Accurate Fault Modeling and Fault Simulation of Resistive Bridges

Vijay Sar-Dessai

Dept. of Electrical Engineering
Texas A&M University
College Station, TX 77843
Ph: (409) 845-0531

e-mail: vijay@ee.tamu.edu

D. M. H. Walker

Dept. of Computer Science
Texas A&M University
College Station, TX 77843
Ph: (409) 862-4387
Fax: (409) 847-8578
e-mail: walker@cs.tamu.edu

Contact Author: D. M. H. Walker

Dept. of Computer Science
Texas A&M University
College Station, TX 77843

Ph: (409) 862-4387
Fax: (409) 847-8578
e-mail: walker@cs.tamu.edu

Topic Areas: IC Testing Techniques
Defect and Fault Tolerance Issues

Abstract
This paper presents accurate fault models, an accurate fault simulation technique, and a new fault coverage metric for resistive bridging faults in gate level combinational circuits at nominal and reduced power supply voltages. We demonstrate that some faults have unusual behavior, which has been observed in practice. On the ISCAS85 benchmark circuits we show that a zero-ohm bridge fault model can be quite optimistic in terms of coverage of voltage-testable bridging faults.

1 Introduction

With the increasing density and complexity of VLSI chips, shorts between normally unconnected nodes are expected to be the main type of manufacturing defect [1]. These shorts can be divided into two kinds: intra-gate shorts between nodes within a logic gate, and inter-gate (or external) shorts between outputs of different logic gates [2][3]. Inter-gate shorts, usually called bridging faults, account for about 90% of all shorts [3][4]. Thus in order to accurately estimate the quality of a chip, it is important to have a fault simulator that can simulate realistic bridging faults.

The accuracy of a bridging fault simulator depends on the following factors:

- determination of the voltages at the nodes involved in the bridge as functions of the bridge resistance and power supply voltage
- interpretation of the fault site voltage by gates fed by the bridged nodes.

It is now well accepted that the traditional stuck-at fault model is inadequate for modeling bridging faults [5][6]. Most bridging fault simulators use other alternatives for fault modeling, like the wired-AND, wired-OR, and voting models [7]. Much of the previous work has either used analytical methods [3] to determine the voltages at the bridged nodes and their interpretation by other gates downstream or has used a table-based approach [8] in which pre-computed tables are used to insert voltages or logic values at the fault site. The resistance of the bridge is usually assumed to be 0Ω. However, as shown in [1], many bridges can have significant resistance. Some resistive bridges are only detected under certain sensitization and propagation conditions. Some resistive bridges can degrade the voltage level and circuit timing without affecting the logical function. In order to improve the accuracy of the fault simulator, it is necessary to consider the resistance of the bridge as well.

This paper presents an accurate bridging fault simulation method that models the behavior of bridging faults by using pre-computed data from circuit simulation at the fault site. Once this data has been inserted at the fault site, gate-level logic simulation is done everywhere else. This fault simulator is considered to be accurate because this pre-computed data has been generated for almost all possible bridges involving outputs of pairs of gates in a combinational circuit. It includes some cases which were left unmodeled in [8]. Considering resistive bridges instead of just zero-ohm bridges has also improved the accuracy of the fault simulator.

It is well known that as the power supply voltage is decreased, higher bridging resistances are detected [9][10][11]. This paper shows some cases in which decreasing the power supply voltage could cause a fault which is detected at a higher power supply voltage to be undetected at a lower power supply voltage. Although this behavior has been predicted in [12] and experimentally observed in [13], it has not been proven by specific examples. This work demonstrates such cases with examples and discusses their impact on overall fault coverage.

The remainder of this paper is organized as follows: Section 2 deals with the limitations of previous fault models, explains the fault model used in this bridging fault simulator, and defines the fault coverage metric used in this work. Section 3 describes the construction of look-up tables used in the fault simulator. Section 4 describes the bridging fault simulation algorithm. Fault simulation at decreased power supply voltage is dealt with in Section 5. Section 6 presents some results obtained from benchmark circuits. Limitations of our approach are discussed in Section 7. Concluding remarks are made in Section 8.

2 Bridging Fault Models

A bridging fault model should not only consider the behavior of the gates involved in the bridge, but should include the driven gate behavior. This is because the logical interpretation of the voltage at the bridged nodes depends on the logical threshold of the gate to which the bridged node is connected. In reality, not only do different gates have different thresholds, but each input of a gate has a different threshold. This implies that two gates tied to the same bridged node can interpret the voltage at the bridged node as different logical values, as shown in an example in [8]. This problem is called "The Byzantine General's Problem" [14]. A bridging fault model should also consider the resistance of the bridge, because it is unrealistic to assume that all bridges between gate outputs are pure shorts.

2.1 Previous fault models

A number of bridging fault models have been proposed in previous work, like the wired-AND, wired-OR and voting models. The wired model is inadequate to model bridging faults because the voltages at the bridged nodes not only depend on the activated pull-up and pull-down networks and the transistor model parameters, but also on the bridging resistance. Even though bridging faults in CMOS circuits almost always result in recognizable logic values [15][16][17], the wired model leads to an incorrect description of the bridging fault, as the example in [7] shows. In [7], a voting model has been proposed that uses a table-based approach for deciding the vote. The drawback of this approach is that it neglects the resistance of the bridge, and also ignores the "Byzantine General's Problem". The bridging fault simulator proposed by [8] is based on accurate modeling of fault behavior, but it considers zero-ohm bridges only, and leaves some classes of bridging faults unmodeled. In [18] the bridging resistance was considered in the fault model, but the threshold voltage of gates fed by the bridged nodes was assumed to be $V_{DD}/2$ for all gate inputs. A method of simulating bridging faults using variable gate logic thresholds has been proposed in [19]. A concept called Parametric Fault Model has been proposed in [3], in which the bridging resistance is taken into account, and instead of propagating a faulty logic value to the primary output, the detectable bridging resistance interval is propagated. However, this model is based on determining the detectable resistance by electrical equations rather than by circuit simulation and it did not discuss some cases too.

