

A Timing-Aware Probabilistic Model for Single-Event-Upset Analysis

Thara Rejimon and Sanjukta Bhanja
 University of South Florida, Tampa, FL, USA.
 E-mail: (rejimon, bhanja)@eng.usf.edu

Abstract—

With device size shrinking and fast rising frequency ranges, effect of cosmic radiations and alpha particles known as Single-Event-Upset (SEU), Single-Event-transients (SET), is a growing concern in logic circuits. Accurate understanding and estimation of Single-Event-Upset sensitivities of individual nodes is necessary to achieve better soft error hardening techniques at logic level design abstraction. We propose a probabilistic framework to study the effect of inputs, circuits structure and gate delays on Single-Event-Upset sensitivities of nodes in logic circuits as a single joint probability distribution function (pdf). To model the effect of timing, we consider signals at their possible arrival times as the random variables of interest. The underlying joint probability distribution function, consists of two components: ideal random variables without the effect of SEU and the random variables affected by the SEU. We use a Bayesian Network to represent the joint pdf which is a minimal compact directional graph for efficient probabilistic modeling of uncertainty. The attractive feature of this model is that not only does it use the conditional independence to arrive at a sparse structure, but also utilizes the same for smart probabilistic inference. We show that results with exact (exponential complexity) and approximate non-simulative stimulus-free inference (linear in number of nodes and samples) on benchmark circuits yield accurate estimates in reasonably small computation time.

I. INTRODUCTION

High-energy neutrons present in cosmic radiations and alpha particles from packaging materials give rise to single event upsets (SEUs) resulting in soft errors in logic circuits. When particles hit a semiconductor material, electron-hole pairs are generated, which may be collected by a P-N junction, resulting in a short current pulse that causes logic upset or Single Event Upset (SEU) in the signal value. An SEU may occur in an internal node of a combinational circuit and propagate to an output latch. When a latch captures the SEU, it may cause a bit flip, which can alter the state of the system resulting in a soft error. In current technology, soft errors are of serious concern in memories, whereas in logic circuits soft error rate is comparatively low due to logical, electrical and temporal masking effects. However, as process technology scales below 100 nanometers and operating frequencies increase, the above masking barriers diminish due to low supply voltages, shrinking device geometry and small noise margin. This will result in unacceptable soft error failure rates in logic circuits even for mainstream applications [1].

Soft error susceptibility of a node j with respect to a latch L , SES_{j-Q_L} is the soft error rate at the latch output Q_L , contributed by node j . The propagation of an SEU generated due to a particle hit at an internal node j to an output i which causes a bit flip at the output of a latch L is depicted in Fig. 1.

We model the SEU propagation as follows: Let T_{j-i} be a Boolean variable which takes logic value 1 if an SEU at a node j causes an error at an output node i . Then $P(T_{j-i})$ (measured as the probability of T_{j-i} being equal to 1) is the conditional probability of occurrence of an error at output node i given an SEU at

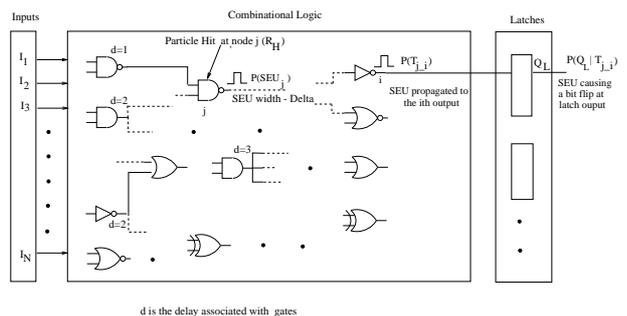


Fig. 1. SEU Propagation.

node j . Let $P(SEU_j)$ be the probability that a particle hit at node j generates an SEU of sufficient strength and let $P(Q_L|T_{j-i})$ be the probability that an error at output node i causes an erroneous signal at latch output Q_L . Mathematically SES_{j-Q_L} is expressed by Eq. 1.

$$SES_{j-Q_L} = R_H P(SEU_j) P(T_{j-i}) P(Q_L|T_{j-i}) \quad (1)$$

where R_H is the particle hit rate on a chip which is fairly uniform in space and time. $P(SEU_j)$ depends on V_{dd} , V_{th} and also on temperature. $P(Q_L|T_{j-i})$ is a function of latch characteristics and the switching frequency.

In this work, we explore $P(T_{j-i})$ by accurately considering the effect of (1) *SEU duration*, (2) *effect of gate delay* and (3) *timing*, (4) *re-convergence* in the circuit structure and most importantly (5) *inputs*.

We model internal dependency of the signals taking into consideration timing issues so that the SEU sensitization probability ($P(T_{j-i})$) captures the effect of circuit structure, circuit path delay and also the input space. A fan-out dependent delay model is assumed where gate delay of each node is equal to its fan-out. We also use logical effort based delay model where gate delays are dependent not only on fan-out but also on input capacitance as well as parasitic capacitance. Due to the temporal nature of SEUs, not all of the SEUs cause soft errors. Let t_h be the time when an SEU originates at a node, δ be the SEU duration, t_s be the time when outputs are sampled and Π be the set of propagation delays (t_d) of sensitized paths from the node to the circuit outputs. Nodes satisfying the following conditions do not cause soft error [4]:

$$t_h + \delta + t_d < t_s \quad \forall t_d \in \Pi. \quad (2)$$

Even though the above empirical formula doesn't take into account of set up and hold time requirements which affect latching window masking, we use this equation for our modeling

because this is pretty accurate as far as logical masking effect, circuit structure and gate delays are concerned.

We use a circuit expansion algorithm similar to that presented in [4], [15] to embed time-related information in the circuit topology without affecting its original functionality. From the expanded circuit, we generate a list of SEUs (possible SEU list) that are possibly sensitized to the circuit outputs at the time frame when output signals are latched. From the expanded circuit and the possible SEU list, we construct an error detection circuit and model SEU in large combinational circuits using a Timing aware Logic induced Soft Error Sensitivity model (TALI-SES), which is a complete joint probability model, represented as a Bayesian Network.

Bayesian Networks are causal graphical probabilistic models representing the joint probability function over a set of random variables. A Bayesian Network is a directed acyclic graphical structure (DAG), whose nodes describe random variables and node to node arcs denote direct causal dependencies. A directed link captures the direct cause and effect relationship between two random variables. Each node is quantified by the conditional probability of the states of that node *given* the states of its parents, or its direct causes. The attractive feature of this graphical representation of the joint probability distribution is that not only does it make conditional dependency relationships among the nodes explicit but it also serves as a computational mechanism for efficient probabilistic updating. Bayesian networks have traditionally been used in medical diagnosis, artificial intelligence, image analysis, and specifically in switching model [2] and single stuck-at-fault/error model [5] in VLSI but their use in timing aware modeling of Single-Event-Upsets is new. We first explore an exact inference scheme also known as clustering technique [13], where the original DAG is transformed into special tree of cliques such that the total message passing between cliques will update the overall probability of the system. We then explore a stochastic inference strategy, named Probabilistic Logic Sampling (PLS) [17], where a full instantiation of the probabilistic network is collected based on a simplified importance function. The sampling is stopped when the probabilities of the nodes converge. *It is worth pointing out that unlike simulative approaches that sample the inputs, importance sampling based procedures generate instantiations for the whole network, not just for the inputs.* These samples can be looked upon as Markov Chain sampling of the circuit state space.

The salient features of modeling SEU by Bayesian Network are as follows.

1. We provide a comprehensive model for the underlying error framework using a graphical probabilistic Bayesian Network based model TALI-SES that is causal, minimal and exact.
2. We can model the effect of timing and transient nature of the SEU's along with the accurate modeling of re-convergence in the circuit.
3. This model captures the data-driven uncertainty in the modeling of soft error that can be used where exact input patterns are not known apriori and also can be used by building a probabilistic model in case data traces are available by learning algorithm [21].
4. We infer error probabilities by (1) exact inference that transforms the graph into a special junction tree structure and relies on local message passing scheme and also by (2) smart stochastic non-simulative inference algorithms that have the feature of any-time estimates and generates excellent accuracy time trade-off for larger circuits.
5. Bayesian Networks are unique tool where effect of an observation at a child node can be used to get a probability space of the parents. This is called backward reasoning. Our model can be used to generate input space for which the SEU occurring at a particular node j might have no impact on the outputs. Note that in such case, hardening techniques will not be needed for node j . Similarly, we can find input space for which SEU at a node j cause high error probability at outputs. If the data trace is similar to the second type of input space, extensive hardening techniques need to be applied to j .

The remainder of this paper is organized as follows. Section II is a summary of the prior works done on soft error modeling and analysis. In section IV we discuss Bayesian inference schemes - both exact and approximate(stochastic) inference. This is followed by section V where we give experimental results using both exact and stochastic inference. Using exact inference we can characterize the input space to achieve zero output error even in the presence of some of the SEUs. The exact inference works well for small circuits. To handle larger circuits we use a stochastic inference scheme and compare our results with logic simulation results and found that our modeling is accurate (close-to-zero error) and efficient.

II. BACKGROUND

An estimation method for soft error failure rates resulting from Single Event Upsets proposed in [1] computes soft error susceptibility of a node based on the rate at which a Single Event Upset (SEU) occurs at the node (R_{SEU}), the probability that it is sensitized to an output ($P_{sensitized}$) and the probability that it is captured by a latch ($P_{latched}$). A model that captures the effects of technology trends in the Soft Error failure Rates (SER), considering different types of masking phenomena such as electrical masking, latching window masking and logical masking, is presented in [3]. Another model to analyze Single Event Upsets with *zero-delay* logic simulation, which is accurate and faster than timing simulators, is presented in [4]. As discussed in the previous section, this model uses a circuit expansion algorithm to incorporate gate delays and a fault list generation algorithm to get a reduced list of SETs. All of the above methods use simulation techniques which are highly input pattern dependant.

Zhao *et al.* proposes a methodology to evaluate softness or vulnerability of nodes in a circuit due to compound noise effects by considering the effects of electrical, logical and timing masking [11]. A selective triple modular redundancy technique (STM) for achieving radiation tolerance in FPGA designs is discussed in [10]. Karnik *et al.* suggests that soft error rate should be considered as a design parameter along with power, performance and area due to its increasing impact on circuits and systems with the scaling of process technology [6]. Effect of threshold voltage on SER of memories and combinational logic has been studied in [7]. Zhang *et al.* in [12] pro-

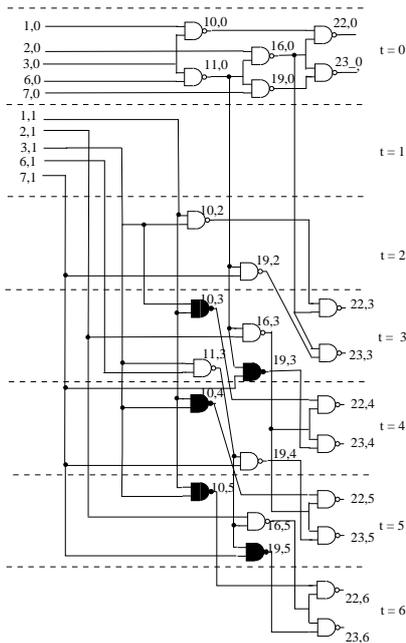


Fig. 2. Time-space transformed circuit of benchmark c17, modeling all SEUs.

posed a composite soft error rate analysis method (SERA) to capture the effect of supply voltage, clock period, latching window, logic depth, circuit topology and input vector on soft error rate. Their method uses a conditional probability based parameter extraction technique obtained from device and logic simulation. In their work, combinational circuits are assumed to have unbalanced re-convergent paths. However, other design considerations usually drive optimal circuit design to have balanced paths by adding buffers wherever re-convergence is necessary. For circuits with balanced paths, soft error analysis based on approximations given in [12] might not be the best choice.

Since all the state-of-the-art techniques have resorted to simulation for logical and device level effects (known to be expensive and pattern-sensitive especially for low probability events), we felt the need to explore the input data-driven uncertainty in a comprehensive manner through a probabilistic model to capture the effect of primary inputs, the effect of gate delay and the effect of SEU duration on the logical masking. There is future scope for these kinds of models to be fused with other models [6], [7], [12] for capturing device effects such as electrical masking, threshold voltage and supply voltage.

III. THE PROPOSED MODEL

In this section, we first focus on handling the timing aware feature of our probabilistic model, followed by the fault list construction. We conclude the section with discussion about the model itself, given the timing-aware graph and the fault list.

A. Timing Issues

We first expand the circuit by time-space transformation of the original circuit, without changing its functionality. The approach is similar to the method discussed in [4], [15]. Fig. 2 is the expanded circuit of benchmark c17. A gate in the original circuit C will have many replicate gates in the expanded circuit

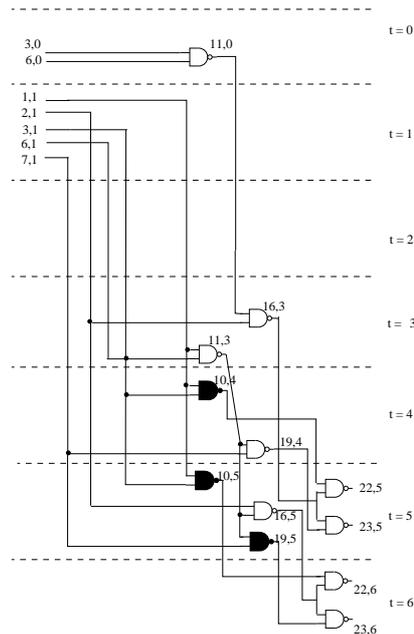


Fig. 3. Modified time-space transformed circuit of benchmark c17, modeling only the possibly sensitized SEUs.

C' , corresponding to different time-frames at which the gate output is evaluated. The output evaluation time $\{T\}$ of each gate in the circuit is calculated based on variable delay model. We assume that the delay associated with a gate is equal to its fan-out. For each gate g whose output is evaluated at time $t \in \{T\}$ a replicate node g, t is constructed. In addition to these replicate gates, we insert some duplicate gates (shown by filled gate symbols in Fig. 2). We explain the reasons for adding these duplicate gates later in this section.

