Soft Error Rate Analysis for Combinational Logic Using An Accurate Electrical Masking Model

Feng Wang, Yuan Xie, R. Rajaraman and B. Vaidyanathan
The Pennsylvania State University, University Park, PA 16802, USA
{fenwang, yuanxie, ramanara, bvaidyan}@cse.psu.edu

Abstract*—Accurate electrical masking modeling represents a significant challenge in soft error rate analysis for combinational logic circuits. In this paper, we use table lookup MOSFET models to accurately capture the nonlinear properties of submicron MOS transistors. Based on these models, we propose and validate the transient pulse generation model and propagation model for soft error rate analysis. The pulse generated by our pulse generation model matches well with that of HSPICE simulation, and the pulse propagation model provides nearly one order of magnitude improvement in accuracy over the previous models. Using these two models, we propose an accurate and efficient block-based soft error rate analysis method for combinational logic circuits.

I. Introduction

Single Event Upset (SEU) is a voltage transient caused by neutron or alpha particles from cosmic ray or package materials [25]. The voltage transients may flip bits in memory cells or latches, causing soft errors. Soft errors in memory can be corrected by ECC (Error Correcting Code) circuitry, and many radiation harden techniques for memory cells or latches have been proposed [25] [10]. However, voltage transients caused by particle strikes can happen on any node in combinational logic. This transient pulse can propagate through logic gates and finally be latched by a sequential element, resulting in a soft error [20]. Cheap solutions to reduce soft errors caused by transients generated in combinational logic are not well understood yet. Fast and accurate analysis of soft error rate for combinational logic circuits is the first step towards the effort of finding efficient solutions [12].

There are three masking effects that can prevent a transient pulse in combinational logic from propagating and being latched by a memory element: *logical masking, electrical masking, and latch window masking* [20]. Logical masking happens when one of the other inputs of a gate is in controlling state (e.g., 0 for a NAND gate), so that the transient is blocked. Latch window masking means that the arrival transient pulse is outside of the latching window for the sequential elements. These two masking effects have been well studied [10] [20] [22] [23] [7] [14]. Electrical masking happens when the voltage transient resulting from a particle strike is attenuated by subsequent logic gates because of the electrical property of logic gates [10].

Electrical masking plays an important role in soft error rate estimation for combinational logic. For example, our experimental result shows that, for a small circuit with logic depth of 5 stages, ignoring electrical masking effect can overestimate the SER by 138%. Many models for the electrical masking effect have been proposed [17] [11] [3] [2] [20] [22] [23] [24] [4] [13]. however, these models introduce large errors for the estimation of transient pulse propagation, resulting in inaccurate soft error analysis for combinational logic. Inaccuracy in transient pulse width estimation can lead to large error in the soft error rate analysis [3]. For example, assuming the transient pulse estimation error is 15%, for the same circuit mentioned afore, 52% error in the SER estimation can be introduced.

In this paper, we propose and validate *transient pulse generation* and *transient pulse propagation* models for accurate electrical masking analysis. Based on these models, we propose an accurate and fast block-based SER analysis method for the combinational logic circuits.

The rest of the paper is organized as follows: Section II reviews related work; Section III provides the background on the soft error rate calculation and discusses the models of transient pulse generation and pulse propagation; Section IV presents our soft error rate estimation methods, Section V shows experimental results on the test circuits. Finally, the conclusions are provided in Section VI.

II. Related Work

Early work in estimating SER in the combinational logic circuits was based on the time-consuming Monte-Carlo Simulations [9] [15]. Recently many attempts have been made to estimate the SER of logic circuits quickly and accurately [20] [22] [23] [24] [4] [19]. Fault simulation methods [14] or BDD based techniques [22] can be used to estimate the logic masking effect. Latch window masking effect is also modeled by Shivakumar et al. [20].

Various techniques [2] [17] [3] [2] [23] [24] [4] [20] [11] [3] have been proposed to capture the electrical masking effect in SEU simulation. These techniques at the circuit level and the logic level can be categorized into three major types. 1) Simple trapezoidal or triangle waveform approximation based approaches [4] [22] [17] [20]. In these approaches, the SEU-induced transients are captured by the simple parameters: the slope, the magnitude, and the width. 2) Equivalent inverter based approaches [11] [2] [23]. In these approaches, the transistor-level models of logic gates are reduced to equivalent inverters. 3) Simple RC model based approaches. Based upon

linear RC models of logic gates, the transient response can be computed using close-formed equations [3] [13].

However, these approaches are not sufficiently accurate to capture the electrical masking effect on the radiation induced transients. Approaches belonging to category 1) are very simple and fast, but these approaches can not cover the possible radiation induced transient waveforms that can be generated due to particle strikes and can not capture the non-linear gate transfer characteristics for SEU-induced transients [13]. Approaches of category 2) provide a good estimation of propagation delay for full swing signals, but the predicted transient responses do not match well with SPICE simulations due to the errors in the current estimation [16]. In addition, current based simulation approaches is relatively slow [16]. Approaches of type 3) are simple, but they fail to account for the non-linear gate transfer characteristics for SEU-induced transients [13]. Thus, accurately modeling the electrical masking effect for combinational logic circuits remains a challenge.

Traditionally, the soft error rate analysis is performed using path based approach [20] [23] [24] [4] [2] [3] [19]. In this work, we use block based approach, which has been widely used in the statistical timing analysis, to achieve fast soft rate analysis. In the path based approach, all possible paths from the nodes in the netlist to the primary outputs are enumerated, and the number of the path is an exponential function of the number of gates. Block based approach has two key advantages over the path based approach: avoiding the re-computation by reusing the pulse estimation result of the gate for all its fan out nodes and low computation complexity.

Our contributions in this paper distinguish itself in the following aspects: (i) we propose a more accurate electrical masking model that takes into account the non-linear properties of MOS transistors and covers all the possible transient waveforms that can be generated due to particle strikes and (ii) we propose a fast block-based soft error analysis method for combinational logic that utilize the accurate electrical masking model.

III. Electrical Masking Modeling for Combinational Logic Circuits

Our electrical masking model consists of the *pulse generation* and the *pulse propagation* model. In this section, we first review the soft error analysis methods considering three masking effects. We then present the MOS transistor modeling and the waveform approximation method for the transient pulse. Based on this approximation method, we present accurate models to estimate the transient pulse generation and attenuation using the table based transistor models.

A. Background

When a high-energy particle strikes a MOSFET device, electron/hole pairs are generated [25] [10]. The electrons and holes move towards opposite directions if there is electric field between the source and drain terminals. This movement generates a transient current pulse, which can be modeled as a double exponential pulse [5]:

$$I_{inject}(t) = I_{peak} \times (e^{\frac{-t}{\tau_a}} - e^{\frac{-t}{\tau_b}}) \tag{1}$$

where $I_{peak} = \frac{Q}{\tau_a - \tau_b}$, in which Q is the charge collected as a result of particle strike, τ_a is the collection time-constant, and τ_b is the ion-track establishment time-constant. τ_a and τ_b are the constants which depend only on process-related factors. The current $I_{inject}(t)$ charges/discharges the capacitance at the output node, generating a transient voltage pulse $Vp(t)$ as shown in Fig. 1. The transient pulse may propagate to the sequential elements at the end of logic chain, and upset the stored values.

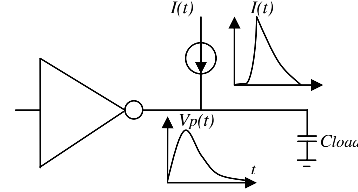


Fig. 1. Transient pulse generation simple modeling.

As mentioned in Section I, there are three masking effects that can prevent a transient pulse in combinational logic from propagating and being latched by a memory element: *logical masking, electrical masking, and latch window masking* [20]. To model the logic masking effect for the path between node n and primary output (PO) POn_i, $Psensitized(n, POn_i, input_pattern)$ is defined as the probability at which the transient error occurs at node n is functionally sensitized to primary out POn_i with a particular $input_pattern$. Fault simulation methods [14] or BDD based techniques [22] can be used to estimate that probability. The latch window masking effect can be modeled as the function of the characteristics of the transient pulse at the latch input, the latch window, and the clock period. The characteristics of the pulse at the latch input is determined by the collected charge generated by the particle strike and the path, through which the pulse propagate. The probability of a pulse is propagated to the primary output POn_i and finally captured by the latch, can be donated as $Perror(n, POn_i, q)$.

