Test Generation for Resistive Opens in CMOS[*]

Arun Krishnamachary and Jacob. A. Abraham
Computer Engineering Research Center
The University of Texas at Austin
Austin, Texas 78712

{arun, jaa}@cerc.utexas.edu

ABSTRACT

This paper develops new techniques for detecting both stuck-open faults and resistive open faults, which result in increased delays along some paths. The improved detection of CMOS open defects is made possible by a new delay fault model which combines the advantages of the gate delay fault model and the path delay fault model. We develop a test generation methodology for this fault model which enables generation of test vectors that test a percentage of the longest sensitizable paths in the design and also test each net for spot defects through their longest sensitizable paths. Real delay values are used to determine the true critical paths in the circuit. The high degree of effectiveness of this fault model under realistic assumptions for process characteristics is first enumerated, and experimental results demonstrate the improved coverage possible with the proposed approach.

Categories and Subject Descriptors

B.8.1 [**Hardware**]: Reliability, Testing and Fault-Tolerance

Keywords

delay testing, resistive opens, defect detection

1. INTRODUCTION

The speed and complexity of the current generation microprocessors have increased rapidly over the last few years. This has been accompanied by a drastic increase in the complexity of test generation for these microprocessors. Even test generation for classical fault models like the stuck-at fault model has become much more complex. Further, the reduction in the process lengths, which has contributed to increases in clock speeds, has brought into prominence new types of defects not effectively modeled by the stuck-at fault model [1, 2]. Hence, there is an increased need to evolve fault models and test generation methodologies to facilitate the detection of these defects.

The new types of defects that are becoming more prominent include opens, bridges and crosstalk. Although bridging faults are the most likely to occur among these defects, opens comprise the bulk of the defects that escape the test phase of the manufacturing flow [3]. This high rate of test escapes reveals an inadequacy in the fault models used to generate test vectors for this defect [4]. Further, interconnect breaks occur with high probability in copper interconnects. One of the main reasons for this is the difficulty of depositing copper without voids and cracks in high aspect ratio trenches and vias [5].

CMOS opens manifest themselves by introducing a memory behavior at the location of the fault [6] or by an increase in the delay of the paths through the fault site [7]. The memory behavior is caused by an open defect with a very high resistance, and this is modeled by the stuck-open and the stuck-on fault models. An open defect with a moderate resistance causes an increase in signal propagation delay of the circuit.

Detecting the memory effect of CMOS opens requires initializing the faulty node to a particular value and then using a second vector to drive a transition on the node and propagate it to the primary outputs. The transition fault model [8] works well to detect this behavior. Reordering stuck-at test sets has also been proposed as a means of obtaining test vectors having good stuck open coverage. The increases in delay due to resistive opens requires at-speed or delay tests to maximize detection. However, the delay fault models that are used commonly (gate delay [9, 10] and path delay [11] fault models) have certain drawbacks that render them unsuitable for test generation for spot defects like resistive opens. The gate delay model has the disadvantage that it does not detect the smallest possible additional delay sizes as it does not propagate the faults through the longest paths in the circuits. The path delay fault model has the disadvantage of a large number of paths to target and the fact that considering just the longest paths does not cover all the nets in the circuit.

It has been shown in [12] that the paths that can be targeted by relaxing the robustness constraint are much longer than the paths targeted using the robustness constraint. Many of these non-robust tests have very low likelihood of failing, thus enabling us to have good confidence in the tests that are generated.

It is thus very important to at-least identify all the longest testable paths in the design even if some of the paths can only be tested non-robustly. In this paper, we present an improved delay fault model which tests for delay faults on all nets for both rising and falling transitions through the longest sensitizable path passing through the particular net while also ensuring that a percentage of the longest sensitizable paths are also tested under the given test set. The sensitizability can be determined by using a technique called vigorous sensitization proposed in [13], where the longest paths are iden-

[*]This research effort was supported by the Semiconductor Research Corporation through contract 99-TJ-715

GLSVLSI'02, April 18-19, 2002, New York, New York, USA.

tified without enumerating all the paths. To identify the longest possible testable paths, we use the non-robust criterion. Our fault model is derived from the Unified Delay Fault model proposed in [14]. This paper first demonstrates the effectiveness of this fault model under realistic assumptions for the process characteristics, proposes an effective test generation approach for the fault model and provides experimental results to demonstrate its effectiveness for resistive opens.

