# Node Sensitivity Analysis for Soft Errors in CMOS Logic

Balkaran S. Gill[1], Chris Papachristou[1], Francis G. Wolff[1], and Norbert Seifert[2]
[1]Department of Electrical Engineering and Computer Science
Case Western Reserve University, Cleveland, Ohio
[2]Logic Technology Development Q&R, Intel Corporation; Hillsboro, Oregon

## Abstract

*In this paper, we introduce an approach for computing soft error susceptibility of nodes in large CMOS circuits at the transistor level. The node sensitivity depends on the electrical, logic, and timing masking. An efficient technique is developed to compute the electrical masking of nodes from the characterization tables and the inverse pulse propagation. We generated these tables for every logic cell of the library using Spice simulations for a 100nm process technology. An additional technique to compute the logic masking of the transistor nodes using an automatic test pattern generation tool is described. Our results show that our approach has Spice like accuracy but it is several orders of magnitude faster than Spice. This approach can be used to analyze the vulnerability of circuits to single event upsets at the chip level and results are provided for ISCAS85 benchmark circuits.*

## 1. Introduction

Single Event Upsets (SEUs) in modern VLSI systems are a major reliability concern. These upsets originate from two primary sources: cosmic ray particles occurring in the space environment and alpha particles emitted from the radioactive decay of uranium and thorium impurities located within the chip itself such as the silicon die, interconnects and ceramic packaging. Soft errors due to SEUs have been a known problem in the semiconductor memories for quite some time. However, due to faster clock rates and shrinking process technologies SEUs are now effecting CMOS logic [2, 6, 15, 17, 19, 22, 23]. A recent study shows that the projected soft error rate in logic will dominate in microprocessors [22].

SEUs occur on the sensitive nodes of transistors of the logic circuits and create transient current pulses which may lead to soft errors. The Soft Error Rate (SER) of a node in a logic circuit is the failure rate of the node due to SEUs. The SER of the individual nodes is non-uniform in nature as it depends on the transistor strength, the capacitive load at the transistor node, $V_{dd}$ value, temperature etc. and furthermore the probability of the input state of the circuit which sensitized distinct data paths. Thus, the SER of the transistor nodes is a measure of the node sensitivity in a circuit. The SER of a circuit can be reduced by using circuit hardening techniques but it can result in unacceptable design overhead. Hence, by analyzing and quantifying the node SER, we can employ appropriate techniques to reduce the SER of highly sensitive nodes which will potentially yield a lower overall product SER.

In this paper, our focus is to develop a new analysis approach to determine the sensitivity of the transistor nodes to soft errors in CMOS logic circuits. To compute the SER of transistor nodes, we have developed a methodology to calculate the electrical and logic masking of particle strike generated pulses. The electrical masking is computed based on the characterization of the transistor nodes of the logic cell library using Spice. The logic masking is determined using an Automatic Test Pattern Generation (ATPG) tool. Our work builds upon the technique developed in [21] for the inverse pulse propagation and complements it by adding a methodology to efficiently incorporate gate-level logic masking and logic cell library characterization.

**Related Work.** The SER of nodes can be computed using Spice simulations of the circuits as discussed in [13] but for large circuits, it can be intractable. In [15], an approach for reducing the soft error failure rate in logic circuits by applying a partial protection technique based on node sensitivity analysis was proposed but an efficient methodology for node sensitivity analysis was not developed. Their approach uses Spice simulations to compute the SER of circuit nodes which can be very time consuming in large circuits. The compound noise (cross talk, ground bounce, substrate coupling noise, radiation induced noise) effects on the gate nodes were analyzed [23] but the details of SEU pulse propagation, latching and SER of transistor nodes were not described.

This paper is organized as follows. In section 2, pulses generated due to SEUs in the logic circuits, masking effects, and soft errors in logic circuits are discussed. In section 3, we describe the details of our approach for the node sen-

sitivity analysis. Section 4 provides the runtime efficiency and accuracy of the approach against Spice simulations and also the sensitivity of nodes in ISCAS85 circuits. Section 5 concludes this paper.

## 2. Soft Errors in Logic Circuits

### 2.1. Background

A Single Event Upset (SEU) in semiconductor devices occurs due to a charged particle strike at a sensitive node. In case of CMOS circuits, a sensitive node is the drain of the OFF-transistor. The particle strike creates electron-hole pairs on its track. The drift and the diffusion of electron-hole pairs generates a current pulse. The duration and the amplitude of the current pulse depends on the striking particle energy, the transistor strengh, the load capacitance, $V_{dd}$ value. This pulse can have positive or negative magnitude depends whether the particle hits at the drain of the OFF NMOS or PMOS transistor. For transient simulations of the circuit for SEUs, the charge collection process is accounted for in our simulations by a current source connected to the circuit node whose failure rate needs to be determined and the substrate [1, 9, 16].