According to Hazucha's model [8], $SER(n, POn_i, input_pattern)$ can be defined as:

$$K \times F \times \frac{AreaS}{AreaC}(\sum_{q=0}^{Q_{max}}(f(q) \times \delta q \times Perror(n, POn_i, input_pattern, q)) \tag{2}$$

where K is a constant; F is the neutron flux; $AreaS$ is the drain area striked by neutron flux; $AreaC$ is the total area of the circuits. The node n is the particle striked node and POn_i is the primary output. The range of the collected charge is defined as from 0 to $Qmax$ where the $Qmax$ can be set to $4Qs$ as in SERA tools [23]. Qs is the charge collection efficiency of a device, which depends strongly on doping and Vdd. We divide the charge value into m equal-size intervals [20]. For each interval, we inject a current pulse

associated with a specific charge q, to obtain the $Perror$. The $f(q)$ is the probability density function of the collected charge q $(1/Qs \times \exp(-q/Qs))$ obtained from Hazucha's empirical model [8]. For a node n in the circuit, the SER value,$SER(n)$, can be calculated as [23]:

$$\sum_{i=1}^{k} SER(n, POn_i, input_pattern) \times Psensitized(n, POn_i, input_pattern) \quad (3)$$

Assume gate n is sensitive to a set of the primary outputs, and this set of the primary outputs are defined as POn_i, where i is from 1 to k.

According to Shivakumar et al. [20], the soft error rate of a combinational logic is the sum of each individual node's susceptibility:

$$SER_{chip} = \sum_{n=0}^{N} SER(n) \quad (4)$$

where N is the total number of nodes in the circuit.

B. Device Modeling

We use two lookup table based device models: *drain current model* and *capacitance model*. We extend the existing current model [6] to *series connected transistors*. We also model the parasitic capacitance using lookup tables. These models effectively capture the nonlinear properties of the MOS transistors. With lookup table based device models, arbitrary precision in modeling the nonlinear devices can be achieved by simply adding more entries to the table. In addition to the nonlinear properties of the MOS transistor, both the stacking effects of series connected MOS transistors and the input patterns [22] have significant impact on the accuracy of the transient error analysis. Both factors have been taken into account in our transistor modeling.

C. MOS Waveform Approximation

In transient analysis, a simple ramp approximation of the waveforms is widely used in static timing analysis. The common approach of the ramp approximation in soft error analysis [20] [22] [4] is using trapezoidal or triangular. This single parameter model fails to accurately capture the large range of waveform shapes in transient analysis. Recent research [19] proposes a parametric waveform model based on the Weibull function, but the model fail to handle the swings above Vdd or below ground because it only matches the values of the pulse between ground and Vdd. In our research, we use discrete values of the waveform to approximate the transient pulse. We define a time step and at each time step the voltage value is sampled. The time step can be adaptively changed according to the voltage change to trade off accuracy and speed.

D. Pulse Generation Modeling Considering the Coupling of the Floating Capacitors

Considering the circuit in Fig. 2, the current pulse is injected at the output of gate N. As shown in Equation 1, the transient current caused by the particle strike is modeled as a current source $I_{inject}(t)$. We first determine the gates that strongly affects the accuracy of the modeling to reduce the computation costs. We adopt the notion *strong coupled nodes*, which are defined as the nodes whose transistors are channel connected [21]. Our experiments show that the nodes that are *not* strongly coupled to the current injected node have little impact on the accuracy of the modeling.