2.2 Description of fault model

In this paper, bridging faults have been modeled by HSPICE [20] circuit simulation of almost all possible bridging fault configurations for all gates included in the gate-level description of the ISCAS85 benchmark circuits. Each circuit was built using basic gates, and no complex gates were used. Each gate was implemented using complementary CMOS logic. We used the SPICE level 3 parameters for the HP CMOS14TB 0.5 μm process, running at a normal V_{DD} of 3.3V. The benchmark circuits contain 22 different types of gates, and by exhaustively simulating different types of bridging faults (explained below) that can occur in various combinations of these gates, we obtain a set of look-up tables containing data that is used at the fault site during fault simulation.

In general, the following types of resistive bridges can occur in a combinational circuit:

Case 1: Bridge between two primary inputs:

We define a primary input as a circuit node that is not at the output of any gate. This type of bridge is not detectable by logic testing, because primary inputs are a source of infinite current, and any bridge between a primary input carrying a logic 1 and another primary input carrying a logic 0 will not affect the functional behavior of the circuit. Hence this type of bridging fault is not modeled.

Case 2: Bridge between a primary input and any other node:

Figure 1 shows a bridging fault between a primary input A and the output of a NAND2 gate, X. Node X feeds into 2 gates having different threshold voltages. The bridge resistance detectable at nodes P and Q depends on the test vector at A, B, C as well as on the logic threshold values of the 2 gates connected to node X.

For example, HSPICE simulation shows that if the applied vector is A,B,C = {0,0,1}, then we can detect a bridging resistance up to 1600Ω at node P and up to 1400Ω at node Q, assuming that other inputs of the AND2 and OR2 gates are held at their non-controlling values. The fault does not propagate along primary input A because of the reason stated earlier.

Case 3: Bridge between outputs of two gates (bridged nodes feeding into different gates):

Figure 2 illustrates a case in which the outputs of a NAND2 and a NOR2 gate are bridged, and the bridged nodes X and Y feed into different gates. The bridge resistance detectable at the outputs of each of these gates depends on the vector at A1, B1, A2, B2 as well as on the logic thresholds of the gates connected to nodes X and Y.

Assuming that the vector at A1,B1,A2,B2 = {1,0,1,1}, HSPICE simulation of this case shows that the bridging fault will propagate along node X, and the resistance detectable is up to 1000Ω at P and up to 1400Ω at Q. Due to the vector used and the thresholds of the INV and the AND2 gate, the fault does not propagate along node Y, and is undetectable at nodes R and S.

Case 4: Bridge between outputs of two gates (bridged nodes feeding into same gate):

Figure 3 illustrates the case in which the outputs of a NAND2 and NAND2 gate are bridged, and the bridged nodes feed into the same AND3 gate. The bridge resistance detectable at node P depends only on

the vector at A1, B1, A2, B2 (assuming that the third input of the AND3 gate is at its non-controlling value).

With a vector of A1,B1,A2,B2 = {1,0,1,1}, HSPICE simulation of this circuit shows that a bridge resistance of up to 800Ω is detectable at node P.

Case 5: Bridge involving primary outputs

If two primary outputs are bridged to each other, the case can be dealt with in a similar manner as in case 3, if we imagine the primary outputs to be nodes feeding gates having a logic threshold value of $V_{DD}/2$, as was done in [18].

If a primary output is bridged to a primary input, this case would fall under case 2. If a primary output is bridged to a node that is neither a primary input nor a primary output, this case would fall under case 3. In both situations we can imagine the primary output to be a node feeding a gate having a logic threshold value of $V_{DD}/2$.

The fault simulator we have built is based on this accurate fault model. By doing HSPICE simulations of all possible gate combinations in all the above cases, we can accurately model the behavior of bridging faults, and insert the data obtained from the simulations into the fault simulator at the fault site.

2.3 Fault coverage metric

Since metal bridging resistance mainly falls in the range from 0Ω to 1000Ω [1], a geometric distribution used in [18] is found to be a good fit. The PDF of the bridging resistance is:

$$P(R_b) = 1-(1-p)^{R_b} \quad (1)$$

where R_b is the bridging resistance and p = 0.00258 for the data in [1].

The normalized fault coverage $c(i)$ for the bridging fault configuration *i* can be computed using:

$$c(i) = \frac{1-(1-p)^{R_b(i)}}{1-(1-p)^{R_{b\max}(i)}} \quad (2)$$

where $R_b(i)$ is the detectable resistance for the bridging fault configuration *i*, and $R_{b\max}(i)$ is the maximum detectable resistance at the fault site for the bridging fault configuration *i*.

The fault coverage of a test vector v is given by:

$$C_v = \frac{1}{N}\sum_{i=1}^{N} c_v(i) \quad (3)$$

where $c_v(i)$ is the normalized fault coverage for the bridging fault i using that test vector v and N is the total number of logic-testable faults in the circuit (assuming equally likely faults).

The cumulative fault coverage of a test vector set is given by:

$$C_c = \frac{1}{N}\sum_{i=1}^{N} c_h(i) \quad (4)$$

where $c_h(i)$ is the highest achieved normalized fault coverage for the bridging fault i.

Since $c_h(i)$ is normalized, C_c is the coverage of all bridging faults potentially detectable by low-speed voltage test, which we refer to as *logic-testable bridging faults.*

3 Construction of Look-up Tables

In order to obtain information about the behavior of the circuit at the fault site for fault simulation, we have built a number of look-up tables. Prior to the construction of the look-up tables, the logic threshold of each type of gate in the ISCAS85 circuits was determined (we assumed that for a given gate, all inputs have the same logic threshold). There are three types of look-up tables, one for each type of bridging fault classified under case 2, case 3, and case 4 explained in the previous section.

