A Novel Approach to Delay-Fault Diagnosis

P. Girard, C. Landrault, S. Pravossoudovitch

Laboratoire d'Informatique de Robotique et de Microelectronique de Montpellier, UA CNRS D01480, Université de Montpellier II : Sciences et Techniques du Languedoc
Place Eugène Bataillon, 34095 Montpellier Cedex 05, FRANCE

Abstract

Different approaches to delay fault diagnosis are discussed in this paper. Then a new method, based on critical path tracing from a six-valued simulation, is presented. This method is an alternative to fault simulation based approaches and provides perfectly reliable results. It does not require timing evaluations and can be very accurate.

1: Introduction

In order to ascertain the correct operation of digital logic circuits, the functional behavior, as well as the timing performance, must be verified. Physical defects that may affect the timing behavior without changing the steady-state logical operation of a system are modeled by delay faults [8], [15].

The diagnosis of failures affecting the functional behavior of a circuit is generally performed by using a fault dictionary [2], [17]. These failures are modeled by stuck–at faults. As for delay faults, diagnosis is obviously different. This difference arises from the fault model associated with timing failures. A delay fault diagnosis method, that is based on an effect–cause analysis, has been developed by Cox and Rajski [4]. However, this method is unrealistic due to the limitations associated with the transition fault model used by the authors (limitations induced by gross delay faults which are not the most predominant in a circuit [16]). Unlike the stuck–at fault model, a delay fault model has to consider the size of delay defects and, therefore, is often harder to define [3], [9], [14].

In this paper, we shall briefly discuss possibilities of delay fault diagnosis based on fault simulation. Then we will detail the proposed approach based on critical path tracing. Firstly, we will present a path tracing process with information provided by a logic simulation. Subsequently, due to the limitations induced by such a simulation, we will present a reliable approach based on a six–valued logic simulation. It requires no delay size based fault models and considers only the fault–free circuit. Consequently, this approach is faster, reliable and requires less memory than conventional fault simulation based approaches.

2: Main concepts on delay–fault detection

In this study, we use the single fault assumption which is the most widely used, particularly in the context of test generation [11], [13]. Furthermore, the fault type considered is that where the falling and rising transition delay on a faulty element is longer than the specified value. A delay fault is, therefore, assumed to be local to a gate (gate delay fault). However, the method presented can be extended, in a straightforward manner, to the case of distributed failures (defects for which the propagation delays along circuit paths exceed the specified values) [15]. In the same way, physical failures affecting only one type of transition (slow-to-rise or slow-to-fall delay faults) [8], are taken into account by considering both falling and rising faulty transitions (which appear, for example, due to a narrow interconnection line). In section 7, we will discuss the degree of diagnosis accuracy according to delay fault models.

Whatever the fault model and the size of a lumped delay fault, two conditions are necessary for the fault to degrade the timing performance of a circuit, and then to be detected during the test.

The sensitization condition implies that a signal change (transition) must occur on the fault location. This condition will be specified for failures affecting only falling or rising transitions (section 7). On the other hand, we emphasize that it is not necessary for the signal change on the fault site to be a single transition. For example, a delay fault on a line with a transient signal can be detected during the test.

The observation condition of a delay fault on a primary output is that at least one fault effect propagation path from the fault site to the primary output must exist.

These two conditions which are not sufficient to detect a delay fault (in this case, the size of the fault and the observation time have also to be considered), will be the only ones to be considered during the diagnosis methodology proposed in this paper. In fact, we assume that the only information available for the diagnosis process is the failed test patterns (input values and fault-free and faulty output values). Naturally, the diagnosis accuracy strongly depends on the test sequence itself and on the observation time. This problem which relies on test pattern generation aspects will not be addressed in this paper.

3: Delay–fault diagnosis based on fault simulation

As for stuck–at faults, delay fault diagnosis could be performed by using a delay fault simulation. However, a fault dictionary is harder and longer to establish with a delay size based fault model. Moreover, such an approach requires a very precise timing simulation [5][12]. So, when we are only interested in locating a delay defect, delay fault diagnosis based on delay fault simulation may not be the best solution.