2. CIRCUIT REPRESENTATION

DEFINITION 1. *The circuit under consideration can be considered to be a four tuple (V, E, w_g, p_g) as in [15] where*

a) *V = the set of gates.*

b) *E = the set of nets.*

c) *w_g for a given gate g is a function that has the domain V and the range as $R^+ \times R^+$ where R^+ is the set of positive real numbers and $w_g = (d_r, d_f)$ where d_r is the delay for a rising transition and d_f is the delay for a falling transition. We use $w_g(r)(w_g(f))$ to refer to delay for rising (falling) transition respectively.*

d) *p_g for a given gate g is a function that determines the inversion parity of the gate, it assumes a value T if the signal is inverted through the gate and assumes a value F if the signal is not inverted through the gate. This function has V as it's domain and $\{T, F\}$ as it's range.*

We deal with synchronous sequential circuits and the generation of two vector test sets for detection of delay faults. The first test vector is applied and the circuit allowed to settle to a stable state; the second vector (the vector enabling the transition on the path) is then applied at time t, the final value is latched on the output latches at time t+L where L is determined by the period of the clock driving the design.

We can consider a sensitizable sub-path P as a set of gates with a linear ordering specified by the relation $i \twoheadrightarrow j$ for gates i and j which specifies the existence of a sensitizable sub path from i to j. Thus for any successive gates (g_1, g_2) present in the set P, $\exists\langle g_1, g_2\rangle \in E$. In this paper, $\langle g_1, g_2\rangle \in P$ implies that the edge $\langle g_1, g_2\rangle$ lies on the path P.

Given a path P given by $\{G_1, \ldots , G_n\}$ where G_1, G_2..., G_n correspond to the gates on the path, the delay of the path is given by the Equation 1. Here, x_i which takes a value from the set {r, f} corresponds to the transition propagating through the gate G_i.

$$delay(P) = \sum_{i=1\ldots n} w_{G_i}(x_i) \qquad (1)$$

Let C_i correspond to the controlling value of the gate i, and NC_i correspond to the non-controlling value of the gate i. Set $CI_i(NCI_i)$ corresponds to the set of the inputs of the gate i which settle to a controlling (non-controlling) value.

3. CMOS OPEN DEFECTS

CMOS opens are a result of various failure mechanisms during fabrication, resulting in actual breaks in the interconnect wires (due to random particle effects), missing vias, partial vias and high-resistance vias [16]. When these resistances have a high value, they result in stuck-opens which cause a memory behavior of the node; moderate resistances can be modeled as delay defects.

In the stuck-open fault model, the transistors are modeled as being permanently open, causing a memory effect in the circuit. Defective circuits thus exhibit a sequential behavior. The transition fault model can be used effectively to detect this behavior because of the relatively large delays. This does not impose a requirement on the length of the path through which the transition is propagated.

On the other hand, resistive opens which result in a finite resistance between the nodes of the open result in much lower delays which are more difficult to detect. This imposes the constraint of having to propagate the transition through the longest path possible, since the additional delay may not cause any problems with shorter paths. The gate delay fault model and transition fault model fail to achieve this objective required to detect such opens. The path delay fault model has the disadvantage that we are required to generate a large number of paths in order to cover all possible fault locations, especially when the circuit is not optimized for timing.