### 2.2. SEU Masking

The transient current pulse generated due to a SEU, results in a voltage pulse at the affected node. A soft error in a combinational circuit occurs when a SEU generated pulse at a node propagates to the observable output(s) and captured by the storage element(s) such as registers, memory, flip-flops, latches. The SEU pulse generated at the struck node can only be captured by the registers if it is not masked by the following three masking effects:

**Electrical Masking:** The SEU pulse will be masked if it attenuates before arriving at the circuit output.

**Logic Masking:** The propagation of the SEU generated pulse is blocked by the state of the circuit inputs.

**Timing Masking:** The SEU pulse arrives earlier or later than the data latching window.

### 2.3. Increasing Soft Errors Susceptibility of Logic Circuits

As the process technology shrinks and supply voltage decreases, the charge stored at nodes of logic circuits reduces because $Q_{node} = C_{node} \times V_{dd}$, which is the primary reason of increasing sensitivity of nodes for soft errors. Additional reasons are due to the reduction in electrical and timing masking. The electrical masking effect decreases due to faster gates and the technology trend towards the reduced logic depth between registers [15]. The reduction in timing masking is due to the high operating frequencies which increases the probability of a SEU pulse being latched. Thus, in Very Deep Sub-Micron (VDSM) technologies the soft errors in logic circuits are becoming a reliability problem [22].

## 3. Node Sensitivity Analysis Approach

We develop an approach to analyze the sensitivity of nodes of a combinational circuit. In combinational circuits, there can be multiple logic paths from a node to observable outputs. The sensitivity metric of a node for a particular path, $S_{node,path}$, is the product of three factors: the SEU rate of a node including electrical masking, the probability when the pulse is not logically masked, and the ratio of latching window to the clock cycle. This can be expressed as:

$$S_{node,path} = U_{node,path} \times P_{node,path} \times T_{node,path} \quad (1)$$

where $U_{node,path}$ is the rate of SEUs when a SEU results in a voltage pulse that is of sufficient amplitude and width to propagate through the functionally sensitized path without significant attenuation. $P_{node,path}$ is the probability of the input vectors which make the path functionally sensitized for the pulse propagation from the node to the observable output. $T_{node,path}$ is the fraction of the time during a clock cycle when the arrived pulse at the observable output can be captured by the flip-flop. Conceptually, equation (1) is similar to the equation described in [15] for the soft error susceptibility of a node and in [23] for the softness of a node. Our approach to determine $S_{node}$ considers all the paths from a node to the observable outputs. This is because a node can have multiple sensitivities based on paths from it. We define the sensitivity of a node, $S_{node}$, as the maximum sensitivity of a path over all paths as shown in equation (2):

$$S_{node} = \max \{S_{node,path}\} \quad (2)$$

In this paper, the two key ideas to determine the node sensitivity are: 1) The inverse pulse propagation through a $path$ from the circuit output to the $node$ under consideration to determine $U_{node,path}$. Using the inverse pulse propagation helps to reduce the time complexity and increase the accuracy to determine $U_{node,path}$. 2) Computing the probability of the observable vectors for a path using an ATPG tool. In the following sub-sections, we discuss our approach to determine $U_{node,path}$, $P_{node,path}$, and $T_{node,path}$.

### 3.1. SEU Rate of a Node for a given path

To calculate the SEU rate of a node, $U_{node,path}$, we need to know three quantities: a) $\sigma_{nseu}(E^{path}_{c,node})$ is the neutron SEU cross-section of the node; b) $W^{path}_{c,node}$ is the critical (minimum) pulse width required at that node which can propagate through the functionally sensitized path without significant attenuation; c) $E^{path}_{c,node}$ is the critical (minimum) energy required to produce the pulse equal to or wider than $W^{path}_{c,node}$.

The $U_{node,path}$ can be calculated using Neutron Cross-Section (NCS) approach proposed in [18] as shown in equation (3):

$$U_{node,path} = \int_{E^{path}_{c,node}} \sigma_{nseu}(E^{path}_{c,node})(\frac{d\mathcal{N}}{dE})dE \qquad (3)$$

The neutron induced SEU cross-section is the probability that a neutron of energy $E$ can produce upset in a device in units of $cm^2/device$. We assume the neutron induced SEU cross-section of a node equal to the area of the node, since the critical energy results in an upset at the node. For the atmospheric differential neutron flux, $\frac{d\mathcal{N}}{dE}$, we will use an analytical approximation for New York City provided in [4, 25]. The critical energy is the minimum energy required to create a voltage pulse at a node which can $propagate$ to the circuit output through the functionally sensitized path without being electrically masked (described in details in section 3.1.2). In the following, we discuss the proposed approach to compute the critical pulse width at every node in the circuit over all paths to the observable outputs and the critical energy required to produce the critical pulse width at a node.

#### 3.1.1 Critical Pulse Width, $W^{path}_{c,node}$

The generated pulse at a node can be electrically masked as it propagates through the gates on the functionally sensitized path. The electrical masking of a pulse depends on the electrical properties of the gates on the path such as the drive strength and the load capacitance. To observe the pulse propagation through the functionally sensitized path, we need to know the pulse attenuation through every gate on the path. As discussed in [2], pulses wider than the logic transition time of a gate will propagate through the gate without attenuation. Pulses smaller than half of the transition time will attenuate. Pulses wider than half of the transition time and less than the transition time will propagate with varying attenuation. In addition to the pulse width, the amplitude of the pulse should be more than the logic threshold of the gate. If the pulse width is measured at the logic threshold of gates then the amplitude can be ignored since we know that it is sufficient to make a transition at the gate output [2]. In our study of pulse propagation through logic paths, we consider the pulse amplitude from rail to rail.

The $W^{path}_{c,node}$ is the critical width of a pulse at a $node$ which will not be electrically masked while propagating through a $path$. The concept of critical pulse width was developed in [21]. We perform the following three steps in order to determine the critical pulse width at every node in the circuit: 1) Flip-flop characterization to find the critical (minimum) pulse width which can be latched into the output flip-flop. 2) The characterization of every gate in the logic cell library to analyze the pulse attenuation through the gate. 3) Inverse pulse propagation and attenuation from the output flip-flop to the node.

1) **Flip-Flop Characterization.** Figure 1 shows the simulation setup to determine the $W^{FF}_c$ for a flip-flop. A narrow pulse is applied at the data input of the flip-flop at different time steps during the clock cycle and the output is monitored for latching the input pulse. This process is repeated in steps with larger pulse widths until the pulse latches. The minimum pulse which latches is the $W^{FF}_c$ for that flip-flop which may not be same as setup and hold time of the flip-flop .

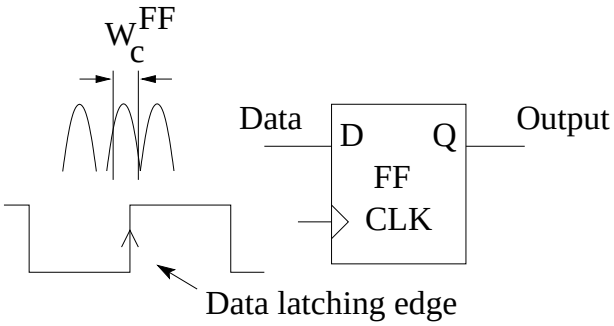



**Figure 1. Simulation setup to calculate $W^{FF}_c$ for a flip-flop**

2) **Logic Cell Library Characterization.** In order to determine the pulse attenuation through gates, we have characterized every gate used in the design using Spice. This gate characterization is based on setting the capacitive load at the output of the gate, applying a pulse at the input of the gate and measuring the pulse width at the output of the gate. If the gate has more than one input, the remaining inputs are considered as controlling inputs except the one where the input pulse is applied. For example, the controlling inputs for a NOR gate are set to logic zero and for an AND gate to logic one. Setting the controlling inputs

to appropriate logic sensitizes the path through the gate.

The attenuation function, $A$, of the gate is obtained using equation (4):

$$W_{out} = A_{gate}(W_{in}, L_{out}) \quad (4)$$

where $W_{in}$ is the width of the pulse applied at the input of the gate, $W_{out}$ is the width of the pulse measured at the output of the gate. $W_{in}$ and $W_{out}$ are measured at the logic threshold of the gate. $L_{out}$ is the capacitive load at the output of the gate. We can simplify the attenuation function of a multiple input gate by assuming all inputs have same attenuation. This can be done by applying $W_{in}$ at the input which has maximum input capacitance. Figure 2 depicts the simulation setup for the characterization of an AND gate. One input of the AND gate (the controlling input) is set to logic one while a pulse of width $W_{in}$ is applied at the other input. $W_{out}$ is the width of the pulse appearing at the output of the AND gate after attenuation through it. Spice simulations are required to obtain $W_{out}$

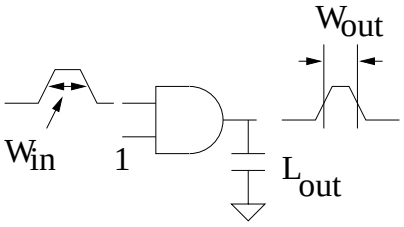



**Figure 2. Simulation setup for the characterization of an AND gate**

by applying various values of $W_{in}$ and $L_{out}$. $W_{in}$ can be varied in steps from zero to the width of the circuit clock cycle and $L_{out}$ can be varied from zero to the maximum load capacitance which can be applied at the output of the gate. An attenuation characterization table is generated for every gate. The column index of the table is the value of $W_{in}$ applied at the input of the gate and the row index is the value of $L_{out}$ at the output of the gate. The values of the table are $W_{out}$ for $W_{in}$ and $L_{out}$.

An example characterization of an AND gate of Figure 2 is shown in Table 1. The first column is the capacitive load, $L_{out}$, at the output of the gate in femto-Farad (fF). The remaining columns show values of $W_{out}$ in pico-seconds (ps) for various values of $W_{in}$. For example, when $W_{in}$ is $55.00\ ps$ and $L_{out}$ is $8\ fF$ then $W_{out}$ is $31.00\ ps$.

3) **Inverse Pulse Propagation/Attenuation.** Our approach to determine the $W_{c,node}^{path}$ of a circuit node is to consider the attenuation effect of $W_c^{FF}$ to the node using the inverse propagation. This is accomplished by finding all the logic paths from a node to the output flip-flops. For every path, the pulse propagation is started from the output of the circuit using $W_c^{FF}$ of the flip-flop towards the $node$. Note that the $W_c^{FF}$ can vary depending on the design of the flip-flop. Starting from the output of a path, the pulse width required at the input of the last gate in the path is calculated using the inverse function of equation (4) as shown in equation (5):

| $L_{out}$ | $W_{in}$ | | | |
|---|---|---|---|---|
| | 50.00 $ps$ | 55.00 $ps$ | 60.00 $ps$ | 65.00 $ps$ |
| 6 $fF$ | 32.75 $ps$ | 41.71 | 55.21 | 60.00 |
| 8 $fF$ | 12.00 | 31.00 | 40.00 | 50.00 |
| 10 $fF$ | 0.00 | 16.00 | 30.50 | 40.50 |
| 12 $fF$ | 0.00 | 0.0 | 9.00 | 20.00 |

**Table 1. An attenuation characterization table of an AND gate.**

$$W_{in} = A_{gate}^{-1}(W_{out}, L_{out}) \quad (5)$$

For the last gate of the path, $W_{out}$ is the $W_c^{FF}$. Based on the $L_{out}$ and $W_{out}$ of that gate, $W_{in}$ is obtained from the table generated from the gate characterization. The critical pulse width for the internal transistor nodes of a gate is considered as the critical pulse width at the output of the gate. Note, that a node can have multiple $W_{c,node}^{path}$ based on the paths from the node to the circuit outputs.

For example, suppose we want to get $W_{in}$ at the input of an AND gate for the given $W_{out}$ = 30 $ps$ and $L_{out}$ = 10 $fF$, then from Table 1, $W_{in}$ will be 60 $ps$. Now, $W_{in}$ at the input of the last gate becomes $W_{out}$ at the output of the second last gate in the path; this process is repeated until all the gates are covered in the path from the circuit output to the node and again for all the paths for that node. This gives the critical pulse width,$W_{c,node}^{path}$, required at the node for the given path which has sufficient width to propagate to the circuit output.

A more detailed example of the inverse pulse propagation in a path of a CMOS logic circuit from a flip-flop to a node is shown in Figure 3. This circuit has only one path from node $N_1$ to flip-flop $F_1$, $Path_1$, as shown in bold lines. The $W_{c,N_4}^{Path_1}$ at node $N_4$ is the $W_c^{FF}$ of the flip-flop $F_1$. The $W_{c,N_3}^{Path_1}$ at node $N_3$ is obtained from the characterization table of inverter using equation (5).

$$W_{c,N_3}^{Path_1} = A_{not}^{-1}(W_{c,N_4}^{Path_1}, L_{out}) \quad (6)$$

The $L_{out}$ in equation (6) is the capacitive load at node $N_4$ in Figure 3. Similarly, $W_{c,N_2}^{Path1}$ at node $N_2$ is obtained from the characterization table of the AND gate and $W_{c,N_1}^{Path1}$ at node $N_1$ is obtained from the characterization table of the OR gate.

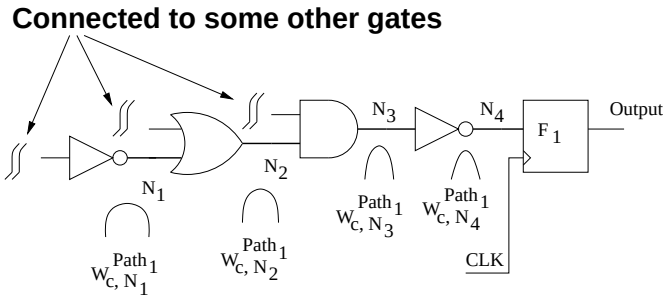



**Figure 3. Pulse propagation in the reverse direction from flip-flop $F_1$ to node $N_1$**

### 3.1.2 Determining Critical Energy, $E_{c,node}^{path}$

The $E_{c,node}^{path}$ is the critical energy of the particle required to produce a voltage pulse of width $W_{c,node}^{path}$ at a $node$ for a $path$ from the node to the observable output. To determine $E_{c,node}^{path}$, we characterize every sensitive node of the transistors in a gate based on equation (7).

$$W_{node,gate} = h_{node,gate}(E_{node,gate}, L_{out}) \qquad (7)$$

where $W_{node,gate}$ is the width of the generated pulse at a $node$ of the $gate$, $E_{node,gate}$ is the energy of the particle and $L_{out}$ is the capacitive load at the output of the gate. Spice simulations are required to obtain $W_{node,gate}$. To simulate particle energy in Spice, we use an analytical model of current pulse shown in equation (8) for neutrons proposed in [10, **?**]:

$$I_{node,gate}(t) \propto \frac{Q_{node,gate}}{T} \times \sqrt{\frac{t}{T}} \times exp(\frac{-t}{T}) \qquad (8)$$

where $Q_{node,gate}$ is the charge collected at a node due to the particle strike and $T$ is the time constant for the charge collection process. $T$ depends on the CMOS process technology used for the device, we will use $T$ = 30$ps$ for 100nm process technology as describe in [8]. $Q_{node,gate}$ is formed by deposition of energy $E_{node,gate}$ given by equation (9):

$$E_{node,gate} = \frac{3.6eV \times Q_{node,gate}}{1.6 \times 10^{-19}C} \qquad (9)$$

where 3.6$eV$ is the energy required to generate an electron-hole pair in silicon [4, 20].

An energy characterization table is generated using Spice for every node of the gate used in the design based on injecting current pulses with different amounts of charge and varying the output load of gate. Figure 4 shows the simulation setup to inject the current pulses at a node (output) of an AND gate and measure the width of the generated voltage pulse. The gate inputs are kept stationary while a current pulse is injected at a transistor node in the gate for the given load and the width of the generated voltage pulse is measured. The column index of the the

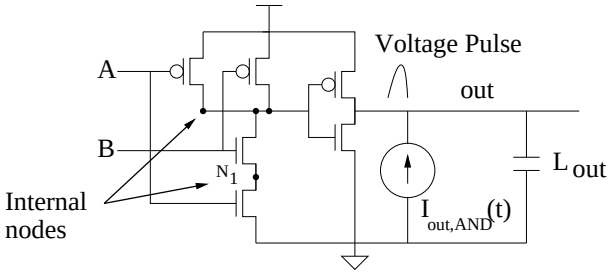



**Figure 4. Simulation setup to inject current pulses at a node of an AND gate**

energy characterization table is the energy of the hitting particle (converted from charge using equation (9)) and the row index is the capacitive load at the output of the gate. The values of the table are the width of the voltage pulse generated at the node.