The inputs of g, t are the replicate nodes of the gates, which are the inputs of g in the original circuit and belongs to the time-frames $t' < t$. We consider the value of signal i at time t by (i, t) . Now the random variable that represents the value of a signal i at time t is denoted by $X_{i,t}$. The circuit outputs reach steady state values, $X_{22,0}$ and $X_{23,0}$ at $t = 0$, after the application of the previous inputs, $\{X_{1,0}, X_{2,0}, X_{3,0}, X_{6,0}, X_{7,0}\}$. Let the new inputs $\{X_{1,1}, X_{2,1}, X_{3,1}, X_{6,1}, X_{7,1}\}$ be applied at $t = 1$. $X_{10,2}$ is the signal value at the output of gate 10 at time instant 2.

We insert a few duplicate gates (example: (10, 4), (10, 5), (19, 5), etc. shown by filled gate symbols) due to the following reasons:

Input signals of certain gates in the circuit might have different arrival time due to the difference in path delays. In order to model the effect of any SEU generated at the junction of the gates at time instants, later than the signal's arrival time, we insert additional duplicate nodes for those internal signals with less path delay. For example, in Fig. 2, input signals to gate 22 have path delays 2 and 5 respectively. The final output signal (22, 6) is evaluated with input signals (16, 5) and (10, 5). If no SEUs originated at the output of gate 10 between time instants 2 and 5, (10, 2) and (10, 5) would be the same. However, in the event an SEU occurs at node 10 at $t = 5$, (10, 2) and (10, 5) may differ depending on the inputs, which can cause a wrong output signal at (22, 6). We model the effect of SEU at (10, 5)

by introducing a duplicate gate (10,5) whose inputs are (1,1) and (3,1). Similarly, (10,3), (10,4), (19,4) and (19,5) are other duplicate gates. Duplicate gates also model the masking effect of some of the SEUs generated in the signal path of the input having lesser path delay. Example: Duplicate gate (10,5) mask the effect of an SEU originated at the output of gate 10, at time $t = 2$. Thus we can arrive at a reduced SEU list which is further explained later in this section.

Steps for constructing the timing-aware expanded circuit, based on fan-out dependent delay model are the following:

1. Arrange gates in the order of levels, with the level of input gates equal to zero.
2. Include all gates that are present in the original circuit. Output signals of these gates represent the steady state signal values at $t = 0$, before the application of new inputs.
3. Add additional input nodes representing new input signal values at $t = 1$;
4. For each level of the circuit starting from level $l_i = 1$, repeat the following step:

For each gate g in level l_i , create replicate gates at time frame $t = t_p + f_g$, where t_p is the maximum time frame of the previously inserted parent gates of g and f_g is the fan-out of gate g . Update time frames of gate g .

Output signals of a circuit are sampled at $t = t_s$, where t_s is the maximum of the latest signal arrival times of the output signals. SEUs which do not satisfy Eq. 2 affect circuit outputs resulting in soft errors. These SEUs are the upsets generated at the output of gates, which are in the fan-in cones of final outputs (outputs evaluated at time t_s). SEUs occurring at certain other gates, which are not in the fan-in cones of the final outputs, may also affect circuit outputs. These nodes arise due to the SEU duration time δ . For example in Fig. 2, we see that the final outputs are generated at time instant $t=6$. If an SEU occurs at signal 19 at 4 ns and lasts for one time unit, it will essentially be capable of tampering the value of node 23 at 6 ns. Note that we assume that δ is one time unit. The fault list will be different if we change the value of δ . Thus we can see that SEUs which are sensitized to outputs at time frames between t_s and $t_s - \delta$ may cause soft errors, depending on the input signals and circuit structure.

Considering the above factors, we modify the expanded circuit by including only those gates that propagate SEUs to the outputs between time instants, t_s and $t_s - \delta$. Thus we get a considerable reduction in the circuit size. Fig. 3 is the modified expanded circuit of c17, which models all SEUs possibly sensitized to a final output.

Next, we discuss how to generate a list of possible SEUs affecting the circuit outputs. Not all gates in Fig. 3 are SEU sensitive. As discussed above, a duplicate node introduces an additional delay of at least one time unit. If the delay introduced by a duplicate gate is greater than or equal to δ , the SEU duration time, the effect of SEUs originated at any of the gates in the fan-in cones of the duplicate gate is nullified and correct signal value is restored at the output of the duplicate gate, and hence those SEUs are effect-less. Thus we create a reduced list of SEUs by traversing the modified extended circuit from each of the circuit outputs at time instants between t_s and $t_s - \delta$, until a duplicate gate or an input node is reached.

TABLE I
GATE DELAYS BASED ON LOGICAL EFFORT

Gate Type	Delay
Inverter	$fanout + P_{inv}$
n-input NAND	$\frac{n+2}{3} * fanout + nP_{inv}$
n-input NOR	$\frac{2n+1}{3} * fanout + P_{inv}$
2-input XOR	$4 * fanout + 4nP_{inv}$

B. Delay Modeling Based on Logical Effort

We extend this work by using logical effort based model which is dependent on fan-out, input capacitance as well as parasitic delay. In this section we explain how gate delays are calculated based on logical effort [20]. Delay of a logic gate can be expressed as the sum of two components, effort delay and parasitic delay. effort delay is the product of logical effort and electrical effort, where logical effort is defined as the relative ability of a gate topology to deliver current and electrical effort is the ratio of output capacitance to input capacitance. Electrical effort is sometimes called fan-out. Mathematically, gate delay is expressed as $d = f + p = gh + p$ where f is effort delay, p is the parasitic delay, g is the logical effort and h is electrical effort. Logical effort is defined to be 1 for an inverter. Hence logical effort is the ratio of input capacitance of a gate to the input capacitance of an inverter delivering the same output current. It can be estimated counting capacitance in units of transistor width. Parasitic delay represents delay of a gate driving no load and it depends on diffusion capacitance. parasitic delay of an inverter, $P_{inv} \approx 1$. From the above considerations, we compute basic CMOS gate delays and use these delay values in our model. Table below shows the delay expressions for basic gates.

Circuit expansion is performed in a similar way as explained in the above section. Each gate is replicated several times corresponding to the time frames at which new gate output signals are evaluated. Here, gate output evaluation time is based on delay values calculated as above. This is illustrated in Figure 3 which shows how benchmark circuit c17 is expanded with logical effort based gate delay model. Delay of a 2-input nand gate with one fan-out is calculated as 3.33 time units and that of a 2-input gate nand gate with 2 fan-out is 4.67 time units. Final output is evaluated at time unit $T_s = 13.67$. From this expanded circuit, we arrive at a reduced circuit by traversing backward from outputs evaluated at T_s and $T_s - \delta$ until a duplicate gate or an input is reached, thereby modeling only the possibly sensitized SEUs.

