Accurate and Efficient Fault Simulation of Realistic CMOS Network Breaks

Haluk Konuk F. Joel Ferguson Tracy Larrabee

Computer Engineering Board of Studies, University of California at Santa Cruz

Abstract – We present a new fault simulation algorithm for realistic break faults in the p-networks and n-networks of static CMOS cells. We show that Miller effects can invalidate a test just as charge sharing can, and we present a new charge-based approach that efficiently and accurately predicts the worst case effects of Miller capacitances and charge sharing together. Results on running our fault simulator on ISCAS85 benchmark circuits are provided.

1 INTRODUCTION

Defects that occur during the IC manufacturing process can be categorized into three classes according to Hawkins et al. [7]. These classes are bridge, open circuit, and parametric defects. Open circuit defects cause breaks in the conducting materials in the layout, and contacts are particularly susceptible to such breaks. Breaks can be divided into two categories: those that physically disconnect one or more transistor gates from their drivers, and those that disconnect transistors from each other in the p-network or n-network of a CMOS cell [12]. We define a **network break** as a break fault in the p-network or in the n-network of a cell that breaks one or more transistor paths between the cell output and Vdd or GND. A **transistor path** is a sequence of transistors physically connected through their drain and source terminals. Note that transistor stuck-open faults form a subset of network break faults. Renovell and Cambon [16], and Champac et al. [1] showed that a transistor stuck-open test set can detect some of the breaks that create floating transistor gates. So, a network break test set is useful not only for detecting network breaks but also other breaks that cause floating transistor gates.

Detection of a network break with voltage measurements requires a two-vector test. Reddy et al. [15] showed that transient paths to Vdd or GND can invalidate a two-vector test in transistor stuck-open testing, and Barzilai et al. showed that charge sharing between the internal nodes of the faulty cell and the high impedance faulty cell output can also invalidate a test. Lee and Breuer [11] proposed a scheme for handling charge sharing in transistor stuck-open fault testing using both IDDQ and voltage measurements, but the resulting test sizes might be prohibitive for IDDQ testing. Barzilai et al. [5] described a fault simulator for transistor stuck-open and stuck-on faults. For handling charge sharing, they partitioned all the nodes in every cell into two classes. Nodes in the first class were assumed to have small enough capacitances so that they could be ignored. If a node in the second class can share charge with the floating cell output, then the test is declared invalidated. Di and Jess [3] developed a fault simulator for network breaks, but they ignored static hazards, and their detecting conditions considered charge sharing only with the nodes on the broken paths. Favalli et al. [6] proposed a set of detection conditions for network breaks, but they considered neither transient paths to Vdd or GND, nor charge sharing.

In this paper we present a new charge-based fault simulation algorithm for network breaks that takes into account the transient paths to Vdd or GND, charge sharing, Miller feedback effect, and Miller feedthrough effect. We demonstrate in Section 2 that Miller capacitances can invalidate a two-vector test for a network break just as charge sharing can. To the best of our knowledge, there is no other published work that considers the Miller effect on test invalidation in network break or transistor stuck-open fault simulation.

Because we have a charge-based approach, the nonlinearity of Miller and p-n junction capacitances are accurately modeled compared to previous capacitance-based approaches. In Sections 2.1 and 2.2, we show that a Miller capacitance and a p-n junction capacitance can vary by more than a factor of five and a factor of two, respectively.

We use only six voltage levels to compute the worst case charge differences, as described in Section 3.2, so the charge equations can be precomputed into a look-up table. Section 4 shows that our CPU times are very competitive with previous less accurate fault simulation methods.

2 DETECTION OF NETWORK BREAKS

To guarantee the detection of a network break with voltage measurements, a two-vector test is necessary. Without loss of generality, let us assume that the break is in the p-network. Then, the first vector should initialize the cell output to GND, and the second vector should activate only the broken paths in the p-network and no other path. **Activating a path** means applying ON voltages to the gates of all the transistors on the path. The second vector will make the faulty cell output high impedance with GND as its initial

Work supported by SRC contract 94-DJ-315 and NSF grants MIP-9158490 and MIP-9158491.

32nd ACM/IEEE Design Automation Conference ®

voltage. If the faulty cell output keeps its logic 0 value until the circuit outputs are sampled, and the second vector is a test for the cell output stuck-at-0 fault, then the network break will be detected. If certain mechanisms, which can raise the high-impedance cell output voltage, are not taken into account, a two-vector sequence may be incorrectly classified as a test for the break.

Two mechanisms that may invalidate a test, *transient paths to Vdd or GND* and *charge sharing*, have been studied by many researchers [15, 9, 20, 2, 5, 11, 3]. In this paper, we show that the gate-drain or the gate-source capacitances of the CMOS transistors can modify the voltage of the faulty cell output when it is at high impedance. We refer to these capacitances as **Miller feedthrough** [14] when they are inside the faulty cell, and as **Miller feedback** [14] when they are inside the fanout cells of the faulty cell.

We now introduce some terminology that will be used in the rest of the paper. Let **time-frame 1** or **TF-1** denote the time interval beginning with the application of the first vector and ending with the application of the second vector, and let **time-frame 2** or **TF-2** begin with the application of the second vector and end with the sampling of the circuit outputs. We assume that all the signals in the circuit will be stable by the end of time-frames 1 and 2.

We use an **eleven-value logic algebra** for the logic values of wires in the two time frames. Let ab denote one of the nine values of our logic algebra, where $a, b \in \{0, 1, X\}$, and a and b are the final values of a wire in TF-1 and TF-2, respectively. Thus, 00 on wire l means that the final value of l is 0 in both time frames. Due to multiple paths from circuit inputs to line l, the value on l may temporarily change to 1 and change back to 0 again, which is called a *static hazard*. As the other two values of our eleven-value logic algebra, we use **S0** to represent a 00 with no static hazard, and **S1** to represent a 11 with no static hazard, and refer to them as stable 0 and stable 1 [19], respectively.

In this paper, our emphasis is on how Miller feedback and feedthrough effects, and charge sharing can invalidate a test. We use the circuit in Figure 1 to demonstrate these test invalidation mechanisms. The cell on the left in Figure 1 with a p-network break in it is an OAI31, and the cell on the right is a NOR gate, both from the MCNC cell library. We used HSPICE to simulate this circuit. We used level 13 (the BSIM model) in HSPICE, because this model guarantees charge conservation. We obtained the BSIM model parameters from MOSIS for the 1.2μ Orbit n-well fabrication process. The 35fF capacitance shown in Figure 1 is used to model a metal-1 wire that is around 160μ long in this 1.2μ process.

2.1 Miller Feedback Effect

We now show that the voltage changes on the drain/source terminals of a transistor can significantly change the voltage of its floating gate (Miller feedback effect). We want to emphasize that a Miller feedback capacitance is not only due to the overlap between the gate and diffusion regions of a transistor, but it is also due to the charge stored in the channel region, and it can go up to half of the

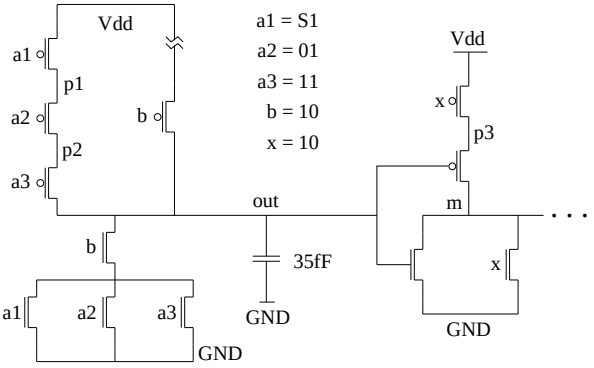

Figure 1: A circuit to demonstrate test invalidation

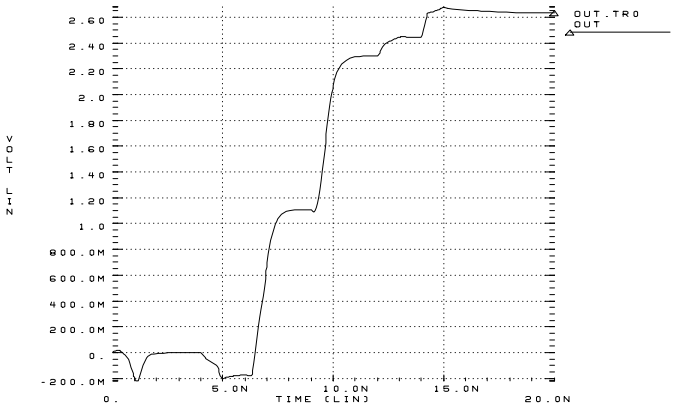

Figure 2: Test invalidation by Miller feedback, charge sharing, and Miller feedthrough

total gate capacitance when the transistor is on. For the pMOS transistor connected to *out* in the NOR gate in Figure 1, the Miller feedback capacitance changes from 4.1fF to 20.8fF according to HSPICE when the transistor gate voltage changes from 5V to 0V with drain and source voltages held at 5V.

Consider the proposed test shown in Figure 1. Table 1 shows the simulated behavior of all the cell inputs in TF-2 and in part of TF-1. We assume that the circuit in Figure 1 is embedded in a larger circuit, and the cell inputs are not the primary inputs. The first transition in TF-2 happens at line b making the OAI31 output floating with a slightly negative initial voltage as shown in Figure 2. The next transition is at x between 6ns and 7ns. Just before this transition, the NOR output m was at 0V, and the internal node $p3$ in the NOR gate was at around 1.2V, which is about the minimum voltage an internal p-diffusion node can acquire in the process we used. After x becomes 0V turning on the pMOS transistor it is connected to, $p3$ and m both rise to around 5V. These rising transitions on $p3$ and m raise the *out* voltage due to Miller feedback to 1.1V from 6ns to 9ns as shown in Figure 2.

In TF-1 x is 0V to first charge up $p3$ to 5V, and then let it drain down to 1.2V when b becomes high impedance.

