 is included in the feedback network.

Rule 1 results in forming two or more chains in the configurable LFSR such that no intersection between these chains and the flip-flops in each chain belong to different secondary compatibility classes. The configurable LFSR consists of K complete LFSRs such that each flip-flop has at most K MUXs. Obviously, the number of these MUXs can be considerable minimized by graph-based algorithm.

Lemma 1: Let any single stuck-at fault in the feedback network of the configurable LFSR be tested. Then the TPG has a self-test property, if SC1-compatibility relation is used during the first phase of the synthesis procedure and Rules 1 and 2 are used.

Proof: Since each stack-at fault in the feedback network is tested, then Rules 1 and 2 assure that the remaining faults in TPG are mapped as single stuck-at faults in the inputs of the modified circuit. According to Definition 2, SC1-compatibility assures that each single stack-at fault in the inputs of the modified circuit is tested. Therefore, each single stuck-at fault in the TPG is tested and the TPG has a self-testing property.

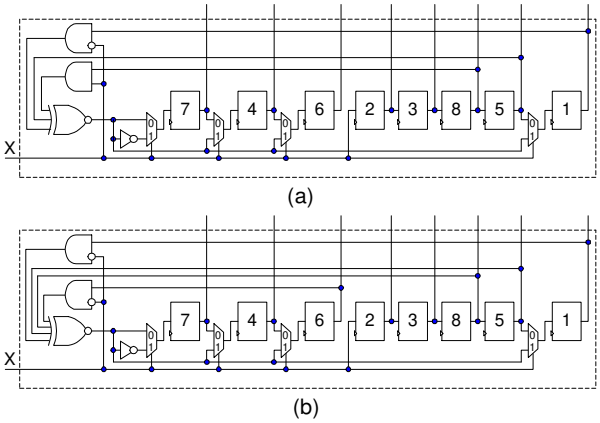

Figure 4: A self-testing TPG

Example 2: Let width compression based on fault set partition result in 8 SC1-compatibility classes and P2-compatibility classes be: {(1,2,4,6,¬7)(3)(5)(8)} and {(1)(2,7)(3,4)(5)(6,8)}. The chains derived by Rule 1 are: <2-3-8-5-1> and <7-4-6> when X=0 and <2-3-8-5>,<7>,<1>,<4> and <6> when X=1. The resultant TPG is shown in Figure 4(a). In this case, chain <7-4-6> has length 3 and is not included in the feedback network. To achieve a self-testing property of the TPG, this chain is included in the feedback network. The resultant TPG is shown in Figure 4(b).

Figure 5 shows a new test-per-scan model of the counter-based BIST where self-testing properties are achieved by connecting the only output of the TPG to the MISR. In this scheme, the mapping logic is implemented by two ROMs. Respectively, ROM1 and ROM2 determine the SC1-compatibility and Pk-compatibility classes of all flip-flops loaded during the current load cycle, where L, N, S and K are the number of

the flip-flops in the longest scan chain, the number of SC1-compatibility and Pk-compatibility classes, and groups of fault set partition obtained by the synthesis procedure. Next, let us consider how to facilitate satisfying the precondition of the test-per-scan model that the i-th flip-flops in all chains are directly compatible for ∀i=(1,2,..,L). Clearly, if the CUT has only one scan chain, then this precondition is always satisfied. Therefore, by choosing a proper value for parameter L, a trade of between the area overhead in the TPG and scan chains can be achieved. For example, by choosing high value for parameter L, we increase the degree of freedom for an ordering and reordering of the scan chains but also increase the size of ROM1. To increase the degree of freedom and keep the size of ROM1 as small as possible, we propose a tree-structure of the scan chains. In this way, if the value of parameter L is properly selected, then this tree-structure allows satisfying the precondition of the test-per-scan model by slight or even without reordering of the scan chains.

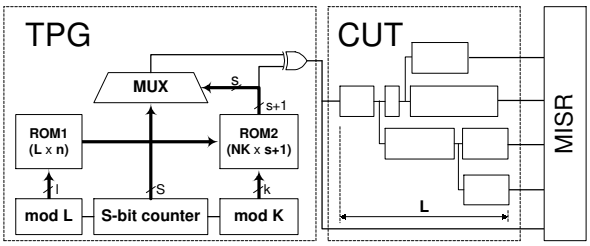

Figure 5: A new test-per-scan model of the counter-based BIST

4.Experimental results

The presented synthesis procedure was implemented using ATPG system SPIRIT[10] and ran on a 1GHz Pentium-III PC. The experimental results for the ISCAS'85[5] benchmark circuits and a full-scan version of the ISCAS'89[6] benchmark circuits are presented in Tables 1 and 2.