C. Bayesian Networks

Bayesian Networks are graphical probabilistic models representing the joint probability function over a set of random variables using a directed acyclic graphical structure (DAG), whose nodes describe random variables and node to node arcs denote direct causal dependencies. In a Bayesian network, the exact joint probability distribution over a set of n variables, $X_1 \cdots, X_n$ in this network is given by Eq. 3.

$$P(x_n, x_{n-1}, \dots, x_2, x_1) = P(x_n | x_{n-1}, \dots, x_1) P(x_{n-1} | x_{n-2}, \dots, x_1) \cdots P(x_1) \quad (3)$$

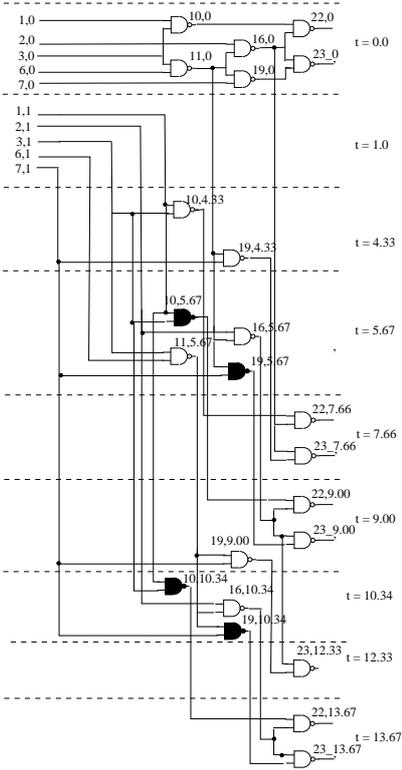


Fig. 4. Time-space transformed circuit of benchmark c17 with Logical Effort Based Delay Model

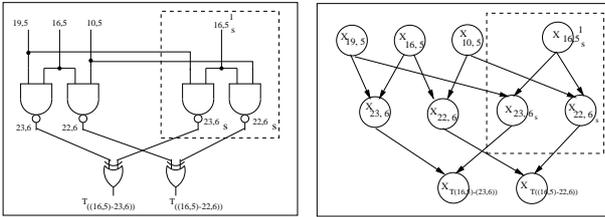


Fig. 5. (a) An illustrative SEU sensitivity logic for a subset of c17. (b) Timing-aware-Logic-induced-DAG model of the SEU sensitivity logic in (a)

Any random variable, X_k is independent of all other variables, given the states of its parent nodes, say, X_{k-1} and X_{k-2} . This *conditional independence* can be expressed by Eq. 4.

$$P(x_k | x_n, x_{n-1}, \dots, x_1) = P(x_k | x_{k-1}, x_{k-2}) \quad (4)$$

Mathematically, this is denoted as $I(X_k, \{X_{k-1}, X_{k-2}\}, \{X_n, \dots, X_{n-1}\})$. Using the conditional independence in directional graph, we arrive at an optimal factorized form that involve conditional probabilities based on the parents (or direct causes, inputs) to a node (effect, output): $P(X) = \prod_{k=1}^m P(x_k | pa(x_k))$. Even though probabilistic inference is worst-case NP-Hard, these factorized forms can reduce complexity significantly for general cases.

D. TALI: Timing-aware-Logic-induced Soft error model

In this section, we first describe the proposed Bayesian network based model, which can be used to estimate the soft error sensitivity of logic blocks. This model captures the dependence

of SEU sensitivity on the input pattern, circuit structure and the gate delays. Note that this probabilistic modeling does not require any assumptions on the inputs and can be used with any biased workload patterns. The proposed model, Timing-Aware-Logic-Induced-Soft-Error-Sensitivity (TALI-SES) Model is a Directed Acyclic Graph (DAG) representing the time-space transformed, SEU-encoded combinational circuit, $\langle C', J \rangle$ where C' is the expanded circuit created by time-space transformation as discussed in section. A and J is the set of possible SEUs (also discussed in section A). The error detection circuit consists of the expanded circuit C' , an error sensitization logic for each SEU and a detection unit T consisting of several comparator gates. We explain it with the help of a small example shown in Fig 5(a), which is the error detection circuit for a small portion of benchmark c17. The error sensitization logic for an SEU at node j consists of the duplicate descendant nodes from j . In Fig. 5(a), the block with the dotted square is the sensitization logic for $16, 5_s^1$ [An SEU¹ at node 16 at time $t = 5$]. It consists of nodes $22, 5_s$ and $22, 6_s$ descending from node $16, 5$ of the time-space transformed circuit. For simplicity, we show the modeling of only one SEU in this example. Our model can handle any number of SEUs simultaneously. Each SEU sensitization logic has an additional input to model the SEU. Example: input $SEU_{16,5}^1$. This input signal value is set to logic one in order to model the effect of a 0-1-0 SEU occurring at node 16 at time frame 5.

As discussed previously in section A, an SEU lasting for a duration δ can cause an erroneous output if its effect reaches the output at any instant between the sampling time t_s and time frame $t_s - \delta$. In this work we assume δ to be one. Hence we get error sensitized outputs at time frame t_s and for some SEUs at $t_s - 1$ also, if there exist re-convergent paths between SEU location and an output. We need to compare the SEU-free output signals evaluated at the sampling time, t_s with the corresponding SEU-sensitized output signals arriving at $t_s - 1$ and t_s . Hence these signals are sent to a detection unit T . The comparators in the detection unit compare the ideal and error sensitized outputs with the corresponding error-free outputs and generate test signals. For example, the test signals for an SEU at node j at time t are $T_{(j,t)-(i,t_s)}$ and $T_{(j,t)-(i,t_s-1)}$. If any of these the test signal value is 1, it indicates the occurrence of an error. The probability $P(T_{(j,t)-i})$, which is a measure of the effect of SEU $(j,t)_s$ on the output node i is computed as a joint probability which is explained below:

Let A be an event that an SEU at node j causes a bit-flip at output i at time t_s and let B be an event that an SEU at node j causes a bit-flip at output i at time $t_s - 1$. $P(A = 1)$ is the probability of occurrence of error and at time t_s . $P(A = 0)$ is the probability that SEU doesn't cause an error at t_s . $P(B)$ can be explained in a similar way. The Error probability due to an SEU at node j at time t w.r.t. output i is the joint probability

$$P(A \cup B) = P(A = 1, B = 0) + P(A = 0, B = 1) + P(A = 1, B = 1) \quad (5)$$

which is expressed as:

$$P(T_{(j,t)-i}) = P(T_{(j,t)-(i,t_s)} \cup T_{(j,t)-(i,t_s-1)}). \quad (6)$$

An SEU can have effect on more than one output. The overall effect of an SEU $(j,t)_s$ on the outputs is computed as $P(T_{(j,t)}) =$

$\max_{\forall i} \{P(T_{(j,t)}^i)\}$. In the example the SEU $(16,5)_s$ is sensitized to outputs 22,6 and 23,6. Hence the two test signals for this SEU are $T_{(16,5)}^-(22,6)$ and $T_{(16,5)}^-(23,6)$.

An SEU occurring at node j at time t , which is either SEU^1 or SEU^0 (but not both), can cause a bit-flip at the output with probability $P(T_{j,t}^1)$ or $P(T_{j,t}^0)$. In order to compute the SEU sensitivity of a node, we take the worst case probability, which is the maximum of the above two probabilities. $P(T_{j,t}) = \max\{P(T_{(j,t)}^1), P(T_{(j,t)}^0)\}$

More than one SEU can originate at a node at different time frames. Considering the effect of SEUs at node j at all time frames, we compute the worst case output error probability due to node j as $P(T_j) = \max_{\forall t} \{P(T_{(j,t)})\}$, which is the maximum probability over all time frames.

These detection probabilities depend on the circuit structural dependence, the inputs, dependencies amongst the inputs, gate delays and the SEU duration. In this work we assume random inputs for experimentation and validation of our model.