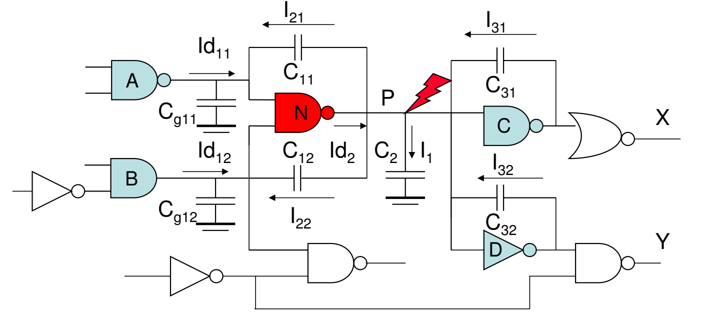


Fig. 2. The Pulse Generation Model Considering the Effects of Coupling Capacitances. Gate N represents the current injected gate; gate A, B, C and D are the strongly coupled gates.

Two types of nodes have the strong coupling with the current injected node: one is the loads of the gate and the other is its fan-ins. As shown in Fig. 2, gate N is the current injected gate, and gate A, B, C and D are the strongly coupled nodes. According Kirchhoff's Current Law (KCL), we have the following equations at the gate N's output node P and its fan in nodes.

For each fan-in node, we have:

$$Id_{1i} + I_{2i} = \frac{\Delta V_{in1i} \times C_{g1i}}{t_{step}} \quad (5)$$

where Id_{1i} is the drain current of the fan-in gate,I_{2i} is the current flowing through the floating capacitor between the fan-in nodes and node P, V_{in1i} is the fan-in gate's output voltage, C_{g1i} is the ground capacitance at the fan-in node, t_{step} is the time step.

At the current injected node P, according to KCL, we have:

$$\sum_{i=0}^{M} I_{2i} + I_1 = Id_2 + I_{inject} + \sum_{j=0}^{K} I_{3j} \quad (6)$$

where Id_2 is the drain current at the node P, M and K are the total number of the fan in nodes and the fan out nodes, I_{3j} is the current flowing through the floating capacitor between the fan-out nodes and node P.

For each capacitor, according to charge conservation law, for each floating capacitor between the fain node and the node P, we have

$$I_{2i} = \frac{(\Delta V_P - \Delta V_{in1i}) \times C_{1i}}{t_{step}} \quad (7)$$

where C_{1i} is the floating capacitor between the fan-in nodes and node P. V_P is the voltage at the node P, where the current is injected.

Similarly, for each floating capacitor between the fan-out node and the node P, we have

$$I_{3j} = \frac{(\Delta V_{out1j} - \Delta V_P) \times C_{3j}}{t_{step}} \tag{8}$$

where C_{3i} is the floating capacitor between the fan-out node and node P, and ΔV_{out1j} the output voltage of the fan-out node.

Finally we have the following equation at the node P.

$$I_1 = \frac{\Delta V_P \times C_2}{t_{step}} \tag{9}$$

where C_2 is the ground capacitance at the node P.

We approximate the ΔV_{out1j} as

$$Id_{3j} \times \frac{t_{step}}{C_{g3j}} \tag{10}$$

where C_{g3j} is the effective ground capacitance at the fan-out node. By solving the equations from 5 to 10, we obtain a closed form result for the voltage change at the node P:

$$\Delta V_P = \frac{Id_2 + I_{inject} + \sum_{i=0}^{i=M} Id_{1i} \frac{C_{1i}}{C_{1i}+C_{g1i}} + \sum_{j=0}^{j=K} \frac{\Delta Vout_{1j} C_{3j}}{t_{step}}}{C_2 + \sum_{i=0}^{i=M} \frac{C_{1i}C_{g1i}}{C_{1i}+C_{g1i}} + \sum_{j=0}^{j=K} C_{3j}} \times t_{step} \tag{11}$$

Thus we have the

$$V_P(T + t_{step}) = V_P(T) + \Delta V_P \tag{12}$$

The initial value of the output voltage, $V_P(0)$, can be obtained as the function of the input pattern and the type of the logic gates.

$$V_P(0) = f(inputpattern) \tag{13}$$

E. MOS Pulse Propagation Modeling Using Simple Overshoot/undershoot Model

In this section, we present the pulse propagation model, which use a simple overshoot/undershoot model to deal with the coupling effects of the floating capacitors. Overshoot (undershoot), as shown in Fig. 3(a), is defined as the transient value of the voltage that exceeds (is lower than) the final value [18]. As shown in Section V, overshoot/undershoot has great impact on the accuracy of estimation of the transient pulse propagation. Thus the overshoot and undershoot have to be taken into account in the transient error modelings .

In this study, we use a simple model to estimate the overshoot and undershoot. As shown in Fig. 3(b), the (C_{miller}) capacitance is the effective parasitic capacitance between the input and output, and (C_l) is the sum of the diffusion capacitances (C_{load}) and input capacitance of the load gates (C_{in}). The drain current (I_{drain}) contributes the change of the output voltage as well as the change of the voltage across the input and output. Thus, the pulse waveform estimation, which includes the overshoot/undershoot effect, can be performed as:

$$V_o(T + t_{step}) = V_o(T) + \frac{C_{miller} \times \delta V_i - I_{drain} \times t_{step}}{C_l} \tag{14}$$

where $\delta V_i = (V_i(T + t_{step}) - V_i(T))$ is the input voltage change, and $I_{drain} = Ids_{pullup} + Ids_{pulldown}$

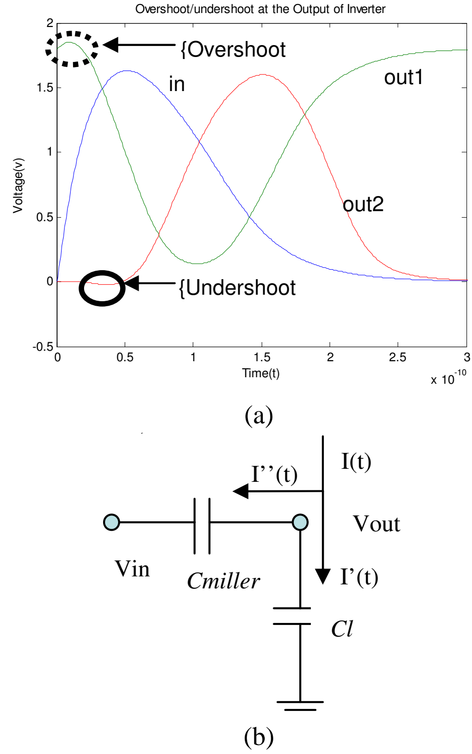


Fig. 3. Overshoot/undershoot in the transient pulse propagation, in is input of the transient pulse, out1 is output of first stage and out2 is the output of 2nd stage. (a) Overshoot/Undershoot Example (b) simple model for the overshoot.

IV. Block-based Soft Error Analysis Method

In this section, we first present block based soft error analysis algorithms based on our transient error modeling and then we show two heuristics to improve the speed of the transient analysis.

SEU ($netlist$){
1. For each random input vector {
2. Find the nodes sensitive to the primary outputs;
3. Update $Psensitized((n, POn_i, input_pattern))$ for the sensitive nodes;
4. }
5. Emask ($netlist$);
6. Calculate the total SER value for the netlist;
7. }

Fig. 4. The Pseduo code of the SER estimation algorithm.

Fig. 4 shows the pseudo code of our SER estimation algorithm. Our algorithm takes the $netlist$ as input; it computes the soft error rate for the entire circuit considering three masking effects. A for loop iterates over the random input vectors. In each iteration, the fault free circuit is first simulated, and the sensitive gates in the gate $netlist$ are determined using the critical path tracing techniques [14] (Line 3). With the sensitive gates determined, we update $Psensitized(n, POn_i, input_pattern)$ for these nodes. Next, we take into account the electrical masking effects and the latch window effects. At Line 5, we estimate the $SER(n, POn_i, input_pattern)$, for each gate n in the $netlist$ by calling the function $Emask(netlist)$. Finally, at Line 6,

the soft error rate of the entire circuit is calculated as the summation of each gate's susceptibility.

```
Emask (netlist){
1.  For each node n in the netlist {
2.     Extract the subcircuits of n to all the possible POs;
3.     Levelize the subcircuits;
4.     For each possible input pattern of gate n {
5.        Generate the pulse at the gate n
6.        Calculate the pulses at the next level until they reach the POs
7.        Calculate and Save SER(n, POn_i, input_pattern);
8.  }}}
```

Fig. 5. Pseudo code for function Emask ($netlist$), which is called by SEU ($netlist$) in Fig. 4.

