DELAY FAULT SIMULATION OF SELF-CHECKING ERROR CHECKERS

Kanji Hirabayashi
ULSI Research Center
Toshiba Corporation
Komukai, Kawasaki, 210, Japan

Abstract

A robustly-tested gate-delay fault model is proposed using 7-valued logic, and applied to the delay fault simulation of self-checking error checkers. The simulated results are compared with those obtained using either a nonrobustly-tested gate-delay fault model or a path-delay fault model. Experiments show that the robustly-tested gate-delay fault model gives the most pessimistic evaluation for delay test effectiveness. The CMOS pass transistor logic implementation of the self-checking error checkers is discussed.

Introduction

As the integration scale of a silicon chip becomes still larger, concurrent error detection circuits which can detect errors concurrently with normal operation will be important. Self-checking error checkers are promising candidates for them. They are usually designed considering a single line stuck fault model [1]. The same also applies for the self-checking CMOS circuits using pass transistor logic[2]. Since delay fault testing is also important, the check circuits should be self-testable for delay faults. Two delay fault models have been proposed earlier: the gate-delay fault model[3] and the path-delay fault model[4],[5]. In addition to robust tests for path-delay faults[4],[5], robust tests for gate-delay faults have been recently discussed[6]. In this paper we propose a robustly-tested gate-delay fault model using 7-valued logic. We simulate the delay faults of two rail and parity checkers for both the robustly/nonrobustly tested gate-delay fault model and the path-delay fault model.

Nonrobustly-tested Gate-delay Fault Model

The slow-to-rise/slow-to-fall fault is a nonrobustly-tested gate-delay fault model[3],[6]. The slow-to-rise fault of a circuit line behaves like a stuck-at-0 fault after the first rise of the line. Similarly the slow-to-fall fault behaves like a stuck-at-1 fault after the first fall. All the stuck faults of the self-checking two-rail checker shown in Fig. 1 are detected by the patterns of Table 1(a). Here the internal structure of the AND and OR gates is assumed to be error free. Several slow-to-rise/slow-to-fall faults cannot be detected by the patterns of Table 1(a). Undetected slow-to-rise/slow-to-fall faults are shown by † (†) in Fig. 1.
Fig. 1: Self-checking two-rail checker

Table 1: Test patterns for two-rail checker

<table>
<thead>
<tr>
<th>l(a)</th>
<th>l(b)</th>
<th>l(c)</th>
<th>l(d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1 a2 b1 b2</td>
<td>a1 a2 b1 b2</td>
<td>a1 a2 b1 b2</td>
<td>a1 a2 b1 b2</td>
</tr>
<tr>
<td>1 0 1 0</td>
<td>1 0 1 0</td>
<td>0 1 0 1</td>
<td>0 1 1 0</td>
</tr>
<tr>
<td>0 1 1 1</td>
<td>0 0 1 1</td>
<td>0 1 0 0</td>
<td>0 1 1 1</td>
</tr>
<tr>
<td>0 1 1 0</td>
<td>1 0 1 1</td>
<td>1 0 1 0</td>
<td>1 0 1 0</td>
</tr>
<tr>
<td>1 0 0 1</td>
<td>1 0 1 0</td>
<td>1 0 0 1</td>
<td>1 0 0 1</td>
</tr>
</tbody>
</table>

Fig. 2: OR, AND and XOR gates of CMOS pass transistor logic

The CMOS pass transistor logic implementation is shown in Fig. 2[2]. In the single-line slow-to-rise/slow-to-fall fault model, if one of paired PMOS and NMOS gate lines might be slow-to-rise or slow-to-fall, such a fault is undetectable. In the standard CMOS technology, however, the paired lines are made by a common silicon gate. Therefore, it is not unreasonable to assume that slow-to-rise/slow-to-fall fault at the PMOS gate occurs simultaneously with that of the NMOS gate and the internal structure of the AND, OR and XOR gates is error free.

In order to find patterns to detect all the slow-to-rise/slow-to-fall faults of the two rail checker of Fig. 1 random sequence of two rail codes may be generated. The number of minimum patterns to detect them is found to be five. An example is shown in Table 1(b).
Test 1

Path-delay Fault Model

A 5-valued logic system may be used for the robustly-tested path-delay fault[5]. The five values, SO, UO, S1, U1 and XX stand for 0→0, 1→0(X→0), 1→1, 0→1(X→1) and X→X transitions, respectively[5]. A robustly-tested path should not include any nondominant gate input along the UO/U1 chain.

