Fault Set Partition for Efficient Width Compression

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 the width 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 \( \{c_1, c_2, \ldots, c_N\} \) be the set of compatibility classes for a given circuit. Let \( \beta_i \) define a partition of the set of compatibility classes into \( S \) blocks, \( \{b_1, b_2, \ldots, b_3\} \) each one consisting of one or more compatibility classes. Let \( \Delta(\beta_i) \) be a set of the redundant faults introduced by partition \( \beta_i \). Then partitions \( \{\beta_1, \beta_2, \ldots, \beta_L\} \) defines a composite \( P_k \)-compatibility relation iff the set of undetected faults, \( \Delta(\beta_i) \cap \Delta(\beta_j) \neq \emptyset \) is empty.

Clearly, the test length for the counter-based testing using the \( P_k \)-compatibility relation is equal to \( 2^{S_1} + 2^{S_2} + \ldots + 2^{S_\beta} \) where \( S_j \) is the number of blocks for partition \( \beta_j \). Also, if the number of blocks of all partitions is equal to \( S \), then the test length of the counter-based testing becomes \( K^2 \).

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 \( c_i \), then ROM1 in address \( (L-i) \) contains an unique \( n \)-bit integer corresponding to class \( c_i \), where \( n=\log_2 N \). Similarly, the third section of the complex counter is a counter by modulo \( K \) that together with ROM2 determines the block of each compatibility class for each partition. For example, if compatibility class \( c_i \) in partition \( \beta_i \) belongs to block \( b_m \), where \( x \in \{1, \ldots, N\} \) and \( i \in \{1, \ldots, K\} \) and \( m \in \{1, \ldots, S\} \), then ROM2 in address \( N(i-1)+(x-1) \) contains an unique \( s \)-bit integer corresponding to the block \( b_m \), where \( s=\log_2 s \). 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 $c_i$ and $c_j$ in partition $\beta$ 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 set $\Delta \lambda(0)$. The output data are partitions $\beta_1, \beta_2, ..., \beta_K$ of the compatibility classes into $S$ blocks such as the set $\Delta \lambda(k)$, i.e., the intersection of the redundant faults introduced by $\beta_1, \beta_2, ..., \beta_K$ is empty.

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

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

2) Optimize partition $\beta_K$ 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 $n=\log_2 N$ and $s=\log_2 S$, $N(12-15)$ 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. For example, if compatibility classes belong to one block and are directly or inversely –

#### Table 2: Comparing the $P_K$-compatibility and reseedings techniques

<table>
<thead>
<tr>
<th>Circuit</th>
<th>$L$</th>
<th>$N$</th>
<th>$S$</th>
<th>$K$</th>
<th>Test length</th>
<th>ROM1</th>
<th>ROM2</th>
<th>CPU time, h</th>
<th>ROM size for the reseeding technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>C7552</td>
<td>194</td>
<td>27</td>
<td>12</td>
<td>5</td>
<td>20K</td>
<td>970</td>
<td>675</td>
<td>0.43</td>
<td>5241</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>4</td>
<td>128K</td>
<td>450</td>
<td>540</td>
<td>0.34</td>
<td>6923</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>12</td>
<td>28K</td>
<td>450</td>
<td>600</td>
<td>0.40</td>
<td>2310</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>12</td>
<td>128K</td>
<td>1050</td>
<td>600</td>
<td>0.40</td>
<td>12350</td>
</tr>
<tr>
<td>S9234</td>
<td>90</td>
<td>30</td>
<td>12</td>
<td>7</td>
<td>28K</td>
<td>450</td>
<td>1050</td>
<td>0.48</td>
<td>6923</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>4</td>
<td>128K</td>
<td>450</td>
<td>600</td>
<td>0.40</td>
<td>2310</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>12</td>
<td>28K</td>
<td>450</td>
<td>600</td>
<td>0.40</td>
<td>12350</td>
</tr>
<tr>
<td>S15850</td>
<td>183</td>
<td>35</td>
<td>12</td>
<td>9</td>
<td>36K</td>
<td>1098</td>
<td>1575</td>
<td>3.17</td>
<td>6528</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>5</td>
<td>160K</td>
<td>1098</td>
<td>875</td>
<td>2.61</td>
<td>2403</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>7</td>
<td>224K</td>
<td>1098</td>
<td>1120</td>
<td>8.77</td>
<td>6721</td>
</tr>
<tr>
<td>S38714</td>
<td>100</td>
<td>32</td>
<td>12</td>
<td>13</td>
<td>32K</td>
<td>500</td>
<td>2080</td>
<td>16.36</td>
<td>24283</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>7</td>
<td>224K</td>
<td>500</td>
<td>1120</td>
<td>8.77</td>
<td>6802</td>
</tr>
<tr>
<td>Total</td>
<td>6153</td>
<td>8398</td>
<td>32.56</td>
<td>42975</td>
<td>14203</td>
<td>50687</td>
<td>23162</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

References:


