

The Optimistic Update Theorem for Path Delay Testing in Sequential Circuits

SOUMITRA BOSE, PRATHIMA AGRAWAL, AND VISHWANI D. AGRAWAL

AT&T Bell Laboratories, Murray Hill, NJ 07974

Received February 5, 1993. Revised April 5, 1993.

Editor: K. Saluja

Abstract. For sequential circuit path delay testing, we propose a new update rule for state variables whereby flip-flops are updated with their correct values provided they are destinations of at least one robustly activated path delay fault. Existing algorithms in the literature, for robust fault simulation and test generation, assign unknown values to off-path latches that have non-steady signals at their inputs in the previous vector. Such procedures are pessimistic and predict low fault coverages. They also have an adverse effect on the execution time of fault simulation especially if the circuit has a large number of active paths. The proposed update rule avoids these problems and yet guarantees robustness.

Keywords: Fault simulation, path delay faults, test generation, timing analysis.

1. Introduction

A robust delay test for a path in a combinational circuit consists of propagating a transition along the path and ensuring that the off-path sensitizing signals do not cause the path destination to change state before the transition has traversed the entire path. The validity of such a test is guaranteed irrespective of other delays in the circuit, and this condition is referred to as *combinational robustness*. In sequential circuits, several latches may be path destinations for many faults activated simultaneously by the same vector. Consider the fault effect propagation in the subsequent vector of one particular fault, referred to as the *primary* fault. For subsequent robust detection of the *primary* fault, it is necessary that none of the other simultaneously activated *secondary faults* mask the primary fault effect propagation. This criterion is often referred to as *sequential robustness*.

Consider a *primary* path delay fault that gets activated by some vector of an input sequence. We will refer to that vector as the *activation* vector for the *primary* fault. For the fault effect to propagate from the destination latch in the next vector, some other latch might be required to have a specified value. However, that latch may be the destination of another *secondary* path fault also activated by the *activation vector*. One might desire that the propagation of the effect of the *primary*

fault is independent of the presence of the *secondary* fault. In such situations, existing algorithms assign unknown values to latches having non-steady signals at their inputs [1], [2]. There are adverse consequences if such a pessimistic update rule is used. Other than predicting low fault coverages, a separate analysis of the fault effect propagation may be necessary for every active fault. In fault simulation, if the number of active faults is large, the entire simulation will be slow. In this article, we present an alternative update rule that can alleviate these problems.

2. Background

To represent the signal state in two consecutive vectors, we use a 7-valued algebra consisting of the 6-valued system of Smith [3] and an additional logic value, X , representing unknown. Smith's logic is suitable to reason about *combinational circuits with fully specified inputs*. The additional value, X , is necessary for sequential circuits and combinational circuits with unspecified inputs.

Smith's algebra consists of 6 values: $p0, p1, s0, s1, -0, -1$ and can be thought of as consisting of tuples (pds, fv) , where pds is *path detectability status* and fv is the *final value*. These logic values describe the timing behavior of the circuit during the interval,

denoted by $\langle V_n, V_{n+1} \rangle$, between any two consecutive vectors V_n and V_{n+1} . During any such interval, a signal may undergo hazards (static or dynamic), but will always settle down to a definite final value (for combinational circuits with fully specified inputs) and this value, which may be 0 or 1, represents the second component of the tuple.

For a signal g , $pds(g) = s$ for the interval $\langle V_n, V_{n+1} \rangle$ if the signal remains stable during the entire interval. It is set to p if there exists at least one path along which a transition propagates and the signal g is guaranteed not to change until the transition along that path has reached g . More than one active path terminating at g may exist, in which case, all transitions along these paths must reach g before g can change. The time of transition will then depend upon the slowest of these paths. Note that a pds value of p does not necessarily mean that a complete path from input to output exists in the combinational circuit. However, it does guarantee that at least one partial path from an input to that signal exists. The pds component is set to $-$ if neither s nor p is true. Thus a logic value of -1 represents either a static-1 hazard or a non-robustly propagating rising transition (as at the output of an OR gate with multiple rising transitions at inputs).