Table 1 presents a comparison of the P2- and C2-comapatibility relation – chosen here as an alternative technique for width compression. Columns 2-7 give the number of inputs before and after width compression based on the dependency matrix, C1-compatibility, C2-compatibility and SC1-compatibility relations. Columns 8 and 9 show the size of counters in the configurable LFSR, and the test length obtained by the P2-compatibility technique. Column 10 compares the test length of the P2- and C2-compatibility techniques. These results demonstrate the effectiveness of the P2-compatibility technique in respect to the C2-compatibility technique. The test length reduction was between 1.3 and 15.06 times in all critical cases, i.e., the cases where after width compression using C1-compatibility technique, the test length is bigger than 2^{20}. However, the proposed and alternative techniques for width compression do not contradict to each other. In fact, they can be applied together to further reduce the test length of the counter-based exhaustive testing.

Table 1: Comparing the P2- and C2-compatibility techniques for width compression

Circuit	#Inputs	#Inputs'	Size of counters					Test length	Test length reduction
			Previous techniques			Proposed technique			
			C1[8]	C1[11]	C1+C2[11]	SC1	SC1+P2		
1	2	3	4	5	6	7	8	9	10
C432	36	36	12	10	9	11	(7:8)	384	1.33
C499	41	41	9	11	9	9	(8:7)	384	1.33
C880	60	45	13	13	10	12	(8:7)	384	2.66
C1355	41	41	11	11	10	11	(10:8)	1,280	0.8
C1908	33	33	13	13	12	13	(10:11)	3,072	1.33
C2670	157*	122	22	22	20	22	(15:16)	98,304	10.66
C3540	50	50	17	18	15	17	(12:10)	5,120	6.4
C5315	178	69	13	16	11	15	(11:11)	4,096	0.5
C6288	32	32	6	8	6	8	(5:4)	48	1.33
C7552	206*	194	28	26	23	27	(17:19)	655,360	12.8
S5378	199	61	16	19	16	22	(14:12)	20,480	3.2
S9234	247	90	30	30	22	30	(20:18)	1,310,720	3.2
S13207	644*	212	27	27	17	32	(14:14)	32,768	4
S15850	599*	183	31	31	26	35	(22:18)	4,456,448	15.06
S35932	2048	15	-	6	6	7	(5:4)	48	1.33
S38417	1742	100	-	30	25	32	(24:23)	25,165,824	1.33

* The isolated lines were removed

Table 2: Synthesis results of the Pk-compatibility technique

Modified circuit	#Faults	Size of counter	#δ(βk) - #untested faults after $K2^S$ test patterns								Test length	ROM size	ROM size [13]	CPU time, h
			K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
C7552	6548	S=12	172	66	25	6	0	-	-	-	20K	1645	2688	0.43
		S=15	116	31	7	0	-	-	-	-	128K	1510		0.34
S9234	6451	S=12	725	234	120	61	25	8	0		28K	1500	2310	0.48
		S=15	389	63	15	0	-	-	-	-	128K	1050		0.40
S15850	10759	S=12	774	301	163	87	57	36	17	2..	36K	2673	2403	3.17
		S=15	511	131	37	14	0	-	-	-	160K	1978		2.61
S38714	26974	S=12	2326	1156	696	490	351	248	172	116..	52K	2580	6802	16.36
		S=15	1247	418	217	93	25	4	0	-	224K	1620		8.77
CPU time distribution, %			56.6	25.3	5.9	4.3	2.4	1.8	1.0	0.7	Total:			32.56

Table 2 presents the experimental results for the Pk-compatibility technique when S=12 and 15. Columns 4-11 show the number of untested faults after $K2^S$ test patterns derived by the Pk-compatibility technique. Columns 12-15 gives the test length, size of ROM and processing time in hours for the proposed BIST technique as well as the size of ROM for the best result for the reseeding technique using also random testing and the width compression method [13] - chosen here as alternative technique. The size of ROM for the proposed BIST technique was calculated by the following formula: $nL + (s+1)KN$ where $n=\lceil \log_2 N \rceil$ and $s=\lceil \log_2 S \rceil$. The values of parameters L, N and S are given in columns 3 and 7 of Table 1 and column 3 of Table 2, respectively. Clearly, the proposed BIST technique needs smaller ROM size than the alternative technique [13] while the degree of freedom for scan chain ordering is much higher. For example, if we use the structure of scan chains proposed in [13], then the mapping logic in Figures 2 and ROM1 in Figures 5 become redundant. Also, the ROM size for the proposed BIST technique slightly depends on the size of the circuits. For the typical core size (10K-100K gates) [16], we may expect $L \leq 512$, $N \leq 64$, $K \leq 16$ and $S \leq 16$, i.e., the test length and ROM size will be less than 1M and 8K, respectively.