Fig. 5 gives the pseudo code of the function $Emask(netlist)$. This function takes the gate net list as its input and calculates the $SER(n, i, input_pattern)$ for each gate n in the netlist. The main body of the function is a for loop, which iterates over all the gate nodes of the circuit. In each iteration, we extract the subcircuit from the gate node n to the primary outputs and calculate the $SER\ (n, i, input_pattern)$ for the set of POs which n can reach. From Line 4 to Line 7, the for loop enumerates all the possible input patterns since the input pattern has large impact on the SEU immunity [22]. The computation complexity of our algorithm can be estimated as $O(n^2)$, where n is the number of the gates in the circuits, while the computation complexity of the path based approach an exponential function of the number of the gates.

In our transient error analysis algorithm, two heuristics are introduced to improve the performance:

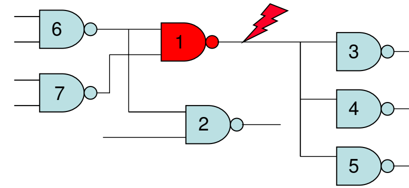
1) **Earlier termination:** The transient pulse propagation computation can be terminated earlier for a particular node if the pulse amplitude of transient voltage at any internal nodes is already less than $Vdd/2$.
2) **Adaptive time step:** The choice of the time step used in the simulation is important to the performance of the SER simulation. The large range of the slope of the transient pulse makes it necessary to use non-uniform time step.

V. Analysis Results

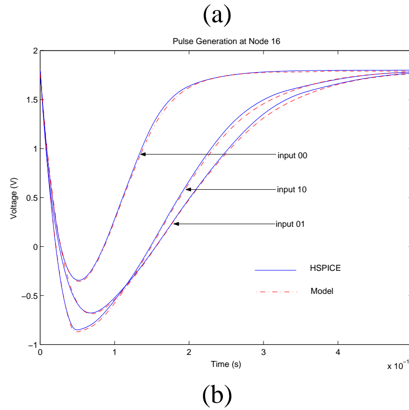
We implement our soft error analysis algorithm with the electrical masking model in C++. In this section, we present our analysis results to show the accuracy and the runtime of the soft error analysis for combinational logics.

We first validate our pulse generation model on a combinational logic circuit, with multiple loads and multiple fanin gates, as shown in Fig. 6(a). Fig. 6(b) shows a comparison between our results and those obtained with HSPICE simulation using Berkeley Prediction Model [1]. In this analysis, three different input patterns are applied to the current injected gate (gate 1). The waveforms generated by our model are perfectly matched with those of HSPICE simulation, as shown in Fig. 6(b).

Next, we perform the transient pulse propagation analysis on two simple logic gate networks: the inverter chain and



(a)



(b)

Fig. 6. The results of our transient pulse generation analysis method compared to that of the HSPICE simulation for a relatively complex logic circuit. a) Test circuit(only the strong coupled nodes are shown) b) Waveform comparison between the transient pulse generation model and the HSPICE simulation with different input combination applied to gate 1.

the two-input NAND chain for 180 nm technology. Table I shows the computation error of the pulse width for transient pulse propagation against the HSPICE simulation. After six stages, the error is less than 3% for the inverter chain for pulse width and 3.22% for the amplitude.The error introduced in our model is less than 6% in pulse width and 2.84% in amplitude after three stages of the propagation for the nand2 chain. After three stages of propagation, the pulse height is less than $Vdd/2$ and the computation is terminated. Fig. 7 shows the comparison between the waveforms computed by our pulse propagation model and those obtained in HSPICE simulation. The computation error for pulse width can be as large as 30% if we do *NOT* include the overshoot (undershoot) model.

