

Fault Set Partition for Efficient Width Compression¹

Emil Gizdarski* and Hideo Fujiwara**

* Synopsys Inc., Mountain View, CA 94043

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

Abstract

In this paper, we present a technique for reducing the test length of counter-based pseudo-exhaustive built-in self-testing (BIST) based on width compression method. More formally, the target faults are divided into K groups such that a binary counter can generate a test set for each group. By selecting the size of the binary counter, this technique allows a trade-off between test application time and area overhead. The experimental results demonstrate the efficiency of the proposed technique. In all cases, this low-overhead BIST technique achieves complete fault coverage for the stuck-at faults in reasonable test application time.

1. Introduction

Logic BIST is gaining acceptance in the VLSI industry because it eliminates the need of expensive test equipment, provides at-speed and in-system testing capabilities. In a core-based BIST strategy, 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 - a multiple-input signature register (MISR). The efficiency of a BIST strongly depends on the abilities to design a low-overhead TPG that achieves high fault coverage with acceptable test lengths. Different design-for-test techniques have addressed this problem, e.g., pseudo-exhaustive testing [11], pseudo-random testing, and deterministic test set embedding [1,7,8,9,10]. Recently, the width compression method [2,5] has been introduced to further reduce the test length of pseudo-exhaustive testing. In general, the width compression method is based on finding compatibility relations between the inputs of the CUT. Unlike pseudo-exhaustive testing, two inputs are considered compatible if they can be connected to the same output of the TPG without any loss of fault coverage. The advantage of the counter-based pseudo-exhaustive testing is the low-area overhead. Despite some recent improvements [5], pseudo-exhaustive testing still has relatively long test length that limits its application to the test-per-clock BIST scheme. In this paper, we address this problem and propose a new compatibility relation to reduce the test length for pseudo-exhaustive testing.

2. New compatibility relation

According to with compression method, two inputs of a CUT are *directly* (or *inversely*) *compatible* if they can be shorted together directly or via inverter without introducing any redundant stuck-at fault in the CUT [2]. A set of inputs forms a *compatibility class* if all these inputs are directly or inversely compatible to one another. As the same (or opposite) logic value is applied to all inputs in a compatibility class during testing, the size of the binary counter used as a TPG is equal to the number of compatibility classes, i.e., the test length for pseudo-exhaustive testing is 2^N where N is the number of compatibility classes.

New compatibility relation: Let $\{c_1, c_2, \dots, c_N\}$ be the set of compatibility classes for a given circuit. Let β_i define a partition of the set of compatibility classes into S blocks $\{b_{i1}, b_{i2}, \dots, b_{iS}\}$ each one consisting of one or more compatibility classes. Let $\lambda(\beta_i)$ be a set of the redundant faults introduced by partition β_i . Then partitions $\{\beta_1, \beta_2, \dots, \beta_k\}$ defines a composite Pk -compatibility relation iff the set of undetected faults, $\Delta\lambda_x = \lambda(\beta_1) \cap \lambda(\beta_2) \cap \dots \cap \lambda(\beta_k) = \emptyset$, is empty.

Clearly, the test length for the counter-based testing using the Pk -compatibility relation is equal to $2^{S_1} + 2^{S_2} + \dots + 2^{S_k}$ where S_i is the number of blocks for partition β_i . Also, if the number of blocks of all partitions is equal to S , then the test length of the counter-based testing becomes $K2^S$.

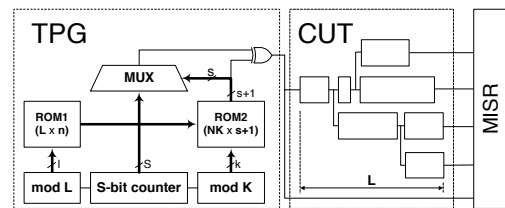


Figure 1: A test-per-scan scheme of the counter-based exhaustive BIST

3. Synthesis procedure

Figure 1 presents the target test-per-scan scheme based on a full-scan approach. Accordingly, in test mode, the CUT is transformed into a combinational circuit and all primary inputs and internal registers are included into one or more scan chains having maximum length L . Also, all flip-flops (FF's) in a position i of all scan chains are directly compatible. This precondition is achieved using pseudo-exhaustive technique [11], i.e., initially, two inputs of CUT are considered as compatible if they do not share a common cone (primary output). The proposed BIST architecture has only one test mode corresponding to the broadcast test mode of the Illinois scan [6]. TPG consists of two ROM's, a complex binary counter and MUX. The first section of the complex counter is a counter modulo L which together with ROM1 determines the compatibility class for each FF in the scan chains. More formally, if the FF's in position i belongs to class c_x then ROM1 in address $(L-i)$ contains an unique n -bit integer corresponding to class c_x where $n = \lceil \log_2 N \rceil$. Similarly, the third section of the complex counter is a counter modulo K that together with ROM2 determines the block of each compatibility class for each partition. For example, if compatibility class c_x in partition β_j belongs to block b_{jm} , where $x \in \{1, \dots, N\}$, $j \in \{1, \dots, K\}$ and $m \in \{1, \dots, S\}$, then ROM2 in address $N(i-1) + (x-1)$ contains an unique s -bit integer corresponding to the block b_{jm} where $s = \lceil \log_2 S \rceil$. In fact, ROM2 has $(s+1)$ -bit word and the most significant bit determines the type of compatibility relation –

¹ This work was partly supported by JSPS under grant P99747

directly or inversely – between the compatibility classes within one block. For example, if compatibility classes c_x and c_y in partition β_i belong to one block and are directly or inversely compatible, then the most significant bits in addresses $N(i-1)+x-1$ and $N(i-1)+y-1$ have equal or different value, respectively.

