A Stimulus-free Probabilistic Model for Single-Event-Upset Sensitivity

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) 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, circuit structure and delay on Single-Event-Upset sensitivity 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.

1 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 which may cause a 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. As process technology scales down, due to high operating frequencies, low supply voltages, shrinking device geometry and small noise margin, soft errors are of serious concern in logic circuits. Soft error susceptibility of a node j with respect to a latch, $SES_{j\_Q_L}$ is the soft error rate at the latch output Q_L, contributed by node j. 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 $P(T_{j\_i}|SEU_j)$ is the probability of occurrence of an error at output node i, given a particle hit at node j causes an SEU. Let $P(SEU_j)$ be the probability that a particle hit at node j generates an SEU at that node 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.

$$SES_{j\_Q_L} = R_H.P(SEU_j)P(T_{j\_i}|SEU_j)P(Q_L|T_{j\_i}) \tag{1}$$

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 $P(T_{j\_i}|SEU_j)$ 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*. Several works have been done on soft error analysis which estimate the overall output signal errors due to SEUs at the internal nodes [5, 6, 8, 9, 10] . Note that our focus is to identify the SEU locations which cause soft errors at the output(s) with high probabilities and **not** on the overall soft error rates. Knowledge of relative contribution of individual nodes to output error will help designers to apply selective radiation hardening techniques. This model can be fused with the modeling of the latches [9] considering latching window, setup and hold time etc and also with the process parameters like V_{th}, V_{dd} [5, 6, 9] for a comprehensive model capturing processing, electrical and logical effect.

We model internal dependency of the signals in timing-aware model such that the SEU sensitization probability ($P(T_{j\_i}|SEU_j)$) captures the effect of circuit structure, circuit path delay, type of SEU (SEU^1 caused by 0_1_0 transition or SEU^0 caused by 1_0_1 transition) and also the input space. To embed time-related information in the circuit topology, we expand the circuit by means of a time-space transformation which is similar to that presented in [4, 14]. Due to the temporal nature of SEUs, all of the generated SEUs do not cause soft errors. We discuss the timing aspects in Section 3.1. From the expanded circuit and a reduced 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 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 [11] in VLSI but their use in timing-aware modeling of Single-Event-Upsets is new. We use two inference schemes for probabilistic updating. An exact method also known as clustering technique [12] and an approximate non-simulative, stimulus-free inference scheme known as Probabilistic Logic Sampling (PLS) [16]. These inference schemes are discussed in detail in section 4.

This paper is organized as follows. Section 2 is a summary of the prior works done on soft error modeling and analysis. In section 3 we discuss our model in detail explaining the timing issues, features of Bayesian network-based modeling and the proposed TALI-SES model. Section 4 deals with Bayesian inference where we discuss both exact and approximate(stochastic) inference schemes. In section 5 we give experimental results. We compare our results with logic simulation results and show that this novel probabilistic estimation technique is accurate (close-to-zero error) and efficient.

2 Background

Effects of Single-Event-Upsets have been studied by several authors in the past [18, 19]. A concurrent error detection scheme to detect and reduce soft error failure rates have been described in [1]. 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]. Results from their model predict that SER per chip will increase nine orders of magnitude from 1992 to 2011 and then it will become comparable to the soft error failure rates of unprotected memory chips. A model to analyze Single Event Upsets with zero-delay logic simulation 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. A mathematical model to estimate the possible propagation of glitches due to transient faults has been presented in [15]. Zhang *et al.* in [9] proposed a composite soft error rate analysis method (SERA). This method uses a conditional probability based parameter extraction technique by resorting to device and logic level simulation. In their work, combinational circuits are assumed to have unbalanced re-convergent paths. This is not true even for small benchmarks such as c17. Another method for soft error tolerance analysis and optimization has been discussed in [7] by modeling glitch generation and propagation characteristics of gates in a circuit. However, they use circuit level simulation results to incorporate logical masking effects.

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), 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 SEU duration on the logical masking. There is future scope for this model to be fused with other models [5, 6, 9, 10] for capturing device effects such as electrical masking, threshold voltage and supply voltage.