4.1. Overhead analysis

For the test-per-clock BIST scheme shown in Figure 1, the hardware overhead includes one MUX per input as well as hardware overhead of the TPG, MISR and local BIST controller. The hardware overhead in the TPG is estimated as N flip-flops and N(K-1) MUXs. Example 2 shows the correctness of this estimation. In this particular case (circuit C6288), N=8, K=2 and the TPG has four MUXs, two 3-input XOR gates and two 2-input AND gates, i.e. less than 8 MUXs.

For the test-per-scan BIST scheme shown in Figure 5, the hardware overhead include two ROMs, a complex counter, S-input MUX and MISR as well as extra hardware in scan-chains. In this case, the hardware overhead in local BIST controller is negligible because the local BIST controller can also use the complex counter of the TPG as a pattern counter and shift counter.

4.2. Comparison with other BIST techniques

For simplicity, let us suppose that two ultimately goals of BIST solution are: complete fault coverage and lowest possible performance penalty. The most promising scan-based techniques able to achieve these goals in reasonable test application time are based on the reseeding of LFSRs [12,13] and the bit-flipping [16,23]. In this analysis, we exclude the test point insertion technique [20] because this technique does not guarantee complete fault coverage [14].

Merits: Clearly, the proposed technique has abilities to achieve much lower area overhead than the alternative scan-based deterministic BIST techniques, and also that the area overhead slightly depends on the size of the CUT. In respect to the previous counter-based BIST techniques, the proposed technique allows a further reduction in both the test length and the computational complexity for width compression. For example, the computational complexity of Procedure 1 based on the Pk-compatibility technique is $O(KN^2)$ while the computational complexity of the Ck-compatibility technique is $O(N^{K+1})$ when $K<<N$. In this way, the proposed technique allows a trade-off between area overhead and test application time to be achieved when complete fault coverage of the single stuck-at faults in both the CUT and TPG is ensured.

Demerit: The weak point of this BIST technique is the computational complexity. The performance results for the first phase of the synthesis procedure - width compression using the C1-compatibility technique - reported in [11] are promising in this direction (CPU time achieved on a 200MHz Pentium Pro PC was less than 1530 sec. for each circuit in the ISCAS'85 and ISCAS'89 benchmarks). To speed up the second phase of the synthesis procedure, a parallel execution and elaboration of Procedure 1 are necessary. For example, a hierarchical partition based on a test set partition using both the Ck- and Pk-compatibility relations before a fault set partition will decrease parameter N and considerably reduce the computational complexity of Procedure 1. Alternatively, the BIST technique can be used as complementary, for example, of the STAR-BIST technique [22]. Since the STAR-BIST technique is able to achieve very high fault coverage by pseudo-random testing, therefore, the target fault set and parameter N will be considerably reduced. According to the last row in Table 2, a fault set reduction will speed up at least 2 times Procedure 1. Also, we expect significant speed up of Procedure 1 because of the

decreasing parameter N - the number of the Ck-compatibility classes derived during the first phase of the synthesis procedure.

5. Conclusions

In this paper, we presented a new technique for reducing the test length of the counter-based BIST based on the width compression method and the divide-and-conquer strategy. We studied different options to apply this technique based on the test-per-clock and test-per-scan models. The experimental results for the ISCAS'85 and ISCAS'89 benchmark circuits demonstrated the effectiveness of the proposed BIST technique. When K=2 (fault set partition into two groups), in all critical cases, much shorter test length was achieved than the previously published counter-based pseudo-exhaustive BIST techniques. A further reduction of the test length was achieved by increasing K (the number of groups for fault set partition). As a result, the proposed BIST technique showed abilities to achieve much lower area overhead than the previously published deterministic scan-based BIST techniques.

