# ATPG for Noise-Induced Switch Failures in Domino Logic 

Rahul Kundu and R. D. (Shawn) Blanton<br>Center for Silicon System Implementation<br>ECE Department<br>Carnegie Mellon University<br>Pittsburgh, PA 15213-3890


#### Abstract

Domino circuits have been used in most modern high-performance microprocessor designs because of their high speed, low transistor-count and hazard-free operation. However, with technology scaling, domino circuits are increasingly susceptible to switch failures due to various noise sources that include crosstalk, charge sharing and leakage. To test for such failures in a manufactured chip, we describe a test pattern generation methodology that generates specific test patterns to target such failures. These test patterns activate noise from multiple sources such that their combined effect causes a switch failure at a domino gate output. In addition, the test patterns propagate the resulting error to an observable output within the duration of the circuit's clock cycle. The methodology has been implemented and validated using a domino multiplier circuit.


## 1 Introduction

Domino logic circuits [1] are extensively used in high performance microprocessors [2-6] since they offer several advantages over static CMOS logic, namely higher speed, reduced transistor-count (resulting in reduced die area) and hazard-free operation. However, with technology scaling, designers find it difficult to deploy dynamic logic [7] because it has an increased susceptibility to switch failure (i.e. an erroneous gate transition) due to noise and process variations. Static CMOS on the other hand is very robust to switch failure and is more likely to exhibit only delay failures due to noise. In this paper, we study noise-induced switch failures in domino logic and describe a methodology to derive test vectors that can be used to test a domino logic circuit robustly for such failures.

## 2 Switch Failures

We define a switch failure [8, 9] as an irreversible and erroneous transition of a domino gate output. Several sources of noise can cause switch failures in domino logic circuits. Of these sources, crosstalk noise caused by capacitive coupling to neighboring lines, subthreshold leakage and charge sharing are especially important. With technology scaling, the importance of crosstalk noise and subthreshold leakage will increase. Although, the effect of charge sharing is not expected to increase [6], it is a significant problem for domino circuits in current technologies and may combine with other sources to cause a switch failure.

To generate a test that satisfies the complicated timing and logic requirements to detect a switch failure, we use the algorithm TEST_GEN outlined in Figure 1. TEST_GEN follows a PODEM-like [12] approach. Initially, logic $x$ is assigned to all primary inputs. Then the algorithm proceeds in a recursive fashion as follows. At every level of recursion, it assigns a logic value at time $t=0$ to a previously unspecified primary input
and then calculates all the logic and timing implications due to that assignment. (The calculation of the timing and logic changes due to any primary input assignment is performed in the Timed_Imply routine of Figure 1.) The newly calculated logic and timing information is used to determine if a switch failure is possible at the targeted gate using a method detailed in Section 3. (This is performed in the conflict_detect routine of Figure 1.) If the failure is still possible, the algorithm assigns another primary input at the next level of recursion. If the failure is no longer possible, the algorithm backtracks from the last decision and returns to the previous level of recursion. The algorithm returns a test pattern if all the conditions for switch failure are satisfied and the resulting error has been propagated to a primary output within the duration of the circuit's specified clock cycle.

```
TEST_GEN (victim gate)
begin
    if (error at PO), return SUCCESS
    if (conflict_detect), return FAILURE
    (k, vk})= objective_domino(
    (j, vj) = backtrace_domino(k, vk}
    Timed_Imply(j,vj)
    if TEST_GEN()=SUCCESS, return SUCCESS
    Timed_Imply }(j,\overline{\mp@subsup{v}{j}{}}
    if TEST_GEN()=SUCCESS, return SUCCESS
    Timed_Imply(j,x)
    return FAILURE
end
```

Figure 1: Pseudocode for the test generation algorithm.
TEST_GEN has two major differences from a standard PODEM based test-generation approach [12]. Each difference is clarified in detail next.

1. The algorithm uses a time-based ATPG as opposed to a logic-only ATPG. This means the algorithm maintains a timing window (i.e. the minimum and maximum time that a signal can transition) with every signal line along with the traditional logic value of either 0 , 1 , or $x$. The timing window changes dynamically during the test generation process and is computed in a manner similar to that described in [11]. We also understand that the gate delays vary as a result of switching activities of neighboring wires (due to the effect of crosstalk on delay), so this effect is calculated dynamically during the ATPG process as well [13].
2. To determine if a switch failure has occurred at a gate output, we calculate the maximum possible noise effect at a dynamic node of a gate from the current signal line values and transition windows. The noise calculation is described in greater detail in the next section.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ICCAD'03, November 11-13, 2003, San Jose, California, USA.