Case 2: For this type of bridge involving a primary input and any other node, construction of the table was done by simulating a bridge between a DC source and the output of a gate. For all test vectors that excite this fault, the maximum detectable resistance was determined by comparing the voltage at the output of the gate with the logic thresholds V_{th} of all gates. Figure 4 (a) illustrates this principle for the circuit in Figure 1. As the resistance of the bridge increases, the voltage at node X approaches its fault-free value. The crossover point between this voltage and the logic threshold of the gate connected to node X determines the maximum detectable resistance. The high gain of the logic gates ensures that before and after this crossover point, the voltages at P and Q are restored to their faulty and good logic values respectively. The table contains the following information at each entry:

(a) vector at primary input and inputs of the gate
(b) logic threshold of gate connected to bridged node
(c) maximum detectable resistance under conditions (a) and (b)

We have built one table for each type of gate in the ISCAS85 circuit. Since there are only 22 different types of gates, we have 22 different tables.

Case 3: For this type of bridge involving outputs of two gates, construction of the table was done by simulating

a bridge between the outputs of two gates for all possible vectors that excite the fault. Referring to Figure 2, for each vector the maximum detectable resistance was determined by comparing the voltage at nodes X and Y with logic thresholds of all gates. As shown in Figure 4(b), as the bridge resistance is increased, the voltages at X and Y approach their fault-free values. The crossover point between the voltage at nodes X and Y and the logic threshold values determine the maximum detectable resistance. Each table entry contains the following information:

(a) vector at bridged gate inputs
(b) fault propagation path (along bridged nodes)
(c) logic threshold of gate connected to propagation path
(d) maximum detectable resistance under conditions (a), (b) and (c)

If we are to build one table per bridged pair of gates, with the 22 different types of gates in the ISCAS85 circuits there could be a total of 253 such tables, if we consider all combinations. However, as pointed out in [17], we can analyze which bridges are the most likely and generate tables only for them. Tables for other bridges which rarely occur could be built during a pre-analysis for each circuit. The approach we have implemented is to generate tables for all combinations of pairs of gates having a fan-in of 5 or less. This is because of the fact that bridges involving outputs of gates with a larger fan-in (NAND8, AND8, NOR8, AND9 in the ISCAS85 circuits) occur extremely rarely (as shown in Table 2, under "Large-case BFs") amongst the target bridges in our implementation of the benchmark circuits, and are left unmodeled. Thus we have generated 171 tables for case 3.

Case 4: For this type of bridge involving bridged outputs of two gates feeding into the same gate, construction of the tables was done by simulating a bridge between the outputs of two gates, with the outputs feeding a third gate. The circuit was simulated for all possible vectors that excite the fault, and the voltage at the output of this third gate was monitored. The bridging resistance at which the voltage at this node changed from its faulty value to its fault-free value was determined to be the maximum detectable bridging resistance. Each entry in the table contains the following information:

(a) vector at inputs of gates whose outputs are bridged
(b) maximum detectable resistance under condition (a).

We need to generate one table for each combination of 3 gates. Since this number is extremely large, the approach we have used is to model only those cases that occur in the ISCAS85 circuits. Again, if the bridges involve outputs of gates having a fan-in larger than 5, these cases are not modeled. Another type of bridging fault falling under this case is that shown in Figure 5. This type of bridge has been found to be extremely rare amongst the target bridges in the ISCAS85 circuits (as shown in Table 2, under the column "Other dropped BFs"), and is left unmodeled. Thus we have generated 20 tables case 4.

Some case 4 bridging faults exhibit anomalous behavior in terms of the maximum detectable resistance. An example is circuit in Figure 6 (a), which showed the behavior depicted in Figure 6 (b) when simulated at low voltage (2V) with a vector A1,B1,A2,B2 = {0,0,0,1}. Instead of the detectable resistance being in the interval of 0Ω to the maximum detectable resistance R_{bmax}, it lies in the interval [R_{bmin}, R_{bmax}]. A similar case involving an XOR2 gate with a low-resistance bridge inserted between its inputs was confirmed in an actual circuit [21]. During the table construction of such cases, the entry in the table corresponding to the maximum detectable resistance is replaced with a resistance interval. (In all other cases, it is implicitly assumed that the lower limit of detectable resistance is 0Ω).

The tables for cases 2 to 4 occupy about 3MB of space, and table construction time was considerable, due to the many HSPICE simulations that had to be performed. One way to reduce the construction time would be to use the property of input equivalence in the pull-up and pull-down networks in any gate. For instance, in the case of a bridge involving a NAND2 gate, simulation with inputs {1,0} and {0,1} would be the same if the pull-up transistors are of the same size.

All entries in each table have been arranged in decreasing order of maximum detectable resistance, so that in the future we can use them for ATPG.

During each HSPICE simulation, the resistance was swept from 0 to 3000Ω. Thus even if the actual maximum detectable resistance exceeded 3000Ω, the table entry showed 3000Ω. We chose this value because using equation (1), we noted that at 3000Ω the detection probability was 99.95%, which was high enough to assume detection of any detectable resistive bridge, as shown by the data in [1].

4 Bridging Fault Simulator

Implementing the bridging fault simulator involved obtaining the fault list and implementing the fault simulation algorithm.

4.1 Fault list preparation

Even though we are considering external bridges only, the list of faults to be considered for logic testing of bridging faults could still be large (of the order of n^2 where n is the number of nodes in the circuit) if we were to consider *all* possible external bridges. Hence a reduced fault set is obtained by randomly choosing bridging faults from the all-pair bridging fault list. The number of bridging faults in this reduced fault list is determined from the reduction ratio (extracted faults/all-pair faults) listed in [22]. Alternatively, a defect simulator [23][24][25] can be used to generate a realistic bridging fault list.

From the reduced fault list, bridging faults between primary inputs are eliminated, because they cannot be tested by logic testing.