TABLE I
Pulse Propagation Computation Errors against HSPICE Simulation

180nm	# of stages	1	2	3	4	5	6
INV	Pulse Width Error(%)	1.8	0	0	0	0	2.8
	Pulse Height Error(%)	0.10	0.85	0.04	1.48	0.98	3.22
NAND2	Pulse Width Error(%)	2.1	3.2	5.4	-	-	-
	Pulse Height Error(%)	1.29	0.45	2.84	-	-	-

We then show the accuracy of our soft error analysis on three small circuits. We compare our estimation results with the soft error estimation using HSPICE. Table II shows the results of soft error rate estimation obtained by our analysis method. The analysis error over HSPICE is within 3%.

Finally, we perform the soft error analysis on large ISCAS benchmark circuits, such as c7552. The runtime of the soft error analysis for large ISCAS benchmark circuits is in the

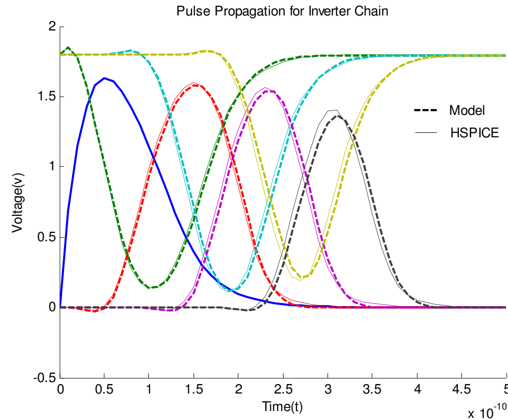


Fig. 7. The results of our transient pulse propagation analysis method compared to that of the HSPICE simulation for a seven stage inverter chain. Waveform comparison between our transient pulse propagation analysis method *with* overshoot modeling and the HSPICE simulation.

TABLE II
ACCURACY OF OUR ANALYSIS METHOD BASED ON OUR ELECTRICAL MASKING MODEL

Test circuit	Error over HSPICE	Run Time
6gates	2.58%	5sec
9gates	2.85%	15sec
12gates	1.8%	9sec

order of hours, which is the same as that of the accurate soft error analysis tool [23]. However, our analysis method with the novel electrical masking model is more accurate. In SERA tools proposed by Zhang [23], the maximum current computed using his method has 12% error over the HSPICE simulation [16], thus this method may introduce large estimation error in some cases since the large current produces large change in the voltage. Bin Zhang's BDD based static soft error analysis tool is very fast [22]. However, the average estimation error introduced in this method can be as large as 12% for small circuits due to his simple ramp approximation electrical masking models.

VI. CONCLUSIONS

Accurate modeling of the electrical masking effect in combinational logic circuits is important to perform accurate soft error rate analysis. Previous proposed electrical masking models are not sufficiently accurate to capture the electrical masking effect on the radiation induced transients. In this paper, we propose and validate the transient pulse generation and propagation models to accurately model the electrical masking effect. A block-based soft error analysis method using our electrical masking model is proposed to compute soft error rate accurately and efficiently in combinational logic.

VII. ACKNOWLEDGMENTS

This work was supported in parts by grants from NSF 0454123 and MARCO/DARPA-GSRC.