## 3 Maximizing Noise

Consider the complex domino gate shown in Figure 2. Assume that at a given stage of ATPG, we know that $\mathrm{A}=1, \mathrm{~B}=0$, and there is a crosstalk glitch present on line B. For this circuit, the crosstalk effect is maximized if the remaining inputs are assigned $\mathrm{C}=1$, either $\mathrm{D}=0$ or $\mathrm{E}=0$, and one of $F$, G and H is 0 . Leakage is maximized when $\mathrm{C}=1, \mathrm{D} \oplus \mathrm{E}=0$, and only two of the signals $\mathrm{F}, \mathrm{G}$ and H are logic 1. Charge sharing is maximized if the circuit values are $\mathrm{C}=\mathrm{H}=0$ and $\mathrm{D}=\mathrm{E}=\mathrm{F}=\mathrm{G}=1$. We can observe that some of the requirements for maximization of one noise effect conflicts with the requirements of another. For example, the requirements for maximum charge sharing conflict with those of leakage. In our work, we independently calculate the maximum noise effect due to each noise source and combine their effects. Obviously, there is some overestimation of the total noise effect when calculating the effect of each noise source independently. However, this overestimation is reduced during ATPG as more and more inputs are specified. Next, we describe how the maximum noise effect due to each source is determined.


Figure 2: A domino gate with partially-specified input values.

### 3.1 Crosstalk

Crosstalk noise is dependent on how many aggressors (i.e. signal lines capacitively coupled to a victim line) transition and the relative timing of their transitions. A detailed procedure for conservatively estimating the impact of crosstalk due to a partially-specified vector is presented in [11]. In addition, we utilize the methodology presented in [11] to calculate the maximum discharge $\Delta Q_{\text {cross }}$ from the dynamic node.

### 3.2 Charge Sharing

To estimate the maximum voltage drop due to charge sharing for a partially-specified vector $V_{i}$, the partially-specified vector is converted to a fully-specified vector $V_{j}$. For example, assume a partially-specified $V_{i}$ has established gate inputs $\mathrm{A}=1$ and $\mathrm{B}=0$ for the gate shown in Figure 3a. The fully-specified vector $V_{j}$ of $\mathrm{A}=\mathrm{C}=\mathrm{D}=\mathrm{E}=\mathrm{G}=1$ and $\mathrm{B}=\mathrm{F}=0$ maximizes charge sharing since this $V_{j}$ connects the dynamic node to the maximum number of intermediate nodes (i.e. 2, 3 and 4 ) without creating a discharge path to ground. (See Figure 3b.)

The conversion of a partially-specified vector to a fully-specified vector can be achieved using two depth-first traversals of a gate's transistor schematic. Once the fully-specified vector is obtained, we calculate the voltage drop due to charge sharing using the model presented in [10]. In [10], the circuit is converted into a network of capacitances, where some of the capacitances (referred to as $C_{V D D}$ ) are connected to $V_{D D}$ and the others (referred to as $C_{G N D}$ ) are connected to ground. The resulting voltage

(a)

(b)

Figure 3: (a) A partially-specified vector $V_{i}$ and (b) a corresponding fullyspecified vector $V_{j}$ constructed to maximize charge sharing.
$v_{D}$ due to charge sharing is then calculated using equation 1 ,

$$
\begin{equation*}
v_{D}=\frac{V_{D D} \times C_{V D D}+Q_{i}}{C_{V D D}+C_{G N D}} \tag{1}
\end{equation*}
$$

where $Q_{i}$ is the initial stored charge at the dynamic node after precharge. In [14], it is shown that this model can accurately predict switch failure due to charge sharing.