Feedback faults are discarded from the fault list. Although feedback bridging faults can potentially be tested by logic testing, some of them may cause oscillations, which makes it difficult to model such faults. We will return to a discussion on feedback bridges in a later section.

From the list of logic-testable bridging faults, bridging faults which are not modeled are dropped. Faults which fall under this class have been explained in Section 3. As shown in Table 2, very few such faults occur.

Bridging faults eliminated in the above steps form a list of faults that can be detected by IDDQ testing. In IDDQ testing, a bridging fault is detected if the nodes involved in the bridge are set to opposite logic values, and the bridging resistance is low enough to cause a detectable IDDQ increase.

4.2 Fault simulation algorithm

The fault simulation procedure is carried out in the following manner:

- For each fault in the set of logic-testable faults, we determine the maximum detectable resistance from the look-up table associated with that bridge. This maximum detectable bridging resistance depends only on the gates whose output nodes are bridged and the gates fed by the bridged nodes, and is independent of the applied test vector. This resistance gives an indication of the best fault coverage we can achieve for this fault.
- For each vector in the test set, the fault-free logic value at each node is determined. A list of excited faults is formed from the logic-testable fault list.
- For each excited fault, the look-up table is used to determine the bridging resistance detectable at the fault site. We follow the convention used in [3] to simulate the faulty circuit. In this convention, the resistance interval, which specifies the range of resistances that can be detected by that test vector, is placed at the fault site. For example, for the case of a primary input bridged to any other node (Figure 1), the interval [0,1600] is inserted at node P (if the non-bridged input of the AND2 gate is at logic 1) and the interval [0,1400] is inserted at node Q (if the non-bridged input of the OR2 gate is at logic 0).
- Fault simulation continues from the fault site towards the primary output with the propagation of the resistance interval at each node. During this forward simulation, the resistance interval can get reduced if two or more nodes carrying resistance intervals feed into the same gate. The resistance interval at the output of such gate can be a union or an intersection of the intervals at the inputs of the gate. This implies that the detectable resistance interval can either remain the same or decrease from its value at the fault site, and may lead to a loss of fault coverage at the primary output.
- Once the resistance intervals at the primary outputs are known, the normalized fault coverage $c(i)$ for each fault excited by this vector is computed using equation (2). If $c(i)$ is equal to 100%, then we have detected the maximum possible bridging resistance, and this fault can be dropped from the logic-testable fault list. Thus, a fault is dropped only if the bridging resistance interval detectable at the primary outputs is the maximum detectable bridging resistance interval at the fault site. We could use a more relaxed criteria for dropping, that is, we could drop a fault if the normalized coverage is within Δ of 100%, in which case more faults will be dropped per test vector.
- We then compute the fault coverage C_v of this test vector using equation (3).
- The above procedure is repeated for each vector. For each fault, we keep track of the best fault coverage obtained so far. This figure is then used to compute the cumulative fault coverage C_c of the entire test vector set using equation (4).

If the PDF of the bridging resistance is not known, then we can decide whether to drop a fault or not by examining the detectable resistance intervals. In this case, a higher detectable resistance implies better fault coverage, assuming that higher resistance bridges are less probable.

5 Fault Simulation at Decreased Power Supply Voltage

It is well known that decreasing the power supply voltage V_{DD} results in a larger bridging resistance being detected [9][10][11][12]. Thus we can obtain an increased overall bridging fault coverage by decreasing V_{DD}. In [12], the authors discussed some cases in which decreasing V_{DD} reduces the detectable resistance. The first case was called a "favorable" case from the point of view of fault coverage improvement, because it resulted in an increase in maximum detectable resistance with decreased V_{DD}. The second case was called "partially favorable" because the maximum detectable resistance first increased and then decreased with decreasing V_{DD}. This indicates that this bridging fault was detectable at higher V_{DD} and undetectable at certain lower V_{DD} values. The implication of this is that fault coverage may not necessarily increase with decreased V_{DD}. However, the authors claimed that since these cases are a "mathematical possibility but never appeared in usual design", the favorable cases are predominant, thus leading to increased overall fault coverage.

We have discovered some cases of bridging faults in common circuit configurations in which the fault is detectable at a higher V_{DD} value but undetectable at decreased V_{DD} values. These cases fall under case 4 as discussed in Section 2, in which outputs of two gates are bridged, with the bridged nodes feeding into the same gate. Figure 7 shows such a case, involving a NAND2 gate having bridged inputs.

Table 1 shows the result of HSPICE simulation of this circuit for different test vectors that excite this fault, along with the maximum detectable resistance R_{bmax} at node Z for two different values of V_{DD}. (An X in the table means that the fault is undetectable).

The results of this simulation indicate that at decreased V_{DD}, the bridging fault is undetectable for some test vectors, even though it is detectable at a higher value of V_{DD}. Some test vectors can detect a higher bridging resistance. In [13], it was shown that some circuits escaped fault detection at low voltage even though the faults were detected at higher V_{DD}.

However, since bridging faults that fall under this case occur relatively few times in the ISCAS85 circuits, the impact of this behavior on overall fault coverage is negligible, as shown in the next section. Even if situations like these do exist in a circuit, the overall fault coverage still improves with decreased V_{DD}, as the results in the following section demonstrate.

6 Results and Discussion

The bridging fault simulator was run on the ISCAS85 benchmark circuits. Table 2 gives some statistics of these benchmark circuits. For each circuit, the table lists the total number of external nodes, the number of all-pair bridging faults, randomly selected (reduced) bridging faults, faults between two primary inputs, feedback bridging faults, faults between outputs of large fan-in gates, bridging faults not modeled because of their special nature, bridging faults which can be potentially detected by voltage testing (which forms the logic-testable fault list), and the number of applied test vectors.

The test vectors were obtained from an automatic test pattern generator for stuck-at faults [26]. This also gave us an indication of how good a fault coverage we can obtain for resistive bridging faults using a stuck-at test set.