Nevertheless, we can imagine another approach to delay fault diagnosis based on stuck–at fault simulation [10]. This is possible due to the relationship existing between the detectability conditions of a delay fault (section 2) and those of a stuck–at fault.

Firstly, if we remind that a necessary condition for the observation of a stuck–at fault on a primary output is that at least one sensitized path from the fault site to the primary output must exist, we can define the following properties:

Property 1: If the observability condition of a stuck–at fault on a circuit line is verified, the observability condition of a delay fault on the same site is also verified.

Property 2: If an error, possibly resulting from a stuck–at fault on a circuit line, is observed on a circuit output, then it may also result from a delay fault on the same site, provided that a signal change exists on this site.

If the opposite property were true, i.e. a delay fault observed on an output implies that a stuck–at fault on the same site is observed, the delay fault diagnosis problem would be solved. Indeed, it
would be sufficient to perform the diagnosis from a stuck-at fault simulation, by considering only the lines with a signal change.

Unfortunately, this opposite property is not valid, particularly in the context of reconvergent fanout circuits. Indeed, some faults in this case can verify the observability condition of a delay fault without affecting the observability condition of a stuck-at fault on the same line. Thus, a delay fault diagnosis based on such an approach can lead to incomplete or erroneous results.

4: Delay-fault diagnosis based on path tracing from a logic simulation

In our attempt to develop an alternative method to fault simulation, we take advantage of the critical path tracing algorithm that was originally developed for stuck-at faults [1].

First of all, we consider a simple logic simulation that provides the list of transitions on each line in the circuit for the applied test sequence. All this information is used by the diagnostic process, in addition to the logical description of the circuit and the failing patterns and primary outputs (information provided by the tester).

Thus, it seems possible to extract some information about fault locations verifying the necessary detectability conditions of a delay fault. This process consists of the following steps:

step 1: After the test, store the generated list of failing patterns and primary outputs.

step 2: Perform a good-machine two-value simulation for one failing pattern and store references about transitions on each line of the combinational circuit.

step 3: From the failing primary outputs associated with the current test pattern, perform a critical path tracing (depth-first-search algorithm) towards primary inputs by marking as critical every sensitized line (line with one or more transitions). In this way, the set of paths having propagated a signal change is determined.

step 4: After this phase, perform a set intersection between the previous failing lines set and the current set constructed with critical lines.

Steps 2, 3 and 4 are repeated for every failing pattern of the test sequence.

The critical lines traced from every failing pattern are possible causes of failure because of the following property:

Property 2: Every delay fault affecting one element belonging to these critical paths verifies the necessary detectability conditions of a delay fault.

Unfortunately, critical path tracing for delay faults based on a two-value logic simulation shows certain deficiencies. These limitations are due to the fact that in real situations output errors may be produced by delay faults affecting internal lines submitted to static hazards which are not handled by a two-valued logic simulation [6]. Thus, the diagnosis process can sometimes provide erroneous results.

However, we can imagine the same approach to diagnosis where the logic simulation is replaced by a six-valued simulation.

5: Delay-fault diagnosis based on path tracing from a six-valued simulation

When we define a set of elements which can be the cause of failure, and if we want to make sure that the fault having induced the error is necessarily on a line belonging to this set, we must have more detailed information about the observability of lines. As we have seen, it is not sufficient to consider transition propagation from the fault-free circuit. All transition propagation possibilities have to be considered, especially when they only exist in the presence of a delay fault.

In order to make transition propagation possibilities appear in fault-free and faulty circuits, we have replaced the classical logic simulation by a six-valued logic simulation based on the H6 algebra [7] with the following symbols: C0 for static 0 signal, C1 for static 1 signal, R1 for 0-to-1 transition, F0 for 1-to-0 transition, P0 for static 0-hazard and P1 for static 1-hazard.

During simulation, the values associated with primary inputs are C0, C1, F0 and R1 for any pattern (for the first vector of the sequence, the values are C0 or C1), provided that the input latches are glichless.