We construct the TALI-SES Bayesian Network of the SEU detection circuit by nodes which are random variables representing signal values of the SEU detection circuit. A signal i in the detection circuit is represented by the random variable X_i in the Bayesian Network.

In TALI-SES DAG structure the parents of each node are its Markov boundary elements. Hence the TALI-SES is a boundary DAG. For definition of Markov Boundary and boundary DAG, please refer to [18]. Note that TALI-SES is a boundary DAG because of the causal relationship between the inputs and the outputs of a gate that is induced by logic. It has been proven in [18] that if graph structure is a boundary DAG D of a dependency model M , then D is a minimal I-map of M ([18]). This theorem along with definitions of conditional independencies, in [18] (we omit the details) specifies the structure of the Bayesian network. Thus TALI-SES DAG is a minimal I-map and thus a Bayesian network (BN).

IV. BAYESIAN INFERENCE

We explore two inference schemes for the TALI-SES. The first inference scheme is cluster based exact inference and the second one is based on stochastic inference algorithm which is an approximate non-simulative scalable anytime method.

A. Junction Tree Based Inference

We demonstrate this inference scheme with a running example shown in Fig 6. The combinational circuit is shown in Fig. 6a and a subset of the time transformed circuit is shown in Fig 6b. The Bayesian Network captures the effect of SEU of “zero” at node 5 at a time instant 2 unit (denoted by the random variable $X_{5,2,0}$) on the output signal 6 at 3 time unit (denoted by random variable $X_{6,3}$). Note that the error in output signal $X_{6,3}$ is $T_{6,(5,2)}$ which is an xor combination of $X_{6,3}$ and $X_{6,3s}$ where $X_{6,3s}$ is the node that captures the effect of SEU at node 5 at 2 time unit. This is the original TALI-SES Bayesian Networks that we further process for exact inference.

The steps involved in the exact inference scheme are described below. Moralization: Create an undirected graph structure called the *moral graph* from the Bayesian network DAG

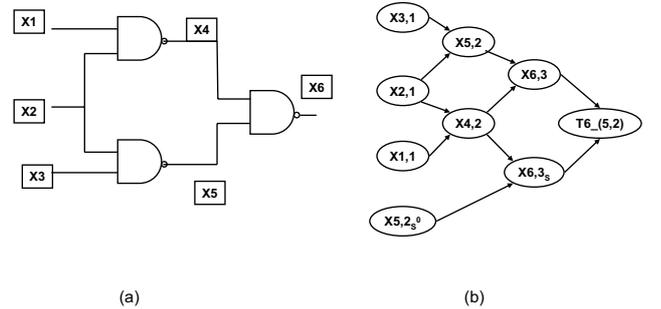


Fig. 6. (a) A small Logic circuit (b) Time transformed Bayesian Network

structure by adding undirected edges between the parents of a common child node and dropping the directions of the links. The moral graph represents the Markov structure of the underlying joint function [19]. The dependencies that are preserved in the original DAG are also preserved in the moral graph [19]. The dashed edges in Fig. 7a are added at this stage. This step ensures that every parent child set is a complete sub graph. Triangularization: In this step, every cycles of length greater than or equal to 4 is reduced to cycles of 3 by inserting additional links (chords) to the moral graph. The moral graph is said to be triangulated if it is chordal [19]. Note that in this particular example, moral graph is chordal and no additional links are needed. Message passing in Junction Tree: A *junction tree* is defined as a tree of cliques (collection of completely connected sub graph) of the choral graph (cliques are connected by unique path as in Fig 7a). Junction tree possesses running intersection property [19] that ensures that if two cliques share a common variable, the variable should be present in all the cliques that lie in the unique path between them. Fig. 7b is the junction tree derived from the chordal graph of Fig. 7a in this example. Interested readers are referred to [2] for a detailed description of how local message passing is performed in junction trees.

Note that since junction tree has no cycle and it is also not directional, we can propagate evidence from any node at any clique and the propagate the evidence in any direction. It is in sharp contrast with simulative approaches where flow of information always propagate from input to the outputs. Thus, we would be able to use it for input space characterization for achieving zero output error due to SEUs. We would instantiate a desired observation in an output node (say zero error) and backtrack the inputs that can create such a situation. If the input trace has large distance from the characterized input space, we can conclude that zero error is reasonably unlikely. Note that this aspect of probabilistic modeling is already used in medical diagnosis but are new in the context of input space modeling for soft error.

This exact inference is expensive in terms of time and hence for larger circuits, we explore a stochastic sampling algorithm, namely probabilistic Logic Sampling (PLS). This algorithm has been proven to converge to the correct probability estimates [17], without the added baggage of high space complexity and has been used in [16].

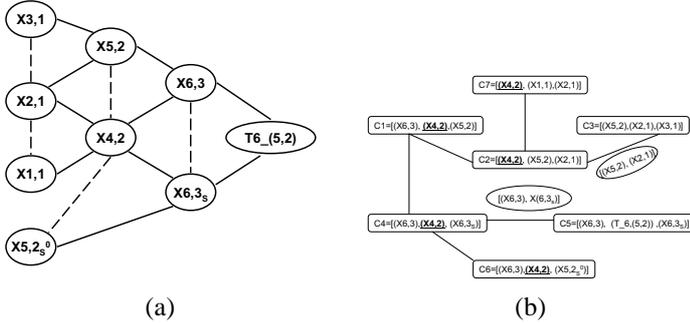


Fig. 7. (a) Chordal Graph (b) Junction Tree

B. Probabilistic Logic Sampling (PLS)

Probabilistic logic sampling is the earliest and the simplest stochastic sampling algorithms proposed for Bayesian Networks [17]. Probabilities are inferred by a complete set of samples or instantiations that are generated for each node in the network according to local conditional probabilities stored at each node. The advantages of this inference are that: (1) its complexity scales linearly with network size, (2) it is an any-time algorithm, providing adequate accuracy-time trade-off, and (3) the samples are not based on inputs and the approach is input pattern insensitive. The salient aspects of the algorithm are as follows.

1. Each sampling iteration stochastically instantiates all the nodes, guided by the link structure, to create a network instantiation.
 2. At each node, x_k , generate a random sample of its state based on the conditional probability, $P(x_k|Pa(x_k))$, where $Pa(x_k)$ represent the states of the parent nodes. This is the local, importance sampling function.
 3. The probability of all the query nodes are estimated by the relative frequencies of the states in the stochastic sampling trace.
 4. If states of some of the nodes are known (evidence), such as in diagnostic backtracking, network instantiations that are incompatible with the evidence set are disregarded.
 5. Repeat steps 1, 2, 3 and 4, until the probabilities converge.
- We adopt the tool GeNie [14] for inference using Probabilistic Logic Sampling.

Complexity: The computational complexity of the exact method is exponential in terms of number of variables in the largest cliques. Space complexity of the exact inference is $n \cdot 2^{|C_{max}|}$ [2], where n is the number of nodes in the Bayesian Network, and $|C_{max}|$ is the number of variables in the largest clique. The time complexity is given by $p \cdot 2^{|C_{max}|}$ [2] where p is the number of cliques.

