University of California
Santa Cruz

Testing for Opens in Digital CMOS Circuits

A dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Engineering

by

Haluk Konuk

December 1996

The dissertation of Haluk Konuk is approved:

F. Joel Ferguson

Tracy Larrabee

John M. Acken

Dean of Graduate Studies and Research

Contents

List of Figures

List of Tables

Testing for Opens in Digital CMOS Circuits

Haluk Konuk

ABSTRACT

Shorts and opens are the most common types of defects in today's CMOS integrated circuits. This dissertation focuses on opens that occur in transistor drain/source connections and in the interconnect wiring.

Compared to prior research, a very efficient and the most accurate, in terms of taking all test invalidation mechanisms into account, fault simulator for opens in drain/source connections is presented. Results show the individual contributions of different test invalidation mechanisms.

How interconnect opens can cause oscillations and sequential behavior is demonstrated for the first time. Necessary conditions for such behavior are likely to occur in many interconnect opens.

A fault simulation algorithm for interconnect opens, which takes into account all known factors that can affect the behavior of an interconnect open, is presented. The estimated run-time for this algorithm is a constant multiple of the run-time required for stuck-at fault simulation.

Empirical evidence from test chips, which contain various floating-gate transistor structures, shows for the first time that the die surface can become a factor in determining the behavior of a floating wire created by an interconnect open.

This dissertation is dedicated to the memory of my mother, Macide Bedir.

Acknowledgements

I gratefully acknowledge F. Joel Ferguson for his constructive comments during this work, both Tracy Larrabee and F. Joel Ferguson for their continuous financial support, John M. Acken for his useful comments on this dissertation, Doug Sojourner from HP for discussions on opens, Don Speck for answering all my VLSI questions at UCSC, Alvin Jee from SDT for pointing me to semiconductor gas detectors, John Caywood from SDT for his assistance in designing our test chips, Vance Tyree and Jen-I Pi from MOSIS and Les Szepesi from HP for information on VLSI fabrication, Aykut Dengi from Motorola for information on capacitance extraction, anonymous reviewers of my papers for their useful comments, Semiconductor Research Corporation and National Science Foundation for their financial support, California Design Center of Integrated Circuit Business Division at HP for letting me use their computer facilities, and all others who contributed in some way to the completion of this work.

1. Introduction

Defects that occur during the integrated circuit (IC) manufacturing process can be categorized into three classes according to Hawkins, *et al.* [12], which provides a very extensive list of references on this topic. These classes are bridge, open circuit (break), and parametric delay defects. Breaks in the conducting materials of a circuit layout cause unintended open circuits. The terms **opens** and **breaks** are also used to mean the same type of defect in this dissertation. Breaks can be categorized into four types based on their location in a layout.

1. A break can disconnect a set of logic-gate inputs from their drivers; thus causing these gate inputs to float. In order for this to happen a break needs to occur in the interconnect wiring.
2. A break can occur inside a CMOS cell affecting transistor drain and source connections [19, 16, 7, 10, 23].
3. A break inside a CMOS cell can affect the connections between the bulk of an n-channel transistor and GND, or the bulk of a p-channel transistor and Vdd.
4. A break can disconnect a single transistor gate from its driver [4, 32].

Among these types, type 1 is the mostly likely one due to 3, 4, or even 5 layers of metal used for interconnect in modern ICs. Vias are especially susceptible to breaks, and the number of vias exceeds the number of transistors in some microprocessor designs [38]. Due to the large number of contacts connecting transistor drain/source terminals to each other, to logic-gate outputs, and to Vdd and GND busses, type 2 is the second most likely type. This dissertation focuses on break types 1 and 2.

Chapter 2 [19, 16] describes a fault simulation algorithm for type 2, which is also called a **network break**. More precisely, a network break is defined to be a break fault in the p-network or in the n-network of a CMOS cell that breaks one or more transistor paths between the cell output and Vdd or GND. Previous work [31, 14, 42, 6, 2, 21, 7], mostly in the context of transistor stuck-open faults, studied test invalidation due to transient paths to Vdd or GND and due to charge sharing. Chapter 2 shows the importance of Miller feedthrough and feedback capacitances in network break test invalidation, which was ignored by previous work. A new fault simulation algorithm for network breaks is presented with the following novel features: First, the electrical charge coming from Miller and p-n junction capacitances is computed using a transistor charge model [35]; this automatically handles the non-linear nature of transistor capacitances accurately, as opposed to assuming constant capacitance values as was done in previous work. Second, this fault simulation algorithm uses only six voltage levels for charge computations, which allows the use of look-up tables that dramatically reduce the computation time.

Using this simulator to analyze test sets for the ISCAS85 circuits, Chapter 2 shows that the charge coming from Miller capacitances has a larger share in test invalidation than the charge from p-n junction capacitances. This simulator spends less time for charge computations than it spends for transient path identification.

Chapters 3 and 4 together with Appendix A cover various aspects of testing for type 1 breaks, which are also called **interconnect opens**.

Chapter 3 [18] shows that interconnect opens can cause oscillations and can add state to the circuit (sequential behavior). It also shows that the conditions for oscillations and added state are likely to occur in many interconnect opens.

Chapter 4 describes a fault simulation algorithm for interconnect opens taking into account all the known factors that can affect the voltage of a floating wire created by an interconnect open.

Appendix A [17] presents experimental evidence that the die surface can act as an RC interconnect, becoming an important factor in determining the voltage of a floating wire created by an interconnect open. It provides a circuit model for this effect that is verified with HSPICE simulations. Appendix A also provides a detailed analysis of potential mechanisms behind this phenomenon, and it provides measurement results for the trapped charge deposited on floating gates during fabrication.

Finally, Chapter 5 summarizes the contributions of this dissertation.

2. Fault Simulation for Network Breaks

A **network break** is 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 [32], and Champac, *et al.* [4] showed that a transistor stuck-open test set can also detect some of the type 4 breaks, which create single floating transistor gates. This chapter introduces the Miller feedback and feedthrough test invalidation mechanisms and describes a novel charge-based fault simulation algorithm for network breaks, which takes into account transient path, charge sharing, and Miller invalidation mechanisms.

2.1 Introduction

Detection of a network break with voltage measurements requires a two-vector test. Reddy, *et al.* [31] showed that transient paths to Vdd or GND can invalidate a two-vector test in transistor stuck-open testing. Barzilai, *et al.* [2] 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 [21] proposed a scheme for handling charge sharing in transistor stuck-open fault testing using both I_{DDQ} and voltage measurements, but measuring both current and voltage may not be feasible during testing. Barzilai, *et al.* [2] 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. They assumed that nodes in the first class have small enough capacitances so that these nodes could be ignored. If a node in the second class can share charge with the floating cell output, then they declare the test invalid. Di and Jess [7] 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.* [10] proposed a set of detection conditions for network breaks, but they considered neither transient paths to Vdd or GND, nor charge sharing.

The fault simulation algorithm of this chapter takes into account transient paths to Vdd or GND, charge sharing, Miller feedback effect, and Miller feedthrough effect. The following is a list of the major contributions of this chapter that distinguishes this work from previous research.

1. Section 2.2 demonstrates that Miller feedback and Miller feedthrough capacitances can invalidate a two-vector test for a network break just as charge sharing can. Furthermore, the experimental results in Section 2.4 show that Miller capacitances have a much greater effect on test invalidation than the p-n junction capacitances considered by previous work on charge sharing. Section 2.3.1 describes a charge-based approach that considers the worst case effects of Miller capacitances and charge sharing together on test invalidation.
2. Because this is a charge-based approach, the non-linear nature of Miller and p-n junction capacitances are more accurately modeled relative to previous capacitance-based approaches. Section 2.2 shows that a Miller capacitance and a p-n junction capacitance can change by more than a factor of five and a factor of two, respectively. A capacitance-based approach needs to use the worst case capacitance value. Our simulator is less pessimistic by using the correct charge value on a transistor capacitance.
3. Our fault simulator uses only six voltage levels to compute the worst case charge differences, as described in Section 2.3.2, so the charge equations can be precomputed into look-up tables. The experimental results in Section 2.4 show that the look-up table based charge computations take less CPU time than transient path identification. The CPU times per vector are better than previous, less accurate, fault simulation methods.
4. The maximum voltage an internal node in an n-network can acquire is about three-fourths of the Vdd voltage, and the minimum voltage an internal node in a p-network can acquire is about one-fourth of the Vdd voltage, as shown by HSPICE simulations using Orbit 1.2μ, HP 0.8μ, and HP 0.6μ process parameters obtained from MOSIS. The assumption here is that an n- or a p-network does not have special circuitry, such

as a charge pump, to pull its internal node voltages up or down. Previous charge sharing approaches assumed that internal nodes can acquire any voltage from GND to Vdd. Again, our simulator is less pessimistic by using the correct maximum and minimum voltage levels for internal nodes.

5. Our fault simulator identifies static hazards on the circuit wires; thus, it can determine whether a faulty-cell internal node has an intermittent or a stable connection to the cell output during charge sharing. This makes a difference, because the resulting voltage when a group of capacitors are sharing charge at the same time is different from the case where the same group of capacitors connect with each other in a certain sequence but not at the same time. This also makes a difference for the worst case Miller feedthrough effects as shown in Section 2.3.2.

2.2 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. The first vector must initialize the cell output to GND, and the second vector must 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 makes the fault-free cell output voltage equal to Vdd, but the faulty cell output is high impedance—retaining its initial GND 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 is detected. If certain mechanisms, which can raise the high-impedance cell output voltage from GND to a higher value, which might be interpreted as logic 1, are not taken into account, then a two-vector sequence may be incorrectly classified as a test for the break. Those mechanisms are said to potentially invalidate a test.

Two mechanisms that may invalidate a test, *transient paths to Vdd or GND* and *charge sharing*, have been studied in the context of transistor stuck-open faults and CMOS opens by many researchers [31, 14, 42, 6, 2, 21, 7]. This chapter shows that Miller effects due to the gate-drain and gate-source capacitances of the CMOS transistors can modify the voltage of the faulty cell output when it is at high impedance. I refer to these capacitances as **Miller feedthrough** [27] when they are inside the faulty cell, and as **Miller feedback** [27] when they are inside the cells driven by the faulty cell's output. Table 2.1 shows in three different fabrication technologies that Miller and p-n junction capacitances have comparable values. Each Miller capacitance has its minimum value when the transistor is off, and has its maximum value when the transistor is fully on, that is, when the gate voltage is 0V with drain and source at 5V. Each p-n junction capacitance has its minimum value when the reverse bias voltage is 5V, and has its maximum value when the bias voltage is 0V.

	Orbit 1.2μ	HP 1.2μ	HP 0.8μ
Miller cap. (fF)	4.2 - 22.5	4.0 - 17.6	6.2 - 17.7
p-n junc. cap. (fF)	13.5 - 29.8	10.6 - 23.0	10.4 - 20.4

Table 2.1: Miller and p-n junction capacitances computed by HSPICE for a 32λ wide pMOS transistor with 3λ diffusion length

Note that only the p-n junction capacitances in the faulty cell can contribute to test invalidation, whereas Miller capacitances in the fanout cells driven by the faulty cell as well as the Miller capacitances in the faulty cell can contribute to test invalidation. This chapter describes all these test invalidation mechanisms in detail, and it shows how our fault simulator handles them efficiently and accurately using a charge-based, instead of a capacitance-based, approach that solves all of the Miller feedthrough, Miller feedback, and the charge sharing problems together.

Using the path-delay fault testing terminology, let **time-frame 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** begin with the application of the second vector and end with the sampling of the circuit outputs. All the signals in the circuit are assumed to be stable by the end of time-frames 1 and 2.

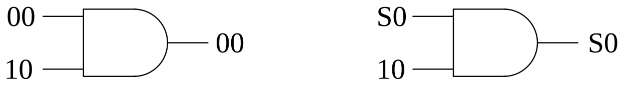

Figure 2.1: An AND gate output with and without a static hazard

Our fault simulator uses an **eleven-value logic algebra** to denote the logic values of wires in the two time frames. Let ab denote one of nine values of this logic algebra, where $a, b \in \{0, 1, X\}$, and a and b are the final values of a wire in time frame 1 and 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* in logic design terminology. The other two values of the eleven-value logic algebra are **S0**, which represents a 00 with no static hazard, and **S1**, which represents a 11 with no static hazard (stable 0 and stable 1 [39], respectively). Figure 2.1 shows two cases for an output assignment of 00 and S0 for an AND gate.

Other researchers studied the effect of transient paths to Vdd or GND on test invalidation extensively [31, 14, 42, 2], here illustrated with an example. Consider the p-network break in Figure 2.2. The cell input assignments shown form a proposed test for this break. Time frame 1 initializes line *out* to 0V, and time frame 2 attempts to charge up *out* to Vdd only through the broken path. In this test, if $a1$ was 11 instead of S1, then $a1$, $a2$, and $a3$ could be logic-0 at the same time momentarily due to glitches on $a1$ and $a3$ after *out* starts floating with b at logic-0. This would momentarily establish a conducting path from Vdd to *out*, and could raise the *out* voltage to a logic-1 value, thus invalidating the test.

In this chapter, the emphasis is on how Miller feedback and feedthrough effects, and charge sharing can invalidate a test. Note that the Miller effect is an entirely different mechanism than charge sharing: Charge sharing is the transfer of charge between two previously isolated electrical nodes. The Miller effect is a result of the charge transfer or charge redistribution from one plate of a capacitor to a plate of another capacitor, where these two plates are connected to the same electrical node.

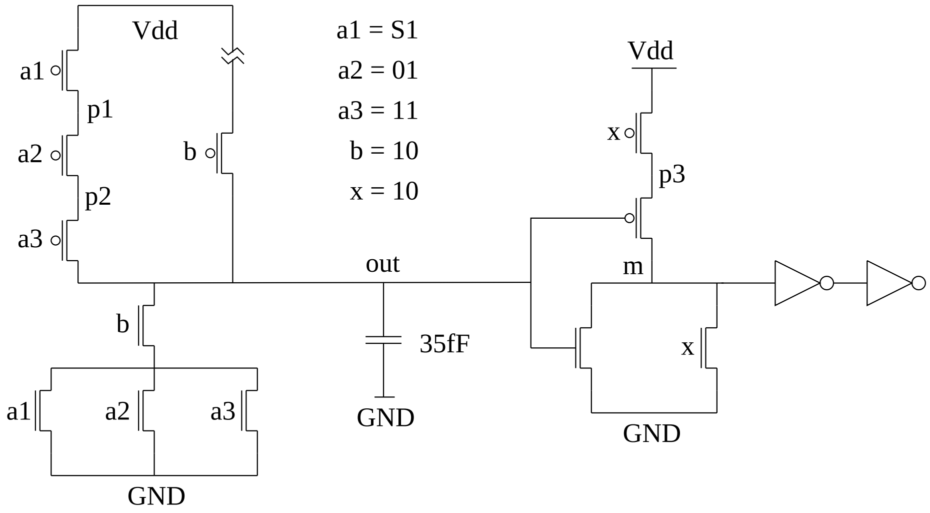

Figure 2.2: The circuit to demonstrate test invalidation for a network break

The circuit in Figure 2.2 demonstrates these test invalidation mechanisms. The cell on the left in Figure 2.2 with a p-network break in it is an OAI31 in the MCNC cell library, and the cell on the right is a 2-input NOR gate, again from the MCNC cell library. I used level 13 (the BSIM model) in HSPICE to simulate this circuit, because this model guarantees charge conservation. I obtained the BSIM model parameters from MOSIS for the 1.2μ Orbit n-well fabrication process. The 35fF capacitance shown in Figure 2.2 is used to model a metal-1 wire that is 160μ long in this 1.2μ process.

2.2.1 Miller Feedback Effect

The voltage changes on the drain/source terminals of the Miller feedback capacitances can significantly change the voltage of a floating node. Note 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 can be up to half of the total gate capacitance when the transistor is on. For the pMOS transistor connected to *out* in the NOR gate in Figure 2.2, 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.

	Part of Time Frame 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 2.2: The simulated behavior of the cell input signals in Figure 2.2

Consider the proposed test shown in Figure 2.2. Table 2.2 shows the simulated behavior of all the cell input signals in time frame 2 and in part of time frame 1, assuming that the circuit in Figure 2.2 is embedded in a larger circuit and the cell inputs are not the primary inputs. The first transition in time frame 2 happens at line b making the OAI31 output floating with a slightly negative initial voltage as shown in Figure 2.3. 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 I 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 2.3.

In time frame 1, x started at 0V in order to first charge up $p3$ to 5V, and then let it drain down to 1.2V at the time b becomes high impedance.

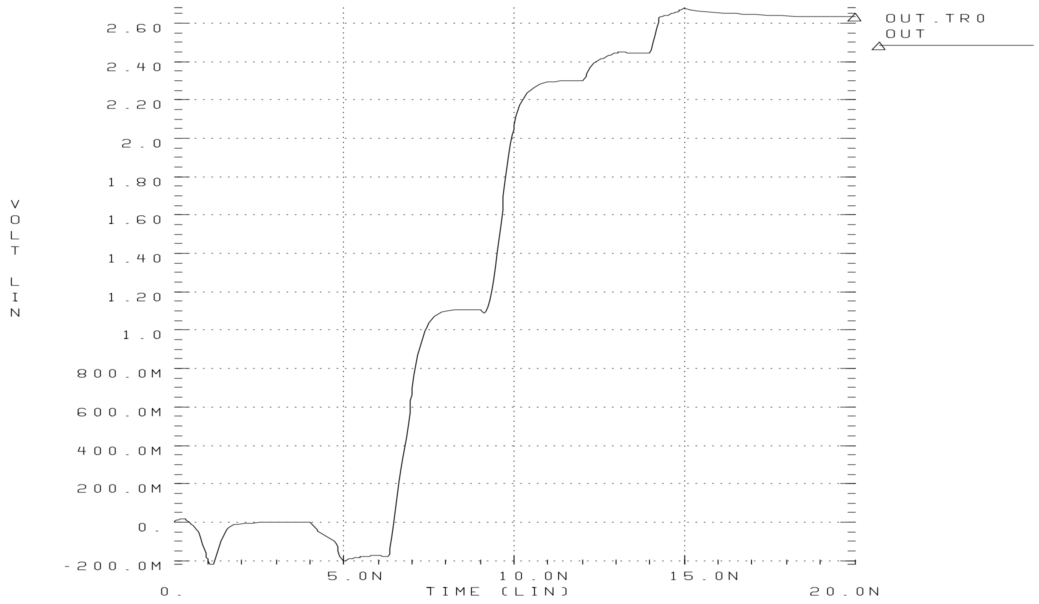

Figure 2.3: Test invalidation by Miller feedback, charge sharing, and Miller feedthrough

2.2.2 Charge Sharing

If in time frame 2 there is a glitch on line $a3$ between 9ns and 10ns, *out* is connected to internal nodes $p1$ and $p2$ in the OAI31 cell. Since $p1$ and $p2$ were initialized to 5V during time frame 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 2.3. 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.

2.2.3 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 $p2$. Please note that the Miller feedthrough capacitance is not only due to the gate-diffusion overlap, but it can go up to half of the total gate capacitance when the transistor is on as in the case of Miller feedback. 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. At this point, the output of the second inverter in Figure 2.2 is a perfect 0V, the same value as in the fault-free circuit, so the test is completely invalidated.

2.3 The Fault Simulation Algorithm

My fault simulation algorithm declares a two-vector sequence to be 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. The first thing to do with a two-vector sequence is to perform gate level simulation using the eleven-value logic algebra. The algorithm is based upon the assumption that if a primary input of the circuit 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 no transient path to Vdd invalidates a test 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 a necessary condition for no transient path, because if a path has no transistor with an S1 at its gate, then that path can be momentarily activated causing current to flow from Vdd to the faulty cell output, making the faulty cell behave like the fault-free one. It is also a sufficient condition, because having at least one pMOS transistor turned off for every possible path in the p-network of the faulty cell throughout time frame 2 guarantees that no current can flow from Vdd to the faulty cell output. 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 difference on the floating faulty cell output. This approach is described in the next section.

2.3.1 A Charge-Based Approach

When a test for a network break is applied, the faulty cell output becomes floating at some point during time frame 2 and stays floating in the rest of time frame 2. I refer to this time period as the **floating period**. I assume that time frame 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 the Miller capacitances can vary up to a factor of five, and the p-n junction capacitances can vary more than a factor of two, as shown in Table 2.1. So, this 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 **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 when the transistor is on [35]. This charge

is a function of the voltages at the terminals of the transistor **t** and the size of t. Some charge is also stored on ds due to the gate overlap capacitance, which is a linear function of the gate-drain or gate-source voltage and the width of t. The charge on ds of t is $\mathbf{Q_{ds,t}}$. This charge is on the diffusion plates of the Miller capacitances in Figure 2.4.

2. Charge is stored in the diffusion regions that make up the transistor terminals connected to fcn, because of the reverse biased p-n junctions between these diffusion regions and the transistor bulks. This charge is a function of the reverse bias voltage and the size of the p-n junctions, denoted as $\mathbf{Q_{junction,fcn}}$. This charge is on the diffusion plate of the p-n junction capacitance in Figure 2.4.

Another component of the charge stored on fcn can be due to a capacitance from fcn to a wire passing over it. The size of this capacitance should be negligible compared to the Miller and p-n junction capacitances. Analyzing such internal nodes in some of the MCNC cells showed that their capacitance to an overhead wire is indeed around 1/100 of the associated Miller and p-n junction capacitances. The following two charge components exist only for the faulty cell output O:

3. Charge is stored on each transistor gate connected to O. This charge is a function of the voltages at the terminals of the fanout transistor **f** and the size of this transistor. This charge is $\mathbf{Q_{g,f}}$, which is on the gate plates of the Miller capacitances in Figure 2.4.

4. Charge is stored on the metal wire that connects the faulty cell to its fanout cells, due to the linear capacitances from this wire to Vdd, to GND, and to nearby wires. The summation of all these capacitances for a wire is the **wiring capacitance**, and the charge on it is $\mathbf{Q_{wiring}}$. Note that voltage changes on the nearby wires during the floating period can affect the voltage on the floating wire. In this chapter, I assume that rising and falling transitions on nearby wires during the floating period cancel each other in the sense that they will not have a net effect on the floating wire voltage.

Let us assume for now that the total charge stored at the nodes in FCN at t_{init} is the

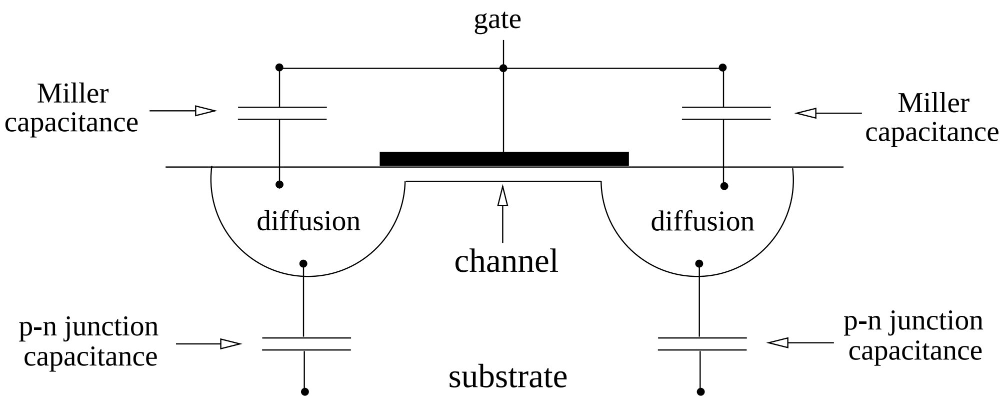

Figure 2.4: Cross section of a CMOS transistor to show charge storage

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 time frame 2. So, I will assume that the net charge difference in the nodes of FCN is zero, that is, charge is conserved during the floating period. I am interested in the worst case charge difference on the wiring capacitance $\mathbf{C_{O,wiring}}$, because this charge difference ΔQ_{wiring} gives us the worst case voltage change on O. Because the net charge difference in the nodes of FCN is zero, any charge difference on the wiring capacitance, which represents only component 4 of the total 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:

$$\Delta Q_{wiring} = -\left(\sum_{fcn \in FCN} \Delta Q_{fcn} + \sum_{f \in F} \Delta Q_{g,f}\right) \tag{2.1}$$

$$\Delta Q_{fcn} = \Delta Q_{junction,fcn} + \sum_{t \in T_{fcn}} \Delta Q_{ds,t}, \tag{2.2}$$

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. Given a circuit, the worst case charge differences are determined only by the worst case voltage differences from t_{init} to t_{final}. Section 2.3.2 describes how to obtain these worst case voltages at t_{init} and at t_{final}

from the elements of the eleven-value logic algebra described in Section 2.2. In Equation 2.2, the $\Delta Q_{junction,fcn}$ 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 2.1, the second summation term is for the Miller feedback effect.

If ΔQ_{wiring} creates a sufficient voltage difference on O, then the test is invalidated. 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 time frame 1, implying a p-network break, then it is assumed that O will reach L0_th at the end of time frame 2, because L0_th is the maximum tolerable voltage without test invalidation. Similarly, if O is initialized to Vdd, implying an n-network break, it is assumed that O will be reduced to L1_th at the end of time frame 2. The test becomes invalidated if

$$C_{O,wiring} * L0\_th < \Delta Q_{wiring} \quad \text{when } O \text{ is initialized to GND, and}$$

$$C_{O,wiring} * (Vdd - L1\_th) < -\Delta Q_{wiring} \quad \text{when } O \text{ is initialized to Vdd.}$$

Otherwise, the test is declared to be valid if there are no transient paths to Vdd or GND that will invalidate the test. Note that a cell in a library will most likely have different logic-0 and logic-1 threshold voltages from another cell in the same library. So, L0_th needs to be the minimum among all the logic-0 thresholds, and L1_th needs to be the maximum among all the logic-1 thresholds. Using individual threshold values for every input of every cell may result in a large number of threshold values, which would make the fault simulation less pessimistic. However, the use of look-up tables as described in Section 2.4 may no longer be feasible.

The following equations, 2.3 through 2.7, are taken from Sheu, Hsu, and Ko [35] 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, the following equations include 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 2.3 to 2.7 need to be negated together with the interterminal voltages.

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

$$Q_g = \frac{cap \cdot zk1^2}{2} \cdot (-1 + \sqrt{1 + \frac{4 \cdot (V_{gb} - zvfb)}{zk1^2}}) \tag{2.3}$$

$$Q_d = Q_s = 0. \tag{2.4}$$

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

$$Q_g = cap \cdot (V_{gs} - zvfb - zphi) \quad \text{with } V_{ds} = 0 \tag{2.5}$$

$$Q_d = Q_s = -0.5 \cdot cap \cdot (V_{gs} - V_{th}) \quad \text{with } V_{ds} = 0. \tag{2.6}$$

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

$$Q_g = cap \cdot (V_{gs} - zvfb - zphi - \frac{V_{gs} - V_{th}}{3 \cdot \alpha_x}). \tag{2.7}$$

The terms V_{th}, α_x, and V_{DSAT} used in the preceding equations are defined as follows [24, 35], but in these definitions the BSIM model parameters $k2$, η, and $U1$ [24, 35] are zero in order to match the definitions in HSPICE [27].

$$V_{th} = zvfb + zphi + zk1 \cdot \sqrt{zphi + V_{sb}}$$

$$\alpha_x = 1 + \frac{g \cdot zk1}{2 \cdot \sqrt{zphi + V_{sb}}}$$

$$g = 1 - \frac{1}{1.744 + 0.8364 \cdot (zphi + V_{sb})}$$

$$V_{DSAT} = \frac{V_{gs} - V_{th}}{\alpha_x}$$

Any term that starts with "z" in the equations above such as $zvfb$ or $zphi$ is a BSIM electrical parameter taking the transistor size into account, computed as follows [24]:

$$zP = P + \frac{P_L}{L - DL} + \frac{P_W}{W - DW}$$

where P is a process parameter such as vfb or phi, P_L and P_W are the length and width sensitivities of parameter P, W and L are the drawn transistor width and length, and DW and DL are the size changes to W and L due to various fabrication steps. The values of P, P_L, P_W, DL, and DW are all determined by the fabrication process. I obtained the values of all the BSIM parameters from MOSIS.

Finally, $cap = C_{ox} \cdot (W - DW) \cdot (L - DL)$ where C_{ox} is the gate-oxide capacitance per unit area.

V_{ds} is assumed to be zero in Equation 2.5, which is used for computing the gate charge of a fanout transistor from the faulty cell. The static current might be non-zero in a fanout cell when O reaches L0_th or L1_th, but this static current will not cause a substantial voltage drop across the drain and source of a transistor in triode region. For example, consider the case when O is initialized to 0V. The final value for O is L0_th; thus the nMOS transistor connected to O in the fanout cell **fc** will be turned on. If the output of fc is sensitized to O, then some static current will be flowing through the nMOS transistor, which is now in saturation region. The output of fc is now at logic-1, because O is at logic-0 even with L0_th voltage on it. Therefore, the pMOS transistor connected to O in fc is in the triode region, and the voltage drop across its channel is about Vdd minus the voltage at fc's output. Since fc's output is at logic-1, this voltage drop can be ignored.

V_{ds} is also assumed to be zero in Equation 2.6, which is used for computing the drain or source part of the channel charge for a transistor in the faulty cell. If this transistor is in the triode region at the beginning or end of the floating period, no drain current will be flowing through this transistor, since there is no conducting path from Vdd to GND. No equation for the drain or source part of the channel charge for a transistor in saturation region is needed, because no transistor in the faulty cell will be in saturation at the boundaries of the floating period.

To compute $\Delta Q_{g,f}$ in Equation 2.1 Equations 2.3, 2.5, and 2.7 can be used, depending on the region the fanout transistor f is in with the initial and final voltages at its terminals. To compute $\Delta Q_{ds,t}$ in Equation 2.2, Equations 2.4 and 2.6 can be used, depending again on the region transistor t is in. I 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 $C_{junction}$. The diffusion region is either the source or the drain of a transistor. From Massobrio and Antognetti [24], $C_{junction}$ can be expressed as a function of the reverse bias voltage V_r as follows:

$$C_{junction} = \frac{C_j \cdot A_{diff}}{(1 + V_r/\phi_j)^{m_j}} + \frac{C_{jsw} \cdot P_{diff}}{(1 + V_r/\phi_j)^{m_{jsw}}},$$

where C_j and C_{jsw} are the capacitances at zero-bias voltage, for unit area and for unit perimeter of the diffusion; m_j and m_{jsw} are the substrate-junction and perimeter capacitance grading coefficient; and ϕ_j is the junction potential. All of these parameters have constant values for the nMOS and pMOS transistors depending on the fabrication process used. Finally, A_{diff} and P_{diff} denote the area and the perimeter of the diffusion. Integrating $C_{junction}$, the charge expression for the p-n junction is as follows:

$$\begin{aligned}\Delta Q_{junction} &= \int_{V_{r,init}}^{V_{r,final}} C_{junction} \cdot dV_r \\ &= \frac{C_j \cdot A_{diff} \cdot \phi_j}{1 - m_j} \cdot \left(1 + \frac{V_r}{\phi_j}\right)^{(1-m_j)} \Bigg|_{V_{r,init}}^{V_{r,final}} + \\ &\quad \frac{C_{jsw} \cdot P_{diff} \cdot \phi_j}{1 - m_{jsw}} \cdot \left(1 + \frac{V_r}{\phi_j}\right)^{(1-m_{jsw})} \Bigg|_{V_{r,init}}^{V_{r,final}} . \qquad (2.8)\end{aligned}$$

The $\Delta Q_{junction,fcn}$ term in Equation 2.2 is computed using Equation 2.8 for node fcn.

2.3.2 Initial and Final Voltages for Charge Computations

This section describes how to determine the worst case voltage values at transistor terminals at t_{init} and at t_{final} in order to compute ΔQ_{wiring} in Equation 2.1. This determination requires only six voltage values as the initial and final voltages of transistor terminals to compute the charge differences given by Equations 2.3 to 2.8. These values are Vdd, GND, L0_th, L1_th, max_n, and min_p, where **max_n** is the maximum voltage an internal node in an n-network can achieve through a path to Vdd without any Miller feedthrough effect, and **min_p** is the minimum voltage an internal node in a p-network can achieve through a path to GND without any Miller feedthrough effect. For the Orbit 1.2μ process, max_n and min_p are approximately 3.3V and 1.2V, respectively, with Vdd equal to 5V. For the HP 0.6μ process, max_n is 2.45V, and min_p is 0.91V with Vdd equal to 3.3V.

The computation of $\Delta Q_{ds,t}$ and $\Delta Q_{junction,fcn}$ in Equation 2.2 requires the gate voltages at t_{init} and at t_{final} for every transistor t connected to fcn, denoted by $\mathbf{V_{g,t,init}}$ and $\mathbf{V_{g,t,final}}$, and the initial and final voltages of fcn, denoted by $\mathbf{V_{fcn,init}}$ 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 three cases to consider, which are briefly described in Table 2.3 together with their subcases. The following paragraphs present a detailed formal analysis of these three cases.

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. This case can be loosely stated as fcn having a constant connection to O. 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). For the sake of brevity, I discuss only 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 2.4 shows the worst case $V_{g,t,init}$ and $V_{g,t,final}$ values for each transistor t connected to node fcn, depending on the logic value at t's gate **gt**. In general, the worst case is when $V_{g,t,init}$ is GND and $V_{g,t,final}$ is Vdd, because the Miller feedthrough effect will increase the voltage at fcn, which has a constant connection to O when below the max_n voltage.

	CASE 1	CASE 2	CASE 3
	fcn has a constant stable connection to O during the floating period.	fcn is guaranteed to be disconnected from O given any time during the floating period.	fcn can have intermittent connections to O during the floating period.
Subcase 1	fcn and break both in n-network.		fcn and break both in n-network.
Subcase 2	fcn in n-network, break in p-network.		fcn in n-network, break in p-network.
Subcase 3	fcn in p-network, break in n-network.		fcn in p-network, break in n-network.
Subcase 4	fcn and break both in p-network.		fcn and break both in p-network.

Table 2.3: Cases for determining the initial and final voltages in a faulty cell.

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 2.4: The worst case gate initial and final voltages for Subcase 1.1

The non-obvious cases in Table 2.4 are when the logic values at gt are 11 and 10. When the logic value is 11, the voltage at gt might be GND at t_{init} due to a glitch. 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 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. In 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 the charge conservation assumption of Section 2.3.1 during the floating period. So, by assuming $V_{g,t,init}$ to be GND, I am effectively moving the beginning of the floating period from t_{init} to the point when this charge transfer is completed; charge conservation still holds. The reason I take $V_{g,t,init}$ to be GND when the logic value at gt is 10 is the same.

Subcase 1.2 : Node fcn is in the n-network, and O is initialized to Vdd. In this case, $V_{fcn,init} = max\_n$. Consider the case where $max\_n \geq L1\_th$. Then, $V_{fcn,final} = L1\_th$, and Table 2.5 shows how the worst case $V_{g,t,init}$ and $V_{g,t,final}$ values are determined for transistor t connected to node fcn, depending on the logic value at t's gate **gt**.

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

When $max\_n < L1\_th$, then $V_{fcn,final} = max\_n$, because max_n is the maximum voltage fcn can acquire while connected to O. Table 2.6 shows the worst case initial and final voltages for the gate of transistor t. The difference between Tables 2.5 and 2.6 is that the initial and final gate voltages for 11 and X1 were both Vdd in Table 2.5, but they changed to Vdd and GND in Table 2.6. The reason is as follows. Due to a glitch during the floating period, gt can make a falling transition absorbing charge from floating O. Because the voltage of O may never go below L1_th during the floating period, and $max\_n < L1\_th$, the charge absorbed may not be transferred back to O when gt rises back to Vdd. Another difference with the case $max\_n < L1\_th$ is as follows: Consider the case when ΔQ_{fcn} in Equation 2.2 comes out to be a negative value, implying that net positive charge will be transferred from fcn to O. For this reason, ΔQ_{fcn} must be conservatively set to zero, because charge transfer from fcn to O is not guaranteed, since O may never go below L1_th during the floating period.

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

Table 2.6: The worst case gate initial and final voltages for Subcase 1.2, $max\_n < L1\_th$

CASE 2 : All the transistor paths from fcn to O have at least one transistor with its gate at S1 value if fcn is in the p-network, and at S0 value if fcn is in the n-network. This case is for when fcn is disconnected from O during the whole floating period; therefore, this fcn does not play any role in disturbing or helping O keep its initial charge.

CASE 3 : The conditions for CASE 1 and CASE 2 are not satisfied. This case is for intermittent connections between fcn and O during the floating period. As in CASE 1, 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 or Vdd. For the sake of brevity, I discuss only two subcases, where fcn is in the n-network. The other two subcases where fcn is in the p-network are similar.

Subcase 3.1 : Node 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 time frame 1, then $V_{fcn,init} = GND$, otherwise $V_{fcn,init} = max\_n$. Note that if the p-network break disconnects the whole p-network from O, only the Miller feedthrough and feedback mechanisms can create an initial voltage of max_n at fcn; therefore, max_n is a pretty pessimistic initial voltage, but not impossible. If fcn is connected to O at the end of time frame 2, then $V_{fcn,final} = L0\_th$, otherwise $V_{fcn,final} = GND$ because even when $V_{fcn,init} = max\_n$, fcn might be connected to O

while O is still at GND voltage, and this may pull down the fcn voltage very close to GND because the total capacitance of O might be much larger than the capacitance of fcn, and fcn may never connect to O again in the rest of time frame 2.

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, I take the initial voltage for gt as GND, and the final voltage as Vdd. This might sound counter-intuitive for the case when 10 is the logic value for gt, but consider the following scenario during the floating period. While the voltage of fcn is GND, a falling transition arrives at gt. This will bring in more charge to the drain or source terminal (whichever is connected to fcn) of t. But, this charge will be coming from the bulk of transistor t due to the forward biased p-n junction between fcn and the bulk. In order to make the charge conservation assumption made in Section 2.3.1 hold, I treat even a 10 at gt as 01, because a 10 can create a rising transition between two falling transitions, and the falling transitions may cause charge transfer from the bulk. Repetitive falling and rising transitions at gt coupled with connections of fcn to O at appropriate times can create an effect of pumping charge from the bulk to node O. But, I ignore this seemingly unlikely effect and leave its detailed discussion to future research. In fact, a similar phenomenon can also happen in Subcase 1.1.

Subcase 3.2 : Node fcn is in the n-network, and O is initialized to Vdd. If fcn is connected to O at the end of time frame 1, then $V_{fcn,init} = max\_n$, otherwise $V_{fcn,init} = GND$. If fcn is connected to O at the end of time frame 2, and $L1\_th < max\_n$, then $V_{fcn,final} = L1\_th$, otherwise $V_{fcn,final} = max\_n$. If fcn is disconnected from O at the end of time frame 2, the actual fcn voltage might be larger than max_n due to Miller feedthrough effect around fcn, but when the fcn voltage exceeds max_n, charge cannot be transferred from O to fcn.

For any transistor t connected to fcn, if the logic value at t's gate gt is neither S0 nor S1, then I take the initial voltage for gt as Vdd, and the final voltage as GND. This is the case even when gt's logic value is 01, because a 01 can create a falling transition between two rising transitions, and during the falling transition the voltage at fcn may be max_n or lower; thus, enabling charge transfer from O to the drain or source (whichever is connected to fcn) of t. However, during the rising transitions the voltage at fcn may be max_n or higher thus preventing charge transfer onto O. 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 3.2.

□

So far in Section 2.3.2, the presentation assumed that fcn was an internal node. How can the initial and final voltages be determined when fcn is the same as node O? When O is initialized to GND, Table 2.4 shows how to determine the initial and final gate voltages of all the transistors, either in the n-network or p-network, connected to O. 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, $\Delta Q_{g,f}$ in Equation 2.1 must be computed for each fanout transistor f of O. For this, the initial and final voltages at the gate, drain, and source terminals of f are needed. There are four cases to consider depending on whether f is an nMOS or a pMOS transistor, and whether O is initialized to GND or Vdd. I discuss only two cases, where f is an nMOS transistor. The other two cases, where f is a pMOS transistor, 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 Figures 2.5 and 2.6 show how to determine the initial and final voltages $\mathbf{V_{ds,init}}$ and $\mathbf{V_{ds,final}}$ for f's drain and source. In the case where O is initialized to GND, when O reaches L0_th at the end of time frame 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

```
GetNodeInitFinal( $V_{ds,init}$, $V_{ds,final}$, static_current_possible )
BEGIN
    static_current_possible = TRUE;
    IF ($O$ is initialized to GND) THEN
        IF (there is at least one path of transistors from $ds$ to Vdd such that
                the gates of all these transistors are S1) THEN
            $V_{ds,init}$ = max_n; $V_{ds,final}$ = max_n;
        ELSE
            $V_{ds,init}$ = GND;
            IF ($ds$ is connected to GND at the end of time frame 2) THEN
                $V_{ds,final}$ = GND;
            ELSE
                $V_{ds,final}$ = max_n;
                IF ($ds$ is disconnected from the cell output at the end of time frame 2 OR
                        the cell output is logic-0 at the end of time frame 2) THEN
                    static_current_possible = FALSE;
                ENDIF
    ELSE
        /* $O$ is initialized to Vdd */
        IF (there is at least one path of transistors from $ds$ to GND such that
                the gates of all these transistors are S1) THEN
            $V_{ds,init}$ = GND; $V_{ds,final}$ = GND;
        ELSE
            $V_{ds,init}$ = max_n;
            IF ($ds$ is connected to Vdd at the end of time frame 2) THEN
                $V_{ds,final}$ = max_n;
            ELSE
                $V_{ds,final}$ = GND;
END
```

Figure 2.5: Determining the initial and final voltages of a drain or source node in a fanout cell

```
Get_MFB_InitFinal()
BEGIN
    GetNodeInitFinal( $V_{drain,init}$, $V_{drain,final}$, drain_SCP );
    GetNodeInitFinal( $V_{source,init}$, $V_{source,final}$, source_SCP );
    IF ($O$ is initialized to GND) THEN
        IF (drain_SCP == FALSE AND $V_{source,final}$ == GND ) THEN
            $V_{drain,final}$ = GND;
        ELSE IF ( source_SCP == FALSE AND $V_{drain,final}$ == GND ) THEN
            $V_{source,final}$ = GND;
END
```

Figure 2.6: Determining the initial and final voltages for the Miller feedback effect

flowing in fc as discussed in Section 2.3.1. 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 2.5 will be replaced by Vdd.

2.4 Implementation and Experimental Results

I implemented the fault simulation algorithm described in the previous section on top of the *Nemesis* single-stuck-at fault simulator [20]. I used the ISCAS85 benchmark circuits implemented with the MCNC standard cell library for my experiments. For charge difference computations, I used the BSIM model parameters for the HP 0.6μ n-well fabrication process. I extracted the wiring capacitance of each wire in a circuit using *Magic* with this 0.6μ technology. Using iterative HSPICE simulations, I computed L0_th to be 1.05V and L1_th to be 1.90V, where Vdd is 3.3V.

For every standard cell used in the ISCAS85 benchmark circuits, I performed the following tasks: I 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. I used an inductive fault analysis tool, *Carafe* [13, 33], to get a list of realistic break faults in the cell, and I eliminated the breaks that are not network breaks.

For each internal node in each faulty cell, the simulator 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. This function is used in determining the initial and final voltages in the faulty cell as described in Section 2.3.2. The simulator first generates the described connection function for each internal node of the fault-free cell. For every faulty cell produced from this fault-free cell with a network break, the simulator lists the faulty cell internal nodes that are identical to the ones in the fault-free cell. Then, the simulator lists the new internal nodes with their connection functions. This method saves memory by generating a connection function only for a new internal node in a faulty cell.

Again for each faulty cell, the simulator generates the connection function between the cell output and either Vdd or GND depending on whether the break is in the p-network or in the n-network. This function is used to determine whether the faulty cell output will float in time frame 2, and whether a transient path to Vdd or GND is possible to invalidate a test.

For each internal node in a fault-free cell, the simulator generates the connection function to the Vdd or GND node depending on whether the internal node is in the p-network or in the n-network. This function together with the connection function to the cell output is used in determining the initial and final voltages for Miller feedback effect as described in Section 2.3.2.

The standard cells are processed as described above only once, not every time before a circuit is fault simulated. Our program performs parallel pattern simulation using the eleven-value logic algebra to determine the logic value on each wire in time frames 1 and 2 in the fault-free circuit. Then, PPSFP (parallel pattern single fault propagation) [41] simulation is performed only in time frame 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 time frame 2 and the wire is logic-0 in time frame 1, then the simulator checks for possible transient paths to Vdd and computes the ΔQ_{wiring} in Equation 2.1 for the p-network breaks in the cell that drives the wire. The n-network breaks are processed similarly.

Even though only the c432 and the c499 have XOR or XNOR gates in their gate level descriptions among all the ISCAS85 benchmark circuits, when these circuits are technology mapped using the MCNC cell library, all the circuits but the c1355 and the c6288 end up having XOR or XNOR gates in their implementations. An XOR gate is implemented using a NOR gate and an AOI21 gate, and an XNOR gate is implemented using a NAND gate and an OAI21 gate in the MCNC library. Figure 2.7 shows an XOR gate with two n-network breaks in it. In the layout of this gate, transistors $T1$ and $T2$ share a diffusion contact to connect to the GND terminal. A break in this diffusion contact causes the two network breaks shown in Figure 2.7. Because of the single break assumption of the fault simulation algorithm described in previous sections, I handle this case as follows: One possible solution is to exercise the AOI21 gate in a fault-free manner so that the network break affects only the NOR gate. The only two-vector sequence that might detect the NOR gate network break is $a = S0$ and $b = 01$. But, this sequence activates the broken path in the AOI21 gate in both time frames 1 and 2, therefore this sequence is not a valid test. The other solution is to exercise the NOR gate in a fault-free manner so that the network break affects only the AOI21 gate. In this case, $a = 10$ and $b = S0$ is the only potential test. This test detects the break fault if the XOR output is observable in time frame 2, and the wire driven by the XOR gate is big enough to handle Miller effects. Two simultaneous breaks in the p-networks of an XOR gate, and two simultaneous network breaks in an XNOR gate are treated similarly.

Because only six voltage levels are used for the charge difference computations, look-up tables can be constructed for all possible combinations of these voltages and different transistor widths used in the cell library. I used fifteen entries per transistor width, with

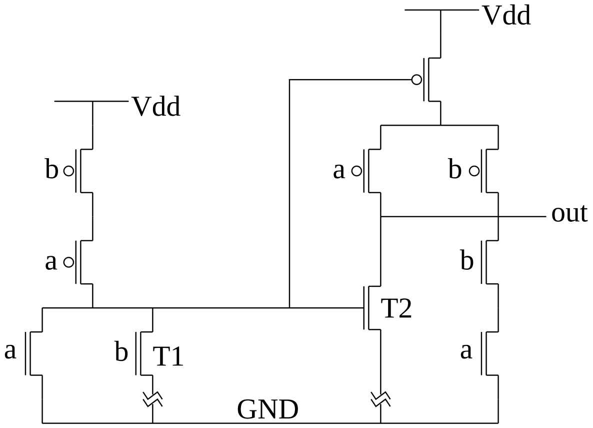

Figure 2.7: Two network breaks in an XOR gate caused by a single contact break in the layout

a total of forty different nMOS and pMOS transistor widths used in the ISCAS85 circuits. Each entry corresponds to a particular value of Equation 2.3, 2.5, 2.6, or 2.7 for a given transistor width and type. These fifteen values cover all possible cases for these equations. Also, the $(1+V_r/\phi_j)^{(1-m_j)}$ and $(1+V_r/\phi_j)^{(1-m_{jsw})}$ terms in Equation 2.8 need five entries each for an nMOS transistor, and five entries each for a pMOS transistor, because an n-network node can take any of the six voltage levels except min_p, and a p-network node can take any of the six voltages except max_n as its initial or final value as explained in Section 2.3.2. This total of twenty entries save me taking the powers of real numbers, which is a computationally expensive operation. Therefore, the total size of the look-up tables is $15 * 40 + 20 = 620$ floating point values, which is a very low memory overhead. I ran the fault simulator with the ISCAS85 benchmark circuits on a DECstation 5000/240 with 128Mb of memory.

Table 2.7 shows the results using uncompacted single-stuck-at (SSA) test sets. A two-vector pattern is formed by using two successive vectors $v1$ and $v2$, and the next two-vector pattern is formed by using $v2$ and $v3$, where $v3$ is the vector following $v2$. I call a wire in a circuit a **short wire** if its wiring capacitance, as I defined in Section 2.3.1, is less than or equal to 15fF. I chose 15fF arbitrarily, because the 35fF wiring capacitance I used in

Figure 2.2 corresponds to 15fF in the HP 0.6μ technology. All circuits but the c1355 and the 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 4fF wiring between them. In Table 2.7 "TP" means transient paths, and "SH" means static hazards. Column 4 gives the fault coverage with both transient paths and charge from Miller and p-n junction capacitances ignored. A value in this column might be greater than the SSA coverage of the circuit. For instance, the value for the c6288 in column 4 is 99.8% while the SSA coverage for this circuit is 99.4%, because most of the undetectable SSA faults in the c6288 are on fanout branches, and the SSA detectability of fanout branches are not relevant in network break detection; only the SSA detectability of fanout stems are important. Column 5 includes only transient paths, and column 6 includes both transient paths and charge for test invalidation.

In Table 2.7, the difference between columns 5 and 6 shows the test invalidation effects of charge from Miller and p-n junction capacitances. Note that it is easier for a test to be invalidated by this charge as the wiring capacitance gets smaller. Circuit c6288 has 9.9% short wires, whereas the c1908 has 35.4% short wires. But, the decrease in coverage due to charge for the c6288 is 18.4 percentage points, while this decrease is only 12.0 points for c1908. This shows that other factors in a circuit in addition to wiring capacitance sizes, such as the number of reconvergent fanouts, types of cells used, etc., can also significantly affect the fault coverage. The low coverage values in column 6 suggest a need for test generation for network breaks.

The last column in Table 2.7 gives network break coverage values with static hazards ignored, but other causes of transient paths, and charge included. Ignoring static hazards during fault simulation means that every 00 is treated as S0, and every 11 is treated as S1. The coverage values jumped up significantly compared to the preceding column, showing how important the static hazard identification is.

Table 2.8 shows the fault coverage results using random vectors. For each circuit, the number of random vectors is ten times the circuit number. For instance, 19080 random

Ct.	# of network breaks	% of short wires	Fault coverage (%) with SSA tests			
			no TP no charge	TP and no charge	TP and charge	SH ignored. TP and charge
c432	931	33.5	91.2	68.1	55.5	69.7
c499	1403	40.3	99.0	70.2	55.0	70.9
c880	1337	22.6	96.6	83.1	72.9	79.2
c1355	2174	6.6	93.2	71.8	56.6	71.9
c1908	2235	35.4	92.3	67.7	55.7	66.5
c2670	3427	19.4	94.8	75.7	66.1	76.0
c3540	4947	17.0	95.8	72.9	64.8	77.4
c5315	7607	22.0	97.3	78.7	71.3	81.8
c6288	10760	9.9	99.8	71.9	53.5	82.4
c7552	9955	22.4	96.4	77.1	68.3	80.9

Table 2.7: Results for ISCAS85 circuits using single-stuck-at test vectors

vectors are simulated for circuit c1908. Transient paths are included from the third through the seventh columns. The fourth column includes only the charge from p-n junction capacitances. The decrease in fault coverage is very small compared to the "no charge" case. The fifth column, labeled as "Miller", includes only the charge from the Miller capacitances. The decrease in fault coverage is significant compared to the "no charge" case. **This shows that Miller capacitances have a much greater effect on test invalidation than p-n junction capacitances have**. Two reasons are as follows: *(i)* Miller capacitances in the fanout cells connected to the faulty cell output as well as the Miller capacitances in the faulty cell can contribute to test invalidation, while the p-n junction capacitances only in the faulty cell can affect test invalidation, and *(ii)* while one terminal of a p-n junction capacitance is always fixed at either Vdd or GND, both terminals of a Miller capacitance can change their voltages.

The sixth column shows the results of the full fault simulation including both Miller

		Transient paths included				1.2μ, TP,
	no TP	no	p-n		Miller and	Miller and
Ct.	no charge	charge	junc.	Miller	p-n junc.	p-n junc.
c432	99.7	91.5	91.0	83.9	84.6	84.8
c499	100.0	75.6	75.6	60.9	62.6	62.7
c880	100.0	97.6	97.5	92.6	93.0	93.2
c1355	100.0	82.2	82.1	65.0	69.4	69.7
c1908	100.0	82.6	82.5	71.1	71.9	72.3
c2670	86.9	81.3	81.3	76.5	76.4	76.9
c3540	98.8	94.1	94.0	90.3	90.7	90.9
c5315	100.0	96.5	96.5	92.2	92.4	92.4
c6288	99.9	89.5	89.4	79.9	80.6	80.8
c7552	95.2	90.2	90.2	84.3	84.4	84.4

Table 2.8: Fault coverage results using random vectors

and p-n junction charges. Fault coverage slightly increased compared to the fifth column where only Miller charge was included. The apparent explanation for this is that the charge difference on the p-n junction capacitances is in many cases in the direction of helping the faulty cell output retain its initial charge, instead of disturbing it. This happens, for instance, when an n-network node fcn in the faulty cell has a stable connection to the cell output through a path of transistors with their gates at S1 value. Assuming that the break is in the p-network, the cell output will be initialized to logic-0 in time frame 1. When a positive amount of charge ΔQ is transferred onto the cell output to increase its voltage, part of this ΔQ will be taken by fcn, because it has a stable connection to the cell output effectively increasing its capacitance, thus helping the cell output retain its initial charge.

The last column in Table 2.8 lists the coverage values using the Orbit 1.2μ technology available through MOSIS, using the full fault simulator as for the preceding column. For each signal in the ISCAS85 circuits, I computed the ratio of that signal's wiring capacitance in the 1.2μ technology to its wiring capacitance in the 0.6μ technology. The average ratio over all the signals was 2.35. The gate-oxide thickness in the 1.2μ technology was 264 Angstroms, whereas it was 100 Angstroms in the 0.6μ technology. The charge on a Miller capacitance, given by Equations 2.3 through 2.7, is proportional to $cap = C_{ox} \cdot (W - DW) \cdot (L - DL)$, where C_{ox} is the gate-oxide capacitance per unit area, W and L are the drawn transistor width and length, and DW and DL are the size changes to W and L due to various fabrication steps. Assuming DW and DL to be zero for a rough calculation, Table 2.9 shows the changes in a Miller and a wiring capacitance going from the 1.2μ process to the 0.6μ process. The ratio of a Miller capacitance to a wiring capacitance grows going from 1.2μ to 0.6μ because of the reduction in the gate-oxide thickness. Note that if the gate-oxide thickness remained the same, then the entry in Table 2.9 for the charge on a Miller capacitance in 0.6μ would be 1 unit instead of 2.64 units. If I used the HP 1.2μ process parameters instead of Orbit's, then this entry would be 2.37, which is still much larger than 1.70.

	Orbit 1.2μ process	HP 0.6μ process
Charge on a Miller cap.	4 units	2.64 units
Charge on a wiring cap.	4 units	1.70 units

Table 2.9: The change in wiring and Miller capacitances with the process

This increase in the relative importance of Miller capacitances going from the Orbit 1.2μ to HP 0.6μ explains why I got slightly better coverage numbers in the last column of Table 2.8.

Table 2.10 shows the CPU times using 1024 random vectors for each circuit. Taking the fact that 2.6 to 3.9 times more network breaks per circuit were simulated, the CPU times per vector are better than the ones reported by Di and Jess [7], where they used an HP-9000/700. Moreover, Di and Jess [7] ignored static hazards, ignored Miller effects, and assumed constant capacitances for internal nodes of a cell. The total time *Carafe* took for break fault extraction for the whole cell library was less than 20 seconds. Note that *Carafe*

Circuit	no TP no charge	TP but no charge	both TP and charge
c432	2.5	5.8	7.9
c499	2.6	6.0	11.0
c880	2.9	3.9	5.5
c1355	3.5	9.2	16.3
c1908	5.4	9.4	16.4
c2670	5.7	9.4	12.6
c3540	10.7	35.7	43.6
c5315	10.1	23.6	32.0
c6288	30.9	221.3	357.0
c7552	13.5	28.4	41.6
Total	87.8 sec.	352.7 sec.	543.9 sec.

Table 2.10: The CPU times in seconds using 1024 random vectors

does not need to be run for every circuit, but once for each cell in the library. Table 2.10 shows that in all the circuits larger than c1908, the CPU time necessary to compute the charge from Miller and p-n junction capacitances is less than the time necessary to identify the transient paths. When the total times from the three CPU time columns are compared, again the charge computation time is less than the transient path identification time.

2.5 Conclusions

The main conclusion from this work is that Miller capacitances play a significant role in test invalidation as demonstrated by Table 2.8 and by the example used to plot Figure 2.3. In fact, Miller capacitances, which until now were not considered as a source of test invalidation, are much more important than charge sharing with p-n junction capacitances. Another important conclusion is that a very accurate fault simulator for network breaks that takes into account transient paths, and Miller and p-n junction capacitances is feasible.

Even though the transient paths to Vdd/GND form the most important test invalidation mechanism as shown by Tables 2.7 and 2.8, Miller and p-n junction capacitances are also important when a significant number of interconnect wires have capacitances that are comparable to these transistor capacitances. The interconnect capacitances are comparable when the wires are relatively short. Even though the interconnect capacitances are not shrinking as fast as the transistor capacitances are shrinking as feature sizes decrease, transistor capacitances can still not be ignored. This is especially true when there are logic blocks in the cell library that are made up of primitive cells packed together tightly using short interconnecting wires. One simple example is an XOR, or an XNOR gate. Careful placement and routing can keep the percentage of short wires used in the interconnect at a substantial level even when there are very long wires in the layout. Finally, the gate-oxide thickness is shrinking as the fabrication technology advances, which has an increasing effect on the Miller capacitances.

3. Oscillation and Sequential Behavior Caused by Interconnect Opens

The remainder of this dissertation deals with interconnect opens. This chapter shows that an interconnect open can create capacitive feedback paths in a CMOS circuit; thus, causing oscillation and sequential behavior. It also shows under what conditions this previously unreported phenomenon will occur.

Capacitive coupling as low as 1 femto-farad between signal lines can activate the feedback path as will be demonstrated in Section 3.1.1. Knowing the cause and necessary conditions for oscillation and added state due to interconnect opens (1) provides the limits to simpler models of interconnect opens, (2) may lead to a more accurate fault grading of interconnect opens, (3) helps in the development of a more effective testing strategy, and (4) may allow this phenomenon to eventually be considered in test pattern generation.

As evidence for sequential behavior occurring in real life defective ICs, Franco, *et al.* [8] observed that 14 out of 128 defective chips showed sequential behavior in their experiments with a test chip. However, they do not report on diagnosing the actual defect causing this behavior.

This chapter also discusses, for comparison purposes, the conditions under which a feedback bridging fault will cause oscillatory or sequential behavior. Finally, three important factors are discussed, that can play a role in the behavior of an interconnect open, which are trapped charge, die surface, and charge collector diodes.

3.1 Oscillations due to Interconnect Opens

This section shows how an interconnect open can cause oscillation. The two types of feedback capacitances responsible for oscillation are wire-to-wire and Miller capacitances to the floating node created by the open.

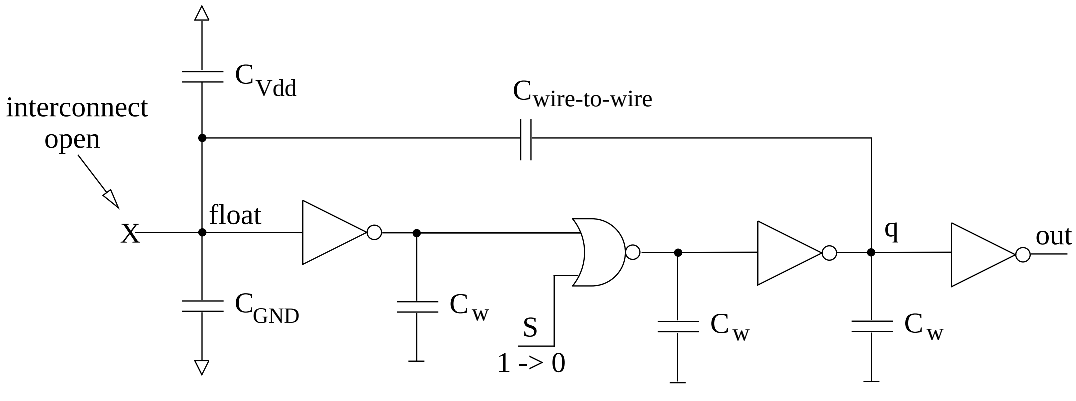

Figure 3.1: Circuit to demonstrate oscillation due to a wire-to-wire capacitance

3.1.1 Feedback via Wire-to-Wire Capacitance

Consider the circuit in Figure 3.1. Node *float* is floating due to an interconnect open defect. $\mathbf{C_{Vdd}}$ represents the total capacitance between *float* and all neighboring nodes that are at the Vdd voltage. These nodes include other signal wires at Vdd, power wires carrying Vdd, and n-wells that are tied to Vdd. Similarly, $\mathbf{C_{GND}}$ represents the total capacitance between *float* and all neighboring nodes that are at the GND voltage. These nodes include signal and power wires at GND, and the p-substrate. The capacitance across the gap created by the interconnect break is included either in C_{Vdd} or in C_{GND} depending on the logic value of node *float* in the fault-free circuit. $\mathbf{C_{wire\_to\_wire}}$ represents the feedback capacitance between the wires q and *float*, and finally C_W represents the total capacitance for the wire it is attached to in Figure 3.1.

Let us now assign some values to these capacitances. To obtain realistic values, the numbers in the MAGIC technology file available from MOSIS for the HP 0.6μ fabrication process is used. I used 20fF for each C_W, which corresponds to a 153μ long minimum width metal-1 wire over substrate. For comparison, the cell height in the MCNC cell library is 17.4μ in this 0.6μ process. I used 12fF for C_{GND}, 8fF for C_{Vdd}, and 1fF for $C_{wire\_to\_wire}$. A 1fF capacitance between two parallel metal-1 wires separated by 0.9μ corresponds to 24μ of metal-1 length. This length decreases to 21μ for metal-2 and 15μ for the metal-3 layer. With a quick glance at the layout of any ISCAS85 circuit, it is easy to find many wires

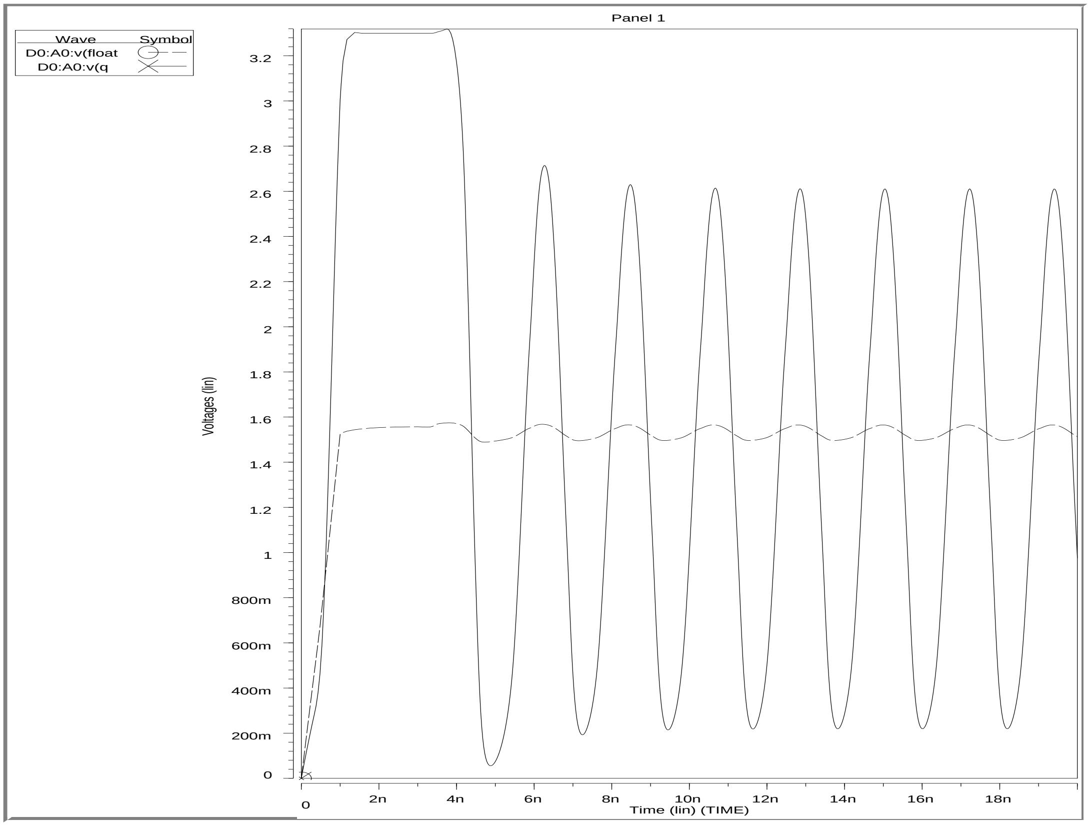

Figure 3.2: HSPICE simulation result for the circuit in Figure 3.1

running in parallel for more than 75μ.

I simulated the circuit in Figure 3.1 using the MCNC cell library with HSPICE with Vdd = 3.3V, and using the BSIM parameters for the HP 0.6μ process from MOSIS. The HSPICE results are shown in Figure 3.2. All the nodes started at 0V, and the circuit is powered up during the first 1ns, that is, Vdd went from 0V to 3.3V. At 3ns, signal S, which is an input to the NOR gate in the middle in Figure 3.1, went from 3.3V to 0V in 1ns. This formed an inverting path from $float$ to q, and node q started oscillating as shown with the solid line in Figure 3.2 due to the electrical feedback created by $C_{wire\_to\_wire}$. Note that this circuit would never oscillate if it were defect free.

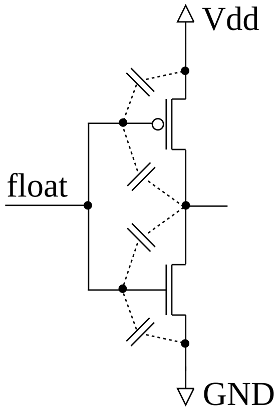

Figure 3.3: Miller capacitances in an inverter.

The voltage around which *float* will oscillate is determined by the values of C_{Vdd}, C_{GND}, and the Miller capacitances in the inverter driven by *float*, as shown in Figure 3.3. Given these capacitances, the size of $C_{wire\_to\_wire}$ determines the $\Delta V_{float}/\Delta V_q$ ratio. If $C_{wire\_to\_wire}$ is not large enough, oscillation will not occur. Since the gate/source and the gate/drain capacitances for the inverter in Figure 3.3, also called Miller capacitances, are non-linear, and the voltage at the output of the inverter has a non-linear relationship to the voltage at its input, HSPICE simulation is used to find out the effects of these Miller capacitances. Simulating the circuit in Figure 3.3 shows that *float* acquires 1.65V with Vdd = 3.3V for the *i1s* inverter in the MCNC library. In general, the floating input of any gate will be forced to a value around Vdd/2 by the Miller capacitances of the p- and n-channel transistors the input drives, when the gate output is sensitized to this floating gate input, except for the XOR and XNOR gates. The Miller capacitances in Figure 3.3 are connected with dotted lines to emphasize that these capacitances are not put in the circuit in addition to the transistors, but they are a byproduct of the transistors.

In Figure 3.2, the *float* oscillates between 1.50V and 1.57V. This 0.07V swing in *float* is sufficient for q to oscillate as the total gain of the path from *float* to q is high enough within the voltage range *float* is moving. If the voltage on *float* was moving around another value, for example, 1.00V, then the total gain of the inverting path might not have been sufficient for an oscillation with *float* changing only 0.07V. Given the transistor sizes in the

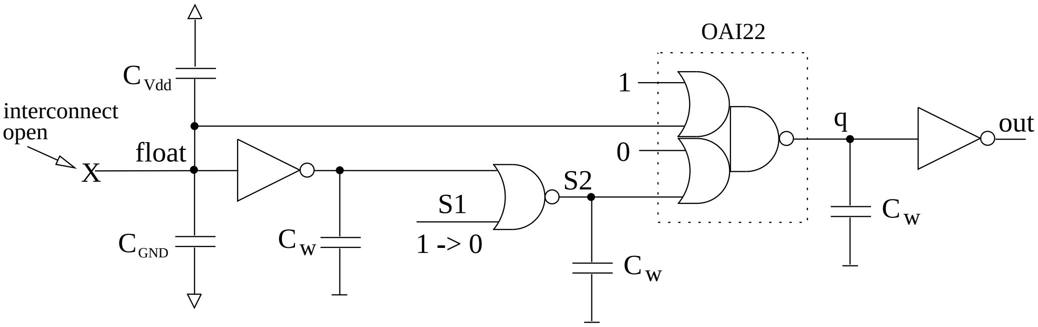

Figure 3.4: Circuit to demonstrate oscillation due to Miller capacitances

*i*1*s* inverter, C_{Vdd} and C_{GND} determine the bias point around which the *float*'s voltage will be oscillating.

C_{Vdd}	10fF	9fF	8fF	7fF	6fF
C_{GND}	10fF	11fF	12fF	13fF	14fF
$C_{wire\_to\_wire,min}$	6.5fF	4.0fF	1.0fF	3.0fF	5.5fF

Table 3.1: Minimum $C_{wire\_to\_wire}$ values for oscillation in Figure 3.1

Assuming that $C_{Vdd}+C_{GND} = 20$fF, Table 3.1 shows the results of HSPICE simulations, where $C_{wire\_to\_wire,min}$ is the minimum capacitance between q and *float* in increments of 0.5fF such that the circuit will still oscillate. Note that as $C_{wire\_to\_wire}$ increases, the $\Delta V_{float}/\Delta V_q$ ratio will also increase. Oscillation is defined to be the case where the swing at *out*'s voltage exceeds the swing at *float*'s voltage. Among the five data points in Table 3.1, $C_{Vdd} = 8$fF and $C_{GND} = 12$fF is the case where the gain in the inverting path is the maximum; thus, even 1fF for $C_{wire\_to\_wire}$ is sufficient for an oscillation as shown earlier. Note that $C_{wire\_to\_wire,min}$ needs to be larger as we move away from the 8fF-12fF point in either direction. Section 3.3 will refer to this table in the discussion about the likelihood of oscillations and sequential behavior.

3.1.2 Feedback via Miller Capacitance

This subsection shows another mechanism that can make an interconnect open oscillate. Consider the circuit in Figure 3.4, which is obtained by removing the $C_{wire\_to\_wire}$ in

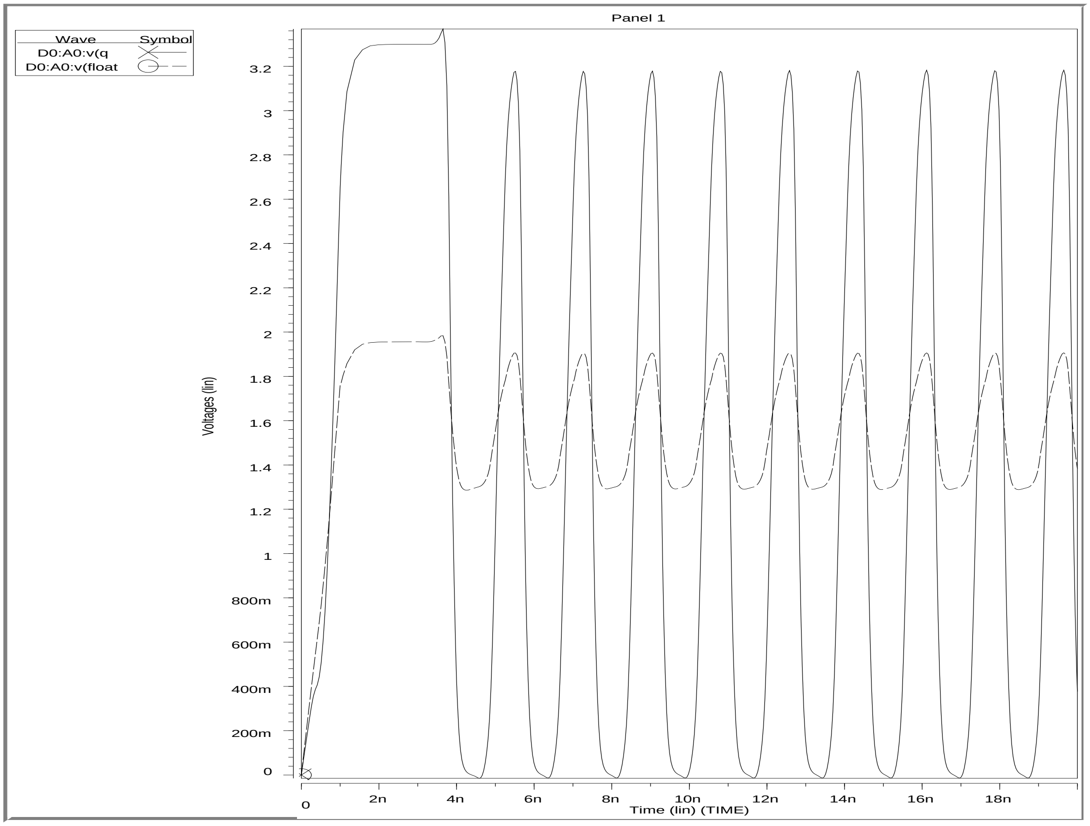

Figure 3.5: HSPICE simulation result for the circuit in Figure 3.4

Figure 3.1 and replacing the inverter that drives node q with an OAI22 (Or-And-Invert) gate from the MCNC library. $C_{wire\_to\_wire}$ was responsible for the feedback loop in Figure 3.1 between q and $float$. In Figure 3.4, however, the feedback from q to $float$ is created by the non-linear Miller capacitances of the transistors inside the OAI22 gate. These Miller capacitances are shown in Figure 3.6, where they are connected with dotted lines to emphasize that they are not additionally inserted into the circuit, but are part of every CMOS transistor. The total Miller capacitance for a transistor can be as large as the total gate-oxide capacitance of that transistor depending on the region the transistor is operating. The interested reader can refer to the "Introduction to Transcapacitance" and the BSIM

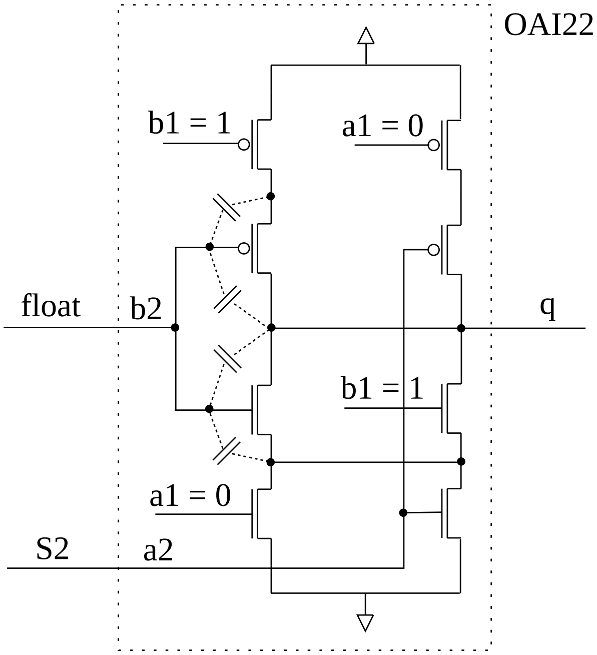

Figure 3.6: Miller capacitances to node $float$ in Figure 3.4

"Charge-Based Capacitance Model" sections in the HSPICE User's Manual [27], and also to Sheu, *et al.*'s work [35].

The HSPICE simulation result for the circuit in Figure 3.4 is shown by Figure 3.5, where C_{Vdd}, C_{GND}, and C_W are 8fF, 12fF, and 20fF, respectively, as for Figure 3.2. The Vdd voltage goes from 0V to 3.3V in the first 1ns, where all nodes in the circuit start from 0V. At 3ns, $S1$ goes from 3.3V to 0V in 1ns sensitizing the inverting path from $float$ to q resulting in an oscillation. Note that this circuit would be a purely combinational circuit, and would never oscillate, if it were defect free.

In order to find the sensitivity of this oscillation to the sizes of the wiring capacitances connected to $float$, HSPICE simulations are used to construct Table 3.2. $C_{Vdd,min}$ and $C_{Vdd,max}$ are the minimum and maximum capacitance values for C_{Vdd} such that the circuit in Figure 3.4 still oscillates. Oscillation is defined to be the case where the swing at *out*'s voltage exceeds the swing at $float$'s voltage. The last two columns in Table 3.2 show that C_{Vdd} needs to be smaller than C_{GND} for an oscillation, but not too small. The last row shows that C_{Vdd} to $C_{Vdd} + C_{GND}$ ratio needs to be 0.46 as $C_{Vdd} + C_{GND}$ gets very large.

$C_{Vdd} + C_{GND}$	$C_{Vdd,min}$	$C_{Vdd,max}$	$\frac{C_{Vdd,min}}{C_{Vdd}+C_{GND}}$	$\frac{C_{Vdd,max}}{C_{Vdd}+C_{GND}}$
20fF	2.0fF	10.0fF	0.10	0.50
30fF	6.5fF	14.5fF	0.22	0.48
40fF	11.5fF	19.0fF	0.29	0.47
50fF	16.0fF	23.5fF	0.32	0.47
60fF	20.5fF	28.0fF	0.33	0.47
400fF	180.0fF	184.5fF	0.45	0.46

Table 3.2: The capacitance ranges for oscillation in Figure 3.4

This implies that the voltage around which *float* needs to oscillate is $0.46 \cdot$ Vdd. This is probably true independent of which MCNC gates are used, because I found out that 1.05V and 1.90V are the maximum logic-0 and the minimum logic-1 voltages, respectively, for the MCNC cell library using the HP 0.6μ BSIM parameters. Note that the mid-point between 1.05V and 1.90V is 1.47V, which is equal to $0.45 \cdot$ Vdd. Gates from other cell libraries are likely to have this property, because n-channel transistors conduct better than the p-channel ones, in general.

C_{Vdd} has two major components; the total capacitance from *float* to other signal wires at logic-1 value, and the total capacitance from *float* to the n-wells (assuming an n-well technology, such as the HP 0.6μ) and to the power wires. On average, the capacitance to signal wires at logic-1 value will be the same as the capacitance to wires at logic-0, and the capacitance to the power wires will be the same as the capacitance to the ground wires. However, n-wells will usually occupy less area than the p-substrate. Therefore, in general it is reasonable to expect that C_{Vdd} will be smaller than C_{GND} on average, but close to it, which is exactly the oscillation requirement discussed in the preceding paragraph as illustrated by Table 3.2.

The last row in Table 3.2 requires a very narrow range for C_{Vdd}, but 400fF corresponds to a very long wire in the HP 0.6μ technology, which would be a more than 3mm long metal-1 wire over substrate. In general, oscillation due to Miller feedback capacitances is more likely when $(C_{Vdd} + C_{GND})$ is small as also shown by Table 3.2, because the Miller feedback capacitance sizes are fixed by the transistor sizes.

3.2 Sequential Behavior due to Interconnect Opens

This section shows how an interconnect open can cause sequential behavior. As in the case of oscillation, the two types of feedback capacitances responsible for sequential behavior are wire-to-wire and Miller capacitances to the floating node created by the open.

3.2.1 Feedback via Wire-to-Wire Capacitance

Consider the circuit in Figure 3.7. This circuit acts like a latch under certain conditions because of the interconnect open. Let $\mathbf{Q_s}$ denote the electrical charge on the $float$ side plate of $C_{wire\_to\_wire}$ plus the electrical charge on the transistor gates of the NOR gate connected to $float$. In Figure 3.8 the Q_s curve is shown with a solid line as a function of V_{float} computed by HSPICE using $C_{wire\_to\_wire}$ = 10fF, and without the assumption that $float$ is floating. This curve is defined to be the **non-floating** Q_s. Note the sudden fall in the non-floating Q_s around V_{float} = 1.7V. This fall is due to the sharp rise of V_q (the voltage on node q) from 0V to 3.3V, which is drawn with a dashed line in Figure 3.8. This sharp rise also happens around V_{float} = 1.7V, causing a sudden positive charge flow away from the $float$ side plate of $C_{wire\_to\_wire}$.

When the assumption that $float$ is actually floating is taken into account, then the following equation needs to be satisfied, also:

$$Q_{init} = Q_s + C_{GND} \cdot V_{float} + C_{Vdd} \cdot (V_{float} - Vdd)$$

where Q_{init} is the trapped charge on $float$ during the fabrication process [15] [17]. If we assume that Q_{init} is zero, then we can rewrite the above equation as follows:

$$Q_s = C_{Vdd} \cdot Vdd - (C_{Vdd} + C_{GND}) \cdot V_{float} \tag{3.1}$$

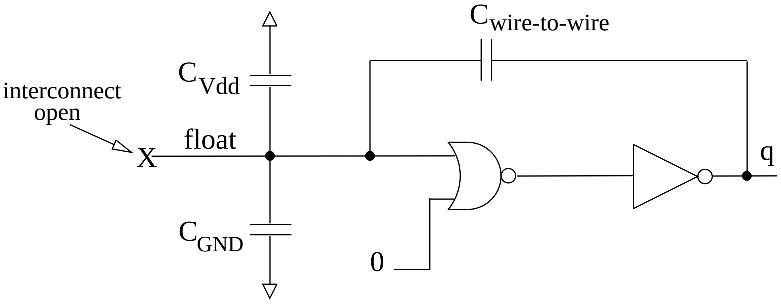

Figure 3.7: Circuit to demonstrate sequential behavior due to a wire-to-wire capacitance

Section 3.4 discusses the effect of Q_{init} not being zero. In Figure 3.8, three straight dotted lines are drawn corresponding to Equation 3.1 with three different $(C_{Vdd} \cdot Vdd)$ values using $C_{Vdd} + C_{GND} = 30\text{fF}$. Note that line 2 intersects the non-floating Q_s at three points, which represent three different solutions. Point *b* corresponds to a metastable state, because even the slightest disturbance on V_{float} will kick the solution point to either *a* or *c*, very much like the metastability in a latch [26]. Therefore, point *b* is not a real solution.

Points *a* and *c* are stable states. The V_q curve in Figure 3.8 shows that the V_{float} values corresponding to points *a* and *c* are interpreted as logic-0 ($V_q = 0V$) and logic-1 ($V_q = 3.3V$), respectively. The straight line for Equation 3.1 moves up as C_{Vdd} increases, and moves down as it decreases, with $(C_{Vdd} + C_{GND})$ a constant determined by a given open. Recall that C_{Vdd} represents the total capacitance between *float* and all neighboring nodes that are at Vdd, which is determined by the vector applied to the circuit for a given open, and that is why I will refer to this straight line as the **vector line**.

If the vector line moves down past line 1 or up past line 3 in Figure 3.8, then it intersects the non-floating Q_s at a single point. Between lines 1 and 3, the real solution is determined by the previous value of V_{float}. V_{float} will remain at logic-0 if the vector line moves below line 1 at least once, and stays below line 3. Similarly, it will remain at logic-1 if the vector line moves above line 3 at least once, and stays above line 1. Therefore, the latch behavior is observed only in the region between lines 1 and 3. C_{Vdd} is 9.6fF and 16.5fF for lines 1 and 3, respectively. Recall that $C_{Vdd}+C_{GND} = 30\text{fF}$, and C_{Vdd} will be smaller than C_{GND}

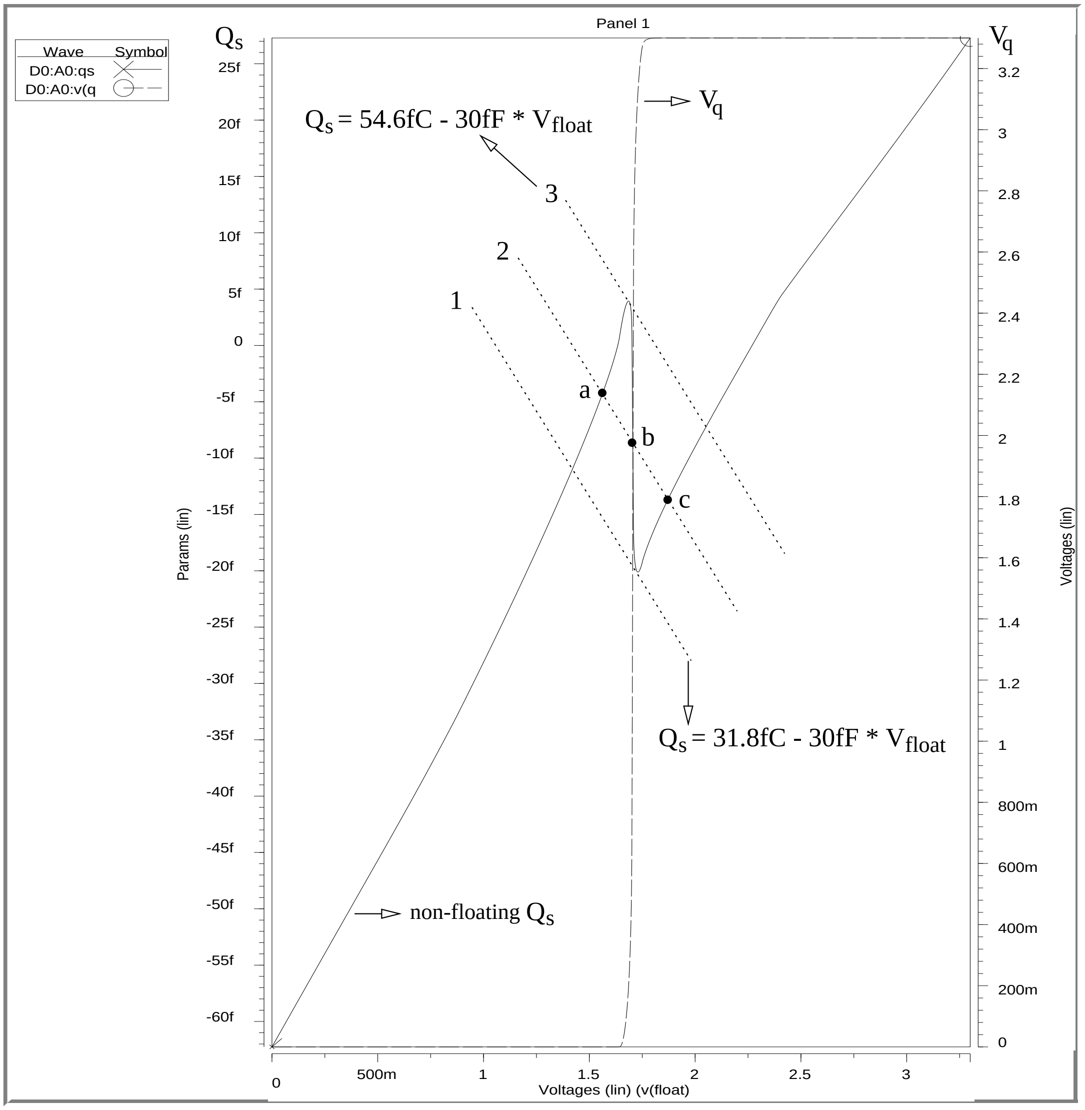

Figure 3.8: Illustration of one metastable and two stable states

but close to it on average as discussed in Section 3.1.2. Therefore, it is reasonable to expect that the vector line will be within lines 1 and 3 in a significant number of vectors applied to the circuit.

In general, for any interconnect open as shown in Figure 3.7 with an even number of inverting gates from $float$ to q, the corresponding curves will look like the ones in Figure 3.8. If we assume a gain of 10 for a single inverting gate, then the cascaded gain for even number of inverting gates will be 100, 10000, etc. Therefore, V_q will make a jump from 0V to Vdd with $\Delta V_{float} =$ Vdd/(cascaded gain) at a critical V_{float} value determined by the type of gate driven by the $float$. This quick jump in V_q is responsible for the sudden drop in the non-floating Q_s, which also marks the transition from logic-0 to logic-1. The size of $C_{wire\_to\_wire}$ together with the transistor sizes connected to the $float$ determines the amount of drop in the non-floating Q_s.

Equation 3.1 shows that the slope of the vector line is -$(C_{Vdd} + C_{GND})$, which is a constant for a given interconnect open. In order for the vector line to intersect the non-floating Q_s always at a single point, it must be steeper than the rate of fall in the non-floating Q_s. In the example in Figure 3.8, this would require $(C_{Vdd}+C_{GND})$ to be larger than 600fF, which corresponds to a metal-1 wire over substrate with a length of over 4.5mm, which is a very long wire. Therefore, as long as $(C_{Vdd}+C_{GND})$ is not extremely large and $C_{wire\_to\_wire}$ is not extremely small, the vector line will intersect the non-floating Q_s at three points for a range of C_{Vdd}, where one point is a metastable state and the other two are for logic-0 and logic-1.

3.2.2 Feedback via Miller Capacitance

The feedback described in the preceding subsection was due to a wire-to-wire capacitance from output to input. I will now show that the same capacitive feedback can occur via the transistor gate-oxide capacitances, more specifically, the Miller feedback capacitances. These two mechanisms were also shown to be responsible for the oscillatory behavior in Sections 3.1.1 and 3.1.2. Consider an XOR gate with one input floating due to an

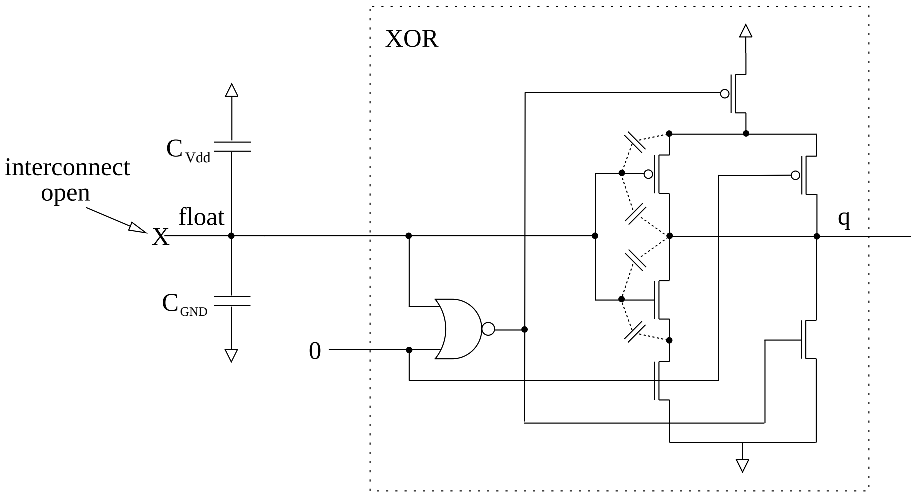

Figure 3.9: Circuit to demonstrate sequential behavior due to Miller capacitances

interconnect open, and the other input at logic-0, as shown in Figure 3.9. The non-inverting path from $float$ to q through the NOR gate inside the XOR gate is analogous to the non-inverting path in Figure 3.7. The Miller capacitances connecting q to $float$ inside the XOR gate form a feedback loop, the same way $C_{wire\_to\_wire}$ does in Figure 3.7. In this case Q_s is simply the total electrical charge on the floating input of the XOR gate. Figure 3.10 shows the non-floating Q_s together with V_q as computed by HSPICE. Note that these curves are very much like the ones in Figure 3.8. Therefore, this XOR gate with a floating input displays a sequential behavior just as the circuit in Figure 3.7 does.

Table 3.3 shows the range of C_{Vdd} for different (C_{Vdd}+C_{GND}) values such that this XOR gate displays sequential behavior, that is, the vector line intersects the non-floating Q_s at three points, one being metastable and the other two being logic-0 and logic-1. Interestingly, the numbers in Table 3.3 are very close to the ones in Table 3.2, showing a duality between oscillation with an OAI22 gate and sequential behavior with an XOR gate in the MCNC library.

Standard cell layouts usually have vias over them to connect signal wires to their inputs and outputs, and vias are particularly susceptible to breaks. I removed one of the input vias for the XOR gate, and extracted all its capacitances using a 0.8μ MAGIC technology file, which had the most detailed extraction information I could find. The floating input had

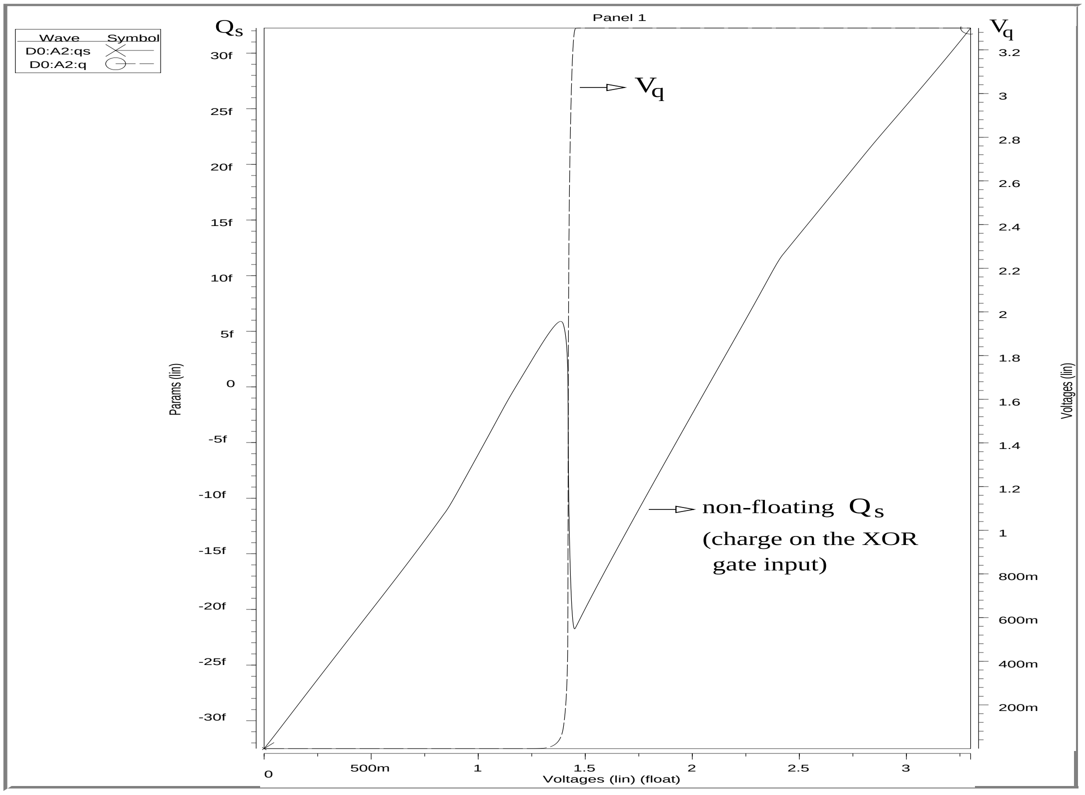

Figure 3.10: Illustration of similarity to the curves in Figure 3.8

some wire-to-wire capacitance to the output of the XOR gate and to the output of the NOR gate inside the XOR. I included these capacitances in my Q_s computation using HSPICE. $(C_{Vdd} + C_{GND})$ was 9.1fF with a 5.1fF capacitance to the substrate and the GND line and a 1.9fF capacitance to the n-well and the Vdd line. So, the minimum and maximum possible C_{Vdd} values were 1.9fF and 9.1 - 5.1 = 4.0fF, which correspond to vector lines that intersect the non-floating Q_s curve at three points. Therefore, this floating input XOR gate will display sequential behavior with any vector applied that makes its fault-free input logic-0 as shown in Figure 3.9. I repeated the same for the other input of the XOR gate, and found the same result.

$C_{Vdd}+C_{GND}$	$C_{Vdd,min}$	$C_{Vdd,max}$	$\frac{C_{Vdd,min}}{C_{Vdd}+C_{GND}}$	$\frac{C_{Vdd,max}}{C_{Vdd}+C_{GND}}$
10fF	0.0fF	6.0fF	0.00	0.60
20fF	2.2fF	10.2fF	0.11	0.51
30fF	6.6fF	14.4fF	0.22	0.48
40fF	11.0fF	18.6fF	0.27	0.46

Table 3.3: Capacitance ranges for sequential behavior in Figure 3.9

3.3 Feedback Activation for Shorts versus Opens

A **feedback bridging fault** is defined to be a short between nodes x and y such that there is a combinational path from x to y. This section discusses the conditions necessary for a feedback bridging fault and an interconnect open to display feedback behavior, where **feedback behavior** means either oscillation or sequential behavior.

In order for a feedback bridging fault between a back wire and a front wire [5] to display feedback behavior, the combinational path from its back wire to the front wire must be sensitized. Similarly, for an interconnect open to display feedback behavior, the combinational path from its floating wire A to at least one other wire B needs to be sensitized, where there is a wire-to-wire or a Miller feedback capacitance between A and B. In this discussion, I assume that this sensitization condition has already been satisfied for a short or an open.

3.3.1 Feedback Activation for Shorts

More Current Paths in the Back Gate

If the gate driving the back wire of a feedback bridging fault wins the drive fight against the gate driving the front wire, then no feedback behavior will be observed. Therefore, the front gate must win for the feedback behavior to occur. Since the path from the back wire to the front needs to be sensitized, the output of the front gate needs to be sensitized to its input on this path. Assuming that this input goes to one n-channel and one p-channel

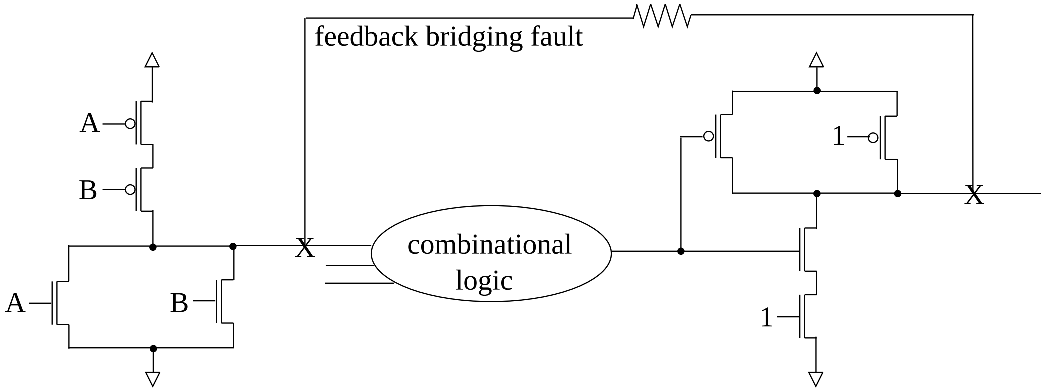

Figure 3.11: Circuit to demonstrate more current paths in the back gate.

transistor inside the front gate, then there is only one current path from Vdd to the front wire or from the front wire to GND inside the front gate, unless multiple paths are sensitized from the back wire to the front wire, which is unlikely. On the other hand, there might be more than one current path from Vdd to the back wire or from the back wire to the GND inside the back gate. For example, in Figure 3.11 when A = B = 1, two current paths exist through the n-channel transistors, in which case the back gate is most likely to win. The side input of the front gate in Figure 3.11 is 1 because of the sensitization condition.

In order to compute the probability of a front gate winning, let's make the following assumptions:

1. 75% of the gates used in a chip can have multiple current paths depending on the gate inputs; unlike an inverter, which can have only one current path.
2. The back gate wins when it has more than one current path.
3. Given a gate that can have multiple current paths, the probability for an input combination that will activate multiple current paths is 25%. For instance, the back gate in Figure 3.11 will have multiple current paths only when A = B = 1;
4. The probability of the front gate winning is 50% when there is a single current path both in the front and back gates. Thus, the implicit assumption is that the bridging fault resistance is zero.

It follows from these assumptions that the probability of a front gate winning is 13/32 = 41%. This probability is actually smaller due to the reasons described in the following, which affect assumption 4 above.

Degraded Voltage on the Back Wire

Due to the drive fight between the back and the front gates, the back wire may not have a rail voltage but an intermediate value between Vdd and GND. If the logic gates from the back wire to the input of the front gate cannot pull this intermediate voltage to a rail value, then the drive strength of the front gate will be diminished, which decreases its chances to win the drive fight against the back gate.

As an example consider the feedback bridging fault in Figure 3.12. The HSPICE simulation result with $R_{short} = 0$ is shown in Figure 3.13, where Vdd and S went from 0V to 3.3V during the first 1ns, and S went to 0V at 4ns. The degraded voltage at q_{back} causes the voltage on $q2$ to be 3.1V instead of 3.3V. This decreases the drive strength of the front gate, and the back gate wins the fight, resulting in no oscillation. However, when I insert two inverters between the NOR gate and the front gate in Figure 3.12, then the circuit oscillates because four stages are sufficient to amplify the degraded voltage on q_{back} to a rail.

I removed the inverter acting as the front gate in Figure 3.12, and made the output of the NOR gate q_{front} in order to verify that sequential behavior is also affected by degraded voltage on q_{back}. The HSPICE simulation with $R_{short} = 0$ showed that q_{front} goes from logic-0 to logic-1 after S switches from 1 to 0, which means that the NOR gate acting as the front gate cannot win the drive fight against the back gate due to the inability of one inverter to pull the degraded back wire voltage to a rail. However, when I insert an inverter between the NOR gate and the front gate in Figure 3.12, then q_{front} stays at logic-0 even after S switches from 1 to 0. This means that the circuit is now acting like a latch, because three stages are sufficient to amplify the degraded logic-0 on the back wire to 3.3V as input to the front gate.

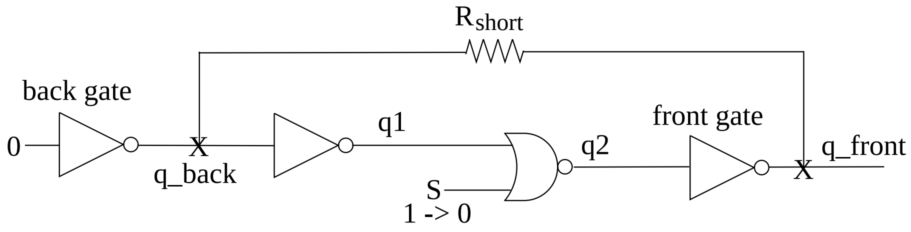

Figure 3.12: Circuit to show the effects of degraded voltage and resistive short

Bridging Resistance

So far the bridging resistance is assumed to be zero, but Rodriguez-Montanes, *et al.* [29] showed that most of the metal bridging resistances fall into the range from 0Ω to 1000Ω in an experimental study. As the bridging resistance increases the voltage on the back wire gets closer to what the back gate is driving; thus, reducing the chances for the front gate to win the drive fight for the back wire.

As explained above, the circuit in Figure 3.12 with $R_{short} = 0$ starts oscillating only when two more inverters are inserted. However, it stops oscillating when $R_{short} \geq 450\Omega$. Similarly, I showed that the same circuit with $R_{short} = 0$ displays sequential behavior only when an additional inverter is inserted. However, the sequential behavior disappears when $R_{short} \geq 427\Omega$. According to the experiments by Rodriguez-Montanes, *et al.* [29], 31% of bridges have a resistance greater than 500Ω. But, resistance distribution for bridges is very much process dependent.

3.3.2 Feedback Activation for Opens

Given a vector applied to the combinational circuit inputs, all the nodes that have a wire-to-wire or a Miller capacitance to the floating wire that is created by an interconnect open fall into two classes: nodes whose voltages depend on, and nodes whose voltages are independent of the floating wire voltage. Adding up the wire-to-wire capacitances to the **independent nodes** that are at logic-1 gives C_{Vdd}, and adding up the wire-to-wire capacitances to the independent nodes that are at logic-0 gives C_{GND}. Capacitances from **dependent nodes** to the floating wire form the feedback capacitances.

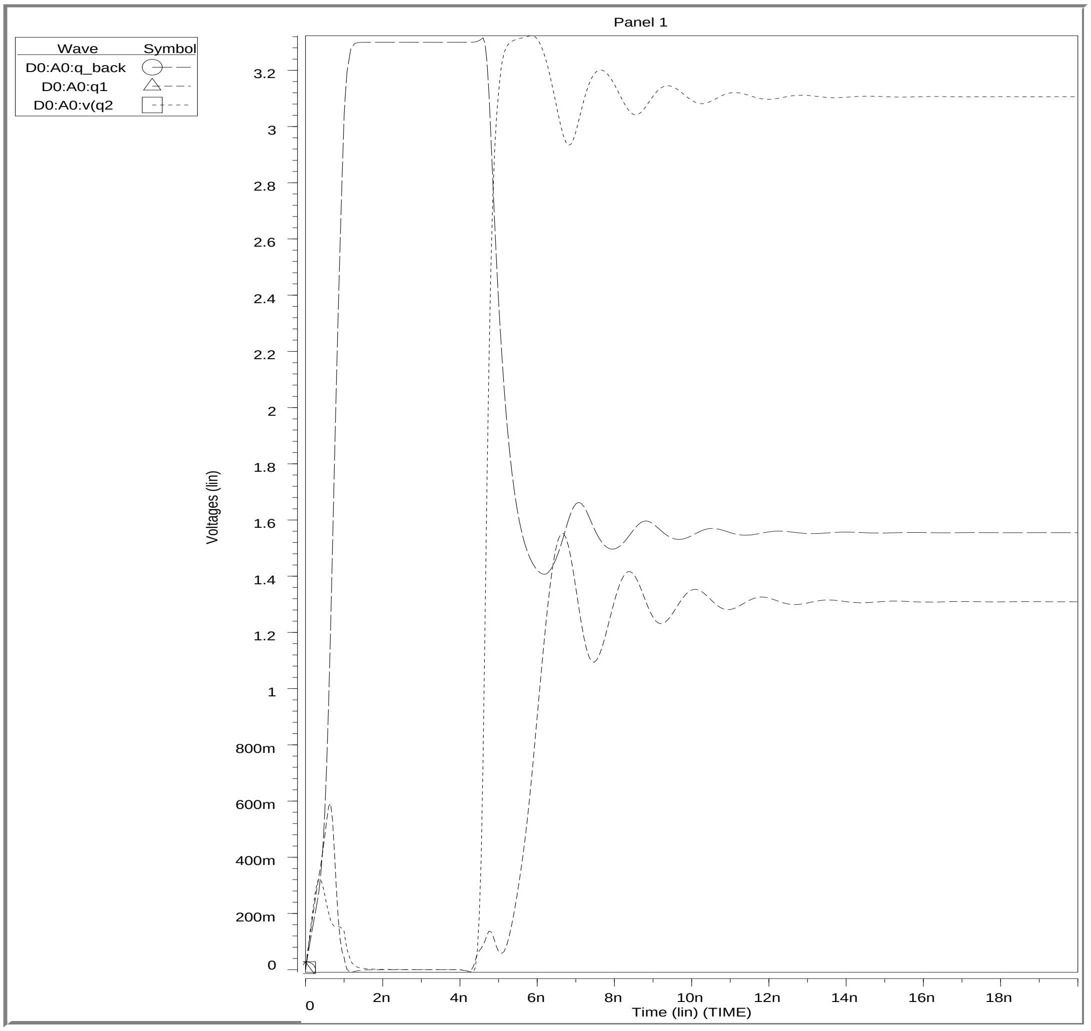

Figure 3.13: HSPICE results for the circuit in Figure 3.12

Dependent nodes become most sensitive to the floating wire voltage when the floating wire voltage is around Vdd/2. For the examples corresponding to Tables 3.2 and 3.3, 0.45*Vdd on the floating wire creates the most sensitivity. A signal wire may cross several other signal wires that run perpendicularly on the metal layer below or above, creating a lot of very small capacitances to it. If we assume N such crossings between the floating wire and other signal wires that are independent of the floating wire voltage, a 0.5 probability for each such signal wire to be logic-1, and a normal distribution for the number of wires at logic-1 among the N signal wires, then the probability $P(x)$ that x wires are at logic-1 is given by the following [1]:

$$P(x) = \sqrt{\frac{2}{N \cdot \pi}} \cdot e^{-2\cdot(x-N/2)^2/N}$$

When $N = 10$, the probability that 4, 5, or 6 wires are at logic-1 is 0.67. When $N = 20$, the probability that the number of logic-1 wires is in the range from 8 to 12 is 0.74. When $N = 30$, the same probability is 0.80 for the range from 12 to 18. Therefore, neighboring signal wires tend to bias the floating wire around Vdd/2 when there are many of them with small capacitances.

In contrast to bridging faults, note that there is no requirement for a gate to outdrive another gate in case of opens to display feedback behavior, because an interconnect open does not cause any drive fight, at all.

Once the floating wire voltage is biased around Vdd/2, then the feedback capacitance(s) need(s) to be large enough to cause a feedback behavior. The size of a feedback capacitance is very much layout dependent. However, wire-to-wire capacitances are growing in importance compared to other capacitances in a layout, because the number of metal layers is increasing, and metal lines on the same layer are getting closer to each other as the smallest feature size decreases. With this trend the probability of feedback behavior from an interconnect open will increase.

Further analysis comparing the probabilities of shorts and opens causing feedback behavior can be done by exploiting the fact that any feedback bridging fault between wires A and B corresponds to a feedback capacitance from B to A when wire A is floating due to an open. This analysis will not be done in this dissertation. As a related reference, Maxwell, *et al.* [25] used total wiring capacitance values estimating the likelihoods of bridging faults to come up with a weighted stuck-at fault coverage measure.

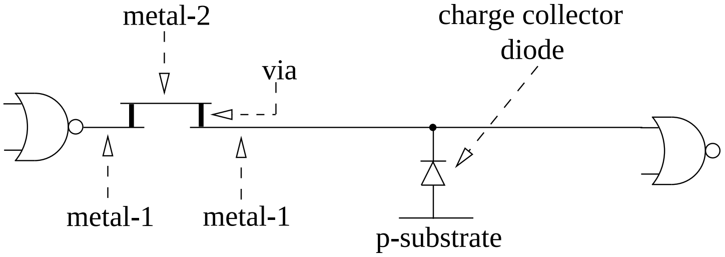

Figure 3.14: Illustration of a charge collector diode.

3.4 Trapped Charge, Die Surface, and Charge Collector Diodes

This section discusses three other factors other than the capacitances to the floating wire that affect the floating wire behavior.

An interconnect open creates a floating metal wire that might collect electrical charge during the fabrication process [15] [22]. Appendix A [17] shows that trapped charge on floating metal wires connected to transistor gates can create voltages in the range between -1V to 1V in a set of experimental chips fabricated with an HP 0.8μ process. However, 75% of our trapped charge voltage measurements were between -0.5V and 0.5V. This trapped charge voltage needs to be added to the floating wire voltage determined by the C_{Vdd}, C_{GND}, Miller and wire-to-wire feedback capacitances discussed in the previous sections. For instance, non-zero trapped charge requires the addition of Q_{init} to the right-hand-side of Equation 3.1 in Section 3.2.1, which moves the vector lines in Figure 3.8 up or down by the magnitude of Q_{init}. The exact value of Q_{init} is very much process dependent.

Another interesting factor in determining the voltage of a floating wire is the RC interconnect behavior of the die surface as I indirectly observed in my experiments described in Appendix A. In this case, the capacitance from the floating wire to the die surface and the die surface resistivity are the important factors. However, because the die surface resistivity depends on the fabrication and packaging technologies used, and usually not measured as part of a manufacturing process, it is difficult to estimate the effect of the RC behavior of the die surface.

In an IC layout, traversing back from a logic gate input towards its driver, one might first find a long metal-1 wire, then a metal-2 wire, and finally another metal-1 wire before reaching the driving gate, as shown in Figure 3.14. During the fabrication of this structure, while metal-1 wires are being etched using plasma, the metal-1 wire connected to one input of the NOR gate on the right in Figure 3.14 will be floating, because the metal-2 wire is not formed, yet. This creates a hazard for the gate-oxide of the transistors the floating metal-1 is connected to when the metal-1 wire is very long, and can collect significant charge from the plasma to create a high voltage (antenna effect [22]). One common technique to bleed this charge is the use of diodes as shown in Figure 3.14, which are called **charge collector diodes**. The reverse-bias current of this diode needs to be sufficient to bleed the charge at the necessary rate during plasma etching. Each IC manufacturer has its own rules determining when and where to add such a charge collector diode.

If one of the vias in Figure 3.14 is broken due to a defect, then the resulting interconnect open will differ from the ones discussed so far because of the reverse-biased diode attached to the floating wire. The reverse-bias current, the size of the floating wire, the size of the transistors driven by the floating wire, and the time allowed on the tester from the application of a test vector to the capture of the circuit response are all determining factors for the behavior of this open. At one end, the existence of the diode can be totally ignored, and at the other end, the diode makes the open always behave like a stuck-at-0 fault (p-substrate is connected to the circuit GND). The percentage of such opens and their exact behavior are governed by the particular fabrication process.

3.5 Summary

This chapter showed that interconnect opens can cause oscillations and state holding (sequential) behavior due to feedback capacitances from nodes sensitized to the floating node back to the floating node. This behavior is exactly like feedback bridging faults, however the conditions necessary to cause feedback are more complicated in the case of opens. The capacitive feedback can come from either wire-to-wire capacitance of as little as 1 femtofarad or from Miller feedback capacitances from within a logic-gate. The range of initial voltages on the floating node to allow oscillation or additional state to occur is in the vicinity of Vdd/2, which is likely to be the voltage the node is charged to due to Miller capacitances during circuit power up.

This chapter then discussed the necessary conditions for a feedback bridging fault to display feedback behavior. This rough analysis implies that an interconnect open might be more likely to display feedback behavior than a feedback bridging fault.

Finally, trapped charge, die surface effect, and charge collector diodes are poined out as important factors to be included in determining the behavior of an interconnect open. However, the extent these factors affect the behavior of an interconnect open depends very much on the fabrication process.

All the information provided in this chapter needs to be used in building an accurate fault simulation tool or an effective test strategy for interconnect opens.

4. A Fault Simulation Algorithm for Interconnect Opens

This chapter describes a fault simulation algorithm for interconnect opens. This algorithm takes into account all the factors, that are covered in this dissertation, and that can affect the voltage of a floating wire created by an interconnect open. These factors are as follows:

1. Capacitances between the floating wire and its neighboring wires (including the substrate),
2. Miller (drain/gate, source/gate) capacitances to the floating wire,
3. charge collector diodes.
4. trapped charge deposited on the floating wire during fabrication,
5. the RC interconnect behavior of the die surface, and

Items 4 and 5 are covered in Appendix A in great detail, where trapped charge voltage measurements on floating wires in a set of experimental chips show a range of -1V to 1V. Figure A.3 in Appendix A models the RC interconnect behavior of the die surface observed in these experiments.

In order to come up with the wiring capacitances to a floating wire, one needs to know the location of the break in the layout. A very good candidate for a break is a via, rather than the metal track itself. So, this chapter assumes that each via in a given layout corresponds to an interconnect open.

Several layout tools that are capable of two-dimensional capacitance extraction exist in both industry (e.g. CELL3 from Cadence) and academia (e.g. MAGIC from UC Berkeley). The problem with two-dimensional capacitance extraction, which is based on the area and the perimeter of the overlap between two conducting surfaces, is its poor accuracy. For example, the capacitance due to a metal-1 wire A crossing a metal-2 wire B perpendicularly vary more than 100% depending on the proximity of other metal-1 and metal-2 wires that interfere with the electric field lines between A and B. This observation is based on the results from a three-dimensional capacitance extraction program, called SPACE3D [40], using the parameters from MOSIS for the HP 0.6μ fabrication process and 0.9μ wire width. Three-dimensional capacitance extraction is sufficiently accurate, but not feasible in terms of CPU time.

One way to deal with this problem is to use two-dimensional capacitance extraction, and to assume that the actual capacitance is within +/- x% of the computed capacitance. This way, a fault simulator for interconnect opens will use capacitance ranges for wiring capacitances to compute a voltage range for a given open. It is almost impossible to find the minimum value for x, such that using a smaller x would not include the actual capacitance for at least one wire. The value used for x should ideally be based on statistical data from two-dimensional and three-dimensional capacitance extractions of the same layout for a given fabrication process. Larger x provides greater accuracy at the expense of a greater indeterminism due to a wider range for the floating wire voltage.

Even though a 100% variation can occur for the cross-capacitance between the two wires of two metal layers, the layout patterns creating the ends of the maximum variation are not likely to occur very often. These patterns are (i) two wires crossing in isolation, that is, the nearest third wire being far away, say, 50μ away, and (ii) the cross-capacitance in the middle of a large tight grid formed by a bundle of parallel wires with minimum separation crossing another bundle of wires with minimum separation on a different layer. Case (ii) might occur more often than case (i) in today's chips routed tightly with automatic place and route tools. Accordingly, the area and perimeter coefficients for two-dimensional capacitance extraction can be adjusted to reflect this fact. Capacitances between a wire and the substrate, and between two wires running in parallel are not affected by the proximity of other wires as much as the cross-capacitances are. Therefore, a value of 30 for x corresponding to a variation of 60% might be reasonable.

The size of a Miller capacitance is determined by the width, length, and thickness of the gate oxide, and the operating point of the CMOS transistor. Since the transistor geometries can be tightly controlled in today's CMOS processes, and Miller capacitances are not affected by surrounding structures, there is no need to use a value range for a Miller capacitance as proposed for a wiring capacitance.

Charge collector diodes are used as described in Section 3.4 in Chapter 3 to protect gate-oxides in cases of very long wires that remain floating temporarily during the fabrication process. The number of charge collector diodes used in a chip depends on the properties of the fabrication process. If the reverse-bias current of a charge collector diode is large enough, then the corresponding interconnect open can be detected by a stuck-at-0 test for that floating wire. If the reverse-bias current is small enough so that it can be ignored during the test application period, then the charge collector diodes can be ignored, and the trapped charge for the corresponding opens will be zero. If the reverse-bias current is somewhere in between such that it may pull the floating wire voltage to logic-0 only towards the end of the test application period, then the test vectors that detect stuck-at-0 on the corresponding opens can be moved to the end of the stuck-at test suite.

It is not possible to estimate the amount of trapped charge on a floating wire. Therefore, my fault simulation algorithm does not make any assumptions for the amount of trapped charge. Note that excessive trapped charge on a floating wire may create a dangerous voltage level during fabrication such that a gate-oxide punch through may occur creating a gate-oxide short. This is different from a conventional gate-oxide short [37, 11], because it is coupled with a floating wire, and the cause of this gate-oxide short is an open. This dissertation does not consider this case, leaving it for further research.

Appendix A provides the experimental evidence for the RC interconnect effect of the die surface. If a particular manufacturing process can guarantee sufficiently large resistivity for the die surface, for instance $R_{surf} > 5 * 10^{13}\Omega$ in Figure A.3 in Appendix A, then the die surface can be ignored. Otherwise, the capacitance between a floating wire and the die surface needs to be considered using a worst case value for the surface voltage.

4.1 Processing the Standard-Cell Library

Before performing fault simulation on a particular standard-cell based design, the standard-cell library needs to be processed. I assume that every gate in the library consists of a network of p-channel transistors and a complementary network of n-channel transistors, such as a CMOS NAND or a NOR gate. If composite gates exist in the library, such as XOR gates, they need to be split into simple inverting gates.

First, the **L0_th** and **L1_th** values need to be determined, which denote the maximum voltage that is still a logic 0 and the minimum voltage that is still a logic 1, respectively, as also described in Chapter 2. Following are some definitions:

An **input combination** for a gate denotes either a single gate input or multiple inputs of the same gate tied together. Tieing the inputs of a 2-input NAND gate does not create a useful structure; therefore, tied inputs will be combinations such as $a1$ and $b1$ inputs or $a1$ and $b2$ inputs of an AOI22 or an OAI22 gate from the MCNC library.

$\mathbf{V_{L0,g,ic}}$ denotes the voltage for the input combination ic of gate g such that the output of g is at L1_th value, and is sensitized to the input combination ic. $\mathbf{Q_{L0,g,ic}}$ denotes the total electrical charge on transistor gates that are driven by the input combination ic, when the voltage on ic is $V_{L0,g,ic}$. $\mathbf{V_{L1,g,ic}}$ and $\mathbf{Q_{L1,g,ic}}$ are defined similarly.

For IDDQ testing, a threshold current $\mathbf{I_{ddq,th}}$ needs to be determined such that a quiescent power supply current larger than $I_{ddq,th}$ indicates a defective chip. $\mathbf{V_{iddq0,g,ic}}$ denotes the logic-0 voltage on input combination ic such that $I_{ddq,th}$ flows through gate g. $\mathbf{Q_{iddq0,g,ic}}$ denotes the total electrical charge on transistor gates that are driven by ic, when the voltage on ic is $V_{iddq0,g,ic}$. $\mathbf{V_{iddq1,g,ic}}$ and $\mathbf{Q_{iddq1,g,ic}}$ are defined similarly.

For every gate g and input combination ic in the library, $V_{L0,g,ic}$, $V_{L1,g,ic}$, $V_{iddq0,g,ic}$, $V_{iddq1,g,ic}$, $Q_{L0,g,ic}$, $Q_{L1,g,ic}$, $Q_{iddq0,g,ic}$, and $Q_{iddq1,g,ic}$ will be computed by (H)SPICE, and be recorded. In addition, the slope and the y-intercept values for the straight line defined by points ($V_{iddq0,g,ic}$, $Q_{iddq0,g,ic}$) and ($V_{L0,g,ic}$, $Q_{L0,g,ic}$) will be recorded to be used for interpolation purposes in my fault simulation algorithm. Similarly, the slope and the y-intercept values for the straight line defined by points ($V_{L1,g,ic}$, $Q_{L1,g,ic}$) and ($V_{iddq1,g,ic}$, $Q_{iddq1,g,ic}$) will be recorded. For example, consider the HSPICE Q-V plot in Figure 4.1 for the a input of the NAND gate in the MCNC library using the HP 0.6μ process parameters. The V_{iddq0}, V_{L0}, V_{L1}, and V_{iddq1} points are marked using $I_{ddq,th} = 50\mu A$, L0_th = 1.05V,

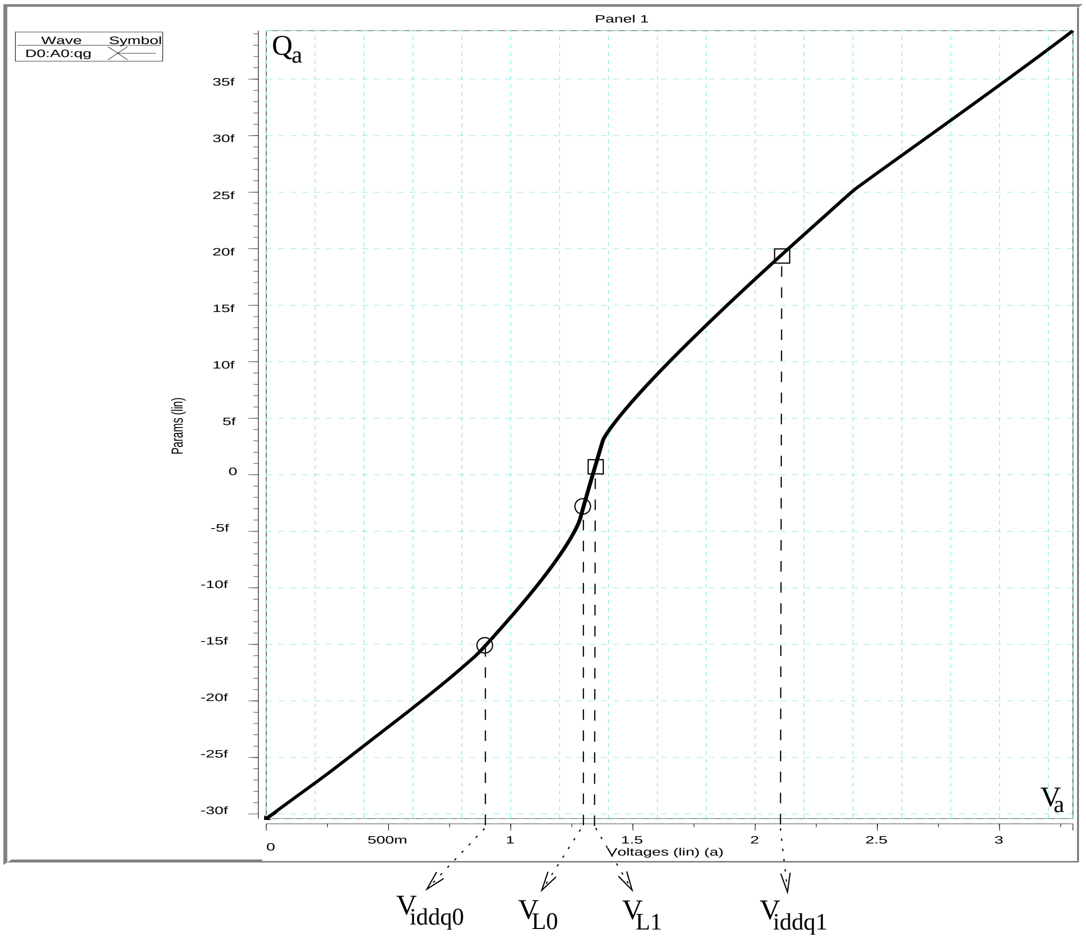

Figure 4.1: Charge versus voltage plot for the *a* input of a NAND gate.

and L1_th = 1.90V as also used in Chapter 2.

4.2 A Fault Simulation Algorithm

Given a test set, the following describes a fault simulation algorithm for interconnect opens using voltage sensing. The circuit is assumed to be full-scan. Since I do not make any assumptions for the amount of trapped charge, the algorithm computes a $Q_{trapped,max,FW,sa0}$ and a $Q_{trapped,min,FW,sa1}$ value for each floating wire **FW** created by an interconnect open. $\mathbf{Q_{trapped,max,FW,sa0}}$ denotes the maximum trapped charge on *FW* with which the given test set can detect the open as a stuck-at-0 fault, and $\mathbf{Q_{trapped,min,FW,sa1}}$ denotes the minimum trapped charge on FW with which the given test set can detect the open as a stuck-at-1 fault.

If a test set for IDDQ measurements exists, I explain how to use this algorithm to find the overall detection intervals of trapped charge for each FW.

This is a worst-case algorithm in the sense that detection is guaranteed if the actual trapped charge falls within the computed detection intervals; however, the open might still be detected even if the trapped charge is outside of these intervals. It would be ideal to compute detection probabilities rather than guaranteed detection intervals, but this is difficult to accomplish. The following describes the steps of my algorithm.

4.2.1 The Steps

STEP 0: Initialize $Q_{trapped,max,FW,sa0}$ to $-\infty$ and $Q_{trapped,min,FW,sa1}$ to $+\infty$ for each FW.