The unknown value, X , represents those signals whose final values are not known. For such signals, the pds components are not relevant and, semantically, X means the final value can be either 0 or 1. The evaluation tables for this 7-valued logic, as shown in Figure 1, are a direct extension of the 6-valued logic of Smith [3].

As an example, we consider the sequential circuit shown in Figure 2. It has two latches, $L1$ and $L2$, one input I , six gates $G1, \dots, G6$, and one primary output $G6$. The circuit is simulated for the following 4-vector input sequence $\langle V_1, V_2, V_3, V_4 \rangle = 1000$. The latches are assumed to have a reset state and are initialized to 0. The first vector, $I = 1$, is applied for sufficiently long interval so that all signals stabilize to their steady states. The subsequent three vectors are then applied at the rated clock interval. We will focus our attention on transitions propagating along the following paths:

$F_1: I, G1, G2$

$F_2: L1, G3, G4$

$F_3: L1, G3, G2$

$F_4: L2, G5, G6$

$F_5: L1, G6$

| AND | s0 | p0 | -0 | s1 | p1 | -1 | X |
|-----|----|----|----|----|----|----|----|
| s0 | s0 | s0 | s0 | s0 | s0 | s0 | s0 |
| p0 | s0 | -0 | -0 | p0 | -0 | -0 | -0 |
| -0 | s0 | -0 | -0 | -0 | -0 | -0 | -0 |
| s1 | s0 | p0 | -0 | s1 | p1 | -1 | X |
| p1 | s0 | -0 | -0 | p1 | p1 | p1 | X |
| -1 | s0 | -0 | -0 | -1 | p1 | -1 | X |
| X | s0 | -0 | -0 | X | X | X | X |

| OR | s0 | p0 | -0 | s1 | p1 | -1 | X |
|----|----|----|----|----|----|----|----|
| s0 | s0 | p0 | -0 | s1 | p1 | -1 | X |
| p0 | p0 | p0 | p0 | s1 | -1 | -1 | X |
| -0 | -0 | p0 | -0 | s1 | -1 | -1 | X |
| s1 | s1 | s1 | s1 | s1 | s1 | s1 | s1 |
| p1 | p1 | -1 | -1 | s1 | -1 | -1 | -1 |
| -1 | -1 | -1 | -1 | s1 | -1 | -1 | -1 |
| X | X | X | X | s1 | -1 | -1 | X |

Fig. 1. Evaluation tables for 7-valued logic.

Figures 2(a) and 2(b) show the state of signals during two vector intervals: $\langle V_1, V_2 \rangle$ and $\langle V_2, V_3 \rangle$. The latches are marked with fault identifiers corresponding to those paths along which transitions propagate robustly. These faults correspond to paths along which all signals have their pds component set to p . For instance, Figure 2(a) shows latch $L1$ marked with F_1 to indicate that this latch will clock in a faulty value if path $I, G1$ and $G2$ is slow. Figure 2(b) has $L1$ marked with F_1 and F_3 while latch $L2$ is marked with two faults: F_1 and F_2 . The reason for marking latch $L1$ with two faults is as follows: if the rising transition at the input of $L1$ (Figure 2(a)) fails to arrive on time, the falling transition at the input of $L1$ in the following vector will also fail to arrive. Therefore, the fault F_1 is propagated to latch $L1$ in the following vector. However, if $L1$ does clock in the correct value, but the path $L1, G3$ and $G2$ is slow, then also $L1$ will latch an incorrect value. Thus, both F_1 and F_3 are in the fault list at $L1$.