The time complexity, based on the stochastic inference scheme, is linear in n , the number of nodes in the expanded circuit, specifically, it is $O(n|N_{SEU}|N)$, where N_{SEU} is the number of SEUs and N is the number of samples.

V. EXPERIMENTAL RESULTS

We demonstrate the modeling of SEU based on TALI-SES using ISCAS benchmark circuits. The logical relationship be-

TABLE II
SIZE OF ORIGINAL AND TIME-EXPANDED ISCAS CIRCUITS FOR FANOUT-DEPENDENT DELAY MODEL

	Gates	Gates expanded	# of nodes (TALI)	Time frames
c432	196	476	1989	55
c499	243	464	1596	30
c880	443	729	2552	51
c1355	587	1440	3388	55
c1908	913	1524	18118	79
c2670	1426	2584	4097	81
c3540	1719	3795	15670	93
c5315	2485	4887	13228	90
c6288	2448	30113	31157	263
c7552	3719	10006	45907	88

node j	SEU^1		SEU^0	
	$P(T_{j-22})$	$P(T_{j-23})$	$P(T_{j-22})$	$P(T_{j-23})$
10	0.2813	0	0.4375	0
11	0.0625	0.2344	0.3125	0.6563
16	0.3125	0.1875	0.4375	0.4375
19	0	0.375	0	0.4375
22	0.4375	0	0.5625	0
23	0	0.4375	0	0.5625

TABLE III
ESTIMATED $P(T_{j-})$ VALUES OF NODES IN BENCHMARK C17 FROM EXACT INFERENCE

tween the inputs and the output of a gate determines the conditional probability of a child node, given the states of its parents, in the TALI-DAG.

In Table II we report the total number of gates in the actual circuit (column 2), total number of gates in the modified expanded circuit (column 3), and the total number of nodes in the resulting TALI-SES (column 4). Column 5 lists the maximum time-frames of the circuits.

A. Exact Inference

In this section, we explore a small circuit c17, with exact inference where we transform the original graph into junction tree and compute probabilities by local message passing between the neighboring cliques of the junction tree as outlined in section IVA. Note that this inference is proven to be exact [18], [19](zero estimation error).

Table III tabulates the results of the TALI-SES of benchmark c17 using the exact inference. In this table, we report the probabilities of error at output nodes 22 and 23 due an SEU at each node j (column 1) namely (10, 11, 16, 19, 22 and 23). Column 2 and 3 of Table III give error probabilities due to SEU^1 (0-1-0 transition) at output nodes 22 and 23 respectively. Similarly 4 and 5 give error probabilities due to SEU^0 (1-0-1 transition) at output nodes 22 and 23 respectively. We compare the error-free outputs at 22 and 23 at sampling time t_s with corresponding error sensitized outputs arriving at time frames $t_s - 1$ and t_s due to SEUs generated at a node at all possible time frames (as discussed in section III D). Columns 2, 3, 4 and 5 of Table III reports the maximum of error probabilities due to SEUs originated at individual nodes at all time frames. From this table it can be seen that for this benchmark circuit SEU^0 s have high impact on the output error probabilities than SEU^1 s. Er-

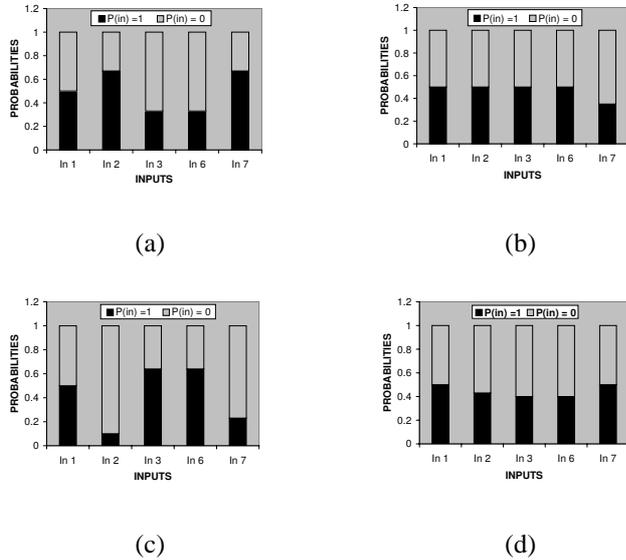


Fig. 8. Input probabilities for achieving zero output errors (at nodes 22 and 23 in presence of SEU's: (a) SEU^0 at node 19 (b) SEU^1 at node 19 (c) SEU^0 at node 11 (d) SEU^1 at node 11 for c17 benchmark

ror probability at output node 22 due to an SEU^1 at node 11, is very low (0.0625) whereas error probability at output 22 due to SEU^0 at 11 is 0.3125. It also shows that the effect of SEUs are not the same over all outputs. For example, an SEU^1 at node 19 causes no error at output 22 whereas error probability due to this SEU at output node 23 is 0.4375. Note that nodes 22 and 23 are the output nodes. SEUs occurring at these nodes at sampling time t_s or time $t_s - 1$ will be latched by an output latch, and are expected to cause very high error probability. However from Table III, it is observed that probability of occurrence of an error due to SEU^1 at node 23 is only 0.4375. Similarly, probability of occurrence of an error due to SEU^1 at node 22 is also 0.4375. This is due to the type of input pattern. In this work, we assume random inputs. This result shows the dependence of input pattern on $P(T_{j\downarrow})$.

A.1 Input Space Characterization

In this section, we describe the input space characterization for a particular observation exploring the diagnostic (backtracking) feature of the TALI-SES model. Note that this feature makes it really unique as instead of predicting the effect of inputs and SEU at a node on the outputs, we try to answer queries like ‘‘What input behavior will make SEU at node j definitely causing a bit-flip the at circuit outputs?’’ or ‘‘What input behavior will be more conducive to no error at output given that there is an SEU at node j ?’’ Resolving queries like this, aids the designer in observing the input space and helps perform input clustering or modeling. Let us take an example of c17 benchmark. We explore the input space for studying the effect of SEU^0 and SEU^1 at node 19 on errors on both the outputs (22 and 23). One can characterize input space for any one of the outputs (or in general effect of SEU at any node on any other subset of nodes). Fig 8a characterizes the input space for an

SEU^0 at node 19 such that no bit-flip occur at the outputs. This is done by setting the output error probability at zero (by giving ‘‘evidence’’ to the detection nodes in the Bayesian Network) and then back propagating the probabilities. We plot the probabilities of each inputs 1, 2, 3, 6 and 7 that gives no output error for an SEU^0 at 19. Each column in the plot represents an input. The lighter color represents the probability of that input = 0 and the black color represents the probability of input = 1 (sum of these two part should always be one). One can see that for obtaining zero output error with an SEU^0 at 19, input 1 can be random, input 2 and 7 have 65% probability of being at logic one and node 3 and 6 has probability of 30% for logic 1. Note that the input space is nearly random ($p(1)=p(0)=0.5$) when SEU^1 at node 19 produces zero output error at both the outputs. Similar characteristics are shown in Fig. 8c, 8d for characterizing the input space with respect to output errors while SEU^0 or SEU^1 occurs at node 11. Once again it can be seen that zero output error for SEU^1 can be more likely by a random inputs than for SEU^0 .

B. Larger Benchmarks

We use approximate inference for larger circuit using Probabilistic Logic sampling [17] which is pattern independent random markov chain sampling and has shown good results in many large industry-size applications.