	Part of TF-1		Time Frame 2									
	initializing $p1$, $p2$, $p3$		out starts floating		Miller feedback		charge sharing		Miller feedthrough			
	0ns	1ns	4ns	5ns	6ns	7ns	9ns	10ns	12ns	13ns	14ns	15ns
x	**0V**	**5V**	5V	5V	**5V**	**0V**	0V	0V	0V	0V	0V	0V
$a1$	**0V**	**5V**	5V	5V	5V	5V	5V	5V	5V	5V	5V	5V
$a2$	0V	0V	0V	0V	0V	0V	0V	0V	**0V**	**5V**	5V	5V
$a3$	5V	5V	5V	5V	5V	5V	**5V**	**0V**	0V	0V	**0V**	**5V**
b	5V	5V	**5V**	**0V**	0V	0V	0V	0V	0V	0V	0V	0V

Table 1: The simulated behavior of the cell input signals in Figure 1

2.2 Charge Sharing

We assume that the next transition in TF-2 is at line $a3$ between 9ns and 10ns due to a glitch. Now, *out* is connected to internal nodes $p1$ and $p2$ in the OAI31 cell. Since $p1$ and $p2$ were initialized to 5V during TF-1 by starting $a1$ at 0V, charge transfer from $p1$ and $p2$ to *out* raises the *out* voltage to 2.3V from 9ns to 12ns as shown in Figure 2. The p-n junction capacitance of node $p2$ changes from 26.7fF to 14.9fF when the voltage at $p2$ changes from 5V to 2.3V. When the voltage at $p2$ drops to 1V, its capacitance drops to 13.2fF.

2.3 Miller Feedthrough Effect

The next event is a rising transition at line $a2$ between 12ns and 13ns. Due to the gate-drain and gate-source (Miller feedthrough) capacitances of the pMOS transistor $a2$ is connected to, this transition raises the voltages on $p1$ and $p2$. The voltage increase on $p2$ enables additional charge transfer from $p2$ to *out* between 12ns and 14ns. The final event is a rising transition at line $a3$ between 14ns and 15ns, which bumps up the *out* voltage to its final value of 2.63V., which is interpreted as logic 1 by the NOR gate. Thus, the test is invalidated.

3 THE FAULT SIMULATION ALGORITHM

Our fault simulation algorithm declares a two-vector sequence a test for a network break if the sequence cannot be invalidated by transient paths to Vdd or GND, Miller feedback and feedthrough effects, and charge sharing. We first perform a gate level simulation using our eleven-value logic algebra. We assume that if a circuit input has the same logic value in time frames 1 and 2, then that input has no static hazard, that is, it is glitch-free. For an AND gate to have an S0 value at its output, at least one of its inputs must be S0, and to have an S1 at its output, all of its inputs must be S1. An OR gate is processed similarly.

In order to guarantee that there will not be any transient path to Vdd for a p-network break, all the paths from the faulty cell output to Vdd in the p-network must have at least one transistor with S1 value at its gate. This is both a necessary and sufficient condition. Similarly, in order to guarantee no transient path to GND for an n-network break, all the paths from the faulty cell output to GND must have at least one transistor with S0 value at its gate.

In order to guarantee that a test will not be invalidated by Miller effects and charge sharing, our fault simulator uses a charge-based approach that computes the worst case charge transfer from Miller and p-n junction capacitances to the floating faulty cell output and vice versa. This approach is described next.

3.1 A Charge-Based Approach

When a test for a network break is applied, the faulty cell output starts floating at some point during TF-2, and stays floating in the rest of TF-2. We refer to this time period as the **floating period**. We assume that TF-2 is short enough so that the transistor leakage currents can be ignored. During the floating period, voltage changes at the gates of the transistors in the faulty cell can displace charge from, or bring in more charge to, the drain and source terminals (Miller feedthrough effect); the output may be connected to some internal nodes in the faulty cell resulting in charge sharing; and voltage changes at the internal nodes of the fanout cells can displace charge from, or bring in more charge to, the gate terminals of the transistors fed by the floating output (Miller feedback effect). Assuming constant values for the Miller and p-n junction capacitances would be too pessimistic or too optimistic, because these capacitances can vary up to a factor of five as shown in Section 2.1. So, our approach is based on computing the worst case changes in electrical charge as a function of the worst case voltage changes at the inputs of the faulty cell and its fanout cells.

Let us now identify the components of the charge stored at the faulty cell output O, and at a faulty cell internal node. Let $\mathbf{I}$ denote the set formed by the faulty cell internal nodes that might be connected to O during the floating period, and $\mathbf{FCN} = I \cup \{O\}$ where FCN stands for the set of Faulty Cell Nodes. The following two components exist for the charge stored on any faulty cell node $fcn \in FCN$.

1. Each transistor drain or source terminal **ds** connected to fcn stores charge in the intrinsic, or channel, area of the transistor **t** when t is on [18]. Some charge is also stored on ds due to the gate overlap capacitance. We denote the charge on ds of t as $\mathbf{Q_{ds,t}}$.
2. Charge is stored in the p-n junction between fcn and the bulk of the transistor, which we denote as $\mathbf{Q_{p-n,fcn}}$.