3 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 SEU list.

3.1 Timing Issues

Due to the temporal nature of SEUs, all of the generated SEUs may not cause soft errors. For example, an SEU effect may reach the output for a short time span, but the output signal can be reinforced to its correct value before it is sampled by the latch. 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]:

$$t_h + \delta + t_d < t_s \quad \forall t_d \in \Pi. \tag{2}$$

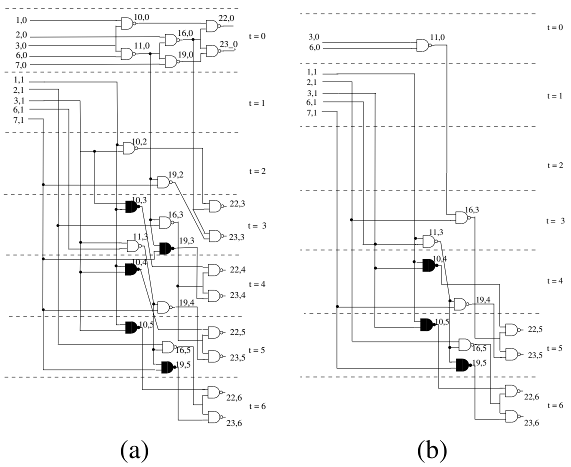


Figure 1. (a) Time-space transformed circuit of benchmark c17, modeling all SEUs. (b) Modified time-space transformed circuit of benchmark c17, modeling only the possibly sensitized SEUs.

We first expand the circuit by time-space transformation of the original circuit, without changing its functionality. A gate in the original circuit C will have many replicated gates in the expanded circuit 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 fanout. For each gate g whose output is evaluated at time $t \in \{T\}$ a replicate node g, t is constructed. 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 $t' < t$.

Fig. 1(a) is the expanded circuit of benchmark $c17$. 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 at the output of gate 10 at $t = 2$.

Input signals of certain gates in the circuit might have different arrival time due to the difference in path delays. We introduce additional duplicate nodes for those internal signals with less path delay in order to model the effect of any SEU generated at the junction of the gates at times later than the signal's arrival. For example, in Fig. 1(a), 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 $(3, 1)$. In Fig. 1(a), it can be seen that (10, 3), (10, 4), (10, 5), (19, 3) and (19, 5) are duplicate gates, represented by filled gate symbols.

Output signals of a circuit are sampled at $t = t_s$, where t_s is the maximum of the latest arrival time of 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 δ. For example in Fig. 1(a), 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.

Based on the above considerations, we modify the expanded circuit by including only those gates that propagate SEU effects to the outputs between time instants, t_s and $t_s - \delta$. Thus we get a considerable reduction in the circuit size. Fig. 1(b) is the modified expanded circuit of c17, which models all SEUs possibly sensitized to a final output.

3.2 Bayesian Networks

A Bayesian network[17] is a Directed Acyclic Graph (DAG) in which the nodes of the network represent random variables and a set of directed links connect pairs of nodes. The links represent causal dependencies among the variables. Each node has a conditional probability table (CPT) except the root nodes. Each root node has a prior probability table. The CPT quantifies the effect the parents have on the node. Bayesian networks compute the joint probability distribution over all the variables in the network, based on the conditional probabilities and the observed evidence about a set of nodes.

The exact generalized joint probability distribution over the n variables in a Bayesian Network is given by Eq. 3.

$$
\begin{array}{c}
P(x_n, x_{n-1}, \cdots, x_k, \cdots, x_3, x_2, x_1) \\
= P(x_n | x_{n-1}, \cdots, x_k, \cdots, x_3, x_2, x_1) \\
\cdots\cdots P(x_k | x_{k-1}, x_{k-2}, \cdots, x_2, x_1) \\
\cdots\cdots P(x_2 | x_1) P(x_1)
\end{array}
\tag{3}
$$

Here x_k denotes some value of the variable X_k. Let $pa(x_k)$ be a set of values for X_k's parents. In a Bayesian