Each circuit was simulated at V_{DD}=3.3V, 2.4V and 1.2V, and for different resistance distributions. The results of some of the simulations are displayed in Figure 8 and Figure 9.

Figure 8 shows the percentage of faults completely detected (dropped) by the test vector set, with simulation done at V_{DD}=3.3V and V_{DD}=1.2V, and for an average resistance distribution using equation (1) (realistic bridges) and a zero-ohm resistance distribution (zero-ohm bridges). Figure 9 shows the fault coverage for the entire test vector set.

In the case of zero-ohm bridges, a fault is considered detected with a 100% fault coverage and is dropped if any resistance interval associated with that fault propagates to a primary output. Thus the fault coverage in this case is the same as the percentage of faults detected. This is similar to the fault dropping criteria used in stuck-at fault simulators. In the case of realistic bridges, if there is a loss of fault coverage as the fault propagates to the primary outputs, the fault is not dropped and the normalized fault coverage will not be 100%.

The following observations can be made with reference to Figure 8 and Figure 9:

- At 3.3V, even though the percentage of realistic bridges completely detected (those for which the fault coverage is equal to the maximum possible fault coverage) is low (Figure 8), the fault coverage (Figure 9) is high. This is because for the faults which escaped complete detection, the maximum resistance detected was high, though not equal to the maximum detectable resistance. Figure 10 shows the distribution of fault coverage for faults which escaped 100% detection. For most circuits, more than half of these faults had a fault

coverage exceeding 90%. The figure also shows that very few faults had a 0% fault coverage.

- For zero-ohm bridges, the percentage of detected faults is equal to the fault coverage, since every fault that contributes to the overall fault coverage has a 100% fault coverage.
- At 3.3V and 1.2V, the percentage of zero-ohm bridges completely detected (Figure 8) and their fault coverage (Figure 9) was higher than the corresponding numbers for realistic bridges. This is due to the fact that for zero-ohm bridges, the fault is dropped and the fault coverage is 100% for any resistance interval at the primary output.
- For realistic bridges, the percentage of completely detected bridges at 1.2V is more than the corresponding number at 3.3V (Figure 8). This is also the case for fault coverage of realistic bridges (Figure 9). This confirms previous results that fault coverage improves with decreasing V_{DD}, and also suggests that even if there are a few isolated faults in which the fault is undetectable at lower V_{DD} but detectable at higher V_{DD}, the overall fault coverage still improves with decreased V_{DD} due to the relatively fewer number of such faults.
- However, for zero-ohm bridges, there are 2 circuits (c2670 and c3540) for which the fault coverage at 1.2V is *lower* than the fault coverage at 3.3V (Table 3 and Figure 9). This is because the fault coverage is high at 3.3V due to the large number of detected faults, and at 1.2V those few faults which escape detection cause the overall fault coverage to drop. This anomaly occurs only for these two circuits because these circuits have a relatively higher number of case 4 bridges than the other circuits, and as explained in section 5, case 4 bridges are responsible for decreased fault coverage at lower V_{DD}.
- At 1.2V, the maximum detectable resistance at the fault site as well as the detectable resistance at the primary outputs was very high (> 3000Ω) in most bridging faults. However, since in our HSPICE simulations we placed a limit of 3000Ω for the resistance sweep, almost all faults were completely detected. This explains the very high percentages for the data at 1.2V in Figure 8. This tends to make the data slightly optimistic. If there was a much higher limit in our resistance sweep, then, for example, the maximum detectable resistance at the fault site could have been 10KΩ, and we could have detected up to 6KΩ at the primary output. The fault would not have been dropped in this case, but it is dropped in our approach.

Figure 11 shows how the fault coverage improves as the fault simulation progresses (for the first 30 vectors) for the c1355 circuit at 3 different values of V_{DD}, assuming realistic bridges. It is clear that the fault coverage increases as V_{DD} decreases. However, Figure 10 also shows that the coverage for the first 10-15 vectors is lower at 1.2V than at 2.4V and 3.3V. The reason for this is that our coverage metric is relative. The resistance interval detected rises with decreasing voltage, but the maximum detectable resistance rises even faster, and occurs for fewer sensitization and propagation conditions. Thus the probability of obtaining the best coverage is lower for each vector.

Figure 12 shows how the fault coverage improves as the fault simulation progresses (for the first 30 vectors), for the c1355 circuit at V_{DD}=3.3V, with 4 different bridging resistance distributions. The $\pm$25% cases are for a $\pm$25% change in the mean value of the real resistance distribution. As expected, lower resistive bridges have higher coverage. The curves also show that a vector that is good for one resistance distribution is also good for other distributions.

However, as shown in Figure 13, the coverage of realistic resistive bridges remains lower than that of zero-ohm bridges even as the coverage of zero-ohm bridges goes to 100%. This result is similar to what has been found comparing zero-ohm bridges to stuck-at faults [27].

7 Limitations

Like other bridging fault simulators built on a table-based method, our fault simulator has certain limitations. The first is that we have to create a large number of look-up tables prior to running the fault simulator. The number of look-up tables depends on the type of gates used in the circuits, and increases with the number of different types of gates. In addition, with a change in device parameters, a new set of look-up tables has to be created. If the fault simulator is to be run at several different power supply voltages, again a new set of look-up tables has to be created for each value of V_{DD}.

The second limitation is that although almost all bridging fault situations have been dealt with during the creation of the look-up tables, there are a few which either cannot be modeled or are computationally expensive to model. These cases have been explained in Section 3. Although we have determined that these cases occur relatively rarely in our implementation of benchmark circuits, inclusion of these cases during fault simulation can result in an improvement in accuracy.

A third limitation of our simulator is that feedback bridging faults are not considered for logic testing,

(although feedback faults involving a primary input are included in the logic-testable fault list). Prior work [28][29] showed that only under special and rare situations do some feedback faults result in oscillations, and so inclusion of feedback faults in the simulator would result in better accuracy.