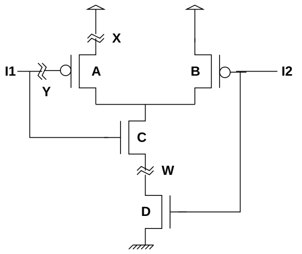

Figure 1: 2-input NAND gate with Breaks

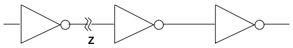

Figure 2: Chain of inverters with Break

Consider the example shown in Figure 1, which shows the transistor model of a two-input NAND gate. The stuck-open fault implies any one of the four transistors could be stuck-open. For example, in the case where transistor A is stuck open, the pull-up through this transistor is affected as the transistor never switches on, and this causes a sequential behavior. However, this is not always the case when one of the terminals of the transistor is floating. In such cases where the opens are resistive opens, the resistance combines with the circuit capacitance to cause an increase in the delay of signal propagation in the circuit. Table 1 shows delay caused in a 2-input NAND gate for defects at locations X and Y as shown in Figure 1; this also shows increases in delays for various resistance values for a 0.35 micron process. A rising transition caused by a falling transition on I1 has a greater delay than in the fault-free case. Similarly, defect W in the pull-down path of the gate can be tested by driving a rising transition on the appropriate transistor. Thus, a methodology that drives both rising and falling transitions through the gate and through all the inputs of the gate enables the activation of all the defects in the gate. In this paper, we show results on circuits with primitive gates AND, OR, NAND, NOR and inverters. These gates are similar structurally to the example given above and can be tested using similar techniques.

Table 1 also shows the increase in delay due to a break (or partial break) in the interconnect between gates as shown in Figure 2. The phenomenon is illustrated with a series of inverters with a defect at location Z. Detecting these defects requires the propagation of both rising and falling transitions through these nets. Further, delay

Table 1: SPICE Results

Fault location	Delay (ns)			
	Fault free	100Ω	500Ω	1KΩ
X	0.17	0.6	2.2	3.2
Y	0.17	0.18	0.19	0.20
W	0.15	0.22	0.88	1.68
Z	0.6	0.77	0.79	0.82

tests are required to enable the detection of these resistive opens. For example, an increase in resistance of 100Ω causes the delay to increase by 170 picoseconds. This increase will only cause a problem if the path has a slack less than this value. If there are multiple paths including this faulty node, propagating a transition along the longest possible path will guarantee the detection of the defect, if the increased delay would affect the correctness of the result. The need to propagate transitions along the longest possible path forms the basis of our fault model.

3.1 Process Parameter Variations

Process parameter variations result in distributed delay faults which cause the circuit to fail with respect to timing. This is a result of minor positive variations on all the gates over an area of the chip. These variations on multiple gates on a given path accumulate to cause the path delay of the path to exceed the circuit timing specifications.

This kind of defect is different from the contamination related defects like the resistive opens. The inter-die parameter variations act on the entire chip or functional block such that each device on one chip or block shows uniform process variation. These parameter variations are caused by systematic effects like process gradients over the wafer. Errors in which the process variations are not uniform over the entire die are called intra-die variations.

4. AN IMPROVED UNIFIED FAULT MODEL

The delay fault model tests each net in the circuit for both rising and falling transition through the longest path through the net and also the set of all paths that are longer than a given threshold. This combines the advantages of the path delay fault model and the gate delay fault model.

4.1 Defect Coverage

Test generation for our fault model has two phases.

- **Phase 1** targets the paths whose delays exceed a given threshold.
- **Phase 2** targets the nets that are not covered in Phase 1 through their respective longest sensitizable paths.

The test vectors that are generated in Phase 1 also enable us to target the effects of process variations that occur across the chip apart from targeting point defects on individual nets. In current device technologies, the intra-die process parameter variation has a very low probability of occurrence. Thus for a given chip there is very high probability that in the case of variation, the process parameters vary uniformly. This is especially relevant as individual blocks considered are usually combinational blocks within latch boundaries in the case of synchronous sequential circuits. Thus we can assume an increase in the delay of all the gates forming the combinational block.

Thus testing the top few paths under these conditions ensures the detection of moderate process parameter variations even if the ordering of the long paths in the design cannot be guaranteed. This is because a path reordering that could cause these tests to fail must elevate a path under the threshold to a high enough value in the presence of process parameter variations and this would be detected by the process monitors.

Further, the combination of vectors that are generated over both the phases drive both rising and falling transitions through all the nets in the circuit along their respective longest paths. This enables us to detect the effect of point defects. The nets covered by the vectors generated in Phase 1 need not be targeted in Phase 2 as shown in Theorem 1.