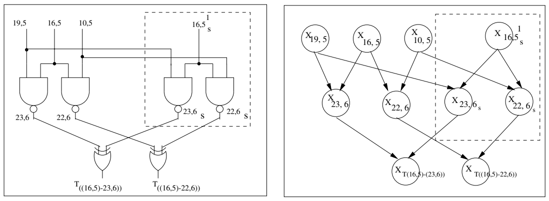


Figure 2. (a) An illustrative SEU sensitivity logic for a sub-circuit of c17. (b) Timing-aware-Logic-induced-DAG model of the SEU sensitivity logic in (a)

Network, the node X_k becomes conditionally independent of all the other nodes given only its parents. Using this conditional independence, a minimal factored representation of exact joint probability distribution over n random variables can be expressed as in Eq. 4.

$$P(X) = \prod_{k=1}^{n} P(x_k|pa(x_k)) \tag{4}$$

We adapted the theoretical understanding based on [17]. The reader is referred to these works for details.

3.3 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. 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, $< C', J >$ where C' is the expanded circuit created by time-space transformation as discussed in section. 3.1 and J is the set of possible SEUs (also discussed in section 3.1). 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 2(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. 2(a), the block with the dotted square is the sensitization logic for $16, 5^1_s$ [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 $SEU^1_{16,5}$. 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 3.1, 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 time frame $t_s - 1$ also. 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, (represented as $(j,\ t)_s$) are $T_{(j,\ t)\_\ (i,\ t_s)}$ and $T_{(j,\ t)\_\ (i,\ t_s-1)}$. If any of these test signal values 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 $P(T_{(j,\ t)\_\ i}) = \max\{P(T_{(j,\ t)\_\ (i,\ t_s)}), P(T_{(j,\ t)\_\ (i,\ t_s-1)})\}$. 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 $T_{(16,5)\_\ (23,6)}$.

More than one SEU can originate at a node at different time frames. Considering the effect of both types of SEUs, SEU^0 and SEU^1, at node j at all time frames, we compute the worst case output error probability due to node j as $P(T_j) = \max_{\forall t}\{P(T^1_{(j,t)}), P(T^0_{(j,t)})\}$, which is the maximum probability over all time frames of SEU^0 and SEU^1.

In TALI-SES DAG structure the parents of each node are its Markov boundary elements. Hence TALI-SES is a boundary DAG. For definition of Markov Boundary and boundary DAG please refer to [17]. 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 [17] that if graph structure is a boundary DAG D of a dependency model M, then D is a minimal I-map of M ([17]). This theorem along with definitions of conditional independencies, in [17] (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).

Quantification of TALI-SES-BN: TALI-SES-BN consists of nodes that are random variables of the underlying probabilistic model and edges denote direct dependencies. All the edges are quantified with the conditional probabilities making up the joint probability function over all the random variables. The overall joint probability function that is modeled by a Bayesian network can be expressed as the product of the conditional probabilities. Let us say, $X^{'} = \{X^{'}_{1}, X^{'}_{2}, \cdots, X^{'}_{m}\}$ are the node set in TALI-SES Bayesian Network. Then,

$$P(X^{'}) = \prod_{k=1}^{m} P(x^{'}_{k}|Pa(X^{'}_{k})) \qquad (5)$$

where $Pa(X^{'}_{k})$ is the set of nodes that has directed edges to $X^{'}_{k}$. A complete specification of the conditional probability of a two input AND gate output will have 2^3 entries, with 2 states for each variable. These conditional probability specifications are determined by the gate type. By specifying the appropriate conditional probability we ensure that the spatial dependencies among sets of nodes (not only limited to just pair-wise) are effectively modeled.

4 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.