References:

[1] M.Abramovici, M.Breuer and A.Friedman, "Digital Systems Testing and Testable Design", IEEE Computer Science Press, 1990.

[2] S.Akers and W.Jansz, "Test Set Embedding in a Built-in Self-Test Environment," Proc. IEEE ITC, 1989, pp.257-263.

[3] P.H.Bardell, W.McAnney, "Self-Testing of Multichip Logic Modules, Proc. IEEE ITC, 1982, pp.200-204.

[4] P.H.Bardell, W.McAnney and J.Savir, "Built-In Test for VLSI: Pseudorandom Techniques," John Wiley & Sons, NY, 1987.

[5] F.Brglez and H.Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortran," Proc. IEEE ISCAS 1985, pp. 663-698.

[6] F.Brglez, D.Bryan and K.Kozminski, "Combinational Profiles of Sequential Benchmark Circuits," Proc. IEEE ISCAS 1989, pp.1929-1934.

[7] K.Chakrabarty, B.Murray, J.Liu and M.Zhu, "Test Width Compression for Built-in Self Testing," Proc. IEEE ITC, 1997, pp.328-337.

[8] C.Chen and S.K.Gupta, "A Methodology to Design Efficient BIST Test Pattern Generators," Proc. IEEE ITC, 1995, pp.814-823.

[9] F.Corno, M.Reorda, G.Squillero and M.Violante, "CA-CSTP: A New BIST Architecture for Sequential Circuits", Proc. IEEE ETW, 2000, pp.167-171.

[10] E.Gizdarski and H.Fujiwara, "SPIRIT: A Highly Robust Combinational Test Generation Algorithm," Proc. IEEE VTS, 2000. (to appear)

[11] I.Hamzaoglu and J.H.Patel, "Reducing Test Application Time for Built-in Self-Test Test Pattern Generators," Proc. IEEE VTS, 2000, pp.369-375.

[12] S.Hellebrand, J.Rajski, S.Tarnick, S.Venkataraman and B.Courtois, “Built-in Test for Circuits with Scan Based on Reseeding of Multiple-Polynomial Linear Feedback Shift Registers,” IEEE Trans. on Computers, vol.C-44, No.2, Feb. 1995, pp.223-233.

[13] S.Hellebrand, H.Liang and H.Wunderlich, “A Mixed Mode BIST Scheme Based on Reseeding of Folding Counters,” Proc. IEEE ITC, 2000, pp.778-784.

[14] G.Hetherington, T.Fryars, N.Tamarapalli, M.Kassab, A.Hassan and J.Rajski, “Logic BIST for Large Industrial Designs: Real Issues and Case Studies”, Proc. IEEE ITC, 1999, pp.358-367.

[15] M.Inoue, E.Gizdarski and H.Fujiwara, “A Class of Sequential Circuits with Combinatorial Test Generation Complexity under Single-Fault Assumption,” Proc. IEEE ATS, 2000, pp.398-403.

[16] G.Kiefer, H.Vranken, E.Marinissen and H.Wunderlich, “Application of Deterministic Logic BIST on Industrial Circuits,” Proc. IEEE ITC, 2000, pp.105-114.

[17] B.Koenemann, “LFSR-Cored Test Patterns for Scan Designs,” Proc. IEEE European Test Conference, 1991, pp.237-242.

[18] A.Krasniewski and S.Pilarski, “Circular Self-Test Path: A Low-cost BIST Technique for VLSI Circuits,” IEEE Trans. on CAD, vol.8, No.1, Jan. 1989, pp.46-55.

[19] E.J.McCluskey, “Verification Testing – A Pseudoexhaustive Test Technique,” IEEE Trans. on Computers, vol.C-33, No.6, June 1984, pp.541-546.

[20] N.Tamarapalli and J.Rajski, “Constructive Multi-Phase Test Point Insertion for Scan-Based BIST,” Proc. IEEE ITC, 1996, pp.649-658.

[21] N.A.Touba and E.J.McCluskey, “Synthesis of Mapping Logic for Generating Transformed Pseudo-Random Patterns for BIST,” Proc. IEEE ITC, 1995, pp.674-682.

[22] K.Tsai, J.Rajski and M.Marek-Sadowska, “Statr Test: The Theory and Its Application”, IEEE Trans. on CAD, vol.19, No.9, Sept.2000, pp.1052-1063.

[23] H.Wunderlich and G.Kiefer, “Bit-Flipping BIST,” Proc. IEEE ICCAD, 1996, pp.337-343.