The self-checking two-rail checker was simulated using the path-delay fault model. The paths, f to s2 and g to a1, of Fig. 1 cannot be robustly tested by the patterns of Table I(b), which detect all the slow-to-rise/slow-to-fall faults. All the path-delay faults are detected by the six patterns of Table I(c).

Robustly-tested Gate-delay Fault Model

We use 7-valued logic for the robustly-tested gate-delay fault. Value propagation tables are shown in Table 2, where SO, UO, S1, U1 and XX are the same as in the path-delay fault model. D0 and D1 stand for fall and rise delay faults, respectively. In the fault injection, UO/U1 of a good machine is replaced by D0/D1 at a faulty gate. The gate-delay fault are robustly detected if D0/D1 is propagated to an external output.

Table 2: Value propagation tables

<table>
<thead>
<tr>
<th>NOT</th>
<th>AND</th>
<th>XOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SO</td>
<td>U0</td>
<td>S1</td>
</tr>
<tr>
<td>U0</td>
<td>U0</td>
<td>U1</td>
</tr>
<tr>
<td>S1</td>
<td>SO</td>
<td>U0</td>
</tr>
<tr>
<td>U1</td>
<td>U0</td>
<td>U1</td>
</tr>
<tr>
<td>XX</td>
<td>S0</td>
<td>U0</td>
</tr>
<tr>
<td>D0</td>
<td>D0</td>
<td>D1</td>
</tr>
<tr>
<td>D1</td>
<td>D1</td>
<td>D0</td>
</tr>
</tbody>
</table>

The self-checking two-rail checker was simulated using the robustly-tested gate-delay fault model. Twenty of 36 rise/fall delay faults cannot be detected by the 5 patterns of Table I(b), and 23 rise/fall delay faults by the 6 patterns of Table I(c). All the rise/fall delay faults are detected by the nine patterns of Table I(d). The nine patterns, on the other hand, can detect all the slow-to-rise/slow-to-fall and path-delay faults. Therefore, the robustly-tested gate-delay fault model may be said to give the most pessimistic evaluation for delay test effectiveness.
For other examples, 4 and 5 bit self-checking odd parity checkers were simulated. They are implemented as

\[
f = a_1 \oplus a_2, \quad g = a_3 \oplus a_4 \quad \text{for 4 bit, and} \\
g = (a_1 \oplus a_2) \oplus a_3, \quad g = a_4 \oplus a_5 \quad \text{for 5 bit.}
\]

All the slow-to-rise/slow-to-fall faults of the 4 bit odd parity checker can be detected by the four patterns of Table 3(a) if the internal structure of the XOR gate is assumed to be error free. The rise-delay fault at \( a_3 \) and fall-delay faults at \( a_1, a_2, \) and \( a_4 \), however, cannot be detected in the robustly-tested gate-delay fault simulation. All the rise/fall delay faults are detected by the five patterns of Table 3(b).

All the slow-to-rise/slow-to-fall faults of the 5 bit odd parity checker can be detected by four patterns, while the detection of all the rise/fall delay faults needs 13 patterns.

### Table 3: Test patterns for 4 bit odd parity checker

<table>
<thead>
<tr>
<th>3(a)</th>
<th>3(b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_1 )</td>
<td>( a_2 )</td>
</tr>
<tr>
<td>0 1 1 1</td>
<td>1 0 1 1</td>
</tr>
<tr>
<td>1 1 0 1</td>
<td>1 1 0 1</td>
</tr>
<tr>
<td>0 0 1 0</td>
<td>1 0 1 0</td>
</tr>
<tr>
<td>0 1 1 1</td>
<td>1 0 1 0</td>
</tr>
</tbody>
</table>

**Conclusion**

Self-checking error checkers have been examined using three delay fault models: the nonrobustly-tested gate-delay fault model, the path-delay fault model and the robustly-tested delay fault model. Experiments show that the robustly-tested delay fault model gives the most pessimistic evaluation for delay test effectiveness.

**References**