The first advantage of six-valued simulation is that it takes into account the possibility for a given line to be stable in the fault-free circuit and unstable in the presence of a delay fault. Another advantage is that it is independent of any gate propagation delay or delay fault size. Thus, such a simulation does not require the consideration of timing specifications.

By applying the critical path tracing algorithm (as described in section 4) with 6-valued simulation results, we can assert that every line not in the potential failing lines set is necessarily fault-free. In this process, starting from a failing primary output and trying to extend the critical state of the primary output as far as possible towards the primary inputs, all lines with the values F0, R1, P0, P1 are considered to be on a fault effect propagation path. The lines with the values C0 or C1 stop the depth search process.

Consequently, the proposed delay fault diagnosis method consists of the following steps:

step 1: Preprocessing the combinational circuit model to determine a levelled order.

step 2: 6-valued simulation of the first sequence's pattern with the symbols C0 or C1 on every primary input.

step 3: 6-valued simulation of one failing test pattern and determination of the signal values on each line.

step 4: Critical path tracing for delay faults. This is a backtracking procedure that identifies the critical lines (and hence the potential failing lines) for the test simulated in step 3. For every gate evaluation, the critical path tracing is carried out from inputs having the value F0, R1, P0 or P1. Conversely, the trace is not extended through lines having the value C0 or C1.

step 5: Execute a set intersection between the potential failing lines set and the set constructed with critical lines from the current test (we use the single fault assumption).

Steps 3, 4 and 5 are repeated for every failing test pattern of the test sequence.

Although we can improve diagnosis accuracy (section 6), this 6-valued simulation based approach is reliable. Indeed, the fault location associated with the cause of failure is always contained in the potential failing lines set.

Examples: Let us consider the test pair (0111,1010) which, when applied to the circuit of figure 1 gives two failing outputs (O2 and O3).

Figure 1: Diagnostic by critical path tracing from a six-valued simulation

The critical path tracing performed from O2 provides the set (11,12,11,1,1,1,1,1,1,1,1,0,2), and the one obtained from O3 is (12,14,1,1,1,1,1,1,1,1,0,3) (dotted lines). As a result, the set of potential failing lines generated after the test and resulting from the set intersection is (12,1,1,1,1,1) (bold and heavy lines). The line
associate with the cause of failure (L1) is actually contained in the resulting set of potential lines.

6: Improvement of diagnosis accuracy

In order to improve diagnosis accuracy without altering reliability, we have implemented a heuristic whose objective is to reduce the number of critical lines traced during the backward procedure. These improvements are obtained by considering fault propagation possibilities through every gate according to transition polarity on the gate inputs. Let us consider an OR gate with transitions on its inputs (Figure 2).

![Figure 2: Illustration of delay fault propagation](image)

In the first case (2.a), a slow-to-fall delay fault on one of the two inputs is propagated to the gate output. Due to their fault effect propagation ability, these inputs are called "sensitive lines" (they are marked by dots). In the second case (2.b), line i2 is a sensitive input because a delay on this line can be propagated throughout the gate. Conversely, a delay on i1 cannot be observed at the gate output (after the normal stabilization time of the circuit Tmax) due to the dominant final value of input i2 (2.c). During the diagnosis process, the search will be carried out from the sensitive input i2 only, reducing the number of lines to be considered and improving diagnosis accuracy. In the case of gates with no sensitive inputs (2.d), called "non-sensitive gates", a delay fault can only be propagated throughout the gate if the two inputs are affected simultaneously by the fault. Due to the single fault assumption, this is only possible if the failure is located on a line belonging to a common part of the paths descending from these inputs. So, in the diagnosis process, the critical path tracing is stopped at this gate, and then, we have to look for a reconvergence of the paths issuing from these inputs (only inputs with signal FO or P0 (AND, NAND, and R1 or P1 OR, NOR)). If at least one such reconvergence exists (associated fanout stem), the search for potential fault locations in the circuit will go on from the fanout stems found. By not marking as critical the lines belonging to the fanout region comprised between the fanout stem and this non-sensitive gate, the diagnosis is improved again.