The following two charge components exist only for the faulty cell output O:

3. Charge is stored on the gate of each fanout transistor **f** connected to O. We denote this charge as $\mathbf{Q_{g,f}}$.
4. Charge is stored on the metal wire that connects the faulty cell to its fanout cells, due to the linear capacitance to GND and to nearby wires. In this paper, we

ignore the interwire capacitances, and consider only the capacitance to GND, which we refer to as the **wiring capacitance**. We denote this charge as $\mathbf{Q_{wiring}}$.

Let us assume for now that the total charge stored at the nodes in FCN at t_{init} is the same as the charge stored at t_{final}, where $\mathbf{t_{init}}$ denotes the beginning of the floating period, and $\mathbf{t_{final}}$ denotes the end of the floating period, which is also the end of TF-2. So, we will assume that charge is conserved during the floating period. We are interested in the worst case charge difference on the wiring capacitance $\mathbf{C_{O,wiring}}$, because this charge difference ΔQ_{wiring} will give us the worst case voltage change on O. Because of charge conservation, any charge difference on the wiring capacitance, which represents only component 4 of the charge stored on O, must come from the charge differences on the remaining three charge components of O and from the charge differences in the nodes of I. Therefore, ΔQ_{wiring} can be expressed as follows.

$$\Delta Q_{wiring} = -\left(\sum_{fcn \in FCN} \Delta Q_{fcn} + \sum_{f \in F} \Delta Q_{g,f}\right) \quad (3.1)$$

$$\Delta Q_{fcn} = \Delta Q_{p-n,fcn} + \sum_{t \in T_{fcn}} \Delta Q_{ds,t} \quad (3.2)$$

where F is the set of transistors whose gates are connected to O, and T_{fcn} is the set of transistors whose drain or source terminals are connected to fcn. The worst case charge differences are determined by the worst case voltage differences from t_{init} to t_{final}. Section 3.2 describes how we obtain these worst case voltages. In Equation 3.2, the $\Delta Q_{p-n,fcn}$ term is for charge sharing between nodes fcn and O, and the summation term is for the Miller feedthrough effect of the transistors in T_{fcn}. In Equation 3.1, the second summation term is for the Miller feedback effect.

Let **L0_th** and **L1_th** denote the maximum voltage that is still a logic 0 and the minimum voltage that is still a logic 1, respectively. If the faulty cell output O is initialized to 0V in TF-1, implying a p-network break, then we assume that O will reach L0_th at the end of TF-2, because $L0\_th$ is the maximum tolerable voltage without test invalidation. Similarly, if O is initialized to Vdd, implying an n-network break, we assume that O will be reduced to L1_th at the end of TF-2. The test becomes invalidated if

$$C_{O,wiring} \cdot L0\_th < \Delta Q_{wiring} \text{ when } O_{init} = GND, \text{ and}$$

$$C_{O,wiring} \cdot (Vdd - L1\_th) < -\Delta Q_{wiring} \text{ when } O_{init} = Vdd$$

Otherwise, the test is declared to be valid if there are no transient paths to Vdd or GND that will invalidate the test.

The following equations, 3.3 through 3.7, are taken from Sheu, Hsu, and Ko [18] to express the charge stored on a transistor gate, denoted by Q_g, and the charge stored by the source and the drain terminals in the channel of a transistor, denoted by Q_d and Q_s. Additionally, we included the sensitivity of model parameters to transistor lengths and widths. These equations are for an nMOS transistor. For a pMOS transistor, the right hand sides of Equations 3.3 to 3.7 need to be negated together with the interterminal voltages.

Subthreshold region, $V_{gs} \leq V_{th}$ and $V_{gb} > zvfb$:

$$Q_g = \frac{cap \cdot zk1^2}{2} \cdot \left(-1 + \sqrt{1 + \frac{4 \cdot (V_{gb} - zvfb)}{zk1^2}}\right) \quad (3.3)$$

$$Q_d = Q_s = 0 \quad (3.4)$$

Triode region, $V_{gs} > V_{th}$ and $V_{ds} \leq V_{DSAT}$:

$$Q_g = cap \cdot (V_{gs} - zvfb - zphi) \quad \text{with } V_{ds} = 0 \quad (3.5)$$

$$Q_d = Q_s = -0.5 \cdot cap \cdot (V_{gs} - V_{th}) \quad \text{with } V_{ds} = 0 \quad (3.6)$$

Saturation region, $V_{gs} > V_{th}$ and $V_{ds} > V_{DSAT}$:

$$Q_g = cap \cdot (V_{gs} - zvfb - zphi - (V_{gs} - V_{th})/(3 \cdot \alpha_x)) \quad (3.7)$$

A term that starts with "z" in the equations above such as $zvfb$ or $zphi$ is a BSIM electrical parameter taking the transistor width and length into account [13]. $cap = C_{ox} \cdot (W - DW) \cdot (L - DL)$ where C_{ox} is the gate-oxide capacitance per unit area, W and L are the drawn transistor width and length, DW and DL are the changes to W and L due to various fabrication steps. For the definitions of V_{th}, α_x, and V_{DSAT}, and for the reason we assumed V_{ds} to be zero in Equations 3.5 and 3.6, please see our technical report [10].

