An Efficient ATPG for Crosstalk-Induced Delay Faults

Sunghoon Chun, Taejin Kim, Yongjoon Kim, Myung-Hoon Yang and Sungho Kang
Department of Electrical and Electronic Engineering Yonsei University
134 Shinchon-dong Seodaemoon-gu, Seoul, Korea
Tel: +82-2-2123-2775, Fax: +82-2-313-8053
<shchun, tj2010, yjkim, mhyang, shkang>@yonsei.ac.kr

Abstract

For short interconnection line in logic circuits, in this paper, we propose a new test generation method for delay faults considering crosstalk-induced delay effects, based on a conventional delay ATPG technique in order to reduce the complexity of previous ATPG algorithm for crosstalk delay faults and to consider multiple aggressor crosstalk faults to maximize the noise of the victim line.

Keywords: delay faults, crosstalk-induced delay, ATPG, fault modeling, timing analysis

I. Introduction

In current trends in integrated circuit design, it is inevitable to eliminate errors caused by crosstalk because of stringent area and performance requirements in recent designs. These noises due to the crosstalk could be eliminated by resizing drivers, shielding interconnect techniques, rerouting signals and repeater insertion techniques. However, these redesign techniques using the analysis of the crosstalk-induced noise may be very expensive in terms of time and design efforts. In addition, since the process variation cannot be expected for all cases, these redesign techniques cannot guarantee the high noise immunity of the circuits. Therefore, automatic test pattern generation (ATPG) for crosstalk faults is an important method to verify and test that a chip can meet performance requirements [1-9].

Therefore, in this paper, we propose a new test generation method for delay faults considering crosstalk-induced delay effects, based on a conventional delay ATPG technique in order to reduce the complexity of previous ATPG algorithm for crosstalk delay faults and to consider multiple aggressor crosstalk faults to maximize the noise of the victim line. Since the proposed ATPG for crosstalk-induced delay faults uses the test patterns obtained from the conventional path delay ATPG, the proposed ATPG can reduce the search space of the backward implication of the aggressor’s constraints and it is helpful for reducing the time cost of the ATPG than previous works. In addition, since the proposed technique targets on the critical path for the original delay test as the victim lines, it can improve test effectiveness of delay testing.

For a sensitizable victim path, a new algorithm is proposed which appropriately activate its associated aggressor-victim pairs for maximization of crosstalk. The proposed algorithm uses the parasitic information such as coupling capacitance between a node of the victim and an aggressor and the timing information such as static timing window and the crosstalk-induced noise delay model. Using the parasitic and timing information, the proposed algorithm handles multiple aggressors coupled to a victim lying along a path and activates the aggressors in the best possible way to induce maximum crosstalk slowdown along a path.

II. Selection of Target Faults

To reduce the complexity of the test pattern generation for crosstalk-induced delay faults, the methods to prune the false faults and to determine appropriate target faults are required. In the proposed ATPG for crosstalk-induced delay, three fault-pruning methods are used for selecting target faults.

1. Spatial Pruning

Intuitively, two adjacent lines in the layout are potential candidates for capacitive coupling. From this fact, a spatial pruning can be performed: only adjacent lines may have crosstalk coupling effects. Using the commercial parasitic extraction tools, coupling capacitances between any adjacent lines can be extracted. Any capacitive coupling effect across more than one metal line is insignificant in modern process and is typically ignored.

In the proposed ATPG, first, First, all the possible candidates related to a given victim path are identified and then they are inserted into the aggressor candidate list of the given victim path as the decreasing order of the coupling
capacitance value. If the coupling capacitance value in the candidate list is smaller than the threshold coupling capacitance, then the aggressor candidates can be eliminated from the aggressor candidate list. Therefore, using this criterion, the number of the potential aggressors can be reduced significantly.

2. Temporal Pruning
The static timing analysis (STA) computes the minimum and maximum values of arrival and transition times for rising and falling transitions at each circuit line. These calculated values are called timing windows of circuit lines. The crosstalk-induced slowdown delay is strongly related to the timing window of each line. If the timing window of the victim line overlaps the timing window of the aggressor line, then the victim line may be affected by the aggressor line and the crosstalk-induced noise can be occurred on the victim line. Otherwise, then the victim line will have any crosstalk-induced noise. In this case, the functional information is ignored but the timing information is included. Therefore, detecting the overlap of timing windows between coupled lines is significant to reduce the false faults in the aggressor candidate list obtained from the spatial pruning presented in section 2.1.