If we generalize the considerations established for an OR gate, we obtain the following rules which allow the evaluation of gate input sensitivities:

**Rule 1:** If only one input has the dominant final logic value and if the signal on this line is FO, R1, P0, P1 (unstable signal), then the line is sensitive.

**Rule 2:** If all inputs have the non-dominant final logic value, then each input with a signal FO, R1, P0 or P1 is sensitive.

**Rule 3:** Otherwise, no input is sensitive.

We emphasize that in the case of XOR or NXOR gates, there is no dominant logic value. So, all inputs with the value FO, R1, P0 or P1 are sensitive.

From the computed examples, it appears that the fault localization, which closely depends on the applied test sequence and on the number of failing patterns, may be very accurate in many cases.

The search for one or more associated fanout stems (if any), using these concepts of sensitive lines, is performed during the critical path tracing (step 4 of the overall algorithm), detailed as follows.

**Detailed description of the critical path tracing procedure**

- the failing output is a potential failing line
- determination of the sensitive inputs of the above gate:
  - if sensitive inputs exist, they are added to the set of potential failing lines, and the research process is carried on from these sensitive lines.
  - if there is no sensitive input, the inputs with the value FO, R1, P0 or P1 are labelled if, according to the gate, they have a dominant final value (FO or P0 for AND and NAND gates, R1 or P1 for OR and NOR gates). The fanout stem(s) associated to the gate is (are) determined by a backward propagation of labels through sensitive paths (width first search algorithm), and by the union of labels propagated on every fanout stem:
    - if we can find lines assigned to all the initial labels, these lines are fanout stems associated with the non-sensitive gate. Furthermore, they are added to the potential failing lines set, and the research process is carried on from these lines.
    - if the initial labelled lines do not reach a fanout stem, any descendent line of the non-sensitive gate is potentially faulty. Therefore, the critical path tracing stopped at the non-sensitive gate will never start again from one of these descendent lines.

7: Improvements according to delay-fault models

As mentioned in section 2, the delay fault diagnosis method presented in this paper does not depend on the delayed transition polarity. In other words, a delay fault affecting a falling or a rising transition may be located as well as a delay fault having an effect on both falling and rising transitions. Conversely, we may often distinguish between slow-to-fall (S1F) and slow-to-rise (S1R) delay faults with respect to delay fault test pattern generation [4], [9]. In this case, we can specify the sensitization condition of a delay fault (section 2) in the following manner: A signal change must occur on the fault location, with the final logic value 0 (1) for an S1F (S1R) delay fault sensitization.

Due to the symbolic values associated with a 0 (1) final logic value signal, it is only possible to sensitize a S1F (S1R) delay fault with symbols FO or P0 (P1 or R1). Thereby, assuming that the delay faults affect only one type of transition, we can improve the diagnosis based on critical path tracing from a symbolic simulation. In this case, the lines that are supposed to have both S1F and S1R delay faults will be declared fault-free after the set intersection process (due to the single fault assumption). Thus, this reduction process allows us to minimize the final set of potential failing lines. Taking these considerations into account, the intersection process must be modified in order to remove the lines sensitized for both S1F and S1R delay faults.

Consequently, diagnosis based on critical path tracing from a symbolic simulation can produce different results according to previous considerations. Indeed, diagnosis accuracy is improved by considering only S1F or S1R delay faults. However, delay faults affecting both falling and rising transitions (Slow-to-Rise&Fall delay faults) are not taken into account in this case. So, if we are interested in considering all types of delay faults (S1F, S1F and S1R), we cannot remove any S1R&F delay fault during the set intersection process. Here, we can only provide some information about the type of faults associated with lines belonging to the potential failing lines set. This information is given by the following symbols:
Moreover, sequence 8:

**8: Practical Results**

Delay fault diagnosis accuracy is strongly dependent on the test sequence applied to the circuit and on the delay defect size. Moreover, diagnosis accuracy depends on the fault discernability (equivalent delay faults). Nevertheless, we present significant results which show the efficiency of the reduction process based on sensitive lines, compared to a symbolic simulation only process.