STEP 1: Repeat STEP 2 for each test vector T in the test suite.

STEP 2: Repeat STEP 3 for each FW. The algorithm assumes a single interconnect open in the faulty circuit.

STEP 3: If T detects a stuck-at-0 fault on FW, then do STEPs 4 through 8. If T detects a stuck-at-1 fault on FW, then do STEPs 4 through 8 with stuck-at-1 detection in mind, otherwise pick the next FW, and go to the beginning of this STEP.

STEP 4:

As also described in Section 3.3.2 in Chapter 3, all nodes that have a wire-to-wire or a Miller capacitance to FW fall into two categories: (i) nodes whose voltages *depend* on, and (ii) nodes whose voltages are *independent* of the FW voltage.

Identify the dependent nodes by observing the effects of flipping the logic value on FW. Let **inverting** and **non-inverting dependent nodes** denote the dependent nodes whose logic value is the inverse of FW's and the same as FW's, respectively. This information is readily available from the FW stuck-at-0 fault simulation process performed in STEP 3.

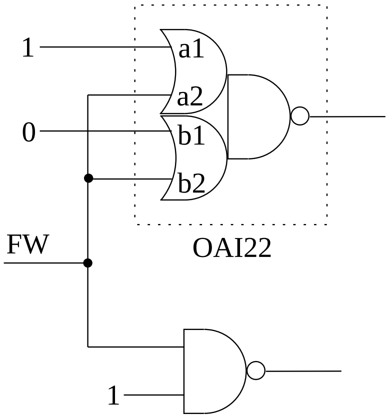

Figure 4.2: An example for a sensitized input combination

Inverting dependent nodes are the source of any oscillation, and non-inverting dependent nodes are the source any sequential behavior as described in Chapter 3.

STEP 5:

For each gate g driven by FW, find the input combination to which the output of g is sensitized. For example, the sensitized input combination for the OAI22 gate shown in Figure 4.2 consists of only its $b2$ input, because its $a2$ input is blocked by $a1 = 1$. Let $\mathbf{V_{L0,FW,T}}$ denote the minimum $V_{L0,g,ic}$ over all such sensitized input combinations.

If the voltage on FW is less than or equal to $V_{L0,FW,T}$, then test vector T will detect the open. Since the algorithm does not make any assumption about the amount of the trapped charge, the maximum amount of trapped charge that will produce a voltage of $V_{L0,FW,T}$ on FW, when T is applied, needs to be computed.

Set $V_{FW} = V_{L0,FW,T}$.

STEP 6:

The total charge on FW has two components: (i) the charge on the transistor gates driven by FW, denoted as $\mathbf{Q_{gate}}$, and (ii) the charge on the wiring part of FW due to capacitances between FW and other nodes, including the substrate and the die surface, denoted as $\mathbf{Q_{wire}}$.

$$Q_{gate} = \sum_{i \in S\_IC} Q_i(V_{FW}) + \sum_{j \in US\_INP} Q_{gate,j} \tag{4.1}$$

where **S_IC** denotes the set of all sensitized input combinations, and **US_INP** denotes the set of all remaining gate inputs driven by FW, which are unsensitized. Recall that V_{FW} is set to the minimum V_{L0} value over all the sensitized input combinations. Therefore, in order to find $Q_i(V_{FW})$ in Equation 4.1 for input combination i, whose V_{L0} value is larger than V_{FW}, the equation for the straight line passing through its (V_{iddq0}, Q_{iddq0}) and (V_{L0}, Q_{L0}) points will be used. Recall that the slope and y-intercept values for this line were recorded during the library processing. Therefore, computing $Q_i(V_{FW})$ involves a floating-point multiplication and an addition (fast operations in today's microprocessors).

The $Q_{gate,j}$ term in Equation 4.1 denotes the charge on the p-channel and n-channel transistor gates connected to logic gate input j, which is unsensitized. Equations 2.3 and 2.5 in Chapter 2 are used to compute $Q_{gate,j}$. The saturation region equation, Equation 2.7, will not be used, because no current is flowing through the transistors connected to j due to its unsensitized state.

Equation 4.2 shows the Q_{wire} portion of the total charge on FW. **CWW0** and **CWW1** denote the set of capacitances from FW to other nodes that are at logic-0 and at logic-1 state, respectively. Recall from the beginning of this chapter that the algorithm uses a range for each wiring capacitance due to the accuracy limitations of two-dimensional capacitance extraction. Since a logic-0 at a neighboring node helps stuck-at-0 detection by T, I assume the worst case by using the minimum value for the wiring capacitance C_{w0} in Equation 4.2. Similarly, a logic-1 at a neighboring node works against stuck-at-0 detection by T, I assume the worst case by using the maximum value for the wiring capacitance C_{w1}.

If the die surface does not have a big enough resistivity, then the capacitance between FW and the surface needs to be considered, also. Given a voltage range $V_{surf,min}$ and $V_{surf,max}$ the die surface can acquire, the last term in Equation 4.2 is added for modeling the worst case effect of the die surface, assuming that $V_{surf,max} > V_{L0,FW,T}$. Otherwise, $C_{FW-surf,min}$ needs to be used instead of $C_{FW-surf,max}$.

$$\begin{aligned} Q_{wire} &= \sum_{w0 \in CWW0} C_{w0,min} * V_{FW} + \sum_{w1 \in CWW1} C_{w1,max} * (V_{FW} - Vdd) + \\ & \quad C_{FW-surf,max} * (V_{FW} - V_{surf,max}) \end{aligned} \tag{4.2}$$

<u>**STEP 7:**</u>

Let $Q1 = Q_{gate} + Q_{wire}$. At this point, one might be tempted to think that $Q1$ gives the maximum trapped charge on FW with which T can detect the open as a stuck-at-0 fault, denoted as $\mathbf{Q_{trapped,max,FW,T}}$. However, if V_{FW} increases a bit entering the logic-1 region, then all the non-inverting dependent nodes will change from logic-0 to logic-1; thus, dumping charge onto FW. That is, it is possible that FW might be at logic-1 state with smaller trapped charge than $(Q_{gate} + Q_{wire})$ due to non-inverting dependent nodes, which cause the sequential behavior explained in Section 3.2 in Chapter 3.

In order to find whether this is the case, flip the logic values of all the dependent nodes identified at Step 4, and recompute Q_{gate} and Q_{wire}, and let $Q2 = Q_{gate} + Q_{wire}$ using these new values. Then,

$$Q_{trapped,max,FW,T} = min(Q1, Q2)$$

<u>**STEP 8:**</u> If $Q_{trapped,max,FW,T} > Q_{trapped,max,FW,sa0}$ then set $Q_{trapped,max,FW,sa0}$ to $Q_{trapped,max,FW,T}$.

□

Steps 4 through 8 are described for a stuck-at-0 detection. For stuck-at-1 detection, the following changes will be made.

In STEP 5, $V_{L1,FW,T}$, which is the maximum logic-1 threshold over all the sensitized input combinations, will be used instead of $V_{L0,FW,T}$.

In STEP 6, for input combination i, the line passing through its (V_{L1}, Q_{L1}) and (V_{iddq1}, Q_{iddq1}) points will be used. Equation 4.2 will change to

$$\begin{aligned} Q_{wire} &= \sum_{w0 \in CWW0} C_{w0,max} * V_{FW} + \sum_{w1 \in CWW1} C_{w1,min} * (V_{FW} - Vdd) + \\ & \quad C_{FW-surf,max} * (V_{FW} - V_{surf,min}) \end{aligned}$$

In STEP 7, $Q_{trapped,min,FW,T}$ will be computed as: $Q_{trapped,min,FW,T} = max(Q1, Q2)$

STEP 8 will be: if $Q_{trapped,min,FW,T} < Q_{trapped,min,FW,sa1}$ then set $Q_{trapped,min,FW,sa1}$ to $Q_{trapped,min,FW,T}$.

4.2.2 IDDQ Testing

The essence of the algorithm described above is to find the maximum trapped charge value to guarantee a voltage less than a given value on FW with a given test vector, or to find the minimum trapped charge value to guarantee a voltage greater than a given value on FW with a given test vector. For a particular FW and T, the voltage range on FW that will give rise to an IDDQ greater than $I_{ddq,th}$ can be found using the $V_{iddq0,g,ic}$ and $V_{iddq1,g,ic}$ values for the sensitized input combinations driven by FW. Let this range be defined by $\mathbf{V_{iddq0,FW,T}}$ and $\mathbf{V_{iddq1,FW,T}}$. Now, a similar algorithm can be used to compute the corresponding $\mathbf{Q_{trapped,iddq0,FW,T}}$ and $\mathbf{Q_{trapped,iddq1,FW,T}}$ values, such that when the actual trapped charge is within the range defined by these values, IDDQ detection is guaranteed.

For a given FW, if all the IDDQ detection ranges computed for all the IDDQ test vectors completely cover the interval $(Q_{trapped,max,FW,sa0}$, $Q_{trapped,min,FW,sa1})$, then this FW is guaranteed to be detected by either IDDQ measurements or by a stuck-at test. If $Q_{trapped,max,FW,sa0} \geq Q_{trapped,min,FW,sa1}$ then FW is guaranteed to be detected as a stuck-at fault, and it will not be considered for IDDQ detection.

If the trapped charge value is known a priori, then a check whether the stuck-at and/or IDDQ detection ranges contain this value will be sufficient to determine whether the corresponding open will be detected.

Note that charge collector diodes are not mentioned in my fault simulation algorithm, because their effect can be incorporated as follows: If the reverse-bias current of a charge collector diode is sufficiently large, then the corresponding FW will be marked detected by a stuck-at-0 test vector. If this current can be ignored during the test application time, then the diode itself will be ignored, and the actual trapped charge value will be set to zero if sufficient time elapses between the end of wafer fabrication and the beginning of test. If the reverse-bias current is somewhere in between, then a stuck-at-0 test vector for the corresponding FW will be moved to the end of the test suite expecting that the diode will have drained FW to a logic-0 state by that time.

4.2.3 Complexity of the Algorithm

It is certain that this algorithm requires more CPU time than regular stuck-at fault simulation. This section attempts to estimate how much more. Assuming that each via represents a potential interconnect open, Table 4.1 compares the number of vias against the number of collapsed stuck-at faults computed by *Nemesis* [20] in the seven largest ISCAS85 circuits to give an idea about the size of the fault space for an interconnect open fault simulator. In Table 4.1, the number of vias is roughly twice the number of collapsed stuck-at faults.

Circuit	c1355	c1908	c2670	c3540	c5315	c6288	c7552
# of vias	2562	2404	4687	6405	11165	12678	13436
# of SAFs	1882	1246	2237	3185	4865	8748	6291

Table 4.1: Number of vias vs. number of collapsed stuck-at faults.

My fault simulator does not drop faults as it proceeds, because every vector in the test suite might have a contribution to the detection intervals. The stuck-at fault simulation in STEP 3 is performed for every test vector and FW pair. Assuming that the average number of fault simulations per stuck-at fault is some fraction f of the number of test vectors in regular stuck-at fault simulation, my algorithm will perform $1/f$ times more stuck-at fault simulation per fault (open).

Equations 4.1 and 4.2 in STEP 6 require floating point arithmetic for every logic gate driven by FW and every wire FW has a capacitance to. The average fanout in a circuit is independent of the circuit size, but depends on the functionality of the circuit. Also, the average number of capacitances a wire has to its neighboring nodes is independent of the circuit size, and again depends on the functionality of the circuit. The total number of realistic bridging faults for each ISCAS85 circuit reported by Chess and Larrabee [5] shows that this number more or less has a linear trend with the circuit size. Since a bridging fault indicates the existence of a wire-to-wire capacitance between the potentially shorted wires, the average number of wire-to-wire capacitances per wire remains more or less the same.

Modern microprocessors that are equipped with floating point units can perform floating point arithmetic not much slower than integer operations.

All the facts listed above indicate that the run-time of this fault simulation algorithm will be a constant multiple of the run-time of regular stuck-at fault simulation for the same circuit.

5. Summary and Conclusion

This chapter summarizes the contributions of this dissertation.

- This dissertation showed that Miller capacitances play a significant role in test invalidation of network break test vectors. In fact, Miller capacitances, which until now were never considered as a source of test invalidation, are much more important than charge sharing with p-n junction capacitances. Another important conclusion is that a very accurate fault simulator for network breaks that takes into account transient paths, Miller and p-n junction capacitances is feasible, as shown in Chapter 2.
- Even though the transient paths to Vdd/GND form the most important network break test invalidation mechanism, Miller and p-n junction capacitances are also important when a significant number of interconnect wires have capacitances that are comparable to these transistor capacitances. Even though the interconnect capacitances are not shrinking as fast as the transistor capacitances are shrinking as feature sizes decrease, transistor capacitances can still not be ignored. This is especially true when there are logic blocks in the cell library that are made up of primitive cells packed together tightly using short interconnect wires. One simple example is an XOR, or an XNOR gate. Careful placement and routing can keep the percentage of short wires used in the interconnect at a substantial level even when there are very long wires in the layout. Finally, the gate-oxide thickness is shrinking as the fabrication technology advances, which has an increasing effect on the Miller capacitances.
- This dissertation showed for the first time that interconnect opens can cause oscillations and state holding (sequential) behavior due to feedback capacitances from nodes sensitized to the floating node back to the floating node. This behavior is exactly like feedback bridging faults, however the conditions necessary to cause feedback are more complicated in the case of opens. The capacitive feedback can come from either wire-to-wire capacitance of as little as 1 femtofarad or from Miller feedback capacitances from within a logic-gate. The range of initial voltages on the floating node to allow

oscillation or additional state to occur is in the vicinity of Vdd/2, which is likely to be the voltage the node is charged to due to Miller capacitances during circuit power up.

- Chapter 3 presented a discussion about the necessary conditions for a feedback bridging fault to display feedback behavior. This rough analysis implies that an interconnect open might be more likely to display feedback behavior than a feedback bridging fault.
- This dissertation presented a fault simulation algorithm for interconnect opens, which takes all the factors, that are discussed throughout the dissertation and affect the voltage of a floating wire, into account. The run-time estimate for this algorithm is a constant multiple of the run-time required by the stuck-at fault simulation for the same circuit.
- This dissertation presented for the first time the experimental evidence that the die surface can act as an RC interconnect; thus, capacitively coupling a floating wire, which is created by an interconnect open, to all other signals in a chip. The resistance range for the die surface necessary for this effect is large enough so that the die surface is a perfect insulator for the fault-free operation of the chip. I presented a circuit model for the RC interconnect effect of the die surface. HSPICE simulations with this circuit model produced the same floating-wire behavior I have observed in my experiments. My experiments and HSPICE simulations show that the passivation layer or the nitrogen gas inside the die cavity is too resistive to cause the die surface act as an RC interconnect. There are two other potential candidates to explain the reduced die surface conductivity. One is a hygroscopic contaminant on the die surface that may form during the time period from the wafers are fabricated to they are cut and packaged. The other is adsorption of water molecules or some other molecules from air by the passivation layer surface. Further study is necessary to identify the actual mechanism.

- Appendix A presented trapped charge voltage measurements on floating-gate transistors with poly or metal extensions. Floating gates with poly extensions always showed negative trapped charge values, up to -4V. Floating gates with metal extensions showed both positive and negative trapped charge values within the -1V to 1V range. As another factor to be considered for predicting the behavior of a floating wire, Chapter 3 pointed out to charge collector diodes.
- The trapped charge measurement results given in Appendix A together with the ones reported by Johnson [15] show the unpredictability of trapped charge values, which is taken into account in the fault simulation algorithm presented in Chapter 4.

Appendix A. Die Surface and Trapped Charge in Testing for Interconnect Opens

The electrical charge trapped on the floating wires created by opens during fabrication is important, because it is one of the factors that determines the voltage on a floating gate [32, 4, 23, 12, 36], thus determining the behavior of the cell this floating-gate transistor is in. Johnson [15] designed and performed trapped charge measurements on test structures that consist of floating-gate p-channel and n-channel transistors with varying lengths of poly extensions. These measurements showed that there was always a positive charge on the floating poly, and the voltage created by this charge ranged from 0.1V to 2.3V.

The floating-gate test structures were built to measure the effects of different lengths and different layers of floating metal lines connected to transistor gates, because interconnect opens most of the time create floating metal wires. This also provided the chance to try erasing trapped charge via ultra-violet light (as is done for EPROMs).

My experiments took a very unexpected turn when I noticed that the floating-gate voltages in our hermetically sealed packages were behaving differently from the ones in other packages with their dies exposed to air. The floating-gate voltages in the sealed packages displayed a swing of at least 0.5V with a short rise time and a long fall time, while the other packages did not. I had to conduct several experiments to identify what was really going on. All the experimental data point to the conclusion that the die surfaces in the hermetically sealed packages are conducting sufficient electrical charge forming an RC path from all other signals in the chip to the metal wires connected to the floating gates. The time constant of this path is varying from a couple of seconds to a couple of minutes. Because, a part typically spends from a couple of seconds to a couple of minutes on a tester, the die surface becomes a determining factor on the behavior of a floating gate during the period the part is tested. The capacitance of a wire in a VLSI chip today is typically between 10^{-15}F to 10^{-12}F. In order to obtain a time constant of 1 second, a resistance of $10^{12}\Omega$ to $10^{15}\Omega$ is needed, which is a perfect insulator for normal operation of the chip, but conductive enough to affect the voltage of a floating-gate.

Until now, the voltage of a wire connected to one or more floating gate transistors due to a break in the interconnect is thought to be determined by only two factors: *i)* the coupling capacitances to neighboring wires and to other terminals of the floating gate transistors, and *ii)* the amount of the trapped charge deposited on the floating gates during fabrication. This appendix presents several pieces of experimental evidence that the die surface conduction forms a third factor in determining the voltage of a floating wire. Section A.2 presents the phenomenon I have observed that led me to the die surface, and then Section A.3 analyzes several mechanisms that might be responsible for die surface conduction and presents evidence for and against these mechanisms including HSPICE simulation results matching the floating-wire behaviors I have observed.

Section A.4 presents the trapped charge measurement results. The following section describes our test chip and the measurement technique we used.

A.1 The Test Chip and the Measurement Technique

The test chip was fabricated using the HP 0.8μ CMOS n-well technology through MOSIS. Twelve of the packages were hermetically sealed with metal lids, and thirteen of them had their die cavities covered by taped plastic lids so that these lids could be easily removed by peeling off the tape to be able to expose the die to ultra-violet light. All of the packages are ceramic.

In the test chip, there are three n-channel transistors with 125μ, 250μ, and 500μ long metal-2 wires attached to their gates. These wires are not driven by any other device, and they are all 1.5μ wide. For each such floating-gate wire g, two metal-2 wires are running on both sides of g with a separation distance of 1.5μ from g. Connected to these two metal-2 wires, a metal-1 wire is running just below g, forming an electrical node pg (pseudo gate). The voltage on floating gate g can be controlled by controlling the voltage on pg because of the capacitance between pg and g. We use an identical n-channel transistor in order to measure the voltage on g.

Figure A.1 shows our measurement circuitry. The dotted rectangle represents the chip boundary. Everything outside the dotted rectangle is off-chip. The n-channel transistor on the left in Figure A.1 represents the floating gate transistor, and the one on the right represents the identical size reference transistor. All the transistors on the test chip are 0.8μ long and 5μ wide. The basic idea to measure the floating-gate voltage is to apply the same drain-source voltage to both transistors, and to have the same drain current flowing through both transistors. In this state, the measured gate voltage on the reference transistor will be the same as the gate voltage on the floating-gate transistor. This method is also used by Johnson [15], but our measurement technique shown in Figure A.1 keeps the drain-source voltage fixed at 0.2V, thus eliminating the hot electron effect that might otherwise alter the amount of the trapped charge, as explained in Section A.1.2. The op-amps in Figure A.1 also make sure that the same current is flowing through both transistors.

The voltage on the floating gate when all the chip pins are grounded is the **trapped charge voltage**. The basic idea to measure the trapped charge voltage is to measure at least two points on the $V_g - V_{pg}$ plane, and extrapolate to $V_{pg} = 0$ assuming a linear relationship between V_g and V_{pg}. The actual relationship is

$$V_g = K1 * V_{pg} + K2 * V_{ds} + V_{tc} \quad \text{where} \tag{A.1}$$

$$K1 = \frac{C_{pg-fw}}{C_{pg-fw} + C_{fw} + C_{gate}}, \quad K2 = \frac{C_{gd}}{C_{pg-fw} + C_{fw} + C_{gate}}.$$

C_{pg-fw} and C_{fw} are the capacitances from pg to the floating wire in Figure A.1 and from the floating wire to the substrate, respectively. The gate capacitance is the sum of the gate-to-drain, gate-to-source, and gate-to-bulk capacitances, that is, $C_{gate} = C_{gd} + C_{gs} + C_{gb}$. V_{tc} is the trapped charge voltage, and V_{ds} is the drain-source voltage, which is kept fixed at 0.2V. Section A.1.1 shows that $K2 * 0.2$ is less than 15mV. The value of $K1$ changes slightly as the transistor enters the linear region from the cut-off region, introducing a slight extrapolation error, as explained in Section A.1.1.

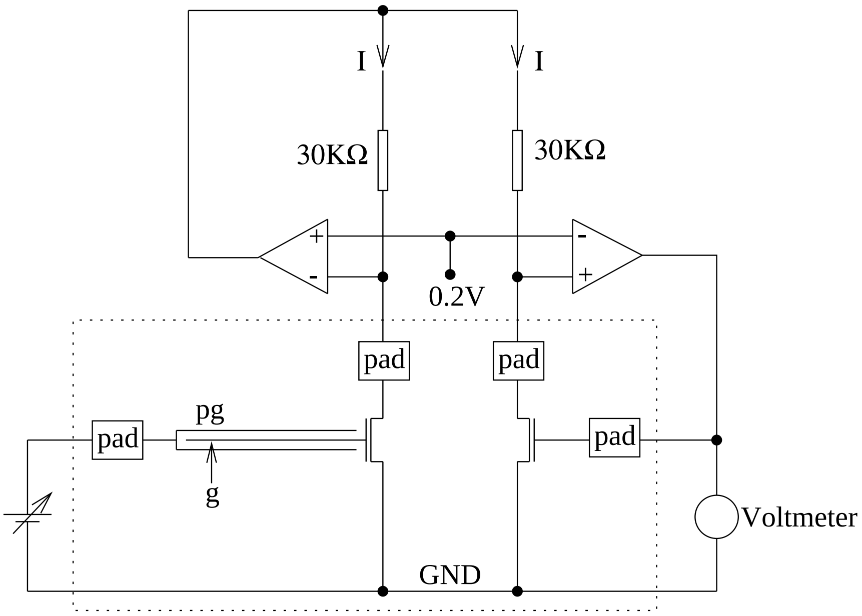

Figure A.1: Circuitry to measure the floating gate voltage controlled by *pg*

In the test chip, there are also three n-channel floating-gate transistors with poly extensions of 42μ, 21μ, and no extension. No metal is attached to these floating gates, and the width of the poly is 1μ. Over both the 42μ and 21μ poly extensions, we have a metal-1 rectangle that is 12μ wide, creating the *pg* node shown in Figure A.1.

There are three p-channel floating-gate transistors with metal-1, metal-2, and metal-3 wires attached to their gates. Each such wire is 250μ long. Both the metal-1 and metal-2 wires are 1.5μ wide, but the metal-3 wire is 3μ wide, which is the minimum width for metal-3 according to MOSIS rules. These metal wires are not surrounded by any other wire, that is, the floating-gate voltage can be controlled only by controlling the source-drain voltage. This is also the technique used by Johnson [15]. Figure A.2 shows our measurement circuitry for this technique. A similar circuit is also used by Pricket, *et al.* [28]. The main reason I did not create a *pg* node around the floating wires of these three p-channel transistors is that estimating the capacitance of a floating wire to *pg* and to substrate becomes very hard based on the capacitance parameters given by MOSIS, because these parameters do not include wire-to-wire capacitance on the same layer, and the given capacitance parameters assume a wire-on-a-plane structure. So, it is very easy to estimate the substrate capacitance of a floating wire not surrounded by any other wire. Knowing the total capacitance of the floating wire will allow the computation of the amount of the trapped charge from the V_{tc} measurements.

The main difference in the technique used in Figure A.2 compared to Figure A.1 is that the source-drain voltage is not constant, and it is used to control the floating-gate voltage. Note that the drain-source voltages for the floating and the reference transistors are kept the same by the op-amp. All of the op-amps have their outputs connected to a 10μF capacitor to prevent oscillation, which is not shown in Figures A.1 and A.2.

In Figure A.2, the basic idea to measure the trapped charge voltage is to measure at least two points on the $V_g - V_{source}$ plane, and extrapolate to $V_{source} = 0$. The assumption is

$$V_g = K3 * V_{source} + V_{tc} \quad \text{where} \quad K3 = \frac{C_{gs} + C_{gb}}{C_{fw} + C_{gate}} \tag{A.2}$$

The bulk (n-well) of the transistor is tied to the source terminal in our p-channel transistors. Section A.1.1 shows the extrapolation errors for this case.

We also have three other p-channel floating-gate transistors with poly extensions the same as the three n-channel transistors described earlier. The only difference is that the metal-1 over the poly extension is grounded, and the measurement circuitry in Figure A.2 is used. Table A.1 summarizes the twelve floating-gate (**FG**) transistors we used. The measurement technique for transistor 6 is similar to the one used for p-channel transistors, because the setup for transistor 6 does not have a *pg* node.