In Fig. 9(a), we plot the number of gates and the number of possibly sensitized SEUs for ISCAS benchmarks. This reduced SEU list was created based on fanout-dependent delay model and assuming an SEU duration δ equal to one time unit. We get a considerable reduction in the number of listed SEUs compared to the number of gates in a circuit. This is because reduced SEU list is generated by traversing backward from the final outputs evaluated at sampling time t_s and $t_s - 1$ and only those gates that lie between the final outputs and duplicate gates need to be considered for SEU sensitivity analysis. Depending on the input pattern and the circuit structure, only a few of these SEUs actually cause soft errors. Based on the estimated SEU sensitivity $P(T_j)$ as explained in Section III D, we classify the SEU sensitive gates in a circuit into three categories, gates where $P(T_j)$ is (i) less than or equal to 0.3 (ii) between 0.3 and 0.6 and (iii) above 0.6. This is plotted in Fig. 9(b). These results are helpful to apply selective redundancy measures or to modify $P(SEU_j)$ (by changing device features) by giving higher priority to nodes those are in the high sensitivity range than those in the lower sensitivity ranges. From Fig. 9(b), it can be seen that the SEU sensitive nodes of circuit c432 are equally distributed within the three probability ranges (i), (ii) and (iii), whereas all the SEU sensitive nodes in circuit c1355 lie within the middle range where $P(T_j)$ is between 0.3 and 0.6. Results of c7552 shows that $P(T_j)$ of most of the SEU sensitive nodes is in the lowest range (less than or equal to 0.3), which indicates that gates in this circuit do not require extensive hardening techniques, whereas majority of SEU sensitive gates in c2670 requires extensive hardening techniques since $P(T_j)$ is very high (above 0.6) for these nodes.

We implemented the SEU simulator based on the work done in [4] with a fanout-dependent delay model for the ground truth. We performed the simulation with 500,000 random vectors obtained by changing seed after every 50000 vectors to

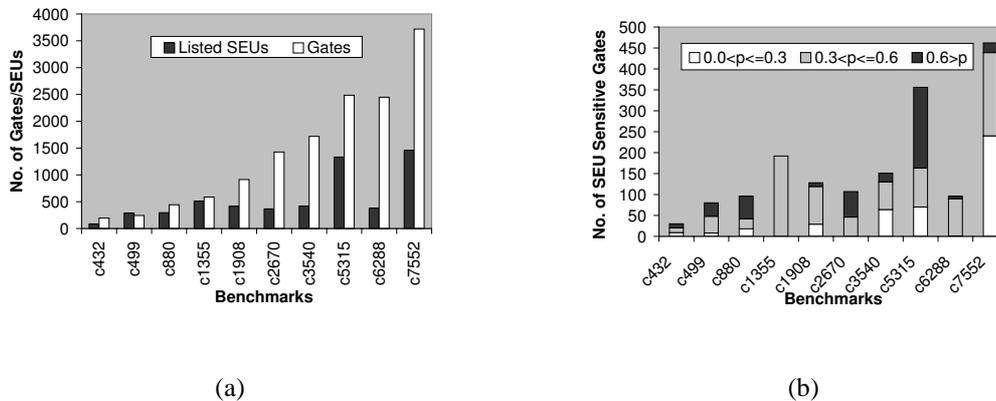


Fig. 9. (a)SEU List-Fanout Dependent Delay Model (b)SEU Sensitivity Range-Fanout Dependent Delay Model, with Delta=1; Input Bias=0.5

TABLE IV
SEU SENSITIVITY ESTIMATION ERRORS AND TIME FOR 9999 SAMPLES.

	(E_{mean})	(E_{max})	$T_{bn}(sec)$
c432	0.0031	0.0069	18.57
c499	0.0024	0.0198	13.43
c880	0.0027	0.0090	27.58
c1355	0.0027	0.0120	28.84
c1908	0.0028	0.0120	176.63
c2670	0.0034	0.0130	34.70
c3540	0.0023	0.0101	148.07
c5315	0.0045	0.0112	121.62
c7552	0.0035	0.0100	513.05

get the ground-truth SEU probabilities. For our probabilistic framework, we use Probabilistic Logic Sampling [17] inference scheme. We compute the SEU sensitivities P_j of gates in ISCAS benchmark circuits using Probabilistic Logic Sampling (PLS) [17] with 9999 samples and compare our results with ground-truth simulation results. Table IV gives the average estimation error E_{mean} and maximum estimation errors E_{max} . Here E_{mean} of a circuit is the average of difference between the SEU detection probabilities (or SEU sensitivities) obtained from simulation and estimated probabilities from PLS sampling over all possible SEU sensitive nodes in the circuit. Similarly E_{max} of a circuit is the maximum of difference between the SEU sensitivities obtained from simulation and estimated SEU sensitivities from PLS sampling over all possible SEU sensitive nodes in the circuit. We estimated the SEU sensitivities all the ISCAS'85 benchmarks with an average belief propagation time of 140.49 sec, whereas the average time taken for logic simulation of these circuits is 33 hours. Estimation error over all benchmarks is below 0.0034 which shows excellent accuracy-time trade-off. T_{bn} is the total elapsed time, including memory and I/O access.

C. Results with Delay Model based on Logical Effort

In this section we give estimation results from our model with logical effort based gate delay modeling. In Table V, we list the number of nodes in TALI Bayesian network and the estimation time in seconds for some of the ISCAS benchmarks. Number of TALI nodes depends on the SEU list as well as the circuit size, whereas estimation time directly depends on the number

of nodes and the number of samples. We show results for Probabilistic Logic Sampling (PLS) with 9999 samples.

Figure 10(a) shows the number of possibly sensitized SEUs vs. the number of gates in ISCAS benchmarks. From this graph, it can be seen that the number of SEUs in the reduced SEU list is low compared to fanout dependent delay model. This is due to high gate delay values with logical effort based delay modeling since we take into account the input capacitance as well as parasitic delay in addition to fanout. Due to increased gate delays the relative effect of an SEU at an internal gate on a primary output during latching period is less since most of the signals get enough time to restore to their ideal values. Figure 10(b) shows the SEU sensitivity ranges of gates in the circuits, with an input bias of 0.5 and SEU width equal to one time unit. As with fanout-dependent delay modeling, here also we classify the SEU sensitive gates in a circuit into 3 categories. Gates with estimated sensitivity values (1) less than 0.3, (2) between 0.3 and 0.6 and (3) above 0.6. Given any delay library for a logic circuit, our model can be used to classify the gates in the circuit in the order of their SEU sensitivity values capturing logical masking effect, circuit structure, input pattern and SEU duration.

Please note the above estimated probability values are relatively high when we consider the overall soft error susceptibility of individual gates. To get a comprehensive model, the electrical masking effect, latching window masking effect and also the SEU generation and propagation characteristics of individual gates are to be incorporated with our model. Modeling electrical masking effect needs circuit level simulation techniques, which we are trying to integrate with our current approach as a future direction.