8 Concluding Remarks

An accurate bridging fault simulator based on an accurate bridging fault model has been developed in this work. The main factors contributing to this accuracy are exhaustive simulation of almost all potential bridging fault situations in a circuit, and inclusion of resistive bridges as opposed to zero-ohm bridges. It has been confirmed that fault simulation done at reduced power supply voltage leads to an increase in overall fault coverage. Certain situations where reducing the power supply voltage causes a fault to go undetected at lower V_{DD}, despite being detected at the higher V_{DD} have been presented. The fault model and simulation results imply that ATPG targeting resistive bridges has the potential for improving fault coverage beyond that obtained by a stuck-at fault test. We also show that as with stuck-at faults, the zero-ohm bridge fault model is optimistic relative to realistic resistive bridges.

References

[1] R. Rodriguez-Montanes, E.M.J.G. Bruls and J. Figueras, "Bridging Defect Resistance Measurements in a CMOS Process," in *Proc. Int. Test Conf*., 1992, pp. 892-899.

[2] M. Renovell, P. Huc and Y. Bertrand, "CMOS Bridging Fault Modeling," *IEEE VLSI Test Symp*., pp. 392-397, 1994.

[3] M. Renovell, P. Huc and Y. Bertrand, "The Concept of Resistance Interval: A New Parametric Model for Realistic Resistive Bridging Fault," *IEEE VLSI Test Symp*., pp. 184-189, 1995.

[4] J.J.T. Sousa, F.M. Goncalves, J.P. Teixeira, "IC Defects-Based Testability Analysis," in *Proc. Int. Test Conf*., 1991, pp. 500-509.

[5] T.M. Storey and W. Maly, "CMOS Bridging Fault Detection," in *Proc. Int. Test Conf*., 1990, pp. 842-851.

[6] T.M. Storey, W. Maly, J. Andrews and M. Miske, "Stuck Fault and Current Testing Comparison Using CMOS Chip Test," in *Proc. Int. Test Conf*., 1991, pp. 311-318.

[7] Steve D. Millman and James P. Garvey, Sr., "An Accurate Bridging Fault Test Pattern Generator," in *Proc. Int. Test Conf*., 1991, pp. 411-418.

[8] Jeff Rearick and Janak H. Patel, "Fast and Accurate Bridging Fault Simulation," in *Proc. Int. Test Conf*., 1993, pp. 54-62.

[9] Yuyun Liao and D.M.H. Walker, "Fault Coverage Analysis for Physically-Based CMOS Bridging Faults at Different Power Supply Voltages," in *Proc. Int. Test Conf*., 1996, pp. 767-775.

[10] Hong Hao and Edward J. McCluskey, "Very-Low-Voltage Testing for Weak CMOS Logic ICs," in *Proc. Int. Test Conf*., 1993, pp. 275-284.

[11] Jonathan T.-Y. Chang and Edward J. McCluskey, "Quantitative Analysis of Very-Low Voltage Testing," *IEEE VLSI Test Symp*., pp. 332-337, 1996.

[12] M. Renovell, P. Huc and Y. Bertrand, "Bridging Fault Coverage Improvement by Power Supply Control," *IEEE VLSI Test Symp*., pp. 338-343, 1996.

[13] Siyad C. Ma, Piero Franco and Edward J. McCluskey, "An Experimental Chip to Evaluate Test Techniques Experiment Results," in *Proc. Int. Test Conf*., 1995, pp. 663-672.

[14] J.M. Acken and S.D. Millman, "Fault Model Evolution for Diagnosis: Accuracy vs. Precision," in *Proc. IEEE Custom Integrated Circuits Conf*., 1992, pp. 13.4.1-13.4.4.

[15] G. Freeman, "Development of Logic level CMOS Bridging Fault Models," , Center for Reliable Computing Technical Report 86-10, Stanford University, 1986.

[16] John M. Acken, "Deriving Accurate Fault Models,", CSL-TR-88-365, Computer Systems Laboratory, Stanford University, Oct. 1988.

[17] Chennian Di and Jochen A.G. Jess, "An Efficient CMOS Bridging Fault Simulator: With SPICE Accuracy," *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems*, Vol. 15, No. 9, September 1996, pp.1071-1080.

[18] Yuyun Liao and D.M.H. Walker, "Optimal Voltage Testing for Physically-Based Faults," *IEEE VLSI Test Symp*., pp. 344-353, 1996.

[19] P. C. Maxwell and R. C. Aitken, "Biased Voting: A Method for Simulating CMOS Bridging Faults in the Presence of Variable Gate Logic Thresholds," in *Proc Int. Test Conf*., 1993, pp.63-72.

[20] Meta-Software, *HSPICE Manual*, Campbell, CA: Meta-Software Inc. 1990.

[21] H. Balachandran, Texas Instruments Inc., *Private Communication.*

[22] Tzuhao Chen, Ibrahim N. Hajj, Elizabeth M. Rudnick, Janak H. Patel, "An Efficient Iddq Test Generation Scheme for Bridging Faults in CMOS Digital Circuits," *IEEE Int. Workshop on Iddq Testing*, pp. 74-78, 1996.

[23] D.M.H. Walker, VLASIC System User Manual Release 1.3, Carnegie Melon University, Pittsburg, PA, 1990.

[24] D.D. Gaitonde and D.M.H. Walker, "Heirarchical Mapping of Spot Defects to Catastrophic Faults - Design and Applications,", *IEEE Trans. on Semiconductor Manufacturing*, May 1995, pp. 167-177.

[25] Alvin Jee, "Carafe: An Inductive Fault Analysis Tool for CMOS VLSI Circuits," *Technical Report UCSC-CRL-91-24, University of California at Santa Cruz, Computer Engineering Dept.*, February 1990.