Consider simulation of the vector pair $\langle V_3, V_4 \rangle$. Propagation of the fault effect of F_3 from latch $L1$

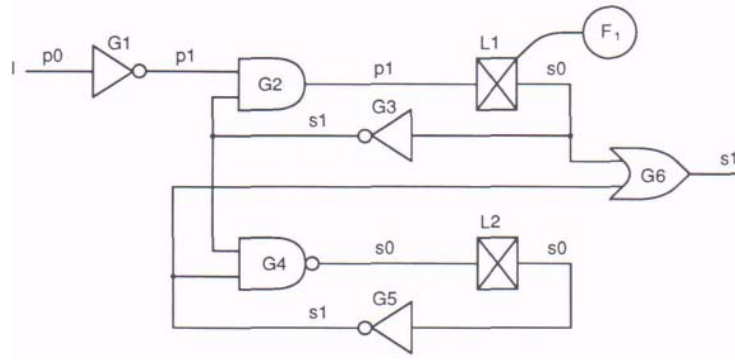
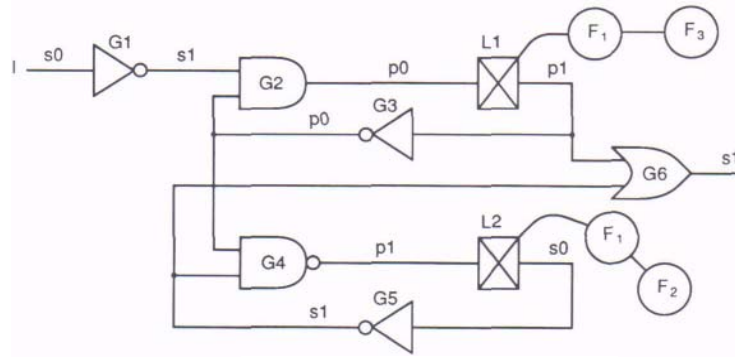
(a) Simulation of vector pair $\langle V_1, V_2 \rangle$ (b) Simulation of vector pair $\langle V_2, V_3 \rangle$

Fig. 2. Signal values in first 3 vectors.

depends on the logic state in L_2 which in turn, depends on faults F_1 and F_2 . Existing algorithms [1], [2] assign an unknown value, X , to latches like L_2 while considering the propagation of fault effects from latches like L_1 . This is shown in Figure 3(a). Assigning an X to L_2 for the fault F_3 drives the primary output to X . Hence F_3 is not detected. Now consider the fault effect of F_2 from latch L_2 . Latch L_1 is assigned a value X , which drives the primary output G_6 to X . This is shown in Figure 3(b). Hence, fault F_2 is also undetected. Now consider fault F_1 , which causes incorrect values in both latches. It is easy to show that only F_1 is declared detectable by the given input sequence. In reality, it can be easily verified that the presence of any combination of these five faults will cause the circuit to fail. This example demonstrates that updating latches with unknown values causes a low (pessimistic) fault coverage prediction.

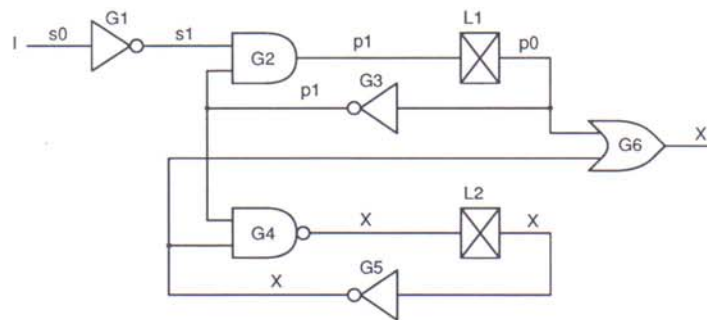
3. Optimistic Update of Latches

Consider an update rule in which all latches with non-steady signals at their inputs are updated with their