Table 2 shows an example of an energy characterization table of an AND gate. The first column is the capacitive load applied at the output of the gate and the second column show the width of generated pulses at the output of the gate, $W_{out,AND}$ in pico-seconds (ps), for different loads when a current pulse equivalent to 10 $MeV$ energy is applied at the output of the gate. The inputs, $A$ and $B$, of the gate are kept stationary at the logic one. Similarly, the remaining columns show the width of the generated pulses for different energies.

From Table 2, the energy required to produce a voltage pulse based on the load capacitance at a node can be determined using the inverse of equation (7) as follows:

$$E_{node,gate} = h_{node,gate}^{-1}(W_{node,gate}, L_{out}) \qquad (10)$$

Thus, for $W_{c,node}^{path}$ of a $node$ of the gate for a $path$, $E_{c,node}^{path}$ can be determined from equation (10) as shown in equation (11).

$$E_{c,node}^{path} = h_{node,gate}^{-1}(W_{c,node}^{path}, L_{out}) \qquad (11)$$

A gate can have multiple sensitive transistor nodes, for example, an AND gate has two nodes which are output of

| $L_{out}$ | $E_{out,AND}$ | | | |
|---|---|---|---|---|
| | 10 $MeV$ | 20 $MeV$ | 30 $MeV$ | 40 $MeV$ |
| 6 $fF$ | 78 $ps$ | 217 | 332 | 467 |
| 8 $fF$ | 0 | 72 | 232 | 390 |
| 10 $fF$ | 0 | 0 | 84 | 308 |
| 12 $fF$ | 0 | 0 | 0 | 203 |

**Table 2. Energy characterization table for output node $out$ of the AND gate for Figure 4**

the gate (i.e. drain of NMOS and PMOS transistors) and four internal nodes as shown in Figure 4. The output of the gate is always sensitive to the particle strike, independent of the state of inputs i.e. it can be either drain of OFF-PMOS transistors or drain of OFF-NMOS transistors. However, the internal nodes (transistor nodes) of the gate may not be always sensitive, meaning either the transistor is not OFF or there is no pulse propagation path from the struck node to the output. This depends on the state of inputs of the gate. To make node $N_1$ of Figure 4 sensitive to the particle strike and pulse propagation path from it to output $out$, the gate inputs $A$ and $B$ must be 0 and 1. To characterize this node, current pulses of different charges are injected at it for different loads at the output $out$ and width of voltage pulse at $out$ is recorded. We have generated energy table for every node of the gate used in the design

### 3.2. Probability of Inputs, $P_{node,path}$, for Pulse Propagation

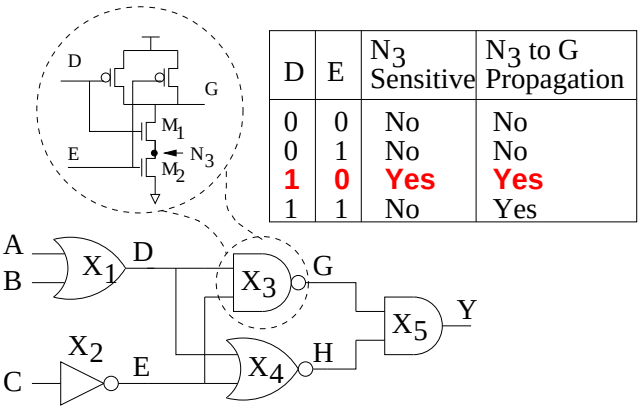
The pulse generated due to a particle strike will not propagate to the observable output if there is no functional sensitized path to the output (i.e. it can be logically masked). The pulse propagation from a node to the observable output depends on the state of the inputs. The input vectors required to make the functional sensitized path from node and the probability of those vectors then the probability of pulse propagation can be computed using an ATPG tool [7]. If the input vectors are considered uniformly distributed then, for a given node the ratio of the set of input vectors required to make a functional sensitized path to the total set of input vectors will give the probability of the pulse propagation from that node. A more accurate analysis can be performed if the input vectors are considered as application specific i.e. the population of input vectors is known.

We divide the computation of probability of the pulse propagation from a node into two parts: the local and the global pulse propagation. In the following, we describe a methodology to find the vectors for local and global pulse propagation.

#### 3.2.1 Local Pulse Propagation

The local pulse propagation concerns the propagation within the internal transistor nodes of a gate. Some internal nodes of a gate are not always sensitive to the particle strike and even if the pulse is generated at these nodes, it may not propagate to the gate output. Thus we need to justify the node, that is, make it sensitive to the particle strike and propagate the pulse from the node to the gate output.