THEOREM 1. *All nets covered in Phase 1 are covered through their longest sensitizable path.*
Proof: *Phase 1 targets the top few longest paths in the design for test generation. Consider a net n covered in the Phase 1 of the test generation methodology. Let the set of all paths covered by the Phase 1 of the methodology be given by I. Let path* $P \in I$ *and* $n \in P$. *Because of the definition of the targeted paths of Phase 1, if there is another path* P_1 *that has more delay than P then* $P_1 \in I$. *Thus all the nets covered by paths identified in Phase 1 are covered by their longest paths.*

Point defects such as resistive opens result in increases in delay of a particular net/gate. When we deal with primitive gates, enabling both rising and falling transitions on all the inputs of the gates enables detection of all resistive opens in the gate as has been outlined in Section 4.1. Our fault model drives transitions through each of the gate inputs as shown in Theorem 2.

THEOREM 2. *Covering both transitions on all the nets in the circuits ensures both transitions on all inputs of all the gates in the circuit.*
Proof*: In our fault model the set of nets includes PI, PO and the set of outputs of each gate in the circuit. Further, if there is a fanout in any one of the nets, all the fanout stems are targeted during test generation. The set of inputs of all gates is completely contained in this set and it follows that all the inputs of all the gates are covered.*

Further, propagating all the tests through the longest path possible through the net for the particular transition minimizes the increase in delay due to the delay faults that can be detected.

There are two major cases that we need to consider when generating tests to detect effects of resistive opens in the CMOS circuits. The first case is when we assume that there is a maximum of one resistive open defect in the circuit under consideration. The more general case concerns the possible presence of multiple resistive open defects in the circuit.

The fact that we generate vectors to drive both rising and falling transitions through each net in the circuit and propagate it through the longest testable path in the circuit, enables the detection of all possible single resistive open defects that can occur in the given combinational block. In the presence of multiple resistive open faults, each resistive open fault can be considered independently of the other resistive open faults in the circuit. Thus the tests that are generated are still effective. The defect occurrence not targeted is the occurrence of multiple resistive opens, many of which should be included in a given path to be detected. Invalidation of tests can also occur due to the presence of delay on the side input which has a $C_i \rightarrow NC_i$ when the on-path input has a $NC_i \rightarrow C_i$ transition.

This model differs from the line delay fault model proposed in [17] in the fact that we target paths and generate tests under the non-robustness criterion to enable an improvement in the detection probability, and also due to the fact that we target the longest paths exceeding a given threshold which gives us a way to target cumulative effects of delays along a path.

5. TEST GENERATION METHODOLOGY

Test generation for the fault model is derived from the vigorous sensitization procedure outlined in [13].

5.1 Implicit Identification of Critical Paths

Consider a combinational block C, consisting of gates of types AND, OR, NAND, NOR, NOT. The methodology incrementally identifies sub-paths that are sensitizable from the primary inputs. Once the sub-paths are identified, all the sensitizable paths associated with any currently identified sub-path are evaluated to identify the longest path through the current subpath. This provides the advantage of not having to repeatedly determine the sensitizability of a given sub-path as they might occur in multiple paths in the combinational block. The procedure uses a five valued logic consisting of $\{0, 1, X, R, F\}$; R(F) refer to rising(falling) transitions.

The progressive identification of the longest path in the combinational block is guided by the *sink delay* of the candidate gates which are chosen from the set of fanouts of the gate under consideration (G_i).

DEFINITION 2. *Sink Delay of a gate in a given combinational block is the delay of the longest path from the gate to one of the primary outputs.*

The gate identified as a candidate gate to follow the current gate in a path is called the *successor gate*. The conditions that need to be satisfied for a fanout to be classified as a successor are

a) the gate has ($C_{G_i} \rightarrow NC_{G_i}$) transition with a settling time before the on-path transition or a NC_{G_i} or an X value at the off-path input, and

b) the gate has the maximum sink delay among all the gates satisfying the above condition.