3. Functional Pruning
Applying the proposed spatial and temporal pruning might be insufficient to reduce false aggressor candidates and search spaces for the proposed ATPG. Just as the proposed temporal pruning can be used in timing analysis to eliminate false aggressors that are never responsible for the crosstalk-induced delay, functional pruning can be used in logic simulation to eliminate those signals that can be never be responsible for noise problems because of their functional relationship.

In the proposed method, any functional pruning does not performed separately. Instead, false aggressor candidates that have no functional relationship of the victim path, which means that the aggressor line cannot satisfy the logical condition for causing crosstalk-induced slowdown, are eliminated during the test pattern generation procedure for crosstalk-induced slowdown faults.

III. Test Pattern Generation Using Delay Tests

After pruning the false aggressor lines in the aggressor candidate list, in the proposed ATPG, the crosstalk fault models are generated as the target of the proposed ATPG. In order to generate test patterns to maximize crosstalk-induced delays for given paths, a crosstalk path delay fault (XPDF) is defined as a critical path and a set of aggressor candidates coupled to it. Critical paths refer to paths whose delay is longer than a given percentage of the longest propagation delay in the circuit and are the same as a general path delay faults.

\[ T_{\text{d}_i} \]: a original delay test pattern of the victim path \( p \)
\[ XFL \]: the fault list of the XPDFs
\[ XPDF \]: its XPDF
\[ \text{AGG}_1 \]: the total number of target aggressors
\[ \text{AGG}_k \]: \( k \)th aggressor line in the XPDFs
\[ \text{VIC}_k \]: \( k \)th victim line coupled to \( \text{AGG}_k \)
\[ C_c \]: the coupling capacitance value of \( k \)th candidates in XPDF
\[ C_v \]: the total coupling capacitance which affects to the victim path \( p \)
\[ T_{\text{x}} \]: the generated test pattern for the XPDF

\[
\text{Test\_generation\_for\_XPDF}() \{
\text{while} (XFL \text{ is not empty}) \{
\text{XPDF} = \text{select\_XPDF}(XFL);
\text{sort\_aggressor\_candidates\_in\_XPDF}(XPDF);
\text{AGG} = 0;
\text{for}(k=0; k<\text{the number of candidates}; k++) \{
\text{flag} = \text{apply\_constraints}(\text{AGG});
\text{if} (\text{flag} == \text{SUCCESS}) \{
\text{flag} = \text{backward\_implication}(\text{AGG});
\text{T}_x = \text{forward\_implication}(\text{XPDF});
\text{if} (\text{flag} == \text{SUCCESS}) \{
\text{AGG} = \text{AGG} + 1;
\}
\text{else} \{
\text{XPDF} = \text{eliminate\_candidate}(\text{AGG}, \text{VIC});
\text{continue;}
\}
\}\}
\text{if}(\text{AGG}>0)
\text{T}_x = \text{forward\_implication}(\text{XPDF});
\}
\}
\}
\]

Figure 1. Test generation algorithm of the proposed ATPG

The goal of the proposed ATPG for XPDF faults is to find a maximal test pattern for path \( p \) with a conventional delay test pattern. To find a worst case test for a given victim path \( p \), the don’t care values can be used to apply some extra constraint transitions which makes crosstalk-induced noise maximal. In [3], a genetic algorithm is used to find a worst case test for the victim path after a conventional path delay fault ATPG process without justification process. However, since it does not care whether or not the target fault is activated finally, the generated test pattern in [3] is actually not a real test for target fault. This fact leads to difficulties in diagnosis. In addition, the ATPG based on the genetic algorithm tends to be very time consuming. Therefore, unlike previous works, the proposed ATPG requires a process to find possible worst case test patterns for XPDF faults using don’t care values in delay test patterns generated by a conventional path delay fault ATPG.
The overall algorithm of the proposed test generation for XPDF faults is shown in Fig. 4.8. The proposed test generation algorithm consists of five steps: (1) logic simulation with the generated delay test pattern of a victim path, (2) sorting the aggressor candidates as decreasing order of the coupling capacitance value, (3) applying a constraint for each aggressor candidate, (4) backward implication from the aggressor’s constraints and (5) forward implication to verify the generated test pattern.

IV. Experimental Results

The proposed ATPG was implemented on C and was run on ISCAS 85 and ISCAS 89 benchmark circuits. All results are obtained for 0.13um technology.