[26] H.K. Lee and D.S. Ha, "On the Generation of Test Patterns for Combinational Circuits," *Technical Report No. 12_93, Dept. of Electrical Eng., Virginia Polytechnic Institute and State University.*

[27] Li-C. Wang, M. Ray Mercer and Thomas W. Williams, "On Efficiently and Reliably Achieving Low Defective Part Levels," in *Proc. Int. Test Conf.*, 1995, pp. 616-625.

[28] Y.-J. Kwon and D.M.H. Walker, “Yield Learning via Functional Test Data,” in *Proc. Int. Test Conf.*, 1995, pp. 626-635.
[29] Haluk Konuk and F. Joel Ferguson, “Oscillation and Sequential Behavior Caused by Interconnect Opens in Digital CMOS Circuits,” in *Proc. Int. Test Conf.*, 1997, pp. 597-606.

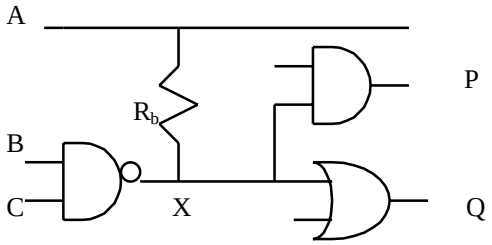

Figure 1. Bridging fault between a primary input and any other node.

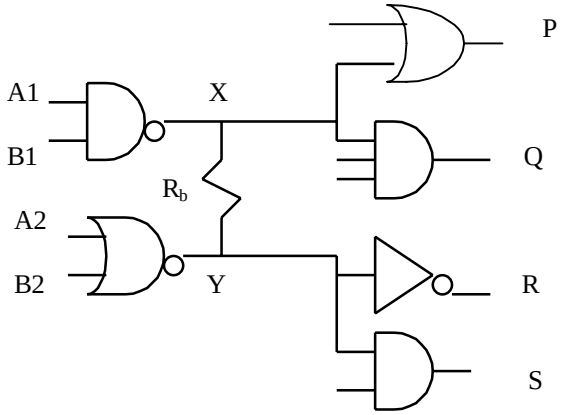

Figure 2. Bridging fault between outputs of two gates.

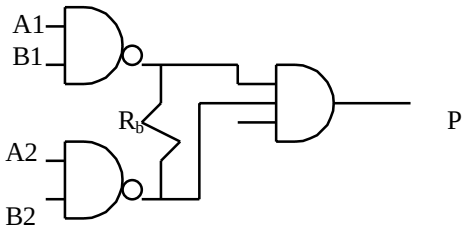

Figure 3. Bridging fault between outputs of two gates, with bridged nodes feeding into same gate.

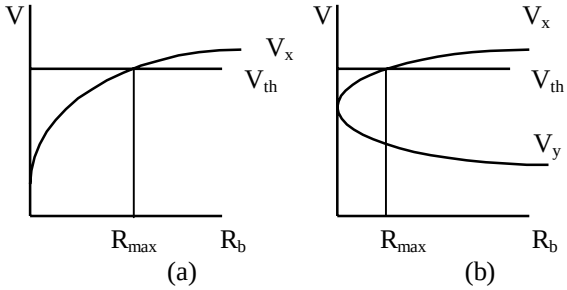

Figure 4. Determination of maximum detectable bridging resistance.

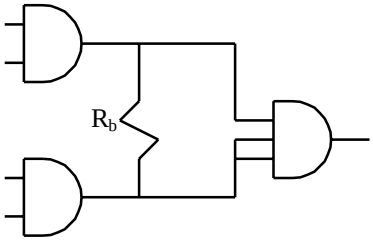

Figure 5. Bridging fault between inputs of gates tied together.

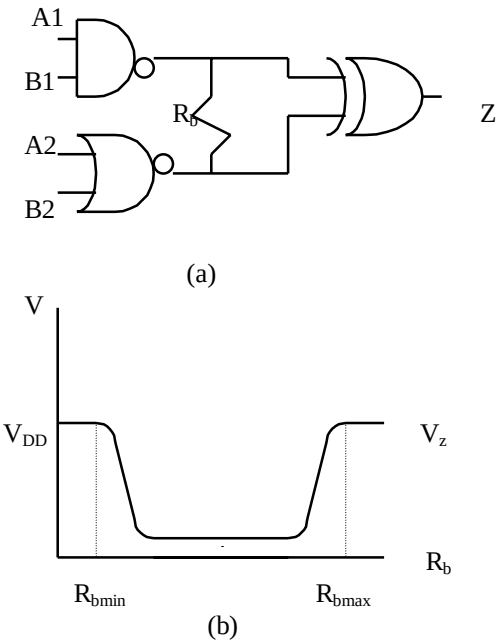

Figure 6. XOR gate with inputs bridged.

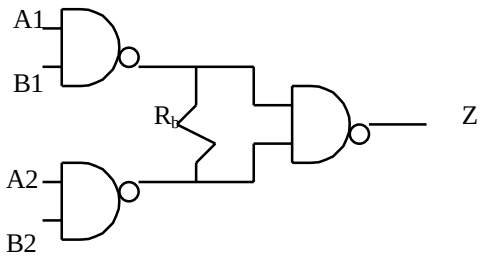

Figure 7. NAND2 gate with inputs bridged.

Test vector A1B1A2B2	R_{bmax} at V_{DD}=3.3V	R_{bmax} at V_{DD}=1.2V
0 0 1 1	2000Ω	> 6000Ω
1 1 0 0	1800Ω	> 6000Ω
1 0 1 1	1200Ω	X
0 1 1 1	1200Ω	X
1 1 0 1	1000Ω	X
1 1 1 0	1000Ω	X

Table 1. Maximum detectable resistance at different V_{DD} values for the circuit in Figure 7.