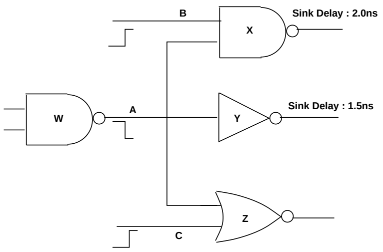

Figure 3: Successor Selection

Figure 3 gives an example gate with a fanout of 3. There is an option of propagating the transition either through gate X or gate Y. If the transition on line B stabilizes before the transition on line A, gate X is chosen because it has larger sink delay. If the transition on line B stabilizes after A, gate Y is chosen.

Thus a transition initiated at a particular primary input is progressively propagated through subsequent successor gates till a primary output is reached. The side input of the successor gates are justified and implication is performed such that the transition is propagated through the successor gate. When we exhaust evaluating all the candidate gates, backtracking is performed where a recently assigned primary input is inverted and the implication is performed.

For the off-path gates in the circuit which have a transition at the inputs, the following conditions are imposed. Let us consider gate G_i which does not lie on the path being tested by the two-vector delay test. Let N (N $>$ 1) represent the number of inputs of G_i. Let $ST_{max}(C_{G_i} \rightarrow NC_{G_i})$ and $ST_{min}(C_{G_i} \rightarrow NC_{G_i})$ represent maximum and minimum stabilization time for $C_{G_i} \rightarrow NC_{G_i}$ transitions on the input of the gate. $ST_{max}(NC_{G_i} \rightarrow C_{G_i})$ and $ST_{min}(NC_{G_i} \rightarrow C_{G_i})$ represent the same for $NC_{G_i} \rightarrow C_{G_i}$ transitions. The conditions that mark the stabilization times for the off-path gates with transitions in the case of the two-vector test include the following.

a) When N - 1 inputs of G_i have NC_{G_i} with one input having a $C_{G_i} \rightarrow NC_{G_i}$ or a $NC_{G_i} \rightarrow C_{G_i}$ with a stabilization time ST, the stabilization time at the output of the gate is (ST + $w_{G_i}(x_{G_i})$).

b) When the number of $C_{G_i} \rightarrow NC_{G_i}$ is greater than 1 with no $NC_{G_i} \rightarrow C_{G_i}$ transitions the stabilization times of the outputs can be given by ($ST_{max}(C_{G_i} \rightarrow NC_{G_i})$ + $w_{G_i}(x_{G_i})$).

c) When the number of $NC_{G_i} \rightarrow C_{G_i}$ is greater than 1 with no $C_{G_i} \rightarrow NC_{G_i}$ transitions, the stabilization times of the outputs can be specified by ($ST_{min}(NC_{G_i} \rightarrow C_{G_i})$ + $w_{G_i}(x_{G_i})$).

d) When the number of $NC_{G_i} \rightarrow C_{G_i}$ transitions and the number of $C_{G_i} \rightarrow NC_{G_i}$ transitions are greater than zero, the stabilization time at the output is given by ($ST_{min}(NC_{G_i} \rightarrow C_{G_i})$ + $w_{G_i}(x_{G_i})$).

x_{G_i} corresponds to the transition propagating through the gate. This procedure uses an ATPG engine based on the PODEM algorithm.

The identification of paths through individual nets is achieved by adding a biasing parameter $\rho \in R^+$ to the rising/falling delays of the gate. This affects the sink delay seen by all the gates on all paths passing through the gate and that appear before the gate under consideration in the paths. The biasing parameter is applied to the rising and falling transition delay separately to drive particular transitions through the nets. The size of ρ can be determined to be a value greater than the delay of the longest sensitizable path in the combinational block under consideration.

When identifying the longest path, in order to identify all the critical paths that are above a given threshold, all the fanout points are identified. Once the paths are identified, the fanout stems at these fanout points which have the required delay characteristics (*sink delay* being above the chosen threshold) are targeted by adding biasing parameters to these “interesting” fanout points.

In the first phase, all the critical paths above a given threshold are identified. The two-pattern test vectors that sensitize the critical paths are also generated. The nets that are not covered by the first phase of the process are then identified. The biasing parameter is added corresponding to the target nets and to the rising or falling transition based on the transition required at the particular net. This biases the vigorous sensitization approach to generate the longest sensitizable path through the target net.

5.2 Determining the Real Delay Values

In our methodology, a technique to estimate the real delay values of the individual gates is used to enable the identification of the true critical paths in the design. This eliminates the need to use unit delay values for each gate which does not yield the true critical paths in the design [18].