From practical point of view, two different tasks for width compression based on divide-and-conquer strategy are possible: (1) minimize K the number of partitions when S the size of the counter is fixed and (2) minimize test length when $K=2$ [4]. Here, we present the synthesis procedure for the first task.

Procedure 1: The input data are the CUT, the compatibility classes, parameter S , and the target fault set $\Delta\lambda(0)$. The output data are partitions $\beta_1, \beta_2, \dots, \beta_k$ of the compatibility classes into S blocks such as the set $\Delta\lambda(k)$, i.e., the intersection of the redundant faults introduced by $\beta_1, \beta_2, \dots, \beta_k$, is empty.

$K=0$; while $\Delta\lambda(k) \neq \emptyset$ do the following:

2) $K=K+1$; derive partition β_k using a greedy strategy to minimize $\Delta\lambda(k)$. Initially, the number of blocks of partition β_k is equal to N – the number of compatibility classes. Next, if two blocks are merged then the number of blocks in partition β_k decreases by 1. This step continues until the number of blocks in partition β_k becomes equal to S .

3) Optimize partition β_k by checking whether each compatibility class can be included to another block so that $\Delta\lambda(k)$ is minimized.

4. Experimental results

The presented synthesis procedure was implemented using ATPG system SPIRIT[3] and ran on a 1GHz Pentium-III PC. The experimental results are presented in Tables 1 when $S=12$ and $S=15$. Columns 2-9 show the results for the proposed BIST technique: the length of scan chains (L), the number of compatibility classes (N), blocks (S) and partitions (K) as well as the test length of counter-based testing, the ROM size and the CPU time for Procedure 1. In this case, the ROM size for the proposed BIST technique was calculated by the following formula: $nL + (s+1)KN$ where $n=\lceil \log_2 N \rceil$ and $s=\lceil \log_2 S \rceil$. Columns 12-15 give ROM size of the best-published results based on reseeding. Obviously, the proposed technique achieved higher compression of test data than the techniques based on reseeding. In fact, some of these techniques [7,8,10] used also Pseudo Random Test Generation (PRTG) and width compression method. For example, if we used the structure of scan chains based on compatibility relations method proposed in [8], then ROM1 would be redundant. These experimental results show that the size of test data (ROM's) for the proposed BIST technique slightly depends on the size of the circuits. For the typical cores

(10K-100K gates) [9], we may expect $L \leq 512$, $N \leq 64$, $K \leq 16$ and $S \leq 16$, i.e., the test length and the size of test data will be less than 1M and 8K, respectively.

5. Conclusions

We presented a new technique for reducing the test length of the counter-based BIST using the width compression method. The experimental results for the ISCAS'85 and ISCAS'89 benchmark circuits demonstrated the effectiveness of the proposed technique. When $K=2$ (fault set partition in two groups), much shorter test length was achieved than the previously published counter-based pseudo-exhaustive BIST technique [4]. A further reduction of the test length was achieved by increasing parameter K . As a result, the proposed BIST technique achieved the higher compression of test data than all previously published BIST techniques based on reseeding. Also, the size of the test data slightly depends on the size of the CUT.

References:

- [1] K. Chakrabarty and S. Swaminathan, "Built-In Self-Testing of High-Performance Circuits using Twisted-Ring Counters," Proc. IEEE ISCAS, 2000, pp.72-76.
- [2] C. Chen and S. K. Gupta, "A Methodology to Design Efficient BIST Test Pattern Generators," Proc. IEEE ITC, 1995, pp.814-823.
- [3] E. Gizdarski and H. Fujiwara, "SPIRIT: A Highly Robust Combinational Test Generation Algorithm," Proc. IEEE VTS, 2000, pp.346-351.
- [4] E. Gizdarski and H. Fujiwara, "Fault Set Partition for Efficient Width Compression," Technical report of IEICE, FTS2000-88, Feb.2001, pp.17-24.
- [5] 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.
- [6] I. Hamzaoglu and J. H. Patel, "Reducing Test Application Time for Full Scan Embedded Cores," Proc. IEEE FTCS, 1999, pp.260-267.
- [7] 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.
- [8] 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.
- [9] G. Kiefer, H. Vranken, E. Marinissen and H. Wunderlich, "Application of Deterministic Logic BIST on Industrial Circuits," Proc. IEEE ITC, 2000, pp.105-114.
- [10] H. Liang, H. Wunderlich and S. Hellebrand, "Two-Dimensional Test Data Compression for Scan-Based Deterministic BIST," IEEE ITC, 2001, pp 894-902.
- [11] E. J. McCluskey, "Verification Testing – A Pseudoexhaustive Test Technique," IEEE Trans. on Computers, vol.C-33, No.6, June 1984, pp.541-546.

Table 2: Comparing the P_k -compatibility and reseeding techniques

Circuit	L	N	S	K	Test length	ROM1	ROM2	CPU time, h	ROM size for the reseeding technique			
									[7]	[8]	[1]	[10]
1	2	3	4	5	6	7	8	9	10	11	12	13
C7552	194	27	12	5	20K	970	675	0.43	5241	2688	-	4788
			15	4	128K		540	0.34				
S9234	90	30	12	7	28K	450	1050	0.48	6923	2310	12350	3800
			15	4	128K		600	0.40				
S15850	183	35	12	9	36K	1098	1575	3.17	6528	2403	6721	3360
			15	5	160K		875	2.61				
S38714	100	32	12	13	52K	500	2080	16.36	24283	6802	31616	11214
			15	7	224K		1120	8.77				
Total						6153	8398	32.56	42975	14203	50687	23162