<table>
<thead>
<tr>
<th>circuits</th>
<th># of faults</th>
<th># of detected faults</th>
<th>fault coverage (%)</th>
<th># of detected faults</th>
<th>fault coverage (%)</th>
<th># of detected faults</th>
<th>fault coverage (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>c880</td>
<td>94</td>
<td>6</td>
<td>6.38</td>
<td>41</td>
<td>43.62</td>
<td>51</td>
<td>54.26</td>
</tr>
<tr>
<td>c1908</td>
<td>608</td>
<td>28</td>
<td>4.61</td>
<td>114</td>
<td>18.75</td>
<td>148</td>
<td>24.24</td>
</tr>
<tr>
<td>c2670</td>
<td>340</td>
<td>93</td>
<td>26.88</td>
<td>32</td>
<td>9.25</td>
<td>144</td>
<td>41.62</td>
</tr>
<tr>
<td>c3540</td>
<td>624</td>
<td>15</td>
<td>2.89</td>
<td>29</td>
<td>4.65</td>
<td>49</td>
<td>7.87</td>
</tr>
<tr>
<td>c5315</td>
<td>936</td>
<td>102</td>
<td>10.90</td>
<td>579</td>
<td>40.49</td>
<td>609</td>
<td>65.06</td>
</tr>
<tr>
<td>c7552</td>
<td>1000</td>
<td>0</td>
<td>0.00</td>
<td>302</td>
<td>30.20</td>
<td>520</td>
<td>52.00</td>
</tr>
<tr>
<td>s1423</td>
<td>398</td>
<td>24</td>
<td>6.03</td>
<td>145</td>
<td>36.43</td>
<td>195</td>
<td>48.99</td>
</tr>
<tr>
<td>s5378</td>
<td>916</td>
<td>564</td>
<td>61.57</td>
<td>120</td>
<td>13.10</td>
<td>666</td>
<td>75.98</td>
</tr>
<tr>
<td>s9234</td>
<td>1000</td>
<td>18</td>
<td>1.80</td>
<td>61</td>
<td>6.10</td>
<td>85</td>
<td>8.50</td>
</tr>
<tr>
<td>s13207</td>
<td>1000</td>
<td>264</td>
<td>26.40</td>
<td>91</td>
<td>9.10</td>
<td>380</td>
<td>38.00</td>
</tr>
<tr>
<td>s15850</td>
<td>1000</td>
<td>83</td>
<td>8.30</td>
<td>89</td>
<td>8.93</td>
<td>187</td>
<td>18.78</td>
</tr>
<tr>
<td>s35932</td>
<td>1000</td>
<td>0</td>
<td>0.00</td>
<td>957</td>
<td>95.70</td>
<td>1000</td>
<td>100.00</td>
</tr>
<tr>
<td>s38584</td>
<td>1000</td>
<td>140</td>
<td>14.00</td>
<td>198</td>
<td>19.80</td>
<td>352</td>
<td>35.20</td>
</tr>
</tbody>
</table>

In our experiment, to compare the efficiency of the proposed ATPG with previous works, fanout weighted delay model is used to obtain timing information for experimental evaluation. In addition, due to lack of layout information, coupling capacitances are not considered in fault selection. In this experiment, assuming that single aggressor line is coupled to the target victim path, single aggressor line is randomly selected for the proposed ATPG and the previous works in [7] and [8]. The maximum number of target paths for each benchmark circuit is 1000 paths, while some small benchmark circuits have smaller number of paths than 1000. Table 4.1 shows the comparison with the proposed ATPG and the previous works for single aggressor lines coupled to the victim paths. Note that the maximum number of backtrack is set to 10000 during test generation.

V. Conclusion

In this paper, we propose a new test generation method for delay faults considering crosstalk-induced delay effects, based on a conventional delay ATPG technique in order to reduce the complexity of previous ATPG algorithm for crosstalk delay faults and to consider multiple aggressor crosstalk faults to maximize the noise of the victim line. Since the proposed ATPG for crosstalk-induced delay faults uses the test patterns obtained from the conventional path delay ATPG, the proposed ATPG can reduce the search space of the backward implication of the aggressor’s constraints and it is helpful for reducing the time cost of the ATPG than previous works. In addition, since the proposed technique targets on the critical path for the original delay test as the victim lines, it can improve test effectiveness of delay testing.

Reference