We incorporate actual delay values in our model by lumping the delays of gates into delay buffers which are added at the inputs of the gates. This lumping of the delays at the gate inputs allows us to account for the input-output path delays that are associated with the actual delay values. The Standard Delay Format (SDF) description of the circuit is used to identify these actual delay values.

5.3 Effectiveness of Non-robust tests

We deal with synchronous sequential designs. The timing in these designs is fixed by determining the longest sensitizable path in the design. A given two-vector test detects a delay fault on a path if the effect of the point/distributed delay defect on the path exceeds the slack of the path. Considering the longest paths that pass through a net detects the defects on the lines with maximum probability.

With an aim to try and determine the smallest possible delay defects in the circuits, we identify the sensitizability of the paths based on the non-robust sensitizability criterion. Various invalidation conditions have been outlined for delay faults in [19]. We outline the conditions to overcome these invalidation mechanisms in the context of our fault models.

THEOREM 3. *The suppression of transitions on the targeted path in the absence of delay faults is eliminated in our test generation methodology.*
Proof: *The actual delay values used in conjunction with the settling time conditions outlined in the Section 5.1 enable us to accurately specify the arrival time of transitions on the side inputs of the on-path gates in the absence of other delay faults. The conditions imposed on the side inputs of the gates of candidate on-path gates enable them to settle to NC_i values before the on-path input. Thus the suppression of transition on the path is eliminated.*

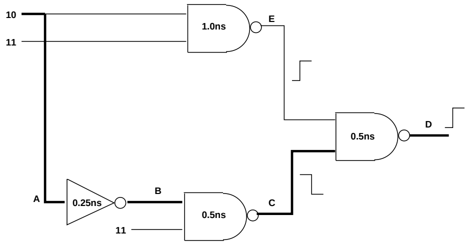

Figure 4: Transition Suppression

We use an example provided in [19] to illustrate this case. The example circuit along with the input vectors is shown in Figure 4; the path under test is shown in bold. The invalidation for the test shown for the path $\{$A, B, C, D$\}$ happens when the $C_{NAND} \rightarrow NC_{NAND}$ on E occurs after the $NC_{NAND} \rightarrow C_{NAND}$ transition on C. Our methodology avoids this case because of the conditions imposed on the selection of the successor gates outlined in Section 5.1.

LEMMA 1. *Early launching of transitions is also avoided in the test generation methodology.*
Proof: *Getting an accurate estimate of these final settling times is achieved by the conditions imposed to evaluate the settling time of gates not on the path being tested. These conditions which were outlined in Section 5.1 enables the last possible settling time to be considered for the purpose of test generation. The requirement for the side inputs to settle to NC_i values eliminates the condition under which this early launch of transitions occurs.*

The other major cause for the invalidation of delay tests is the presence of path delay faults on the side inputs of the gates on the target path. This is the case only when the side inputs have a $C_i \rightarrow NC_i$ transition with the on-path inputs having a $NC_i \rightarrow C_i$ transition. This delay fault could occur due to a variety of causes. These include the presence of a distributed delay defect on the side inputs of the paths and the presence of an additional resistive open on the path leading to the side input.

The scenarios under which a distributed delay defect on a path leading to the side input leads to the invalidation of the delay test is considerably reduced because the test set also targets the longest paths in the given combinational block. Only conditions where the defects due to these variations do not affect the longest paths but still cause a reordering of transitions at the sub-path levels could result in the delay tests being invalidated. One condition is a non-uniform process parameter variation across the combinational block under consideration which as mentioned earlier we can safely assume not to happen. Another contributing condition is a presence of very small process parameter variation. The masking of delay tests due to multiple delay faults with the individual delay faults remaining undetected remains fairly remote.

This delay fault model has a good tradeoff between the test quality and the test application difficulty. The test application difficulty can gauged by the number of vectors required to test the given circuit, this difficulty is minimized because the number of vectors in our fault model is proportional to the number of nets in the circuit rather than the number of paths. The test quality is reflected in the size of the delay defects that can be detected by the test set unlike the classical stuck-at fault model, and our fault model targets this aspect effectively.