### 3.3 Leakage

Using the BSIM2 MOSFET model [15], the expression for subthreshold leakage current through a transistor is given by

$$
\begin{equation*}
I_{l}=A \frac{W}{L} e^{q / k T\left(v_{g s}-V_{T H 0}-\gamma \times v_{s b}+\eta \times v_{d s}\right)} \times\left(1-e^{-v_{d s} /\left(\frac{k T}{q}\right)}\right) \tag{2}
\end{equation*}
$$

where $W$ and $L$ are the channel width and length of the transistor, respectively, $A$ is a constant, $v_{g s}$ is the gate-to-source voltage of the transistor, $V_{T H 0}$ is the threshold voltage, $v_{d s}$ is the drain-to-source voltage across the transistor, $K T / q$ is the thermal voltage, $\gamma$ is the body-effect coefficient, and $\eta$ is the DIBL coefficient. From this equation, we can observe that the leakage increases with $v_{d s}$. Thus, when multiple off-transistors are stacked in series, their leakage decreases significantly since $v_{d s}$ across each transistor is reduced. Thus, the maximum leakage current through a transistor $t_{i}$ occurs when all other transistors in series with $t_{i}$ are turned on. In addition, the leakage depends only on the $W / L$ ratio of the off transistor $t_{i}$. Since $L$ is identical for all transistors in digital circuits, the problem reduces to maximizing the cumulative $W$ of all off transistors $t_{i}$ with the remaining series transistors being on. This optimization is easily mapped to identifying the maximum cut of a graph model of the evaluate chain. We have shown that the max-cut of a graph corresponding to a domino logic evaluate chain can be found in polynomial time [14].

## 4 Combination of Noise Sources

Our methodology for integrating noise due to multiple noise sources is based on the following observation: If the reduction of voltage at the dynamic node is not too severe, then the various noise mechanisms act independently. For example, in Figure 5 some amount of charge $\Delta Q_{\text {cross }}$ is removed from the dynamic node due to crosstalk. Simultaneously, there is

(a)

(b)

Figure 4: (a) An example domino gate with various $W$ values; the width of each transistor is shown next to the transistor. (b) The corresponding graph for estimating maximum leakage. The quantities in parentheses indicate the transistor width $W$ as edge weight. The max-cut of the graph is shown using a dotted line.
also a reduction of the dynamic node voltage due to charge sharing among circuit nodes $1,2,3$ and 5 . If the amount of voltage reduction at the dynamic node due to noise leaves the victim transistor MNA in saturation, the current due to crosstalk is independent of the voltage at the dynamic node. From Figure 5, it is also observed that the victim transistor affected by a crosstalk glitch is always outside any charge sharing path, meaning that crosstalk discharge occurs through a separate path to ground. In other words, crosstalk discharge does not involve any of the nodes that participate in charge sharing. Thus, the charge loss due to crosstalk is independent of the charge redistribution due to charge sharing. Given that the initial charge stored in the dynamic node is $Q_{i d}$, crosstalk drains $\Delta Q$ from the dynamic node and charge sharing causes a redistribution of the dynamic node charge, causing a reduction of the voltage by some factor $K$. The two effects can be combined to give $v_{D}=\frac{Q_{i d}-\Delta Q}{C_{i n v}} \times K$, where $v_{D}$ is the voltage at the dynamic node and $C_{i n v}$ is the input capacitance of the static inverter connected to the dynamic node. Hence, to obtain the final voltage due to the combined effect of all three noise sources, we independently derive the charge loss $\Delta Q_{\text {cross }}$ due to crosstalk, $\Delta Q_{\text {leak }}=\int I_{l} d t$ due to leakage, $C_{V D D}$ and $C_{G N D}$ due to charge sharing, and combine them using equation 3 .

$$
\begin{equation*}
v_{D}=\frac{V_{D D} \cdot C_{V D D}+Q_{i}-\Delta Q_{\text {cross }}-\Delta Q_{\text {leak }}}{C_{V D D}+C_{G N D}} \tag{3}
\end{equation*}
$$



Figure 5: An example domino gate that has both crosstalk discharge due to a glitch on line A and charge sharing due to device capacitances at nodes 1, 2, 3 and 5.

If the dynamic node voltage $v_{D}$ predicted by equation 3 is less than or equal to the switching threshold of the output inverter, the failure is possible. Otherwise, the failure is not possible and the test generation process has to backtrack from a previous circuit input assignment.


Figure 6: Hspice simulation results showing the voltage at the dynamic node due to crosstalk only, charge sharing only, and crosstalk and charge sharing together.