REFERENCES

[1] http://www.eas.asu.edu/p̃tm/.
[2] H. Cha, E. M. Rudnick, J. H. Patel, R. K. Iyer, and G. S. Choi. A gate-level simulation environment for alpha-particle-induced transient faults. *IEEE Trans. Computers*.
[3] P. Dahlgren and P. Liden. A switch-level algorithm for simulation of transients in combinational logic. *Proc. Intl. Fault-tolerant Computing Symposium*, pages 207–216, 1995.
[4] Y. S. Dhillon, A. U. Diril, and A. Chatterjee. Soft-Error Tolerance Analysis and Optimization of Nanometer Circuits. *Proceedings of the conference on Design, Automation and Test in Europe*, 2005.
[5] A. Dharchoudhury et al. Fast timing simulation of transient fault in digital circuits. *Proc. Intl. Conference on Computer-Aided Design*, pages 719–726, 1994.
[6] T. Shima et al. Table Look-Up MOSFET Modeling System Using a 2-D Device Simulator and Monotonic Piecewise Cubic Interpolation. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, (2):121–126, April 1983.
[7] B.S. Gill, C. Papachristou, F.G. Wolff, and N. Seifert. Node sensitivity analysis for soft errors in CMOS logic. *Proc. International Test Conference (ITC)*, pages 964–972, 2005.
[8] P. Hazucha and C. Svensson. Impact of CMOS technology scaling on the atmospheric neutron soft error rate. *IEEE Transactions on Nuclear Science*, (6):2586–2594, December 2000.
[9] T. Juhnke and et al. Calculation of the soft error rate of submicron CMOS logic circuits. *IEEE Journal of Solid-State Circuits*, pages 830–834, July 1995.
[10] T. Karnik, P. Hazucha, and J. Patel. Characterization of Soft Errors Caused by Single Event Upsets in CMOS Process. *IEEE Transaction on Dependable and Secure Computing*, (2):128–143, 2004.
[11] N. Kaul, B. L. Bhuva, , and S. E. Kerns. Simulation of SEU transients in CMOS ICs. *IEEE Trans. Nuclear Science*, pages 1514–1520, Dec. 1991.
[12] S. Mitra, T. Karnik, N. Seifert, and M. Zhang. Logic soft errors in sub-65nm technologies design and CAD challenges. *Design Automation Conference (DAC)*, June 2005.
[13] K. Mohanram. Simulation of transients caused by single-event upsets in combinational logic. *Proc. International Test Conference (ITC)*, 2005.
[14] K. Mohanram and N. A. Touba. Cost-Effective Approach for Reducing Soft Error Failure Rate in Logic Circuits. *Proc. International Test Conference (ITC)*, pages 893–901, 2003.
[15] P. C. Murley and G. R. Srinivasan. Soft-error Monte Carlo modeling program, SEMM. *IBM Journal of Research and Development*, (1):109–118, January 1996.
[16] A. Nabavi-Lishi and N. C. Rumin. Inverter models of CMOS gates for supply current and delay evaluation. *IEEE Trans. On Computer-Aided Design of Integrated Circuits and Systems*, pages 1271–1279, October 1994.
[17] M. Omana, G. Papasso, D. Rossi, and C. Metra. A model for transient fault propagation in combinatorial logic. *Proc. Intl. On-line Testing Symposium*.
[18] J. Rabaey, A. Chandrakasan, and B. Nikolic. *Digital Integrated Circuits: A Design Perspective*. Prentice Hall, NJ, second edition, 2003.
[19] R. Rao, K. Chopra, D. Blaauw, and D. Sylvester. An Efficient Static Algorithm for Soft Error Rate Analysis of Combinational Circuits. *ACM/IEEE Design Automation and Test in Europe Conference (DATE)*, March 2006.
[20] P. Shivakumar, M. Kistler, S.W. Keckler, D. Burger, and L. Alvisi. Modeling the Effect of Technology Trends on Soft Error Rate of Combinational Logic. *International Conference on Dependable Systems and Networks*, pages 389–398, June 2002.
[21] J. Vlach and K. Singhal. *Computer Methods for Circuit Analysis and Design*. John Wiley & Sons, Inc., second edition, 1993.
[22] B. Zhang, W. S. Wang, and M. Orshansky. FASER: Fast Analysis of Soft Error Susceptibility for Cell-Based Designs. *ISQED*, 2006.
[23] M. Zhang and N. Shanbhag. A Soft Error Rate Analysis Methodology. *Proc. ICCAD*, pages 111–118, 2004.
[24] C. Zhao, X. L. Bai, and S.Dey. A scalable soft spot analysis methodology for compound noise effects in nano-meter circuits. *Proceedings of the 41st annual conference on Design automation*, pages 894–899, 2004.
[25] J. Ziegler. IBM experiments in soft fails in computer electronics. *IBM Journal of Research and Development*, (1):3–18, 1996.