6. RESULTS

As described above, the test generation tool for the fault model was obtained by modifying the timing verification tool, CRITIC. In order to enable the use of real delay values in determining the longest paths in the circuit, we first generate the Standard Delay Format (SDF) file and then use it to generate the delay buffers in the file which is in a format suitable for the tool that we use. The experiments are based on a 0.35 micron process.

We generated tests for some of the ISCAS combinational benchmarks. We first determined the structurally longest path in the circuits. The threshold for determining the longest sensitizable critical paths in Phase 1 was fixed at 95% of the delay of the structurally longest path. This yielded only very few paths because of the fact that the circuits were not timing optimized. The times were measured on a Sun UltraSparc II with 1 GB of RAM.

The nets that were not covered in the paths that were identified in Phase 1 in our methodology form the fault set for Phase 2. In this phase, weights are assigned to the delay buffers according to the nets that have to be covered by the appropriate longest sensitizable path. The maximum number of faults in this case, and the number of paths identified in this phase, is twice the number of nets, since we target both the rising and falling transitions for each net.

Table 2 gives the results of the test generation. Column 2 gives the number of nets in the design, Column 3 gives the fault coverage which is the percentage of the nets covered for rising and falling transitions.. The cases where the fault coverage is less than 100% are a result of no sensitizable paths being found and, in some cases, the test generation tool aborting due to limited backtrack. Column 4 gives the percentage of the nets for which the test generation tool aborted, Column 5 gives the percentage of nets for which no sensitizable path could be found and Column 6 gives the CPU time taken to generate the test vectors.

Further, in all the paths that were identified, the percentage of side inputs of the gates in the path that made the test generated non-robust were less than 2% of the total number of side inputs in all gates having a $NC_i \rightarrow C_i$ transition at the on-path input, only c1908 had a value higher than this at 7%. This directly points to a very low probability of a path failing due to delay faults (distributed

Table 2: Test Generation Results

Circuit	# of Nets	Cov (%)	Aborts (%)	Untestable (%)	Time (mins)
c880	598	100	0	0	3.5
c1908	603	86.23	8.14	5.63	363.15
c2670	1118	94.82	0	5.18	263.88
c5315	2751	98.33	0	1.67	311.85
s1196	865	99.77	0	0.23	70.25
s1238	886	95.94	0	4.06	107.56
s1423	951	96.21	3.05	0.74	233.45
s5378	2165	98.57	0.65	0.78	299.88
s38584	17936	98.28	0.25	1.47	9256.14

and point) at the side inputs.

As a comparison with existing approaches, Table 3 gives the percentage of the longest paths identified by our fault model that are detected by test patterns targeting sequence dependent defects. We show the coverage for four representative sets of test patterns generated using the following approaches.

A. Single stuck-at (SSA) test generation.

B. Tests generated using the transition fault model.

C. SSA patterns padded with zeros.

D. SSA patterns padded with ones.

E. Union of all the above patterns.

As can be seen, only a fraction of the longest paths identified are exercised by these test sets. This is the case even with a test set consisting of the union of patterns generated under the first four methodologies. This demonstrates the inadequacy of existing techniques for detecting delays due to resistive opens, and the usefulness of the proposed approach.

Table 3: Longest Path Coverage

Circuits	Path Coverage (%) Test Generation Methodology				
	A	B	C	D	E
c880	19.13	25.56	23.62	21.52	31.24
c2670	11.80	22.49	14.99	15.59	29.48
s1196	33.44	45.56	32.31	31.01	68.66
s1238	27.32	48.11	46.53	31.41	59.29
s1423	16.87	23.20	17.58	19.29	27.40
s5378	32.52	51.58	34.32	35.77	61.17

7. CONCLUSION

In this paper we have presented an improved delay fault model suitable for generating tests for resistive CMOS opens, a class of defects which is presenting problems for the new manufacturing processes. The model considers delays, and covers defects on all nets in the given circuit while also detecting the cumulative effect of delays along a path. This treatment helps in achieving complete stuck-open coverage, and maximizes resistive open coverage.