4.1 Junction Tree Based Inference

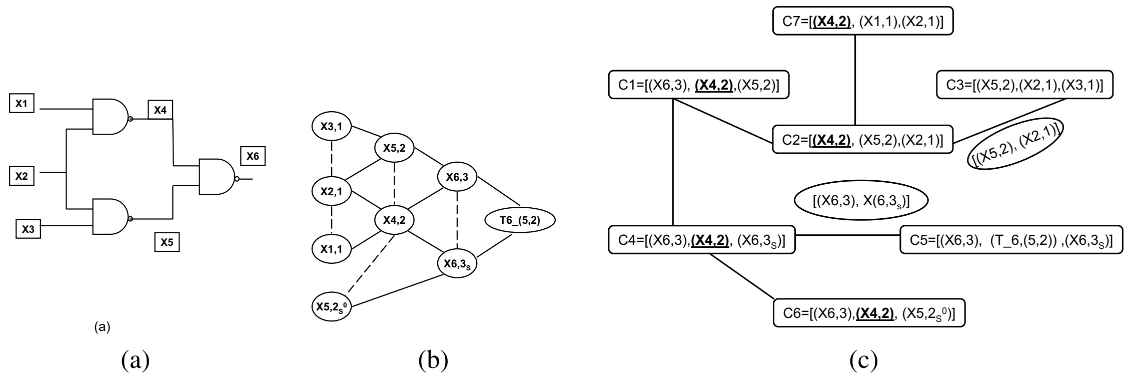
We demonstrate this inference scheme with an example. A small combinational circuit is shown in Fig. 3a. We first construct a Bayesian network for a subset of the time transformed circuit of Fig. 3a that captures the effect of SEU of "zero" at node 5 at a time instant 2 unit (denoted by the random variable $X_{5,2}s^{0}$) on the output signal 6 at 3 time unit(denoted by random variable $X_{6,3}$) Fig. 3b is the moralized graph (discussed below) derived from the Bayesian Network. Note that the error in output signal $X_{6,3}$ is determined by $T_{6\_(5,2)}$ 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 Network 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 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 [17]. The dependencies that are preserved in the original DAG are also preserved in the moral graph [17]. The dashed edges in Fig. 3b are added at this stage. This step ensures that every parent child set is a complete sub graph. Triangularization: In this step, every cycle of length greater than or equal to 4 is reduced to a cycle of 3 by inserting additional links (chords) to the moral graph. The moral graph is said to be triangulated if it is chordal [17]. 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 3c). Junction tree possesses running intersection property [17] 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. 3c is the junction tree derived from the chordal graph of Fig. 3b in this example. Interested readers are referred to [2] for a detailed description of how local message passing is performed in junction trees. **Probabilistic Backtracking:** 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 propagate it 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 output error in presence of SEU(s)) 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 backtracking 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 [16], without the added baggage of high space complexity.

4.2 Probabilistic Logic Sampling (PLS)

Probabilistic logic sampling is the earliest and the simplest stochastic sampling algorithm proposed for Bayesian Networks [16]. 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.


X1
X2
X3
X4
X5
X6
(a)
X3,1
X5,2
X2,1
X6,3
X4,2
T6_(5,2)
X1,1
X6,3s
X5,2s0
C7=[(X4,2), (X1,1),(X2,1)]
C1=[(X6,3), (X4,2),(X5,2)]
C3=[(X5,2),(X2,1),(X3,1)]
C2=[(X4,2), (X5,2),(X2,1)]
[(X5,2), (X2,1)]
[(X6,3), X(6,3s)]
C4=[(X6,3),(X4,2), (X6,3S)]
C5=[(X6,3), (T_6,(5,2)) ,(X6,3S)]
C6=[(X6,3),(X4,2), (X5,2S0)]

(a) (b) (c)

Figure 3. (a) A Small Circuit (b) Chordal Graph (Same as Moral Graph in this example) (c) Junction Tree

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.

The above scheme is efficient for predictive inference, when there is no evidence for any node, but is not efficient for diagnostic reasoning due to the need to generate, but disregard samples that do not satisfy the given evidence. We adopt the tool GeNie [13] 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 $n.2^{|Cmax|}$ [2], where n is the number of nodes in the Bayesian Network, and $|Cmax|$ is the number of variables in the largest clique. The time complexity is given by $p.2^{|Cmax|}$ [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.

5 Experimental Results

We demonstrate the modeling of SEU based on TALI-SES using ISCAS benchmark circuits. The logical relationship between 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 1 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.