Figure 6 shows the Hspice waveforms of a representative domino gate. Here crosstalk alone removes about $\frac{1}{3}$ of the initial charge from the dynamic node and charge sharing alone causes a reduction of the initial node voltage by $\frac{2}{3}$. When crosstalk and charge sharing occur together, our model predicts a reduced voltage of $\frac{4}{9} V_{D D}$ which is corroborated by the Hspice simulation shown in Figure 6.

## 5 Simulation Results

We applied our method to a dual-rail domino Wallace tree multiplier circuit [16], implemented in a 2-metal, $0.18 \mu \mathrm{~m}, 1.8 \mathrm{~V}$ technology. The multiplier consists of 1806 transistors, arranged in 43 identical adder cells that formed a total of 172 domino gates. A layout of the multiplier was generated automatically using an industrial place and route tool. A netlist containing parasitic capacitances was extracted from the layout using Space [17].

In order to validate our methodology for combined analysis of charge sharing, crosstalk and leakage, we inserted several noise failures into the multiplier circuit. The method for inserting a noise failures is as follows. We first select a test vector that creates a small crosstalk glitch on a victim line. No charge sharing exists at the destination gate of the victim line since device capacitances are initially too small. Also, the glitch by itself is not large enough to cause a switch failure at any of the destination gates of the victim line. Next, we incrementally increase the device capacitances of the transistors of the victim gate to increase charge sharing. The increase of the device capacitances is continued until we observe an error at one of the destination gates. Once we observe the error in Hspice, the modified netlist that exhibits the failure is used as input to our test generation tool. If the test-generation methodology is sound, we expect the test generation tool to identify the failure. In addition, the tool should generate all the tests, including the original input vector used for creating the failure, that can both activate the failure and propagate the resulting error to a circuit output.

The outcome of test generation for three failures is listed in Table 1. Column one of Table 1 shows the failure identification number. Column two shows the total number of test vectors generated for the failure. Column three shows the number of generated vectors that succeeded in detecting the switch failure. For each failure, our test generator identifies the failure at the victim line. The test generator also provides a set of test vectors for each failure and in every case, the test set subsumes the original test

| Failure id <br> number | Number of tests <br> generated | Number of successful <br> tests |
| :---: | :---: | :---: |
| 1 | 9 | 2 |
| 2 | 2 | 1 |
| 3 | 8 | 6 |

Table 1: Test vectors obtained for switch failures in the presence of crosstalk, charge sharing and leakage.
vector. It is observed that the percentage of tests successful in creating a failure varies from $20 \%$ to $75 \%$ since (1) our test generation methodology always overestimates the amount of noise and (2) has inherent inaccuracy in the timing calculation since the effects of interconnect are not dealt with comprehensively.

Next, our test generation tool was used to analyze the entire multiplier circuit. The nominal multiplier circuit does not contain any testable switch failures, so we introduced manufacturing variations into the circuit. The methodology for incorporating manufacturing variations is described in [14]. Here, we use a parameter $\alpha$ to quantify the level of manufacturing variations, where $\alpha=0.0$ represents the nominal design and increasing values of $\alpha$ denote an increasing amount of manufacturing variations. We performed our analysis for $\alpha$ values ranging from 0.0 to 0.9 . Table 2 shows the number of detected switch failures for each value of $\alpha$. The analysis includes the combined effect of crosstalk, charge sharing, leakage and the delay variation of all lines in the circuit due to coupling. For each failure, all the test vectors were generated. Column one of Table 2 shows the value of $\alpha$ considered. Column two shows the number of testable failure sites identified for each value of $\alpha$. Column three shows the number of test vectors generated for all the failure sites for each value of $\alpha$. Column four shows the total amount of CPU time taken by timed ATPG for generating all the tests for each value of $\alpha$. Because the total number of test vectors generated is large, we only validated the test vectors for $\alpha=0.3$ using Hspice simulation of the entire netlist. Of the 15 test vectors generated for $\alpha=0.3$, five were successful in activating a failure and propagating the resulting error to a circuit output. The remaining ten vectors were not successful because our analysis of delay in the presence of crosstalk accounts for the simultaneous switching of multiple gate inputs in a minmax fashion only, and therefore introduces uncertainty in the computed delays within timed ATPG.

| 最 | No. of <br> failures | No. of <br> tests generated | CPU <br> time (secs) |
| :---: | :---: | :---: | :---: |
| 0.0 | 0 | 0 | 0.0 |
| 0.1 | 0 | 0 | 0.0 |
| 0.2 | 2 | 15 | 19.2 |
| 0.3 | 3 | 31 | 46.1 |
| 0.4 | 5 | 41 | 68.0 |
| 0.5 | 6 | 84 | 98.7 |
| 0.6 | 9 | 101 | 116.4 |
| 0.7 | 12 | 168 | 170.1 |
| 0.8 | 14 | 180 | 192.2 |
| 0.9 | 20 | 268 | 294.4 |

Table 2: Number of testable failures for each value of $\alpha$.