Figure 5 shows an example of determining internal node justification of a gate. The NAND gate, $X_3$ has one internal node $N_3$ (shown in the large circle). To make $N_3$ sensitive to the particle strike, the transistor $M_2$ should be turned OFF and to propagate the generated pulse from $N_3$ to the output $G$ the transistor $M_1$ should be ON. This condition can $only$ be satisfied when the inputs $D$ and $E$ of $X_3$ are at logic 1 and 0, respectively, as shown in the table in Figure 5. Thus to compute the probability of pulse propagation from $N_3$ to $G$, we need to know all the common vectors which can set simultaneously $D$ to logic 1 and $E$ to logic 0. In this case, the vectors are <011>, <101>, and <111> which are the only vectors when the node $N_3$ is sensitive to the particle strike and the pulse generated at it can propagate to the gate output $G$.




| D | E | $N_3$ Sensitive | $N_3$ to G Propagation |
|---|---|---|---|
| 0 | 0 | No | No |
| 0 | 1 | No | No |
| **1** | **0** | **Yes** | **Yes** |
| 1 | 1 | No | Yes |

**Figure 5. Local pulse propagation**

#### 3.2.2 Global Pulse Propagation

The global pulse propagation occurs when a SEU generates a pulse at the output of the gate and it propagates through a logic path to the output of the circuit. The path can

contain gates with more than one input so to make the path functionally sensitized the controlling inputs of all the gates on the path need to be set to appropriate value.

The example shown in Figure 6 explains observable vectors for paths from node $N_2$ to the circuit outputs. There are three paths from $N_2$ to the observable outputs $F_1$ and $F_2$. These paths are as $path_1(N_2)$ = $(N_2, N_3, F_1)$, $path_2(N_2)$ = $(N_2, N_3, F_2)$, $path_3(N_2)$ = $(N_2, N_4, F_2)$. The observable vectors for $path_1(N_2)$ are $\{<$01xxx$>, <$110xx$>\}$.

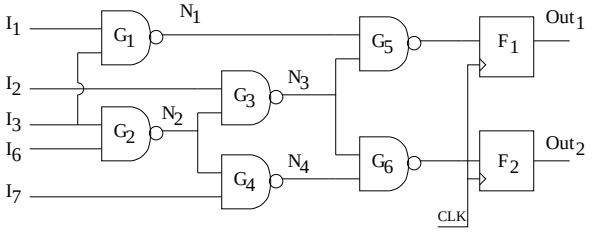



**Figure 6. Global pulse propagation**

To find all the input vectors which set an input of a gate to logic 0 or 1, an ATPG tool can be used which gives all the justification vectors of that input for stuck-at-1 or stuck-at-0 fault respectively. We use the ATPG tools as provided in [12].

To compute the probability of the pulse propagation from an internal node of a gate to the observable output of the circuit, we find all the input vectors which are common to justify local and global pulse propagation. To compute the probability of the pulse propagation from the output node of a gate to the observable output of the circuit, we just need all the vectors for global pulse propagation from the gate output to the observable output.

Since we know the probability of occurrence of each set of input vectors so the ratio of the sum of all the input vector sets required to sensitize the path from a node to observable output to the total input vectors set gives the probability of the input vector for that node, $P_{node,path}$. For example, in Figure 6 if we assume that each set of the input vectors have equal probability (i.e. the probability of each set is 1/32) then $P_{N_2,path_1}$ is $0.375$. Similarly $P_{N_2,path_2}$ and $P_{N_2,path_3}$ is 0.3125.

### 3.3. Determining $T_{node,path}$

For the normal operation of the circuit the data has to be stable during the setup time ($T_s$) and the hold time ($T_h$) of the flip-flop. The pulse can be latched into the output flip-flops if it arrives during the latching window also known as timing window of vulnerability, $T_{wov}$. The $T_{wov}$ can be defined as $T_s$ + $T_h$ + $W$ where $W$ is the width of the pulse. If the pulse arrives outside the $T_{wov}$ then it cannot be latched into the flip-flop hence timing masking occurs. Figure 7 shows the latching window of a flip-flop when the arrived SEU pulse during a clock cycle can be latched. $T_{node,path}$ is expressed as the ratio of $T_{wov}$ to the clock cycle time (i.e. $0 \leq T_{node,path} \leq 1$). To determine $T_{wov}$ of nodes, we use the approach proposed in [17] for static logic gates i.e. $T_{wov}$ = $T_s$ + $T_h$ + $W$.

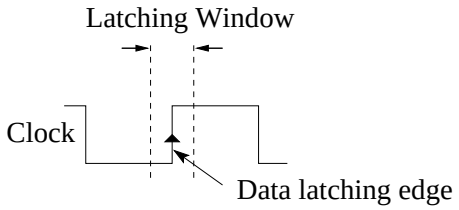



**Figure 7. Latching window of a flip-flop**

## 4. Results

The node sensitivity analysis approach for soft errors described in section 3 was implemented in PERL and C. Using this approach, we calculated node sensitivity of various ISCAS85 circuits and two adders. All the circuits were implemented in 100nm process technology using scaled MOSIS layout design rules. The Spice parameters were obtained from [3, 5] and the supply voltage for these circuits was 1.2v. We used Spice simulations for two adder circuits to compute the node sensitivity for accuracy and simulation (run) time comparison purposes between Spice and our approach. These simulations were performed on Sun Blade 1000 workstation. Adders, $Add_1$ and $Add_2$ have similar structure to 74182 and 74283 circuits. Table 3 shows comparisons between Spice simulations and our approach. Third row from top shows the percentage error in accuracy for node sensitivity of all the nodes calculated using Spice as reference. For $Add_1$, the node sensitivity calculated using our approach was 10.08% more than Spice and for $Add_2$, it was 5.98% less than Spice. The speedup factor using our approach against Spice was $9 \times 10^3$ for $Add_1$ and $6.3 \times 10^3$ for $Add_2$.