To compute $\Delta Q_{g,f}$ in Equation 3.1 we use Equations 3.3, 3.5, and 3.7 depending on the region the fanout transistor f is in. To compute $\Delta Q_{ds,t}$ in Equation 3.2 we use Equations 3.4 and 3.6 depending again on the region transistor t is in. We also include in $\Delta Q_{g,f}$ and $\Delta Q_{ds,t}$ the charge difference due to the gate-diffusion overlap capacitances.

The reverse biased p-n junction between the diffusion region and the bulk of a transistor forms the capacitance C_{p-n}, whose expression is given in Massobrio and Antognetti [13] as a function of the reverse bias voltage V_r. Integrating C_{p-n} from $V_{r,init}$ to $V_{r,final}$, we obtain the following charge expression for the p-n junction.

$$\Delta Q_{p-n} = \frac{C_{jsw} \cdot P_{diff} \cdot \phi_j}{1 - m_{jsw}} \cdot \left(1 + \frac{V_r}{\phi_j}\right)^{(1-m_{jsw})} \Bigg|_{V_{r,init}}^{V_{r,final}} + \frac{C_j \cdot A_{diff} \cdot \phi_j}{1 - m_j} \cdot \left(1 + \frac{V_r}{\phi_j}\right)^{(1-m_j)} \Bigg|_{V_{r,init}}^{V_{r,final}} \quad (3.8)$$

where A_{diff} and P_{diff} denote the area and the perimeter of the diffusion, and the other new parameters are determined by the fabrication process used. The $\Delta Q_{p-n,fcn}$ term in Equation 3.2 is computed using Equation 3.8 for node fcn.

3.2 Initial and Final Voltages

In this section, we describe how we determine the worst case voltages at transistor terminals at t_{init} and at t_{final} in order to compute ΔQ_{wiring} in Equation 3.1. We use only six voltage levels, which are Vdd, GND, L0_th, L1_th, max_n, and min_p, where **max_n** is the maximum voltage an n-network internal node can achieve through a path to Vdd without any Miller feedthrough effect, and **min_p** is the minimum voltage a p-network internal node can achieve through a path to GND without any Miller feedthrough effect. For the 1.2μ process we used, max_n was around 3.3V, and min_p was around 1.2V with Vdd equal to 5V.

In order to compute $\Delta Q_{ds,t}$ and $\Delta Q_{p-n,fcn}$ in Equation 3.2, we need the gate voltages at t_{init} and at t_{final} for every transistor t connected to fcn, which we denote as $\mathbf{V_{g,t,init}}$ and $\mathbf{V_{g,t,final}}$, and we need the initial and final voltages of fcn, which we denote as $\mathbf{V_{fcn,init}}$ and $\mathbf{V_{fcn,final}}$. Let us assume that node fcn is an internal node in the faulty cell, and not the output node. There are two cases to consider:

CASE 1 : There is at least one path of transistors from fcn to O such that the gates of all these transistors are S0 if fcn is in the p-network, and S1 if fcn is in the n-network. Under this, there are four subcases depending on whether fcn is in the p-network or in the n-network, and whether O is initialized to GND (p-network break) or Vdd (n-network break). Due to lack of space, we only discuss the two subcases where fcn is in the n-network. The other two subcases where fcn is in the p-network are similar.

Subcase 1.1 : Node fcn is in the n-network, and O is initialized to GND. In this case, $V_{fcn,init} = GND$, and $V_{fcn,final} = L0\_th$. Table 2 shows how the worst case $V_{g,t,init}$ and $V_{g,t,final}$ values are determined depending on the logic value at t's gate **gt**.

Logic value at gt	$V_{g,t,init}$	$V_{g,t,final}$
01, 11, 0X, X1, XX, 1X	GND	Vdd
S0, 00, 10, X0	GND	GND
S1	Vdd	Vdd

Table 2: Worst case gate voltages for Subcase 1.1

The non-obvious cases in Table 2 are when the logic values at gt are 11 and 10. When the logic value is 11, it is possible due to a glitch that the voltage at gt is GND at t_{init}. Even when the voltage of gt at t_{init} is Vdd, the following scenario might occur after t_{init}: While O is at GND voltage, a glitch causes a falling transition at gt, which forces the voltage at fcn go below GND, which makes the p-n junction between fcn and the bulk of t forward-biased, because the bulk of an nMOS transistor is connected to GND. This way, positive charge is transferred from t's bulk to node fcn. Note that this charge transfer is happening during the floating period, which will violate our charge conservation assumption of Section 3.1 during the floating period. So, by assuming $V_{g,t,init}$ to be GND, we are effectively moving the beginning of the floating period from t_{init} to the point this charge transfer is completed, this way we can still assume charge conservation. The reason we take $V_{g,t,init}$ to be GND when the logic value at gt is 10 is the same.

In the following cases and subcases, we will omit the reasoning behind our decisions due to lack of space. Our technical report [10] explains the reasons for these decisions.