## 6 Summary

In this paper, we described how vector-dependent effects of multiple noise sources can be combined to cause erroneous operation (i.e. switch failure) in a domino logic circuit. Specifically, we demonstrated how test input vectors can be derived to activate and observe switch failures using a PODEM-based timed test-generation framework that combines the effect
of crosstalk, charge sharing and leakage. Application of the technique to a multiplier circuit showed that effective test vectors can be obtained using reasonable CPU resources.

## References

[1] R. H. Krambeck, C. M. Lee and H. F. S. Law, "High-speed Compact Circuits with CMOS," IEEE Journal of Solid-State Circuits, vol. SC-17, no. 3, pp. 614619, June 1982.
[2] R. Heald, "A Third-generation SPARC V9 64-b Microprocessor," IEEE Journal of Solid-State Circuits, vol. 35, no. 11, pp. 1526-1535, Nov. 2000.
[3] D. R. Bearden, "A 780 MHz PowerPC/sup TM/ Microprocessor with Integrated L2 Cache," in International Solid-State Circuits Conference, pp. 90-91, Aug. 2000.
[4] J. Silberman et. al., "A 1.0-GHz Single-issue 64-bit PowerPC Integer Processor," IEEE Journal of Solid State Circuits, pp. 1600-1608, Nov. 1998.
[5] D. W. Bailey, "High Performance Alpha Microprocessor Design," Tech. Rep., Compaq Computer Corporation, 2000.
[6] R. Kumar, "Interconnect and Noise Immunity Design for Pentium 4 Processor," in Intel Technology Journal, Jan. 2001.
[7] M. Allam, M. Anis and M. Elmasry, "Effect of Technology Scaling on Digital CMOS Logic Styles ," in Custom Integrated Circuits Conference, pp. 401-408, 2000.
[8] R. Kundu, and R. D. Blanton, "Identification of Crosstalk Switch Failures in Domino CMOS Circuits," in International Test Conference, pp. 502-509, Oct. 2000.
[9] D. Somasekhar, S. H. Choi, K. Roy, Y. Ye, and V. De, "Dynamic Noise Analysis in Precharge-Evaluate Circuits ," in Design Automation Conference, pp. 243246, June 2000.
[10] K. Heragu, M. Sharma, R. Kundu, and R. D. Blanton, "Testing of Domino Circuits Based on Charge Sharing," in VLSI Test Symposium, pp. 396-403, April 2001.
[11] R. Kundu and R. D. Blanton, "Timed Test Generation for Crosstalk Switch Failures in Domino CMOS," in VLSI Test Symposium, pp. 379-385, April 2002.
[12] P. Goel, "An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits," in IEEE Transactions on Computers, vol. C-30, no. 3, pp. 215-222, March 1981.
[13] R. Kundu and R. D. Blanton, "Timing Analysis of Domino Logic in the Presence of Crosstalk," in Technical Report, Department of Electrical and Computer Engineering, Carnegie Mellon University, no. CSSI 02-21, June 2002.
[14] R. Kundu, "Test Generation for Noise-Induced Switch Failures in Domino CMOS Circuits," in PhD Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, Aug. 2003.
[15] Avant! Corporation, Star-HSPICE Manual. Santa Clara, CA, 2000.
[16] B. Ramasubramanian, H. Schmit and L. R. Carley, "Mixed-swing Quadrail for Low Power Dual-rail Domino Logic," in 1999 International Symposium on Low Power Electronics and Design, pp. 82-84, Aug. 1999.
[17] N. P. Van der Meijs and A. J. Van Genderen, "An Efficient Finite Element Method for Submicron IC Capacitance Extraction," in Design Automation Conference, pp. 678-681, June 1989.