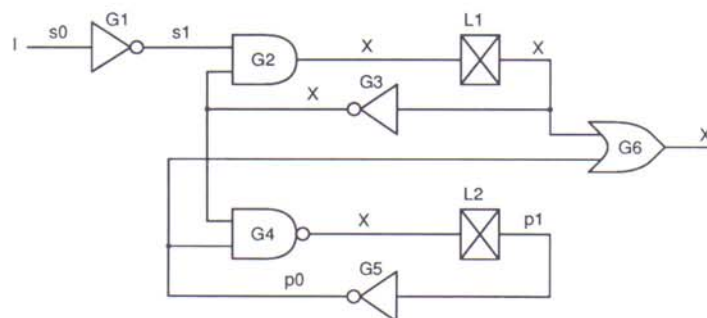
fault-free circuit values, provided they are destinations of at least one robustly activated path. This condition can be verified easily (in constant time) by ensuring that the pds component for the latch input signal was p in the previous vector. Latches with static hazards and dynamic hazards due to non-robust propagation are updated with an unknown value (X). This rule can be stated as follows:

Optimistic Update Rule: For any latch whose input is not steady at either 0 or 1, assume fault free value in the subsequent vector, provided the latch is the path destination of at least one robustly activated fault.

We will apply the *optimistic update rule* to the example of Section 2. Figure 4 shows the signal values for the vector pair $\langle V_3, V_4 \rangle$. Note that using optimistic values for both latches, all five path delay faults are found detectable. The fault effects for F_1 and F_2 at latch L_2 are propagated robustly along the path G_5 and G_6 . In addition, the path consisting of L_2 , G_5 and G_6 (fault F_4) is also sensitized. Fault effect of F_3 is propagated through G_6 to the output. Hence all five faults are detected.



(a) Vector pair $\langle V_3, V_4 \rangle$: propagation of F_3



(b) Vector pair $\langle V_3, V_4 \rangle$: propagation of F_2

Fig. 3. Signal values during propagation in fourth vector.

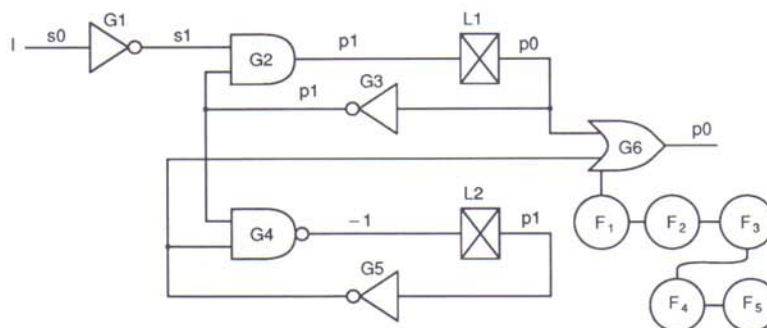


Fig. 4. Signal values with optimistic update.

There is another important consequence of the *optimistic update rule*. In the previous section, propagation of fault effect F_3 at $L1$ required a resimulation with $L2$ initialized to X . Similarly, propagation of fault effect F_2 at $L2$ required another resimulation with $L1$ initialized to X . If there are many latches in the circuit and many faults get sensitized, propagation of each fault effect requires a separate simulation. To deal with large number of paths, schemes for compact and implicit path fault identifiers have been suggested [4], [5], [2] but such algorithms are still computationally inefficient

because the number of simulations required may be of the same order as the number of activated paths in the circuit.

4. Optimistic Update Theorem

We consider the *activation vector* of a path fault during an arbitrary cycle in an input sequence. We assume that all latches in the circuit, including the off-path latches, have already been updated using the *optimistic*

update rule in all prior vectors. The signal value at the path destination at the end of the activation vector may be incorrect due to two possible reasons. Either the activated path is slow or the state of the circuit is incorrect, thus preventing the off-path signals to sensitize the fault. The latter situation occurs only if the optimistic update rule has been applied to a latch in a previous vector, and the state of the latch is indeed incorrect due to a delayed transition. In either case, it is possible to show that at the end of the activation vector, the path destination has a faulty value. Since we assume fault free values (optimistic update) only if a latch is a destination of at least one robustly activated path, the incorrectness of the circuit state (in the presence of faults that were robustly activated) is also independent of other delays in the circuit. Subsequent to activation, the fault may eventually be detected at a primary output. Similar to the concept of fault activation in combinational circuits [3], such a failure in a sequential circuit is not diagnosable. However, the condition for fault detectability can be stated as follows:

Optimistic Update Theorem: Any fault found detectable by using optimistic update of latch variables, if present in the circuit, is guaranteed to cause failure and cannot be masked by other path delay faults in the circuit.

We first prove some auxiliary properties about signal values specified by the optimistic update rule. We will then use these properties to prove correctness of the optimistic update theorem. We assume that all signal values at nodes in the circuit have been obtained from the 7-valued algebra of Figure 1. Since the optimistic update rule has been applied for all latches throughout the entire input sequence, the signal values may be different from the values h_i the actual physical circuit which may have one or more path delay faults. The auxiliary properties deal with the relationship between the signal values in the actual (faulty) circuit and those evaluated by the 7-valued algebra and the optimistic update rule.

Lemma 1: A node in a sequential circuit with a stable value ($s1$ or $s0$), as specified by applying the 7-valued algebra and the optimistic update rule, attains this value (in the actual physical circuit) irrespective of the presence or absence of any delay fault.

Proof. Examination of the truth table reveals that for a gate to have a stable output, either at least one input must have a stable controlling value or all inputs have

steady non-controlling values. Hence, for a stable gate output to have a logic value in the real circuit that is different from the value in the fault-free circuit, it is necessary that at least one stable input is affected by the fault. We use induction arguments first on the levels of the circuit and then on the successive vectors. Since the inputs to the combinational network (primary inputs and latch outputs) remain unchanged at their stable values irrespective of slow paths, it follows by induction on the levels of the circuit that no gate of the network with a stable value is affected by delay faults. Since this is true in the first two time frames, it follows by induction on successive vectors that the lemma is true universally. •

Lemma 2: Any signal in the circuit that has a final value of 0 (1), as specified by applying the 7-valued algebra and the optimistic update rule, may have a final value of 1 (0) in the actual physical circuit only if some robustly activated path fault caused an erroneous latch state in a previous vector.

Proof. Intuitively, this lemma says that if the actual circuit (with one or more faults) is slowed down arbitrarily in vector N , the logic value to which any node stabilizes can be complementary to the final value predicted by the optimistic update rule only if some robustly activated fault caused an incorrect latch state in a previous vector. Note that the comparison here is between the signal in the actual circuit and the value obtained by the 7-valued algebra and the optimistic update rule. This lemma applies only to those signals that do not evaluate to X , i.e., signals that attain a 0 or 1 final value.

We will first prove that the above is true in the time frame immediately after the first fault (or faults) was activated robustly. Then by induction on successive time frames, it will follow that this is true universally. The proof for the first time frame uses induction on the levels of the circuit.

Consider primary and pseudo-primary inputs (latch outputs). Primary inputs cannot have different final values in the actual physical circuit. The only possible logic values for pseudo-primary inputs are $s1$ and $p1$ ($s0$ and $p0$) if the final stabilization value is 1 (0). The lemma does not apply to signals with logic value X and -1 and -0 are never assigned to inputs of the combinational network. From Lemma 4, any signal with value $s1$ or $s0$, attains that value in the actual circuit irrespective of the absence or presence of any delay fault. By definition of the update rule, pseudo-primary inputs with the value $p1$ ($p0$) attain the value $s0$ ($s1$) in

the actual circuit if robustly activated faults cause incorrect latch states. Thus, the lemma is true for pseudo-primary inputs and trivially true for the primary inputs to the combinational network.