Table 1. Size of original and time-expanded ISCAS circuits

	Gates	Gates expanded	# of nodes (TALI)	Time frames
c432	196	476	1947	55
c499	243	464	1596	30
c880	443	729	2486	51
c1355	587	1440	3388	55
c1908	913	1524	18020	79
c2670	1426	2584	4097	81
c3540	1719	3795	15472	93
c5315	2485	4887	13138	90
c6288	2448	30113	31157	263
c7552	3719	10006	45367	88

Table 2. estimated $P(T_{j\_i})$ of nodes in c17 (Exact Inference)

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	0.2188	0.3125	0.5625
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

5.1 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 44.1. Note that this inference is proven to be exact [17](zero estimation error).

Table 2 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 to an SEU at each node j (column 1) namely $(10,\ 11,\ 16,\ 19,\ 22\ and\ 23)$. Column 2 and 3 of Table 2 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 3 3.3). Columns 2, 3, 4 and 5 of Table 2 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^0s have high impact on the output error probabilities than SEU^1s. Error 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 2, 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 $P(T_{j\_i}|SEU_j)$.

5.1.1 Input Space Characterization

In this section, we describe the input space characterization for a particular observation exploring the diagnostic 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 at the 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 4a characterizes the input space for an SEU^0 at node 19 such that no bit-flip occur at the outputs. We plot the probabilities of each inputs $1, 2, 3, 6$ and 7 for observing no output error for an SEU^0 at 19. Each column in the plot represents an input. Now 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 observing no output error for an SEU^0 at 19, input 1 can be random, input 2 and 7 have 67% probability of being at logic one and node 3 and 6 has probability of 33% for logic 1. Note that if the input space is nearly random (p(1)=p(0)=0.5), then an SEU^1 at node 19 causes no output error for both the outputs. Similar plots are shown in Fig. 4c and 4d for characterizing the input space for zero output errors while SEU^0 or SEU^1 occurs at node 11 of c17 benchmarks. It can be seen that zero output error for SEU^1 can be more likely by random inputs than for SEU^0.

Table 3. SEU Sensitivity ranges of gates in ISCAS circuits, when $\delta = 1$ and input bias = 0.5

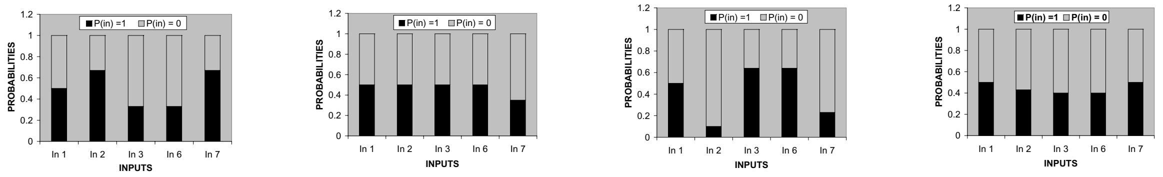
	# of SEU's	$p \leq 0.3$	$0.3 < p \leq 0.6$	$p > 0.6$
c432	86	9	11	10
c499	288	8	40	32
c880	296	18	24	54
c1355	512	0	192	0
c1908	414	29	90	9
c2670	366	0	46	61
c3540	418	75	61	15
c5315	1332	76	87	193
c6288	1332	1	89	6
c7552	1460	240	199	23

5.2 Larger Benchmarks

We use approximate inference for larger circuit using Probabilistic Logic sampling [16] which is pattern independent random markov chain sampling and has shown good results in many large industry-size applications.

In Table 3, we report the number of possible SEU's causing soft errors (column 2). The reduced SEU list was created based on fanout-dependent delay model and assuming an SEU duration period δ equal to one time unit. We report the estimated SEU sensitivity $P(T_j|SEU_j)$ for all gates in the circuits that might cause error. Column 3 of Table 3 lists the number of all gates j where $P(T_j|SEU_j)$ is less than or equal to 0.3, whereas columns 4 and 5 list number of gates for which $P(T_j|SEU_j)$ lies in between 0.3 and 0.6, and above 0.6 respectively. These results are helpful to identify and classify nodes and apply redundancy measures or modify $P(SEU_j)$ (by changing device features) to nodes selectively giving higher priority to nodes in column 5 and than that in lower ranges.