Subcase 1.2 : fcn is in the n-network, and O is initialized to Vdd. In this case, $V_{fcn,init} = max\_n$. Assume $max\_n \geq L1\_th$, then $V_{fcn,final} = L1\_th$, and Table 3 shows how the worst case $V_{g,t,init}$ and $V_{g,t,final}$ values are determined for transistor t connected to node fcn. The case for $max\_n < L1\_th$ is described in our technical report [10].

CASE 2 : The condition for CASE 1 is not satisfied, and there is no path of transistors from fcn to O such that

Logic value at gt	$V_{g,t,init}$	$V_{g,t,final}$
10, 1X, X0, XX	Vdd	GND
S0, 00, 0X	GND	GND
S1, 11, X1	Vdd	Vdd
01	GND	Vdd

Table 3: The worst case gate initial and final voltages for Subcase 1.2, $max\_n \geq L1\_th$

the gates of all these transistors are S1 if fcn is in the p-network, and S0 if fcn is in the n-network. This case is for intermittent connections between fcn and O during the floating period. As in CASE 1, there are again four subcases. Due to lack of space, we only discuss two subcases where fcn is in the n-network. The other two subcases are similar.

Subcase 2.1 : fcn is in the n-network, and O is initialized to GND. In this case, if fcn is connected to GND at the end of TF-1, then $V_{fcn,init} = GND$, otherwise $V_{fcn,init} = max\_n$. If fcn is connected to O at the end of TF-2, then $V_{fcn,final} = L0\_th$, otherwise $V_{fcn,final} = GND$.

For any transistor t connected to fcn, if the logic value at t's gate gt is S0 or S1, then the initial and final gt voltages are both GND or both Vdd, respectively. Otherwise, we take the initial voltage as GND, and the final voltage as Vdd.

Subcase 2.2 : fcn is in the n-network, and O is initialized to Vdd. If fcn is connected to O at the end of TF-1, then $V_{fcn,init} = max\_n$, otherwise $V_{fcn,init} = GND$. If fcn is connected to O at the end of TF-2, and $L1\_th < max\_n$, then $V_{fcn,final} = L1\_th$, otherwise $V_{fcn,final} = max\_n$.

For any transistor t connected to fcn, if the logic value at t's gate gt is neither S0 nor S1, then we take the initial voltage for gt as Vdd, and the final voltage as GND. When gt's logic value is S0 or S1, then the initial and final gt voltages are both GND or both Vdd, respectively. This completes Subcase 2.2.

When fcn is the same node as O, and O is initialized to GND, we determine the initial and final gate voltages of all the transistors, either in the n-network or p-network, connected to O as shown in Table 2. Obviously, $V_{fcn,init} = GND$ and $V_{fcn,final} = L0\_th$ in this case. The case when O is initialized to Vdd is similar.

In order to estimate the worst case Miller feedback effects, we need to compute $\Delta Q_{g,f}$ in Equation 3.1 for each fanout transistor f of O. For this, the initial and final voltages at all the terminals of f are needed. There are four cases depending on whether f is an nMOS or a pMOS transistor, and whether O is initialized to GND or Vdd. Due to lack of space, we only discuss the two cases where f is an nMOS transistor. The other two cases are similar.

Let $\mathbf{V_{g,f,init}}$ and $\mathbf{V_{g,f,final}}$ denote the initial and final voltages at f's gate. Obviously, $V_{g,f,init} = GND$ and $V_{g,f,final} = L0\_th$ when O is initialized to GND, and $V_{g,f,init} = Vdd$ and $V_{g,f,final} = L1\_th$ when O is initialized to Vdd. Let **ds** denote the drain or the source terminal of f. Let us assume that ds is an internal node, that is, it is neither GND nor the output of cell **fc** in which f is located, then routines *GetNodeInitFinal* and *Get_MFB_InitFinal* in Figure 3 show how we determine the initial and final voltages

```
GetNodeInitFinal(V_ds,init, V_ds,final, static_current_possible) {
  static_current_possible = 1;
  IF(O is initialized to GND)
    IF(there is a path of transistors from ds to Vdd
        such that the gates of all these transistors are S1)
      V_ds,init = V_ds,final = max_n;
    ELSE {
      V_ds,init = GND;
      IF(ds is connected to GND at the end of TF-2)
        V_ds,final = GND;
      ELSE {
        V_ds,final = max_n;
        IF(O is logic-0 at the end of TF-2 OR
            ds is disconnected from O at the end of TF-2)
          static_current_possible = 0;
      }
    }
  ELSE
    IF(there is a path of transistors from ds to GND
        such that the gates of all these transistors are S1)
      V_ds,init = V_ds,final = GND;
    ELSE {
      V_ds,init = max_n;
      IF(ds is connected to Vdd at the end of TF-2)
        V_ds,final = max_n;
      ELSE
        V_ds,final = GND;
    }
} /* End of GetNodeInitFinal */

Get_MFB_InitFinal() {
  GetNodeInitFinal(V_drain,init, V_drain,final, drain_SCP);
  GetNodeInitFinal(V_source,init, V_source,final, source_SCP);
  IF (O is initialized to GND)
    IF (drain_SCP == 0 AND V_source,final == GND)
      V_drain,final = GND;
    ELSE IF (source_SCP == 0 AND
            V_drain,final == GND)
      V_source,final = GND;
} /* End of GetNodeInitFinal */
```

Figure 3: Determining drain/source initial/final voltages for the Miller feedback effect

$\mathbf{V_{ds,init}}$ and $\mathbf{V_{ds,final}}$ for f's drain and source. In the case O is initialized to GND, when O reaches L0_th at the end of TF-2, the nMOS transistor f will be weakly turned on. If the output of fc is sensitized to O, then a static current will be flowing in fc. The flag **static_current_possible** in routine *GetNodeInitFinal* is used to determine when it is impossible for fc's output to be sensitized to O due to the logic values at the side-inputs of fc. When ds is fc's output, then the max_n terms in Figure 3 will be replaced by Vdd.