Previous research papers determine the sensitivity of a node by using multilevel nested iterative methods by increasing the energy level iteratively in order to find the critical energy of a node [14, 15]. We achieve run time efficiency by predetermining the critical energy of a node and in addition, we use ATPG tool in order to find the functionally sensitized paths. Thereby eliminating the need

| | $Add_1$ | $Add_2$ |
|---|---|---|
| Simulated nodes | 19 | 36 |
| Accuracy (% Error) | $-10.08\%$ | 5.98% |
| Spice Simulations | $\sim$10 Hours Runtime | $\sim$14 Hours Runtime |
| Our approach | 4 Seconds Runtime | 8 Seconds Runtime |
| Speedup factor | $9 \times 10^3$ | $6.3 \times 10^3$ |

**Table 3. Accuracy and run time comparisons.**

| $S_{node}$ | c7552 | c1355 | c499 | c432 | $Add_1$ | $Add_2$ |
|---|---|---|---|---|---|---|
| $\leq 1$ | 14% | 0 | 0 | 0 | 26% | 29% |
| $\leq 10^{-1}$ | 5% | 0 | 0 | 0 | 74% | 81% |
| $\leq 10^{-2}$ | 26% | 0 | 0 | 36% | 0 | 0 |
| $\leq 10^{-3}$ | 11% | 0 | 0 | 25% | 0 | 0 |
| $\leq 10^{-4}$ | 6% | 76% | 0 | 18% | 0 | 0 |
| $\leq 10^{-5}$ | 5% | 15% | 32% | 12% | 0 | 0 |
| $\leq 10^{-6}$ | 33% | 9% | 68% | 0% | 0 | 0 |
| Analyzed nodes | 3512 | 210 | 202 | 168 | 19 | 36 |
| Number of PIs | 207 | 41 | 41 | 36 | 9 | 9 |
| Runtime | 60 Hours | 26 Hours | 45 Mins | 3 Hours | 4 Secs | 8 Secs |

**Table 4. Normalized sensitivity of nodes.**

of two nested iterative loops which leads to several orders of magnitude faster.

Table 4 shows normalized node sensitivity, $S_{node}$, of four ISCAS85 circuits and two adders ($Add_1$ and $Add_2$). The node sensitivity of a node in the circuit is normalized with respect to the node with largest sensitivity. If a node has multiple paths to the circuit outputs then it has multiple values of sensitivity. In this table we chose the highest value among these sensitivity values of a node. The test vectors generated for a node of a circuit which has large number of primary inputs, can be very large. We use the upper limit on test vector sets generated by the ATPG for a node which is 100,000. This limit is for unexpanded test sets, meaning there might be don't cares in a test set. First column in Table 4 shows normalized sensitivity range and the remaining columns show percentage of nodes of every circuit which comes in normalized sensitivity range of the first column. The bottom three rows show the number of analyzed nodes, number of primary inputs (PIs), and runtime of our approach for every circuit.

It is clear from Table 4 that the sensitivity of the nodes in circuit is not uniformly distributed which is same as discussed in [15, 23]. The SER of a circuit can be reduced by reducing the sensitivity of highly sensitive nodes. One of the techniques to reduce the node sensitivity is the node hardening technique [11, 24].

We used a node hardening technique to reduce the node sensitivity of example circuits - $Add_1$ and $Add_2$. The node hardening technique is based on changing (increasing) the size of transistors connected to node. We applied this node hardening technique on four highly sensitive nodes of $Add_1$ and five highly sensitive nodes of $Add_2$. In this case, the size of transistors, which are connected to the selected node, was arbitrarily increased by 1.5 times. Table 5 shows the distribution of the node sensitivity for $Add_1$ and $Add_2$ before and after applying the hardening technique. The first column shows range of sensitivity. The second and third columns show the distribution of the node sensitivity before applying hardening technique i.e. the percentage of the nodes comes under the sensitivity range of first column. The fourth and fifth columns show the distribution of the node sensitivity after applying hardening technique. The hardening technique reduced the node sensitivity of 21% nodes (We applied node hardening to only 4 nodes) from $\leq 10^{-2}$ to $\leq 10^{-3}$. Similarly, for $Add_2$ in fifth column the node sensitivity reduced from 29% and 81% to 25% and 55% respectively. It is clear from the Table 5 that the soft error rate of the circuit can be reduced by applying node hardening techniques on selected nodes.

| | Before Hardening | | After Hardening | |
|---|---|---|---|---|
| $S_{node}$ | $Add_1$ | $Add_2$ | $Add_1$ | $Add_2$ |
| $\leq 1$ | 26% | 29% | 26% | 25% |
| $\leq 10^{-1}$ | 74% | 81% | 53% | 55% |
| $\leq 10^{-2}$ | 0 | 0 | 21% | 20% |
| $\leq 10^{-3}$ | 0 | 0 | 0 | 0 |

**Table 5. Reduction in the node sensitivity after applying node hardening technique**

## 5 Conclusion

In this paper, we proposed an approach to determine a metric of sensitivity of nodes to soft errors in CMOS logic.