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 get the ground-truth SEU probabilities. For our probabilistic framework, we use Probabilistic Logic Sampling [16] inference scheme. SEU detection probabilities computed by Probabilistic Logic Sampling (PLS) [16] with 9999 samples were compared with



(a) (b) (c) (d)

Figure 4. Input probabilities which give zero output errors for $c17$, in the 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

Table 4. SEU Sensitivity estimation errors and time for 9999 samples.

	(E_{mean})	(E_{max})	$T_{bn}(sec)$
c432	0.0031	0.0069	18.18
c499	0.0024	0.0198	13.43
c880	0.0027	0.0090	28.31
c1355	0.0027	0.0120	28.84
c1908	0.0028	0.0120	175.67
c2670	0.0034	0.0130	34.70
c3540	0.0023	0.0101	146.20
c5315	0.0045	0.0112	120.79
c7552	0.0035	0.0100	507.02

ground-truth simulation results and average and maximum estimation errors, E_{mean} and E_{max} are reported in columns 2 and 3 respectively of Table 4. Propagation time, T_{bn} (column 4) is the time taken by the PLS scheme for belief propagation. We estimated the SEU sensitivities of all the ISCAS'85 benchmarks with an average belief propagation time of 140.49 sec, whereas the average estimation error is below 0.0034 which shows an excellent accuracy-time trade-off. $T_{bn}$is the total elapsed time, *including memory and I/O access*. This time is obtained by the ftime command in the WINDOWS environment on a Pentium-4 2.0 GHz PC.

6 Conclusion

We are able to effectively model the SEU sensitivity of individual nodes capturing spatial, temporal correlations, specially emphasizing the effect of inputs, gate delay, SEU duration and circuit structure. We show results with exact and approximate inferences showing excellent accuracy-time trade-offs. Moreover, we are able to characterize input space with respect to SEU causing error and no error. Future effort includes modeling with biased input patterns, for different SEU width δ and also for other delay models, to study the effect of these factors on SEU sensitivities.

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 Transaction 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. Sonza Reorda and M. Violante, "Accurate and efficient analysis of single event transients in VLSI circuits ," *On-Line Testing Symposium*, pp. 101–105, 2003.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] M. Zhang and N. R. Shanbhag, "A Soft Error Rate Analysis (SERA) Methodology" *International Conference on Computer Aided Design*, November, 2004.

[10] Norbert Seifert *et al.* "Impact of Scaling on Soft-Error Rates in Commercial Microprocessors" *IEEE Transactions on Nuclear Science*,Volume: 49, No. 6, pp. 3100–3106, Dec. 2002.

[11] T. Rejimon and S. Bhanja, "An Accurate Probabilistic Model for Error Detection ," *Proc. IEEE International Conference on VLSI Design*, pp. 717–722, Jan. 2005.

[12] URL `http://www.hugin.com`

[13] `"GeNie", URL http://www.sis.pitt.edu/~genie/genie2`

[14] 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.

[15] M. Omana, G. Papasso, D. Rossi, C. Metra,"A Model for Transient Fault Propagation in Combinatorial Logic," *On-Line Testing Symposium*, pp. 111-115, 2003.

[16] M. Henrion, "Propagation of uncertainty by probabilistic logic sampling in Bayes' networks," *Uncertainty in Artificial Intelligence*, 1988.

[17] J. Pearl, "Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference," Morgan Kaufmann Publishers, Inc., 1988.

[18] J.T. Wallmark and S.M. Marcus,"Minimum size and maximum packaging density of non-redundant semiconductor devices," *Proceedings of IRE*, vol. 50, pp. 286–298, March 1962.

[19] G.H. Johnson, J.H. Hohl, R.D. Schrimpf and K.F. Galloway,"Simulating Single-event burnout in n-channel power MOSFETs," *IEEE Transactions on Electron Devices*, vol. 40, pp. 1001–1008, 1993.