A.1.1 Extrapolation Errors

Equation A.1 is used to compute the trapped charge voltage for transistors 1–5 in Table A.1 by extrapolating to $V_{pg} = 0$. The value of $K2$ is the largest for transistor 5, because C_{pg-fw} and C_{fw} are the smallest compared to transistors 1–4. From the parameters supplied by MOSIS, both C_{pg-fw} and C_{fw} can be computed to be 3fF. Performing an

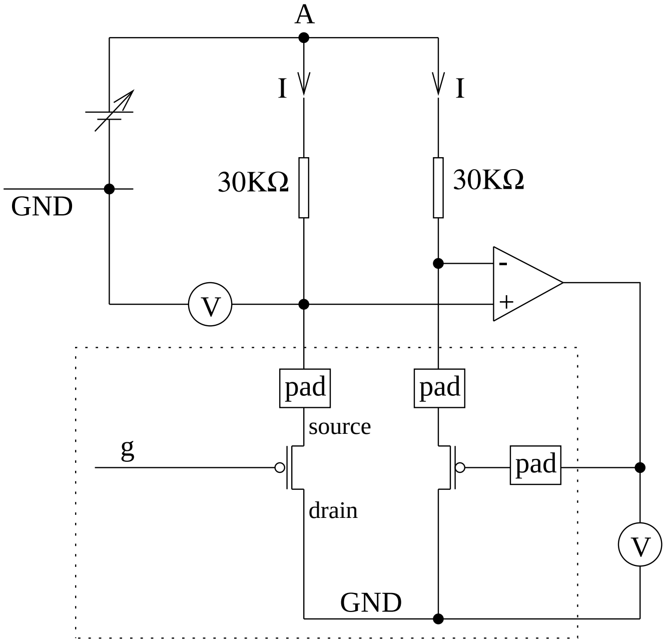

Figure A.2: Circuitry to measure the floating gate voltage controlled by source

HSPICE simulation with $V_{pg} = 0$, $V_{ds} = 0.2V$, and $V_{tc} = 0$, I found $V_g = K2 * V_{ds} = 15mV$, which is a negligibly small number. This number is even smaller for transistors 1–4.

Again with transistor 5, by varying V_{pg} from 0V to 5V in HSPICE simulation, $K1 = 0.27$ until V_g reaches the threshold voltage, and $K1 = 0.24$ afterwards. The measurement technique in Figure A.1 needs the FG transistor be conducting in order to be able to make FG voltage measurements. Therefore, I can only measure FG voltages larger than the threshold value. Using two points computed by HSPICE, $(V_{pg} = 3.75V, V_g = 1.00V)$ and $(V_{pg} = 5.00V, V_g = 1.29V)$ while the transistor is on, extrapolating to $V_{pg} = 0$ results in $V_g = 0.13V$, which is the *extrapolation error*. The reason for the decrease in $K1$ is the decrease in C_{gate} from the cut-off region to the linear region. The extrapolation error decreases as C_{pg-fw} and C_{fw} become larger going from transistor 5 towards 1. Recall that all the transistors in our test chip have the same dimensions: 5.0μ wide and 0.8μ long.

Equation A.2 is used to compute the trapped charge voltage for transistors 7–12. According to HSPICE simulations, the value of $K3$ increases from the cut-off region to the saturation region where the FG transistor of Figure A.2 operates when it is on. This in-

n-channel transistors: FG voltage controlled by *pg* voltage		p-channel transistors: FG voltage controlled by *source* voltage	
trans. #	FG extension	trans. #	FG extension
1	500μ metal-2	7	250μ metal-1
2	250μ metal-2	8	250μ metal-2
3	125μ metal-2	9	250μ metal-3
4	42μ poly	10	42μ poly
5	21μ poly	11	21μ poly
6	no extension	12	no extension

Table A.1: Summary of the twelve floating-gate (FG)transistors in our test chip

crease in $K3$, caused by the change in the three transistor capacitances C_{gd}, C_{gs}, and C_{gb}, creates a negative extrapolation error. HSPICE simulation with $V_{tc} = 0$ and $C_{fw} = 0.5$fF corresponding to transistor 12 gave me an extrapolation error of -0.47V. I could not use $C_{fw} = 0$, because the transistor did not conduct even when $V_{source} = 5V$ with $C_{fw} = 0$. The extrapolation error with $C_{fw} = 34$fF corresponding to transistor 7 with the largest C_{fw} among transistors 7–12 was -0.07V. Therefore, the extrapolation error for transistors 8–11 should be between -0.5V and -0.07V.

Even though Johnson [15] has not reported his extrapolation errors, his should be similar to the ones reported here.

A.1.2 Advantages of Our Measurement Technique

Controlling the FG voltage with a pg terminal, and keeping the drain-source voltage fixed at 0.2V as shown in Figure A.1 ensures that the FG transistor will be in the linear region while taking measurements, whereas controlling the FG voltage with the source or drain terminal of the FG transistor, as also done by Johnson [15], allows the transistor to be only in the saturation region while taking measurements, unless there is a large enough

V_{tc} on the gate. When an n-channel transistor is in the saturation region, hot electrons can be easily injected into the gate oxide, altering the amount of the trapped charge sitting on the transistor gate. In our setup in Figure A.1, electrons crossing the channel do not have enough energy to penetrate the gate-oxide. This is the main advantage of this setup. The other advantage is to have pg as an independent terminal to control the FG voltage, without relying on the drain or source voltage.

The p-channel transistors are not as susceptible to hot carrier effect as the n-channel transistors are, so controlling the FG voltage with the source terminal is safe for p-channel transistors.

A.2 The (Mysterious) Effect of the Vdd-Ring

On our chip, there is a 50μ wide metal-3 wire, called Vdd-ring, that goes all around the die periphery through the pad cells to supply the Vdd voltage to electrostatic discharge protection circuits inside the pad cells. What looked very mysterious was that the FG voltage of any transistor with a metal wire connected to it was quite unstable when I applied 5V to the Vdd-ring. This applies to the transistors 1–3 and 7–9 in Table A.1. More specifically, switching the Vdd-ring voltage from 0V to 5V caused the FG voltages of transistors 1 through 3 go from around 1.0V to around 1.5V within 5 seconds to a couple of minutes depending on the package. The FG voltage would then turn back, and go down to its previous value before 5V was applied to the Vdd-ring, which took about 3 minutes to about an hour. Both the rates of increase and decrease in FG voltage resembled an RC charge-up or an RC discharge. I observed the same type of behavior for transistors 7 through 9. The FG voltages of transistors 7, 8, and 9 in one package increased 0.66V, 0.92V, and 1.43V, respectively, using the setup shown in Figure A.2. In this particular experiment, I kept the voltage of node A constant, but as the gate voltage increased, the drain-source voltage also increased, unlike the setup in Figure A.1, where the drain-source voltage is always 0.2V. The FG voltages reached their peaks within 3-5 seconds again displaying an RC charge-up, and came back to their previous values within 2-3 minutes at the rate of an RC discharge. Transistors 7–9 in other hermetically sealed packages showed the same behavior but only taking a longer time, up to 45 minutes, to complete the cycle of RC charge-up and RC discharge.

The gate voltages of all other *FG* transistors showed only a couple of millivolts of reaction to the Vdd-ring. Note that all of these other *FG* transistors have only poly extensions to their gates, if any, and all the poly extensions are covered with metal-1.

A very important observation is that switching the Vdd-ring voltage from 0V to 5V had no effect on the *FG* voltage of any transistor in the taped-lid packages, even though the dies in all our packages are identical. That is, I observed the effect of Vdd-ring only in the hermetically sealed packages. This shows that there is no direct capacitive coupling between the Vdd-ring and the *FG* wires. Because, if there was, I would observe the same behavior in the taped-lid packages also. Besides, the distance between the Vdd-ring and the long side of any other *FG* wire is between 100μ to 600μ, which is too large a distance to have any coupling capacitance.

I carefully removed the metal lid of an hermetically sealed package to see whether the Vdd-ring effect would disappear. To my expectation, the Vdd-ring effect has indeed completely disappeared. I very carefully drilled a very tiny hole, less than $1mm$ in diameter, through the metal lid of another hermetically sealed package. About 5 minutes after drilling the hole, switching Vdd-ring from 0V to 5V caused a diminished effect on the *FG* voltage of transistor 9. The following day, the Vdd-ring effect has completely disappeared. This was an evidence that air entering the die cavity has something to do with this phenomenon.

I learned from MOSIS [9] that in the case of hermetic sealing the packages are placed in a closed oven, where the chips are cleared of moisture at a temperature of 150°C by passing dry nitrogen through the oven. The packages are then sealed by covering the die cavity with a metal lid soldered using the gold eutectic solder ring around the lid. So, the die surface in touch with air versus the die surface cleared of moisture and in touch with dry nitrogen is making the difference.

I speculate that the die surface is somehow involved in charge transport. I came up

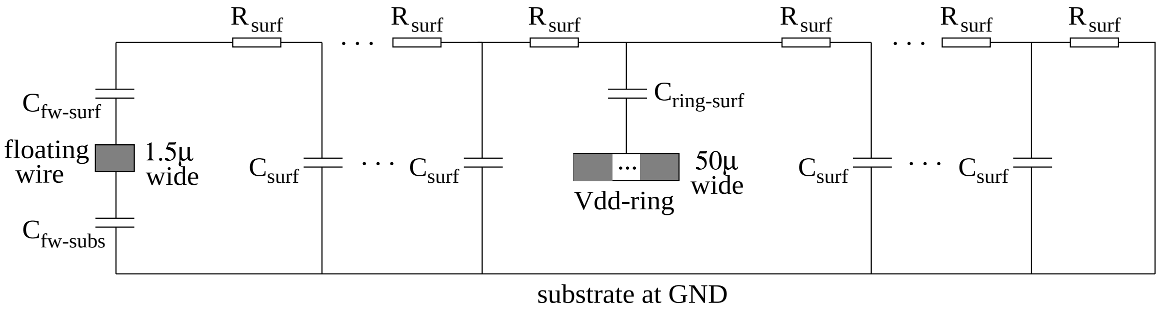

Figure A.3: The circuit modeling the effect of the Vdd-ring

with the circuit shown in Figure A.3 modeling the Vdd-ring effect I observed. The cross sections of the floating wire and the Vdd-ring are shown. The floating wire is connected to a transistor gate, which is not shown in the figure. $C_{fw-subs}$ and $C_{fw-surf}$ denote the capacitances from the floating wire to the substrate and to the die surface, respectively. The RC path $\{C_{ring-surf}, R_{surf}, ..., R_{surf}, C_{fw-surf}\}$ from Vdd-ring to the floating wire in Figure A.3 models the RC charge-up I observed in my experiments. As mentioned earlier, the FG voltages of transistors 7, 8, and 9 in one package increased 0.66V, 0.92V, and 1.43V, respectively, showing that metal-3 is affected the most by the Vdd-ring, followed by metal-2 and metal-1. So, as the floating wire gets closer to the die surface, it becomes more influenced by the voltage change on the Vdd-ring, which supports the die surface conduction speculation. It does not surprise me that Johnson [15] has not reported any such phenomenon, because all of the poly extensions in his FG transistors were covered by metal lines connected to transistor drains or sources, which shielded the poly extensions from the die surface. As explained earlier, the poly extensions of our transistors 4, 5, 10, and 11 in Table A.1 are also covered with metal-1, and these transistors are not affected by the voltage change on the Vdd-ring, which is further evidence of die surface conduction.

The RC path $\{C_{ring-surf}, R_{surf}, ..., R_{surf}, \text{substrate}\}$ in Figure A.3 models the observed RC discharge. This path implies that the surface must have a path to the substrate, which is indeed the case. The passivation layer touches the bare silicon around the periphery of the die. This way, cutting the dies from the wafer is done by cutting through the bare silicon. Otherwise, cutting through the oxide might crack the oxide layers. The die surface is modeled as an RC interconnect, with extremely high resistance values producing RC time constants that are on the order of seconds and minutes. Air is substantially decreasing the resistances in Figure A.3 due to the humidity, resulting in very small RC time constants to observe the Vdd-ring effect.

Even though I refer to the effect of the die surface conduction as the Vdd-ring effect, because I have first noticed it by switching the voltage on the Vdd-ring wire, all the signal wires in a regular chip will be affecting the floating-wire voltage via die surface conduction. The Vdd-ring wire in Figure A.3 can be replaced by any signal wire. Because a signal wire is typically a much smaller wire than the Vdd-ring, the $C_{ring-surf}$ capacitance in Figure A.3 will be much smaller in the case of a signal wire. Because there are thousands or millions of signal wires in a chip, their combined effect will be much stronger than the Vdd-ring.

The following section goes through possible mechanisms for die surface conduction, and show whether they fit my observations. Note that the surface conduction mentioned here has extremely high resistance, but when combined with extremely low capacitances of the wires in the chip, the resulting RC time constant becomes on the order of seconds. The following section will also present HSPICE simulation results with the circuit in Figure A.3, that match the behavior I observed in the experiments.

A.3 Analysis of Possible Mechanisms for the Vdd-ring Effect

A.3.1 Is It the Passivation Layer?

One candidate for charge transport on the die surface is the passivation layer. In the HP 0.8μ technology our chips were fabricated with, two passivation layers are used. First, a 0.35μ silicon oxynitride film is deposited on top of the metal-3 layer, followed by a 0.60μ silicon nitride film. Rabiller, *et al.* [30] reported that the room temperature resistivity varies smoothly from less than $10^{14}\Omega - cm$ for silicon nitride to more than $10^{16}\Omega - cm$ for silicon dioxide by varying the ratios of oxygen and nitrogen in a silicon oxynitride film deposited using plasma enhanced chemical vapor deposition (PECVD). Therefore, the nitride passivation layer should be 100 to 1000 times more conductive than the intermetal dielectric, which is silicon dioxide.

The floating wire of transistor 3 in Table A.1 starts from a point 85μ away from the Vdd-ring, and extends 125μ into the chip as shown in Figure A.4. It is 300μ away from the closest parallel Vdd-ring wire. As a crude approximation, let us assume that charge needs to travel 100μ from above Vdd-ring to above the floating wire through the 0.6μ thick nitride film. Let's call this path the **nitride path**. The metal-2 floating wire of transistor 3 is surrounded by two other metal-2 lines connected to a metal-1 line below, creating the electrical node *pg*, as explained in Section A.1. The distance from the floating wire to either of *pg*'s metal-2 lines is 1.5μ. The metal-2 thickness and the separation between metal-2 and metal-1 layers are both around 1μ. Considering that *pg* has three paths through the oxide to the floating wire, each path being 1.5μ or less long, charge needs to travel roughly 1μ from *pg* to the floating wire through the oxide, which constitutes the **oxide path**. Assuming a 1 to 1000 ratio for the nitride to oxide resistivities, and recalling that the nitride path is 100 times longer than the oxide path, the resistance of the oxide path must be only 10 times larger than the resistance of the nitride path.

Even though I observed a 0.5V increase in the floating wire voltage due to the Vdd-ring within seconds or a couple of minutes, the *FG* voltage returned to its previous value, and stayed there even after an overnight operation with constant *pg* voltage. In other words, in 15 hours I have not observed any RC charge-up through the oxide path, which is at most 10 times more resistive than the nitride path. But, the Vdd-ring effect is observable within seconds, which shows that the nitride cannot be the medium of charge transport for the Vdd-ring effect. Moreover, a substantial portion of the charge traveling through the nitride path will be kept by the surface-to-substrate capacitances shown in Figure A.3, thus delaying the RC charge-up of the floating wire, which is not the case for the oxide path. HSPICE simulation results later in this appendix confirm that the nitride is too resistive to be responsible for the Vdd-ring effect. In addition, it is not clear to me how air would tremendously increase the conductivity of nitride. Recall that the taped-lid packages do not

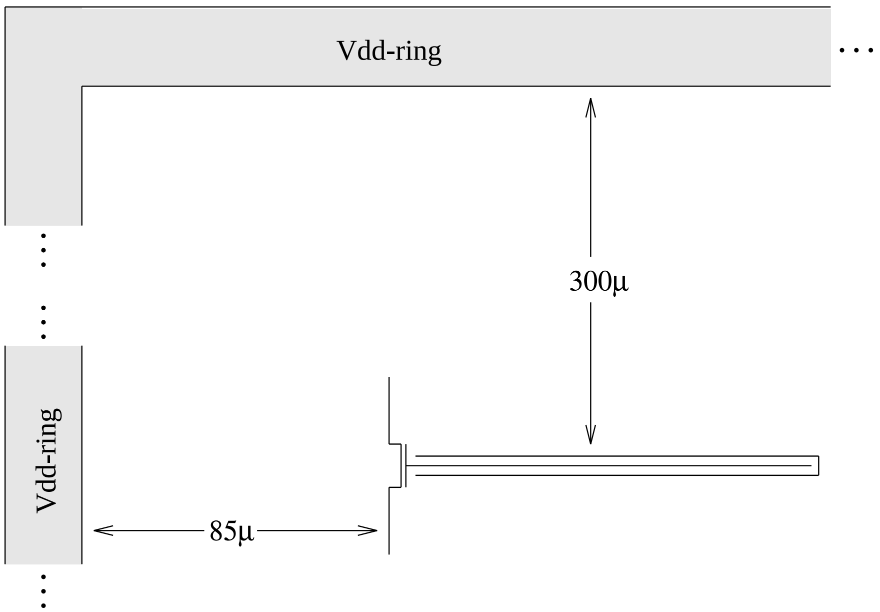

Figure A.4: The distances between the Vdd-ring and transistor 3

possess the Vdd-ring effect most probably due to the reduction in the surface resistances shown in Figure A.3 as a result of contact with air.

A.3.2 Is It the Dry Nitrogen and Air?

Another candidate for the medium of charge transport on or over the die surface is the gas inside the die cavity. This gas is dry nitrogen according to MOSIS in our hermetically sealed packages with metal lids, and ordinary air in packages whose die cavities are covered by taping plastic lids over them. The die cavity is identical in both types of packages, and it is $10mm$ on one side. The distance between the die surface and the plastic or the metal lid is about $1mm$, and the die size is $1.9mm$ on one side. Therefore, applying 5V to the metal lid over the die surface is expected to create a couple of volts of increase in the die surface potential if dry nitrogen is the conduction medium responsible for the Vdd-ring effect. But, I observed only a 1-2mV immediate increase in the FG voltage of transistor 3 after 5V is applied to the metal lid. I observed no further increase at all even after waiting for a couple of minutes, whereas the RC charge-up of the FG voltage of transistor 3 in the same package takes only 30 seconds resulting in a 0.5V increase. Grounding the metal-lid resulted in an immediate 1-2mV decrease in the *FG* voltage, again followed by no further change at all.

I took the package that I drilled a tiny hole through its metal lid as described in Section A.2, and applied 5V to its lid, also. Again, the behavior was exactly the same as described above. Therefore, neither air nor dry nitrogen can transport sufficient electrical charge from/to the die surface within minutes to be responsible for the Vdd-ring effect I observed.

There is an additional evidence against the dry nitrogen. After 3 months, both the RC charge-up and discharge of the *FG* voltages in hermetically sealed packages took about 20 times longer than they used to take initially. For instance, the RC charge-up of transistor 3 took 10 minutes compared to 30 seconds it took 3 months earlier. The RC discharge was also proportionally delayed, but the amplitude remained the same at 0.5V level. If the dry nitrogen is responsible for the charge transport, it is not clear why its resistivity would increase 20 times while it is enclosed under a hermetic seal, and the ambient temperature is about the same. Actually, this observation is evidence against the passivation layer, also.

A.3.3 A Hygroscopic Film on the Die Surface?

When I mentioned that the Vdd-ring effect disappears when air enters the die cavity by either removing the metal lid or drilling a tiny hole through it, Vance Tyree of MOSIS speculated the existence of a hygroscopic contaminant on the die surface. From the time wafers are fabricated to the time they are cut and packaged at a different location and company they are shipped to, a film of hygroscopic material might be formed on the die surface. When the packages are placed in an oven at 150°C, and dry nitrogen is passed through the oven, most of the moisture in this hygroscopic material will evaporate, but sufficient amount of moisture may remain to cause the conduction I observed in the hermetically sealed packages. When a hole is drilled through the metal lid, air re-hydrolizes this material on the die surface, substantially decreasing its resistivity, so that any induced charge on the die surface would leak away to the substrate in less than a second. Recall that the die surface terminates at the substrate on the periphery of the die, as explained in Section A.2.

In 3 months, more moisture from the hygroscopic film may evaporate into the dry nitrogen inside the hermetic seal at room temperature. This explains why the resistivity of the conducting medium has increased 20 times after 3 months.

In order to estimate the resistance this speculated hygroscopic film needs to exhibit in order to produce the RC behavior I observed, I performed HSPICE simulations using the circuit in Figure A.3. I attempted to duplicate my RC charge-up and discharge observations for transistor 8 in a typical hermetically sealed package, where the FG voltage in Figure A.2 increases 0.9V in less than a minute, and comes back to its initial value in less than an hour. In my HSPICE runs, the floating wire in Figure A.3 is connected to a floating gate pMOS transistor as shown in Figure A.2, the source and the bulk of the transistor are connected to a 30KΩ resistor, and the drain is grounded.

The floating wire is a metal-2 line 250μ long and 1.5μ wide. From the capacitance parameters provided by MOSIS for run *n4cp* of the HP 0.8μ process, $C_{fw-subs}$ is computed to be 23fF. Because metal-2 might be slightly closer to the die surface than it is to the substrate in this 3-metal process, and the dielectric constant is 7.0 for silicon nitride and between 3.9 and 7.0 for silicon oxynitride depending on its oxygen-nitrogen composition [30], $C_{fw-surf}$ can be assumed to be 30fF. To compute $C_{ring-surf}$, the metal-1 to substrate capacitance parameters given by MOSIS can be used, assuming that metal-1 to substrate distance is about the same as the distance from metal-3 to the die surface, which is the thickness of the passivation layers deposited on top of metal-3. I multiplied the resulting capacitance value by 6.5/3.9, where 3.9 is the dielectric constant for SiO_2, and 6.5 is my guess for the dielectric constant of the passivation layer that consists of a 0.60μ nitride and a 0.35μ oxynitride film. The resulting capacitance value is 16926fF from the whole Vdd-ring to the die surface.

The orientation of transistor 8 is similar to the one in Figure A.4, with 85μ by 300μ replaced by 355μ by 135μ. Therefore, the bulk of the charge induced over the floating wire is coming from the parallel Vdd-ring wire 135μ away, and the other portions of the Vdd-ring do not have as much contribution. Thus, I took one-fourth of 16926fF, and bumped it up a little bit to account for the proximity of metal-3 to the passivation layer, and used 4400fF for $C_{ring-surf}$.

The Vdd-ring is $1.5mm$ long on one inner side, so the die surface area enclosed by the Vdd-ring is $1.5 * 1.5 = 2.25mm^2$. To estimate the capacitance of the die surface to the substrate, I multiplied 2.25 by $10pF/mm^2$, which is the area capacitance parameter from MOSIS for the metal-3 layer, to obtain 22.5pF. The surface to substrate capacitance between the floating wire and the Vdd-ring wire, which is 135μ away, is less than 10% of 22.5pF, because Vdd-ring is 1500μ on one side, and I took this capacitance to be 2000fF. In the HSPICE runs, I used 10 RC stages between the floating wire and the Vdd-ring as shown in Figure A.3. Therefore, each C_{surf} in Figure A.3 is one-tenth of 2000fF, which is 200fF. The RC network between the floating wire and the Vdd-ring represents the 135μ surface distance, but more surface area will be receiving induced charge from the Vdd-ring. In order to model this, I added another 10 RC stages to the left of the floating wire in Figure A.3.

The only parameters left to be set in Figure A.3 are the value of R_{surf} and the number of RC stages to the right of the Vdd-ring, which model the surface path from the Vdd-ring to the die periphery terminating at the substrate. The distance from the Vdd-ring to the outer boundary of a pad cell is 140μ. The die periphery should be pretty close to the outer boundary of a pad cell. Because the 10 RC stages between the floating wire and the Vdd-ring represent a 135μ surface distance, I expect 10 to 20 RC stages to the right of the Vdd-ring in Figure A.3. Using 15 such RC stages, and setting $R_{surf} = 5 * 10^{11}\Omega$, I obtained the HSPICE simulation result shown in Figure A.5, which is very close to what I observed experimentally as the "Vdd-ring effect". The y-axis in Figure A.5 shows the floating wire voltage, and x-axis is time in seconds. The 0.23V at $t = 0$ is obtained by applying 10V to node A in Figure A.2. At $t = 5sec$, Vdd-ring is switched from 0V to 5V, and stays at 5V throughout the simulation. The RC charge-up takes 20 seconds, and RC discharge is taking

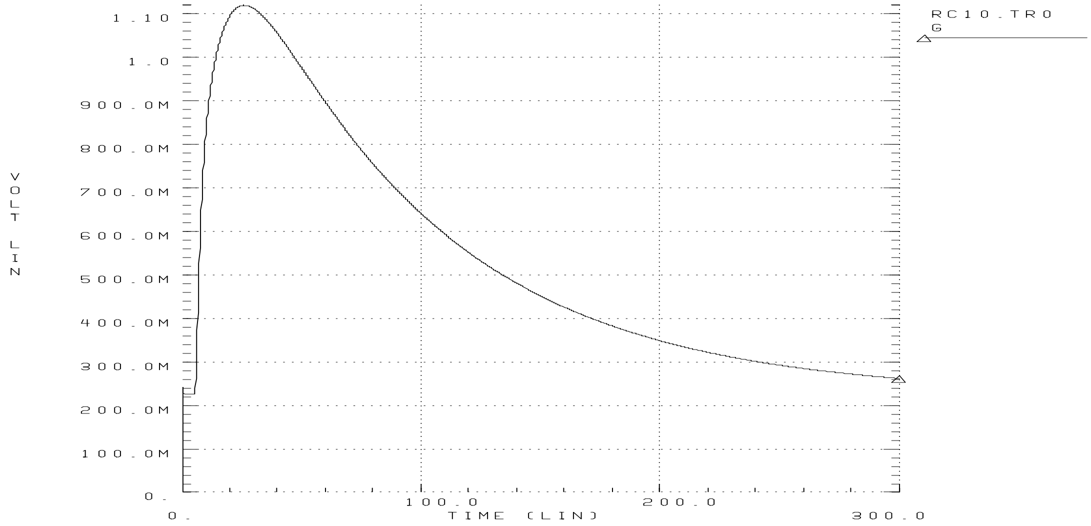

Figure A.5: The HSPICE simulation showing the Vdd-ring effect through surface conduction

more than 275 seconds.