VI. CONCLUSION

We are able to effectively model Single-event-Upsets in logic circuits (ISCAS benchmarks) to estimate the SEU sensitivity of individual nodes in a circuit capturing spatial and temporal signal correlations, specially emphasizing the effect of inputs, gate delay, SEU duration and circuit structure. We show results with exact and approximate inferences. Using exact inference we characterize input space which gives zero output error even in the presence of some SEUs. Results from approximate infer-

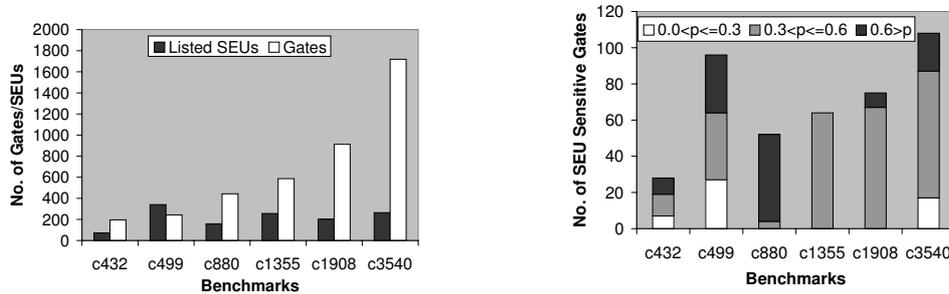


Fig. 10. (a)SEU List-Logical Effort Delay Model (b)SEU Sensitivity Range-Logical Effort Delay Model with Delta =1 and Input Bias = 0.5

TABLE V

SIZE OF TALI-MODEL AND ESTIMATION TIME FOR LOGICAL-EFFORT BASED DELAY MODEL

	# of nodes (TALI)	Estimation Time(s)
c432	2390	22.32
c499	7814	65.75
c880	1097	12.49
c1355	1773	15.092
c1908	2279	22.22
c3540	14370	135.79

ence shows excellent accuracy-time trade-offs. We report SEU sensitivity estimates for fanout dependent delay model as well as for logical effort based delay model. Given an appropriate delay library of gates in a circuit, our model is capable of estimating SEU sensitivities of individual gates in the circuit and these results can be used for classifying gates for application of mitigation schemes. Future effort includes modeling with biased input patterns and also for different SEU width δ , to study the effect of these factors on SEU sensitivities. We are also investigating on the effect of threshold voltage and supply voltage on the electrical masking effect on transient pulses caused by particle bombardment.

REFERENCES

- [1] K. Mohanram and N. A. Touba, "Cost-Effective Approach for Reducing Soft Error Failure Rate in Logic Circuits," *International Test Conference*, pp. 893–901, 2003.
- [2] S. Bhanja and N. Ranganathan, "Switching Activity Estimation of VLSI Circuits using Bayesian Networks" *IEEE Transactions on VLSI Systems*, pp. 558–567, Feb. 2003.
- [3] P. Shivakumar, et al., "Modeling the Effect of Technology Trends on the Soft Error Rate of Combinational Logic," *Proc. International Conference on Dependable Systems and Networks*, pp. 389–398, 2002.
- [4] M. Violante, "Accurate Single-Event-Transient Analysis via Zero-Delay Logic Simulation," *IEEE Transactions on Nuclear Science*, Vol. 50, No. 6, pp. 2113–2118, 2003.
- [5] T. Rejimon and S. Bhanja, "An Accurate Probabilistic Model for Error Detection," *Proc. IEEE International Conference on VLSI Design*, pp. 717–722, Jan. 2005.
- [6] T. Karnik and P. Hazucha, "Characterization of soft errors caused by single event upsets in CMOS processes," *IEEE Transactions on Dependable and Secure Computing*, Volume: 1-2, pp. 128–143, Apr-Jun. 2004.
- [7] V. Degalahal, R. Rajaram, N. Vijaykrishan, Y. Xie and M. J Irwin, "The effect of threshold voltages on soft error rate," *5th International Symposium on Quality Electronic Design*, March 2004.
- [8] Y. S. Dhillon, A. U. Diril and A. Chatterjee, "Soft-Error Tolerance Analysis and Optimization of nanometer circuits," *Proceedings of Design, Automation and Test in Europe*, Volume: 1, pp. 288–293, Mar. 2005.
- [9] S. Krishnaswamy, G. S. Viamontes, I. L. Markov, and J. P. Hayes, "Ac-

- curate Reliability Evaluation and Enhancement via Probabilistic Transfer Matrices", *Design Automation and Test in Europe (DATE)*, March 2005.
- [10] P. K. Samudrala, J. Ramos and S. Katkooi, "Selective Triple Modular Redundancy (STMR) Based Single-Event-Upset (SEU) Tolerant Synthesis for FPGAs," *IEEE Transactions on Nuclear Science*, Vol. 51, No. 5, Oct. 2004.
- [11] Chong Zhao, Xiaoliang Bai and S. Dey, "A scalable soft spot analysis methodology for compound noise effects in nano-meter circuits," *Proceedings of Design Automation Conference*, pp. 894–899, Jun. 2004.
- [12] M. Zhang and N. R. Shanbhag, "A Soft Error Rate Analysis (SERA) Methodology" *International Conference on Computer Aided Design*, November, 2004.
- [13] URL <http://www.hugin.com>
- [14] "GeNie", URL <http://www.sis.pitt.edu/~genie/genie2>
- [15] S. Manich and J. Figueras, "Maximizing the weighted switching activity in combinational CMOS circuits under the variable delay model," *European Design and Test Conference*, pp. 597–602, 1997.
- [16] S. Bhanja, K. Lingasubramanian and N. Ranganathan, "A Stimulus-Free Probabilistic Switching Model for Sequential Circuits," *ACM Transactions on Design Automation and Electronic Systems*, 2006.
- [17] M. Henrion, "Propagation of uncertainty by probabilistic logic sampling in Bayes' networks," *Uncertainty in Artificial Intelligence*, 1988.
- [18] J. Pearl, "Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference," Morgan Kaufmann Publishers, Inc., 1988.
- [19] R. G. Cowell, A. P. David, S. L. Lauritzen, D. J. Spiegelhalter, "Probabilistic Networks and Expert Systems", Springer-Verlag New York, Inc., 1999.
- [20] I. Sutherland, R. Sproull and D. Harris, "Logical Effort: Designing Fast CMOS Circuits", Morgan Kaufmann, February 1999.
- [21] N. Ramalingam and S. Bhanja, "Causal Probabilistic Input Dependency Learning for Switching Model in VLSI Circuits", *ACM GLSVLSI*, 2005.

Thara Rejimon received her BE in Electrical and Electronics Engineering from Kerala University, India and MS in Electrical Engineering from Regional Engineering college, Kerala, India. She also received MS Degree in Electrical Engineering with concentration in Digital Electronics from University of South Florida in year 2002 and is currently pursuing PhD at the University of South Florida. Her research interests are radiation tolerant design, probabilistic computing and error modeling for nano-domain circuits and testing under nano-domain.

Sanjukta Bhanja received BE and MSc(Engg.) in Electrical Engineering from Jadavpur University, Calcutta and the Indian Institute of Science, Bangalore, India, respectively. She received her Ph. D in Computer Science and Engineering from University of South Florida, Tampa in Dec. 2002. She is currently an Assistant Professor in the Department of Electrical Engineering at the University of South Florida, Tampa. Her research interests include VLSI design automation, reliable computing with inherent errors, nano-electronics(Quantum-dot-cellular-automata), etc.