4 IMPLEMENTATION AND RESULTS

We implemented the fault simulation algorithm described in the previous section. We obtained the BSIM model parameters from MOSIS for the 1.2μ Orbit n-well fabrication process. We extracted the wiring capacitances using *Magic*. We took L0_th to be 1.8V and L1_th to be 3.2V.

For every MCNC standard cell used in the ISCAS85 circuits, we used the public domain *ext2spice* program to determine the area and the perimeter of the diffusion region for the drain and source terminals of each transistor in the cell. We used an inductive fault analysis tool, *Carafe* [8, 17], to get a list of realistic network breaks in the cell.

For each internal node in each faulty cell, our program generates the connection function between the internal node and the faulty cell output, where the **connection function** between two nodes in a cell denotes a sum-of-products expression, where each product term describes the condition to activate a transistor path between the two nodes, and a product term exists for every possible transistor path between the two nodes. Please see our technical report [10] for other connection functions we generate.

The standard cells are processed as described above only once, not every time a circuit is fault simulated. Our program performs parallel pattern simulation using our eleven-value logic algebra to determine the logic value on each wire in time frames 1 and 2 in the fault-free circuit. Then, we perform parallel pattern single fault propagation [4] only in TF-2 to determine the stuck-at-0 and stuck-at-1 detectability of the wires. If a stuck-at-0 on a wire is detectable in TF-2 and the wire is logic-0 in TF-1, then our program checks for possible transient paths to Vdd and computes the ΔQ_{wiring} in Equation 3.1 for the p-network breaks in the cell that drives the wire. The n-network breaks are processed similarly.

We ran our simulator on a DECstation 5000/240 with 128Mb of memory. In the experiments shown in Table 4, we kept generating random patterns until a certain number of successive random patterns do not detect any further network break, and that number is proportional to the number of cells used in the circuit. We call a wire in a circuit **short wire** if its capacitance to GND is less than or equal to 35fF. We chose 35fF arbitrarily mostly because the wiring capacitance we used in Figure 1 was also 35fF. All circuits but c1355 and c6288 have double digit short wire percentages, because all these circuits have XOR or XNOR gates in them, and such a gate consists of two primitive gates with about 10fF wiring between them. Note that it is easier for a test to be invalidated by Miller effects and charge sharing as the wiring capacitance gets smaller. The last two columns give the percentages of network breaks (NBs) covered by random vectors and uncompacted single-stuck-at (SSA) test sets, respectively. The low coverage by SSA vectors hint a need for test generation for network breaks.

Because we use only six voltage levels for our charge difference computations, a look-up table can be constructed for all combinations of these voltages in the charge Equations 3.3 to 3.8. We constructed such a look-up table only for the $(1 + V_r/\phi_j)^{(1-m_j)}$ and $(1 + V_r/\phi_j)^{(1-m_{jsw})}$ terms in Equation 3.8, since taking the power of a real number is computationally expensive. Even though we did not construct look-up tables for other equations, we ended up with reasonable CPU times as shown in Table 4, in fact, our CPU times per vector are always better than the ones by Di and Jess [3], where they used an HP-9000/700.

Table 5 shows our fault coverage results using 1024 random vectors for each circuit. The fault coverage numbers in an "SH off" column are obtained by turning the static hazard identification off, that is, every 00 is treated as S0, and every 11 is treated as S1. In an "SH on" column, static hazard identification is on. In Table 5, "charge off" means that

Ct.	# of NBs	% of short wires	# of random vecs	CPU per vec (ms.)	FC (%)	FC (%) with SSA vecs
c432	931	27.7	4000	3.8	87.8	59.0
c499	1403	44.0	5856	7.3	63.4	56.8
c880	1337	20.6	7360	2.0	94.8	76.7
c1355	2174	4.9	9120	9.4	74.5	61.2
c1908	2235	34.0	22528	9.0	75.5	57.8
c2670	3427	16.7	17920	6.2	78.2	69.5
c3540	4947	17.0	29984	13.1	91.6	67.0
c5315	7607	20.3	70528	15.1	94.0	73.6
c6288	10760	7.9	138624	128.2	87.4	61.5
c7552	9955	23.2	90912	22.3	86.5	70.6