8. ACKNOWLEDGEMENTS

We would like to thank Rathish Jayabharathi for the helpful discussions and useful insights.

9. REFERENCES

[1] R. C. Aitken, “Nanometer Technology Effects on Fault Models for IC Testing”, *IEEE Transactions on Computer*, 1999, pp. 46–51.

[2] K. T. Cheng and A. Krstic, “Current Directions in Automatic Test-Pattern Generation”, *IEEE Transactions on Computers*, 1999, pp. 58–64.

[3] W. Needham, C. Prunty and E. H. Yeoh, “High Volume Test Escapes, An Analysis of Defects our Tests are Missing”, *Proceedings of IEEE International Test Conference*, 1999, pp. 25–34.

[4] J. Rearick and P. Maxwell, “Deception by Design: Fooling Ourselves with Gate-level Models”, *Proceedings of IEEE International Test Conference*, 2000, pp. 921–929.

[5] C. H. Lee, K. H. Shen, T. K. Ku, C. H. Luo, C. C. Tso, H. W. Chou and C. Hsia, “CVD Cu Technology Development for Advanced Cu Interconnect applications”, *Proceedings o‘f IEEE Interconnect Technology Conference*, 2000, pp. 242–244.

[6] J. Soden, R. Treece, M. Taylor and C. Hawkins, “CMOS IC stuck-open Fault Electrical Effects and Design Considerations”, *Proceedings of IEEE International Test Conference*, 1989, pp. 423–430.

[7] W. Maly, P. K. Nag and P. Nigh, “Testing Oriented Analysis of CMOS ICs With Opens”, *Proceedings of IEEE International Conference on Computer Aided Design*, 1988, pp. 344–347.

[8] J. A. Waicukauski, E. Lindbloom, B. K. Rosen and V. S. Iyengar, “Transition Fault Simulation”, *IEEE Design and Test of Computers*, Apr. 1987, pp. 32–38.

[9] J. L. Carter, V. S. Iyengar and B. K. Rosen, “Efficient Test Coverage Determination for Delay Faults”, *Proceedings of IEEE International Test Conference*, 1987, pp. 418–427.

[10] A. K. Pramanick and S. M. Reddy, “On the Detection of Delay Faults”, *Proceedings of IEEE International Test Conference*, 1988, pp. 845–856.

[11] G. L. Smith, “Model for Delay Based Upon Paths”, *Proceedings of IEEE International Test Conference*, 1985, pp. 342–349.

[12] A. Pierzynska, “Accounting for Electrical Phenomena in Delay Fault Testing,” *Ph.D. Dissertation, Simon Fraser University*, 1999.

[13] H. Chang and J. A. Abraham, “Vigorously Sensitizable Path Extractor”, *Proceedings of the 30th ACM Design Automation Conference*, 1993, pp. 112–117.

[14] R. Jayabharathi, “Hierarchical Timing Verification and Delay Fault Testing”, *Ph. D. Dissertation, The University of Texas at Austin*, 1999.

[15] W. Li, S. M. Reddy and S. K. Sahni, “On Path Selection in Combinational Logic Circuits,” *IEEE Transactions on Computer-Aided Design*, pp. 56–63, January 1989.

[16] K. Baker, G. Gronthoud, M. Lousberg, I. Schanstra and C. Hawkins, “Defect-Based Delay Testing of Resistive Vias-Contacts : A Critical Evaluation”, *Proceedings of IEEE International Test Conference*, 1999, pp. 467–476.

[17] A. K. Majhi, V. D. Agrawal, J. Jacob and L. M. Patnaik, “Line Coverage of Path Delay Faults”, *IEEE Transactions on Very Large Scale Integration Systems*, Oct. 2000, pp. 610–613.

[18] L. G. Silva, J. P. M. Silva, L. M. Silveira and K. A. Sakallah, “Realistic Delay Modeling in Satisfiability-Based Timing Analysis”, *Proceedings of IEEE International Symposium on Circuits and Systems*, 1998, pp. 215–218.

[19] H. Konuk, “On Invalidation Mechanisms for Non-Robust Delay Tests,” *Proceedings of IEEE International Test Conference*, pp. 393–399, 2000.