The sensitivity of a node to soft errors depends on the electrical, logic and timing masking. We proposed an efficient technique to compute the upset rate of nodes. We proposed an additional technique to determine the logic masking of nodes using ATPG tool. The calculated results using our approach for various circuits show that it has accuracy close to Spice and several orders of magnitude faster than Spice. This approach can be used at the chip level to compute the sensitivity of gate and transistor nodes. One of the advantages of analyzing the sensitivity of nodes to soft errors is that the SER of the logic circuits can be reduced by applying appropriate techniques to reduce the sensitivity of highly sensitive nodes. A common technique to reduce the node sensitivity is by applying node hardening techniques.

## Acknowledgement

This work has been partially supported by the National Science Foundation under grant number 0313004 and by the Wright-Patterson labs under subcontract RSC04007.

## References

[1] L. Anghel and M. Nicolaidis. Cost reduction and evaluation of a temporary faults detecting technique. *Design Automation and Test in Europe Conference and Exhibition 2000*, pages 591–598, 2000.

[2] M. P. Baze and S. P. Buchner. Attenuation of single event induced pulses in cmos combinational logic. *IEEE Transactions on Nuclear Science*, 44(6):2217–2223, 1997.

[3] http://www-device.eecs.berkeley.edu/∼ptm.

[4] P. D. Bradley and E. Normand. Single event upset in implantable cardioverter defi brillators. *IEEE Transactions on Nuclear Science*, 45(6):2929–2940, 2004.

[5] Y. Cao, T. Sato, M. Orshansky, D. Sylvester, and C. Hu. New paradigm of predictive mosfet and interconnect modeling for early circuit simulation. *IEEE 2000 Custom Integrated Circuits Conference*, pages 201–204, 2000.

[6] P. E. Dodd and L. W. Massengill. Basic mechanisms and modeling of single-event upset in digital microelectronics. *IEEE Transactions on Nuclear Science*, 50(3):583–602, 2003.

[7] R. R. Fritzemeier, J. M. Soden, R. K. Treece, and C. F. Hawkins. Increased cmos ic stuck-at fault coverage with reduced i ddq test sets. *IEEE International Test Conference*, pages 427–435, 1990.

[8] B. S. Gill, C. Papachristou, and F. G. Wolff. Soft delay error effects in cmos combinational circuits. *IEEE VLSI Test Symposium*, pages 325–331, 2004.

[9] P. Hazucha, K. Johansson, and C. Svensson. Neutron induced soft errors in cmos memories under reduced bias. *IEEE Transactions on Nuclear Science*, 45(6):2921–2928, 1998.

[10] P. Hazucha and C. Svensson. Impact of cmos technology scaling on the atmospheric neutron soft error rate. *IEEE Transactions on Nuclear Science*, 47(6):2586–2594, 2000.

[11] T. Karnik, S. Vangal, V. Veeramachaneni, P. Hazucha, V. Erraguntla, and S. Borkar. Selective node engineering for chip-level soft error rate improvement. *Symposium On VLSI Circuits Digest of Technical Papers*, pages 204–205, 2002.

[12] H. Lee and D. Ha. Atalanta: an effi cient atpg for combinational circuits, technical report, 93-12. Technical report, Dep't of Electrical Eng., Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 1993.

[13] A. Maheshwari, W. Burleson, and R. Tessier. Trading off transient fault tolerance and power consumption in deep submicron (dsm) vlsi circuits. *IEEE Transaction on Very Large Scale Integration (VLSI) Systems*, 12(3):299–311, 2005.

[14] A. Maheshwari, I. Koren, and W. Burleson. Techniques for transient fault sensitivity analysis and reduction in vlsi circuits. *IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pages 597–604, 2003.

[15] K. Mohanram and N. A. Touba. Cost-effective approach for reducing soft error failure rate in logic circuits. *International Test Conference*, pages 893–901, 2003.

[16] P. C. Murley and G. R. Srinivasan. Soft-error monte carlo modeling program, semm. *IBM J. RES. DEVELOP.*, 40(1):109–118, 1996.

[17] H. T. Nguyen and Y. Yagil. A systematic approach to ser estimation and solutions. *International Reliability Physics Symposium*, pages 60–70, 2003.

[18] E. Normand. Single-event effects in avionics. *IEEE Transactions on Nuclear Science*, 43(2):461–474, 1996.

[19] E. Normand. Single event upset at ground level. *IEEE Transactions on Nuclear Science*, 43(6):2742–2750, 1996.

[20] J. Olsen, P. E. Becher, P. B. Fynbo, P. Raaby, and J. Schultz. Neutron-induced single event upsets in static rams observed at 10 km flight altitude. *IEEE Transactions on Nuclear Science*, 40(2):74–77, 1993.

[21] N. Seifert, P. Shipley, M. D. Pant, V. Ambrose, and B. S. Gill. Radiation-induced clock jitter and race. *accepted for publication at International Reliability Physics Symposium, 2005*, 2005.

[22] P. Shivkumar, M. Kistler, S. W. Keckler, D. Burger, and L. Alvisi. Modeling the effects of technology trends on the soft error rate of combinational logic. *Dependable Systems and Networks, 2002. Proceedings. International*, pages 389–398, 2002.

[23] C. Zhao, X. Bai, and S. Dey. A scalable soft spot analysis methodology for compound noise effects in nano-meter circuits. *Design Automation Conference*, pages 894–899, 2004.

[24] Q. Zhou and K. Mohanram. Transistor sizing for radiation hardening. *IEEE International Reliability Physics Symposium*, pages 310–315, 2004.

[25] J. F. Ziegler. Terrestrial cosmic rays. *IBM Journal of Research and Development*, 40(1):19–39, 1996.