Playing with the value of R_{surf} shows that the amplitude of the RC charge-up does not change with R_{surf}, nor the ratio of RC charge-up time to RC discharge time changes. Only the width of the curve in Figure A.5 changes. Setting $R_{surf} = 5 * 10^8\Omega$ resulted in the simulation result shown in Figure A.6, which shows that the RC charge-up and discharge cycle completed within half a second after the Vdd-ring is switched to 5V at $t = 5sec$. Half a second would not be a sufficient time for me to see the effect of the Vdd-ring while taking measurements on the taped-lid packages, because I used ordinary digital multimeters. Therefore, air increases the conductivity of the speculated hygroscopic film by a factor of 1000 or more.

A.3.4 Adsorption by the Die Surface?

More than 40 years ago, Brattain and Bardeen [3] discovered that gas adsorption onto a semiconductor surface changes the conductance of that surface. This is the operating principle for many semiconducting gas and humidity detectors today [34]. It may be possible

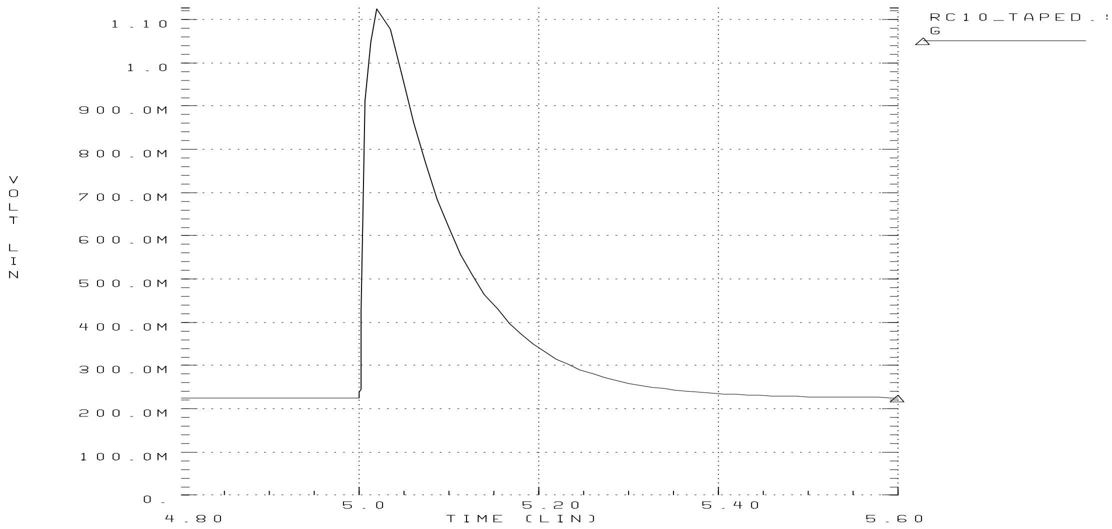

Figure A.6: The floating wire voltage with $R_{surf} = 5 * 10^8 \Omega$h

that either the water molecules themselves or other molecules in the air are adsorbed by the silicon nitride passivation layer, significantly increasing its surface conductivity. When the packages are placed in an oven with dry nitrogen passing through, some of the atoms adsorbed from the air may leave the die surface, but still leaving behind enough atoms to cause the surface conduction I have observed. After the packages are sealed with dry nitrogen inside the die cavity, more atoms from the die surface may diffuse into nitrogen very slowly, causing the surface conductivity shift I have observed in 3 months.

The materials used for gas or humidity sensors are semiconductors, but silicon-nitride is an insulator, so I could not find data in the gas sensors literature within my limited time about the adsorption properties of silicon nitride.

A.3.5 Further Evidence for My Surface Conduction Model

Further evidence supporting the surface conduction model illustrated by Figure A.3 is another experimental observation. Switching the Vdd-ring from 5V back to 0V while the *FG* voltage is increasing does not stop the increase in *FG* voltage immediately. The *FG* voltage continues to increase for a while before it turns back. HSPICE simulation showed the

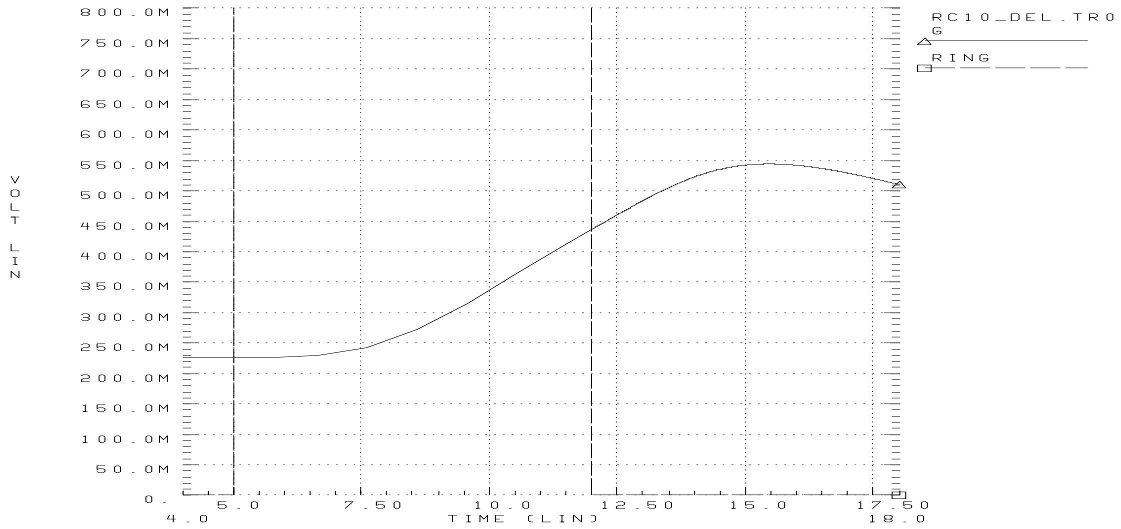

Figure A.7: The delayed response of the floating wire voltage as another evidence

same response as illustrated in Figure A.7. The floating wire voltage continued to increase for 4 seconds after the Vdd-ring voltage is switched to 0V at $t = 12sec$. $R_{surf} = 2 * 10^{12}\Omega$ is used for this simulation mimicing a package with slightly larger hygroscopic film resistivity.

One more piece of evidence for the surface conduction model is the following observation: After setting the pg voltage in Figure A.1 from 0V to 1.51V for transistor 1, the FG voltage starts from 1.18V, and drops 0.11V to 1.07V after 2.5 minutes. This behavior is due to the time necessary for $C_{fw-surf}$ in Figure A.3 to increase the potential difference across it by pushing charge into the substrate through the die surface.

Now consider the 135μ surface distance represented by the 10 RC stages in HSPICE simulations between the floating wire and the Vdd-ring in Figure A.3. Let's compute the resistance of this distance through the silicon nitride layer. Let's assume $5 * 10^{13}\Omega - cm$ for the resistivity of the nitride. Then, the resistance $R_{nitride}$ will be

$$R_{nitride} = \frac{resistivity * length}{cross\_section\_area} = \frac{(5 * 10^{13}) * (135 * 10^{-4})}{(0.6 * 10^{-4}) * (1500 * 10^{-4})} = 7.5 * 10^{16}\Omega \qquad (A.3)$$

Since there are 11 R_{surf} resistors in the 10 stage RC network, each R_{surf} will be

$7.5 * 10^{16}/11 = 6.8 * 10^{15}\Omega$. Recalling from Section A.3.3 that the RC charge-up takes 20 seconds for $R_{surf} = 5 * 10^{11}\Omega$, $R_{surf} = 6.8 * 10^{15}\Omega$ will produce an RC charge-up that takes 272000 seconds, that is 3.15 days! This clearly shows that the silicon nitride is too resistive to be responsible for the Vdd-ring effect.

A.4 Charge Measurement Results

Figures A.8 through A.10 show the trapped charge voltage measurements on 8 taped-lid and 9 hermetically sealed packages. In Figure A.8, the {42 - taped} column shows the trapped charge voltages in our taped-lid packages for transistors 4 and 10 in Table A.1, which have 42μ poly extension. Similarly, {42 - sealed} column is for transistors 4 and 10 in our sealed packages. One main conclusion from these measurements is that floating gate transistors with no or some poly extensions have negative trapped charge voltages sitting on their gates, up to almost -4V. This is in contrast with Johnson's measurements [15], who measured always positive charge on his floating gate transistors, which also had some or no poly extensions. This clearly shows the fabrication process dependence of the trapped charge polarity.

Figures A.9 and A.10 show that the polarity of the trapped charge can be both negative and positive when a metal wire is connected to a floating gate. Also, the magnitude of V_{tc} is smaller compared to the poly-only case in Figure A.8. In Figure A.9, the sealed packages display a trend of larger V_{tc} as the metal layer number increases. But, I believe that this is due to the capacitive coupling of the floating wire through the die surface to other signals in our chip, as shown in Figure A.3 for the coupling between the Vdd-ring and the floating wire. This trend is not visible for the taped packages in Figure A.9, and recall that the Vdd-ring effect does not exist for the taped packages. The substrate capacitance of the 250μ floating metal wire is 34fF, 23fF, and 24fF for metal-1, metal-2, and metal-3 layers, respectively. Considering these capacitance values, Figure A.9 does not show any significant difference in the amount of charge deposited on different metal layers.

In Figure A.10, the V_{tc} values for the 500μ and 250μ floating wires are not present for the

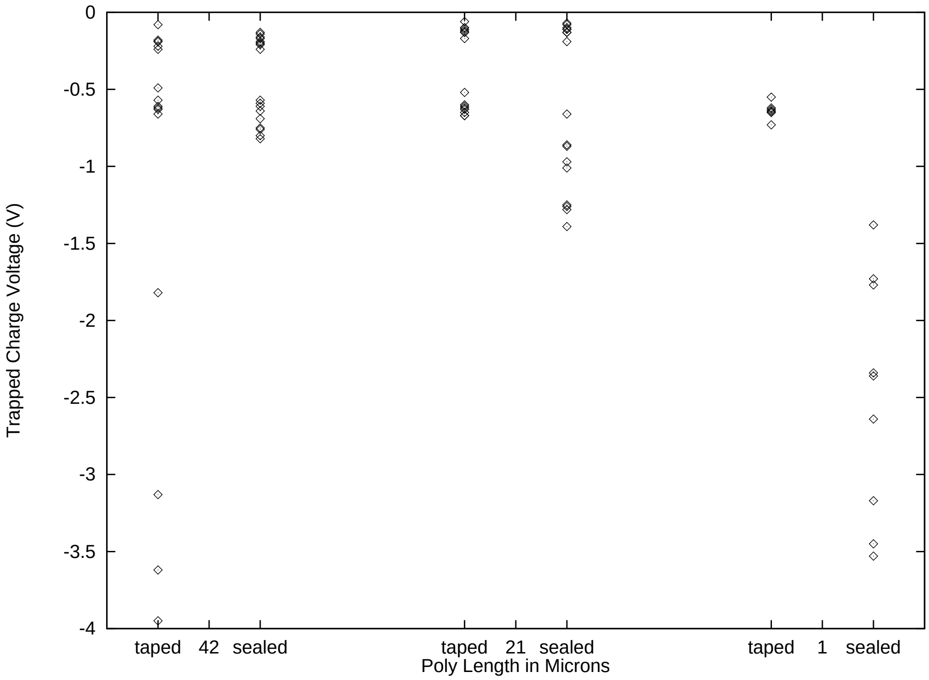

Figure A.8: V_{tc} with different lengths of poly extensions

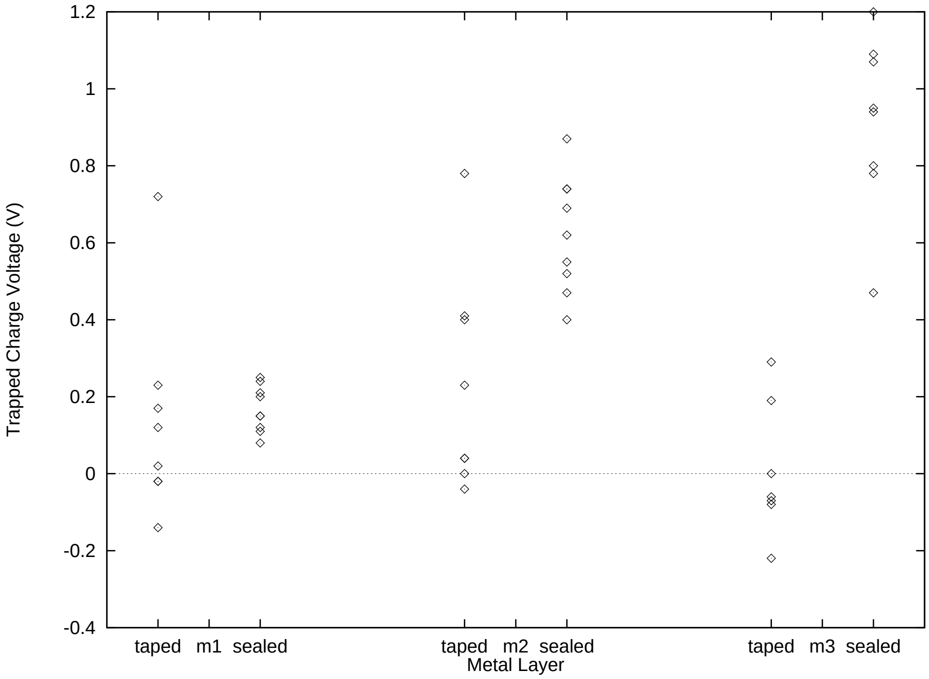

Figure A.9: V_{tc} with floating wires at different metal layers

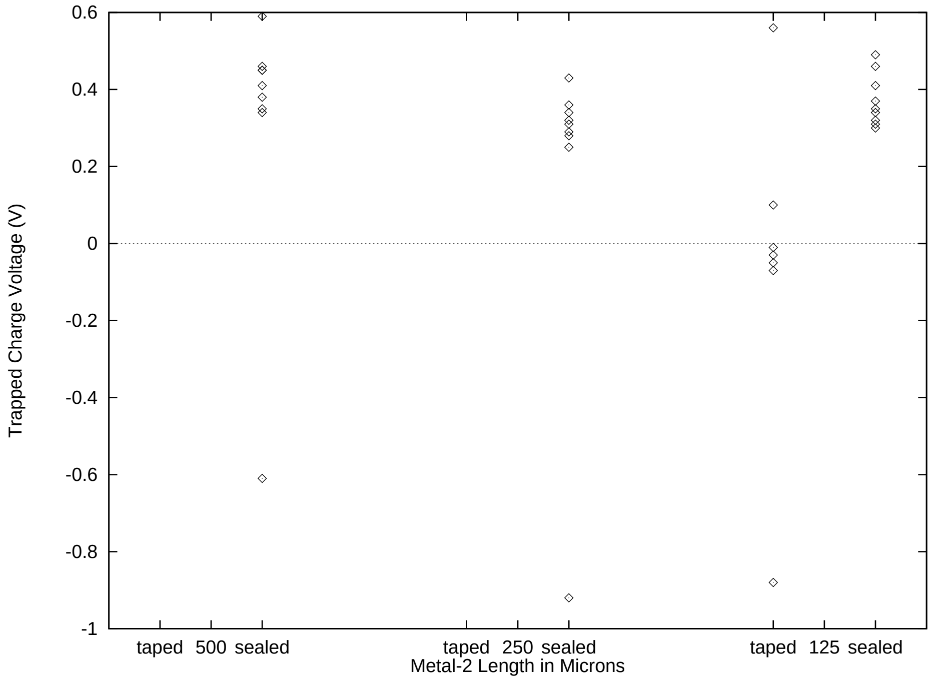

Figure A.10: V_{tc} with different lengths of floating metal-2 wires

taped packages, because the *FG* transistors these wires are connected to did not conduct until V_{pg} in Figure A.1 was around 12V, and the current steadily decreased even though all other voltages were kept fixed. On the other hand, the same *FG* transistors in the sealed packages started conducting with V_{pg} around 1V, and were stable. I have not explained this behavior, yet. The V_{tc} distribution in Figure A.10 shows that there is no noticeable change in the amount of charge deposited per unit length of a metal-2 wire as its total length changes.

I exposed some taped packages to ultraviolet light using an EPROM eraser. The trapped charge leaked away, but I noticed that the discharge rate becomes very slow when V_{tc} is low. For instance, reducing V_{tc} from -0.22V to -0.18V for transistor 4 in Table A.1 took 1 hour of ultraviolet exposure in an EPROM eraser. Therefore, it may not be feasible to zero the trapped charge during fabrication using ultraviolet light.

A.5 Conclusion

This appendix has presented experimental evidence that the die surface can act as an RC interconnect capacitively coupling a floating wire to all other signals in a chip. The resistance range for the die surface necessary for this effect is large enough so that the die surface is a perfect insulator for the fault-free operation of the chip. A circuit model for the RC interconnect effect of the die surface is presented. HSPICE simulations with this circuit model produced the same floating-wire behavior I have observed in the experiments. My experiments and HSPICE simulations show that the passivation layer or the nitrogen gas inside the die cavity is too resistive to cause the die surface act as an RC interconnect. There are two other potential candidates to explain the reduced die surface conductivity. One is a hygroscopic contaminant on the die surface that may form during the time period from the wafers are fabricated to they are cut and packaged. The other is adsorption of water molecules or some other molecules from air by the passivation layer surface. Further study is necessary to identify the actual mechanism.

This appendix has also presented trapped charge voltage measurements on floating-gate transistors with poly or metal extensions. Floating gates with poly extensions always showed negative trapped charge values, up to -4V. Floating gates with metal extensions showed both positive and negative trapped charge values within the -1V to 1V range.

References

[1] H.L. Alder and E.B. Roessler. *Introduction to Probability and Statistics.* W.H. Freeman and Company, 1964.

[2] Z. Barzilai, J.L. Carter, V.S. Iyengar, I. Nair, B.K. Rosen, J. Rutledge, and G.M. Silberman. Efficient fault simulation of CMOS circuits with accurate models. In *Proceedings of International Test Conference*, pages 520–529, October 1986.

[3] W.H. Brattain and J. Bardeen. Surface properties of germanium. *Bell Syst. Tech. J., 32 (1953) 1.*

[4] 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 Computer-Aided Design*, pages 359–369, March 1994.

[5] B. Chess and T. Larrabee. Bridge fault simulation strategies for CMOS integrated circuits. In *Proceedings of Design Automation Conference*, 1993.

[6] H. Cox and J. Rajski. Stuck-open and transition fault testing in CMOS complex gates. In *Proceedings of International Test Conference*, pages 688–694, October 1988.

[7] C. Di and J.A.G. Jess. On accurate modeling and efficient simulation of CMOS opens. In *Proceedings of International Test Conference*, pages 875–882, October 1993.

[8] P. Franco et al. Analysis and detection of timing failures in an experimental test chip. In *Proceedings of International Test Conference*, 1996.

[9] Vance Tyree (Reliability expert at MOSIS). *Private Communication.* May 1995.

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

[11] H. Hao and E.J. McCluskey. Analysis of gate oxide shorts in CMOS circuits. *IEEE Transactions on Computer-Aided Design*, pages 1510–1516, December 1993.

[12] C.F. Hawkins, J.M. Soden, A.W. Righter, and F.J. Ferguson. Defect classes - an overdue paradigm for CMOS IC testing. In *Proceedings of International Test Conference*, pages 413–425, October 1994.

[13] Alvin Jee and F. Joel Ferguson. Carafe: An inductive fault analysis tool for CMOS VLSI circuits. In *Proceedings of the IEEE VLSI Test Symposium*, 1993.

[14] N.K. Jha and J.A. Abraham. Design of testable CMOS logic circuits under arbitrary delays. *IEEE Transactions on Computer-Aided Design*, pages 264–269, July 1985.

[15] S. Johnson. Residual charge on the faulty floating gate MOS transistors. In *Proceedings of International Test Conference*, October 1994.

[16] H. Konuk, F.J. Ferguson, and T. Larrabee. Accurate and efficient fault simulation of realistic CMOS network breaks. In *Proceedings of Design Automation Conference*, pages 345–351, June 1995.

[17] Haluk Konuk and F. Joel Ferguson. An unexpected factor in testing for CMOS opens: The die surface. In *Proceedings of the IEEE VLSI Test Symposium*, 1996.

[18] Haluk Konuk and F. Joel Ferguson. Oscillation and sequential behavior caused by interconnect opens in digital CMOS circuits. *Submitted to Design Automation Conference*, June 1997.

[19] Haluk Konuk, F. Joel Ferguson, and Tracy Larrabee. Charge-based fault simulation for CMOS network breaks. *To Appear in IEEE Transactions on Computer-Aided Design.*

[20] Tracy Larrabee. Test pattern generation using Boolean satisfiability. *IEEE Transactions on Computer-Aided Design*, pages 6–22, January 1992.

[21] K.-J. Lee and M.A. Breuer. On the charge sharing problem in CMOS stuck-open fault testing. In *Proceedings of International Test Conference*, pages 417–425, October 1990.

[22] W. Lukaszek, W. Dixon, and E. Quek. Charging studies using CHARM2 wafer surface charging monitor. *Nuclear Instruments and Methods in Physics Research B74*, pages 301–305, 1993.

[23] W.M. Maly, P.K. Nag, and P. Nigh. Testing oriented analysis of CMOS ICs with opens. In *Proceedings of International Conference on Computer-Aided Design*, pages 344–347, November 1988.

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

[25] Peter C. Maxwell, Robert C. Aitken, and Leendert M. Huisman. The effect on quality of non-uniform fault coverage and fault probability. In *Proceedings of International Test Conference*, pages 739–746, October 1994.

[26] E.J. McCluskey. *Logic Design Principles with Emphasis on Testable Semicustom Circuits.* Prentice Hall, 1986.

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

[28] B.L. Prickett, J.M. Caywood, and R.K. Ellis. Trapping in tunnel oxides grown on textured polysilicon. In *The 21st Annual Proceedings on Reliability Physics*, 1983.

[29] J. Figueras R. Rodriguez-Montanes, E.M.J.G. Bruls. Bridging defects resistance measurements in a CMOS process. In *Proceedings of International Test Conference*, pages 892–899, October 1992.

[30] P. Rabiller, J.E. Klemberg-Sapieha, M.R. Wertheimer, and A. Yelon. Electrical properties of a-SiOxNy:H films prepared by microwave PECVD. In *IEEE 3rd Intl. Conf. on Conduction and Breakdown in Solid Dielectrics*, 1989.

[31] S.M. Reddy, M.K. Reddy, and J.G. Kuhl. On testable design for CMOS logic circuits. In *Proceedings of International Test Conference*, pages 435–445, October 1983.

[32] M. Renovell and G. Cambon. Electrical analysis and modeling of floating-gate fault. *IEEE Transactions on Computer-Aided Design*, pages 1450–1458, November 1992.

[33] Jeffrey Rogenski and F. Joel Ferguson. Characterization of opens in logic circuits. In *Proceedings of IEEE ASIC Conference*, 1994.

[34] G. Sberveglieri. Recent developments in semiconducting thin-film gas sensors. *Sensors and Actuators, B-Chemical*, pages 101–109, February 1995.

[35] B.J. Sheu, W.-J. Hsu, and P.K. Ko. An MOS transistor charge model for VLSI design. *IEEE Transactions on Computer-Aided Design*, pages 520–527, April 1988.

[36] A.D. Singh, H. Rasheed, and W.W. Weber. IDDQ testing of CMOS opens: An experimental study. In *Proceedings of International Test Conference*, 1995.

[37] M. Syrzycki. Modeling of gate oxide shorts in MOS transistors. *IEEE Transactions on Computer-Aided Design*, pages 193–202, March 1989.

[38] K.M. Thompson. Intel and the myths of test. In *The Keynote Address of International Test Conference*, October 1995.

[39] B. Underwood, S. Kang, W.-O. Law, and H. Konuk. Fastpath: Robust path-delay test generator for standard scan designs. In *Proceedings of International Test Conference*, pages 154–163, October 1994.

[40] N.P. van der Meijs and A.J. van Genderen. An efficient finite element method for submicron IC capacitance extraction. In *Proceedings of Design Automation Conference*, June 1989.

[41] J.A. Waicukauski, E.B. Eichelberger, D.O. Forlenza, E. Lindbloom, and Th. McCarthy. Fault simulation for structured VLSI. In *VLSI Systems Design*, pages 20–32, Dec. 1985.

[42] 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. In *Proceedings of Design Automation Conference*, pages 726–729, June 1989.