Now, we consider gates at level two. If the output of a gate, as evaluated by the algebra and the update rule, has a dominant value, i.e., the value is caused when at least one input has the dominant logic value, the signal may have the complementary non-dominant output value in the actual circuit only if there exists at least one input in the actual circuit with the opposite final value. Since this is not possible for primary and pseudo-primary inputs *unless there is an activated fault causing an incorrect latch state*, the lemma holds for all gates at level two. Such an argument can be easily extended to a gate with a non-dominant output final value. For inverters which are single input gates, the lemma is true trivially. •

Consider path fault f and its activation vector N . We assume that latches have been updated *optimistically* in all prior vectors. For a circuit with fault f , robustness is guaranteed in the following sense: for a circuit with fault f , the path destination is guaranteed to latch an incorrect value irrespective of whether the state of the circuit is different from that evaluated by using the optimistic update rule. The destination flip-flop may latch an incorrect value either due to f or due to some other path being slow in vector N (combinational robustness) or due to the failure of off-path signals in sensitizing the fault. The last condition arises due to delayed robustly propagating transitions in previous vectors (sequential robustness). In either case, failures are not diagnosable. Moreover, the faults found detectable by this update procedure, if present in the circuit, are guaranteed to cause failure independent of other circuit delays.

Proof of the Optimistic Update Theorem. The proof considers all possible primitive gate types and transitions (rising and falling) from the on-path input to the output of a gate lying on the path under consideration. We outline the proof for an AND gate, G , lying on the path corresponding to the fault f . The arguments for an OR gate are similar.

Consider a falling transition at the on-path input of the AND gate G . Since the path fault f was robustly activated, the off-path input, as evaluated by the 7-valued

algebra and the optimistic update rule, has to be $s1$. Considering the possibility of this off-path signal *in the actual circuit* to have a value different from $s1$ in the presence of other delay faults, we find that it is impossible as shown in Lemma 1.

We next consider a rising transition on the on-path input of gate G . The off-path input can have a value $p1$, $s1$ or -1 . If the value is $s1$, then this node is invariant with respect to circuit delays, as shown in Lemma 1. If the value is $p1$ or -1 , from Lemma 2, we can assert that if a robustly activated fault caused a delayed transition at a latch input in the actual circuit, the off-path signal may have a final value 0, i.e., logic value $s0$, $p0$ or -0 . Even when the final value of this off-path input is indeed 0, the destination flip-flop of path f is guaranteed to latch in an incorrect value. •

5. Conclusion

The state variable update rule proposed in this article guarantees robustness in the presence of arbitrary delays in the circuit. It has applications to delay fault test generation and fault simulation of sequential circuits. The optimistic update rule can be used in both fault activation and fault effect propagation phases. A fault simulation algorithm using this update rule is described in a recent article [4]. As shown there, its impact on fault simulation is two fold: prediction of a higher and a more realistic fault coverage and reduction in the effort required for fault effect propagation.

References

1. T.J. Chakraborty, V.D. Agrawal, and M.L. Bushnell, "Path delay fault simulation algorithms for sequential circuits," *Proc. Asian Test Symp.* pp. 52-56, November 1992.
2. I. Pomeranz, L.N. Reddy, and S.M. Reddy, "SPADES: a simulator for path delay faults in sequential circuits," *Proc. European Design Automation Conference*, pp 428-435, September 1992.
3. G.L. Smith, "Model for delay faults based upon paths," *Proc. Inter. Test Conf.* pp 342-349, September 1985.
4. S. Bose, P. Agrawal, and V.D. Agrawal, "A path delay fault simulator for sequential circuits," *Proc. Sixth Inter. Conf. on VLSI Design*, pp 269-274, January 1993.
5. K. Fuchs, F. Fink, and E.S. Schulz "DYNAMITE: an efficient automatic test pattern generation system for path delay faults," *IEEE Trans. on Computer-Aided Design*, vol. 10, pp 1323-1335, October 1991.