Table 4: Results using random and SSA vectors

Circuit			charge off		charge off
	SH on	SH off	SH on	SH off	paths off
c432	84.0	89.5	88.0	92.6	98.7
c499	60.4	80.8	73.0	90.1	99.5
c880	89.3	90.6	92.4	93.3	98.6
c1355	69.6	83.3	77.6	87.8	96.9
c1908	54.8	63.5	63.6	70.9	86.5
c2670	71.2	76.5	75.1	79.6	85.7
c3540	77.1	85.6	81.7	88.7	96.6
c5315	83.7	91.0	87.6	93.9	98.9
c6288	76.8	96.0	82.8	97.2	99.9
c7552	72.0	80.7	76.9	84.4	89.9

Table 5: Fault coverage results using 1024 random patterns with varying accuracy levels

the computation of ΔQ_{wiring} is turned off, that is, Miller effects and charge sharing are ignored. The "paths off" term means that transient paths to Vdd or GND are ignored. Note that when all of Miller effects, charge sharing, and transient paths are ignored, detection of a network break is only determined by SSA detection in TF-2 and the value of the cell output in TF-1, so static hazards have no relevance. A fault coverage value in this last column might be greater than the SSA coverage of the circuit. For instance, the coverage value for c6288 in the last column is 99.9% while the SSA coverage for this circuit is 99.4%, because most of the undetectable SSA faults in c6288 are on fanout branches, and SSA detectability of fanout branches are not relevant in network break detection.

Comparing the "SH on" with the "SH off" columns shows that static hazard identification makes a quite significant difference. Another observation is that disabling Miller effects and charge sharing has a significant impact on fault coverage even though the impact of disabling transient path identification is bigger.

5 CONCLUSION

The main conclusions from our results have been that Miller effects and charge sharing are significant enough not to be ignored, transient paths are the main cause of test invalidation, static hazard identification is very important, and test generation for network breaks may be necessary to achieve high fault coverage.

Acknowledgments

We thank Don Speck for stimulating discussions on VLSI, Alvin Jee for his help with *Carafe*, and Andy Burstein for his help with the *ext2spice* software.

References

[1] V.H. Champac, A. Rubio, and J. Figueras. Electrical model of the floating gate defect in CMOS IC's: Implications on IDDQ testing. *IEEE Transactions on CAD*, March 1994.

[2] H. Cox and J. Rajski. Stuck-open and transition fault testing in CMOS complex gates. *ITC*, 1988.

[3] C. Di and J.A.G. Jess. On accurate modeling and efficient simulation of CMOS opens. *ITC*, 1993.

[4] J.A. Waicukauski et al. Fault simulation for structured VLSI. *VLSI Systems Design*, Dec. 1985.

[5] Z. Barzilai et al. Efficient fault simulation of CMOS circuits with accurate models. *ITC*, 1986.

[6] M. Favalli, M. Dalpasso, P. Olivo, and B. Ricco. Modeling of broken connections faults in CMOS ICs. *European Design and Test Conference*, 1994.

[7] C.F. Hawkins, J.M. Soden, A.W. Righter, and F.J. Ferguson. Defect classes - an overdue paradigm for CMOS IC testing. *ITC*, 1994.

[8] A. Jee and F. J. Ferguson. Carafe: An inductive fault analysis tool for CMOS VLSI circuits. *VLSI Test Symposium*, 1993.

[9] N.K. Jha and J.A. Abraham. Design of testable CMOS logic circuits under arbitrary delays. *IEEE Transactions on CAD*, July 1985.

[10] H. Konuk and F. J. Ferguson. Accurate and efficient fault simulation of realistic CMOS network breaks. *Technical Report UCSC-CRL-94-40, Comp. Eng. Dept., UC Santa Cruz. URL: ftp://ftp.cse.ucsc.edu/pub/tr/1994/ucsc-crl-94-40.*

[11] K.-J. Lee and M.A. Breuer. On the charge sharing problem in CMOS stuck-open fault testing. *ITC*, 1990.

[12] W.M. Maly, P.K. Nag, and P. Nigh. Testing oriented analysis of CMOS ICs with opens. *ICCAD*, 1988.

[13] G. Massobrio and P. Antognetti. *Semiconductor Device Modeling with SPICE.* McGraw-Hill, 1993.

[14] Meta-Software. *HSPICE User's Manual: Elements and Models.* 1992.

[15] S.M. Reddy, M.K. Reddy, and J.G. Kuhl. On testable design for CMOS logic circuits. *ITC*, 1983.

[16] M. Renovell and G. Cambon. Electrical analysis and modeling of floating-gate fault. *IEEE Transactions on CAD*, Nov. 1992.

[17] J. Rogenski and F. J. Ferguson. Characterization of opens in logic circuits. *IEEE ASIC Conference*, 1994.

[18] B.J. Sheu, W.-J. Hsu, and P.K. Ko. An MOS transistor charge model for VLSI design. *IEEE Transactions on CAD*, April 1988.

[19] B. Underwood, S. Kang, W.-O. Law, and H. Konuk. Fastpath: Robust path-delay test generator for standard scan designs. *ITC*, 1994.

[20] J.-F. Wang, T.-Y. Kuo, and J.-Y. Lee. A new approach to derive robust tests for stuck-open faults in CMOS combinational logic circuits. *DAC*, 1989.