Circuit	Nodes	All-pair BFs	Reduced BFs	PI BFs	Feed-Back BFs	Large-case BFs	Other dropped BFs	Logic-Test BFs	Applied Vectors
c17	11	55	6	1	2	0	0	3	5
c432	196	19,110	269	7	103	2	0	157	50
c499	243	29,403	191	8	47	0	1	135	53
c880	443	97,903	1086	27	110	0	1	948	48
c1355	587	171,991	1066	5	422	0	0	639	85
c1908	913	416,328	2206	7	521	16	0	1662	118
c2670	1426	1,016,025	4572	110	168	0	0	4294	103
c3540	1719	1,476,621	5315	4	780	100	0	4431	156
c5315	2485	3,086,370	7407	38	237	11	0	7121	120
c6288	2448	2,995,128	4492	1	1275	0	0	3216	34
c7552	3719	6,913,621	12444	40	298	0	0	12106	204

Table 2. Statistics for ISCAS85 circuits used.

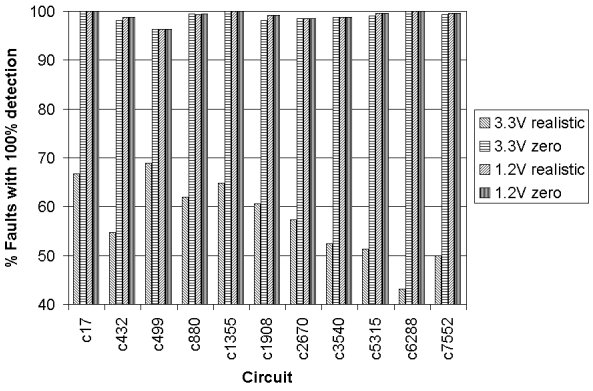

Figure 8. Faults detected for each circuit.

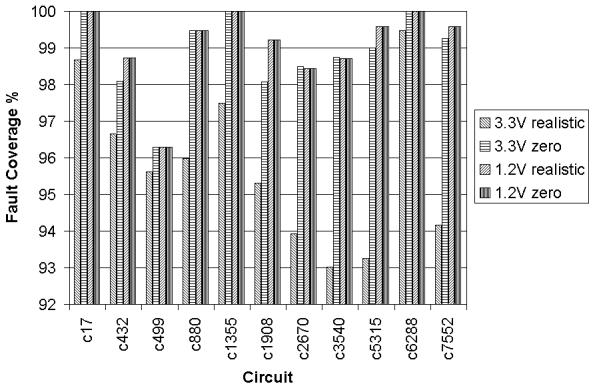

Figure 9. Fault coverage for each circuit.

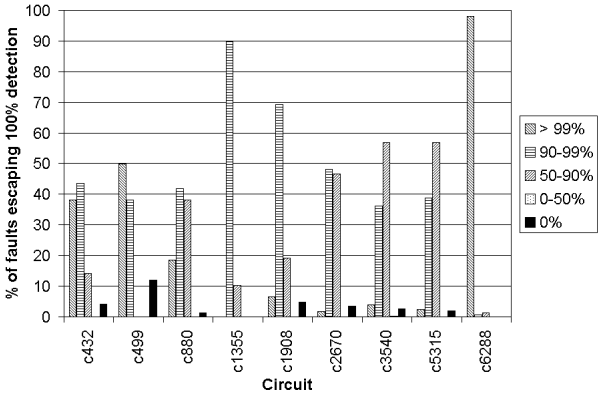

Figure 10. Fault coverage for faults which escaped 100% detection.

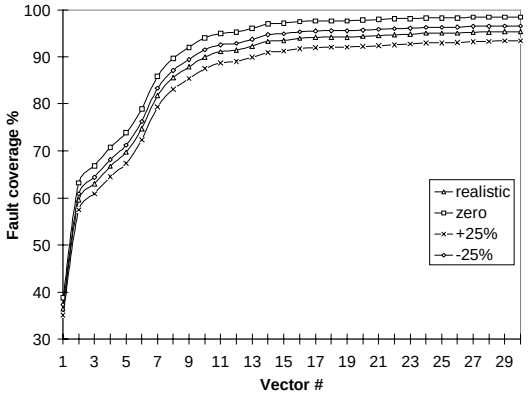

Figure 12. Fault coverage for the c1355 circuit for different distributions.

Circuit	Logic-Testable BFs	BFs detected at 3.3V	BFs detected at 1.2V
c2670	4294	4229	4227
c3540	4431	4375	4374

Table 3. Zero-ohm bridging faults for c2670 and c3540.

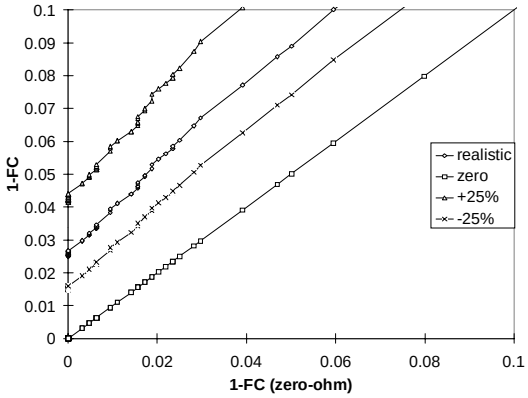

Figure 13. Realistic distribution vs. zero-ohm distribution.

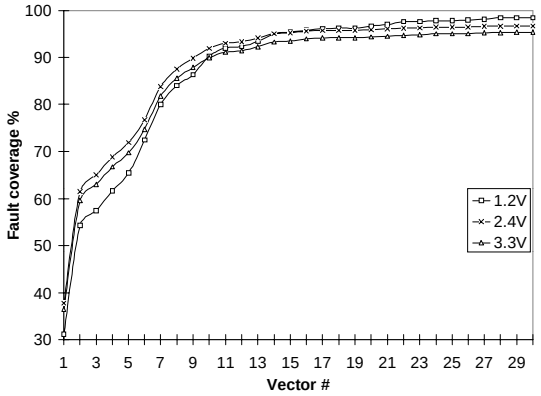

Figure 11. Fault coverage for the c1355 circuit at different V_{DD}.