The following tables summarize the results obtained with test sequences generated for stuck-at faults by "System-Hilo: Hitsat" (GENRAD). Faulty circuits have been simulated using "System-Hilo: Hism" with random delay faults (location and duration). We have assumed that there is a propagation delay of one unit for every gate in the fault-free circuit, and that the propagation delay is the same as the unit delay of the device. The CPU time is given in seconds. This logic simulation produces information normally provided by the tester in the presence of real faulty circuits. These results are given for two ISCAS'85 benchmarks in the number of potential failing lines produced by the following diagnostic processes:

- **D1:** Diagnosis based on critical path tracing from a symbolic simulation without considering sensitive lines.
- **D2:** Diagnosis based on critical path tracing from a symbolic simulation with the reduction process based on sensitive lines.

<table>
<thead>
<tr>
<th>C432</th>
<th>Patterns &quot;fail&quot;</th>
<th>Outputs &quot;fail&quot;</th>
<th>D1</th>
<th>D2</th>
<th>CPU (D1)</th>
<th>CPU (D2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gate: 46 / 1u</td>
<td>2</td>
<td>4</td>
<td>9</td>
<td>2</td>
<td>0.21</td>
<td>0.21</td>
</tr>
<tr>
<td>Gate: 29 / 1u</td>
<td>2</td>
<td>4</td>
<td>13</td>
<td>5</td>
<td>0.21</td>
<td>0.27</td>
</tr>
<tr>
<td>Gate: 144 / 1u</td>
<td>2</td>
<td>2</td>
<td>17</td>
<td>7</td>
<td>0.20</td>
<td>0.19</td>
</tr>
<tr>
<td>Gate: 48 / 2u</td>
<td>6</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>0.32</td>
<td>0.25</td>
</tr>
<tr>
<td>Gate: 27 / 2u</td>
<td>2</td>
<td>3</td>
<td>25</td>
<td>5</td>
<td>0.18</td>
<td>0.24</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C7552</th>
<th>Patterns &quot;fail&quot;</th>
<th>Outputs &quot;fail&quot;</th>
<th>D1</th>
<th>D2</th>
<th>CPU (D1)</th>
<th>CPU (D2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gate: 67 / 1u</td>
<td>3</td>
<td>3</td>
<td>90</td>
<td>15</td>
<td>14.26</td>
<td>12.10</td>
</tr>
<tr>
<td>Gate: 67 / 2u</td>
<td>8</td>
<td>10</td>
<td>64</td>
<td>6</td>
<td>13.58</td>
<td>13.74</td>
</tr>
<tr>
<td>Gate: 141 / 3u</td>
<td>21</td>
<td>23</td>
<td>59</td>
<td>4</td>
<td>26.24</td>
<td>22.61</td>
</tr>
<tr>
<td>Gate: 3498 / 3u</td>
<td>8</td>
<td>8</td>
<td>143</td>
<td>42</td>
<td>15.99</td>
<td>21.03</td>
</tr>
<tr>
<td>Gate: 3495 / 4u</td>
<td>11</td>
<td>11</td>
<td>302</td>
<td>39</td>
<td>22.02</td>
<td>18.80</td>
</tr>
</tbody>
</table>

9: Conclusion

Delay fault diagnosis is quite different from the stuck-at fault diagnosis. At the beginning of this paper, we showed the deficiencies of delay fault diagnosis based on fault simulation. Then we presented an alternative to the fault simulation based approach. The proposed method for delay fault diagnosis is based on critical path tracing from a symbolic simulation, due to the limitations caused by a logic simulation. The principle of such a simulation and the critical path tracing process have been presented. This diagnosis method is perfectly reliable, and does not need timing evaluations. To improve diagnosis accuracy, we have also developed the concept of sensitive lines. Based on the method presented, a fast algorithm has been implemented in C++ on a SUN–space workstation.

**References**


