A Monte Carlo Simulation Flow for SEU Analysis of Sequential Circuits

Meng Li, Ye Wang, and Michael Orshansky
Department of Electrical and Computer Engineering, University of Texas at Austin, USA
{meng_li, lhywang, orshansky}@utexas.edu

ABSTRACT

An efficient methodology for soft error analysis of sequential circuits based on Monte Carlo sampling is proposed. It uses nested sampling for faster statistical convergence: it samples only from the workload space and statically evaluates the conditional probability over the subspace of particle strike and circuit parameters. A novel check on the stationarity of machine state sequence to reduce the number of samples to convergence is introduced. The flow combines logic simulation for latch-level error propagation and stationarity diagnostic and an improved combinational error simulator with a new masking model based on signal controllability.

Experiments show that nested sampling reduces the number of samples by up to 1500X and runtime by up to 25X compared to direct sampling. Stationarity checking allows reducing sampling number by 25%, on average. The new latching window model permits accuracy of within 1% from SPICE, compared to a 12% error with a prior model.

1. INTRODUCTION

Vulnerability of integrated circuits to single event upsets (SEU) caused by extrinsic radiation has become a significant and growing reliability concern. SEUs arise when a high-energy particle, typically, an alpha particle or a by-product of a neutron decay, hits the depletion region of a pn-junction between the drain (source) and the bulk terminals of a MOS transistor. The resulting voltage pulse, known as a single event transient, can produce a bit-flip at circuit registers as it propagates through combinational logic and gets latched into memory elements. It often persists over many cycles and may never be fully masked.

Accurate yet computationally scalable estimation of system-level error rate due to SEUs is challenging as it requires analysis at multiple levels of abstraction. Accurate sequential SEU estimation requires capturing the mechanisms of error propagation and masking at both combinational and sequential levels. At combinational level, much research has identified the importance of several masking mechanisms that modulate the probability of a SET propagating to a memory element [1, 2, 3]. Sequential circuit analysis adds the challenge of estimating fault propagation through multiple cycles efficiently. Several methods have been proposed to model, analyze, and estimate SER in sequential circuits [4, 5, 1, 6]. Signal-probability based methods [4, 5] are based on analytical computations of error probability which allows achieving high computational efficiency. However, these methods are generally not able to capture the complex temporal and spatial dependencies between internal machine states defined by a state transition graph [7, 8]. As a result, the level of accuracy allowed by the purely analytical methods is not generally sufficient. Symbolic methods [1, 6] eliminate explicit enumeration of input vectors and, thus, have high coverage but also cannot capture temporal correlations of internal states and typically have very high memory requirements. Sampling-based methods have also been proposed but without a focus on statistical convergence and ways to improve the rate of convergence [9].

In this paper we describe a soft error prediction flow for sequential circuits that is based on Monte Carlo sampling. Monte Carlo sampling is attractive in several aspects. It provides a feasible approach to dealing with an enormous space of possible workloads via sampling. Perhaps, its main attraction is that it is a flexible framework that is compatible with many requirements needed for accurate simulation. First, it allows ensuring that the faults are injected into legitimate states of a finite state machine (FSM). Second, it allows a static input-specific combinational circuit evaluation of key masking mechanisms.

The basic problem of any procedure based on Monte Carlo sampling is the question of convergence of the estimate. The main contribution of this paper is the flow based on the rigorous analysis of the following questions: (1) how to analyze the variance of the estimate, (2) how to know when the simulation can be terminated, i.e., what are the convergence criteria, and (3) what are the sampling and simulation strategies that are efficient specifically in the sense of error variance reduction.

A sequential circuit Monte Carlo faces the problem of a large sampling space: a state of the circuit and the values of the primary inputs when the strike occurs, the location of a particle strike and its timing within a given strike-cycle, the magnitude of charge deposited by the particle at a circuit node. Sampling the state space directly is difficult. Because in a sequential circuit the state is defined by a history of primary outputs, a random assignment of state bits leads to a highly inaccurate estimate of the error rate. We propose a history-dependent strategy: starting from a random internal state, we sample a sequence of primary inputs, until the internal state transitions approach a stationary state. In particular, we advocate using a check on the stationarity of the machine state sequence aimed at reducing the number of samples to con-

DAC '16, June 05-09, 2016, Austin, TX, USA

DOI: http://dx.doi.org/10.1145/2897937.2897967

vergence. The adopted heuristic check monitors the potential scale reduction factor to infer stationarity: its convergence to unity indicates high degree of stationarity [10]. As results indicate, injecting errors into non-stationary states leads to wasted simulation effort.

With an FSM state correctly sampled in stationary phase, the seemingly most natural sampling strategy, that we refer to as a direct approach, is to sample from the strike parameter space (within-cycle time, location, and charge magnitude) directly to compute the latch-level error probability. We show that the direct sampling strategy is sub-optimal from the convergence point of view. Instead, we advocate a hybrid static-sampling approach that exploits a conditional probability estimation property. In that approach, we sample the workload input space but statically evaluate the conditional expectation of error probability for the given input over the remaining sampling sub-space in its entirety. In effect, a much larger sample space is covered by each sample and the number of workload samples required for convergence is reduced. However, the specific computation occurring after each sampling is now different since the circuit-level simulator needs to compute the conditional probability rather than simply propagate a binary value through a circuit. Thus, the efficiency of the overall procedure is based on our ability to efficiently compute the conditional expectation of bit errors.

At the combinational level, we use a static algorithm that combines electrical cell pre-characterization in the style of static timing analysis and an efficient timing-window analysis to enable an accurate prediction of the probabilities of single event transients being latched: this captures the electrical and timing-window masking mechanisms. Because the input to a combinational circuit is fixed at the time of analysis, we are able to improve the accuracy of latching window (LW) modeling by incorporating signal controllability. This translates into the overall accuracy improvement. A single strike can impact multiple flip-flops and our algorithm captures such an event via a novel strategy of representing impacted flip-flops as sets along with the associated set error probability. This captures the impact of correlated multi-bit errors.

If the error probability for a flip-flop (FF) set is above a threshold, a set of bit-errors is injected and the simulation moves to the purely logic domain. The circuit outputs are monitored to determine if the error persists or is masked. A single simulation run is terminated if no internal latch shows an error (indicating the fault is masked) or after a prespecified number of cycles. The result of each such run is a binary outcome of error injection. The product of combinational error probability and the binary outcome contributes a data point to the Monte Carlo series. The individual estimates are not independent. For correlated samples, the convergence of this series is assessed continuously using the non-overlapping batch method [10].

2. MONTE CARLO FLOW AND ITS CONVERGENCE

In this section we present the proposed Monte Carlo flow and analyze its convergence properties. In particular, we demonstrate that the nested sampling strategies based on the conditional expectation approach achieves faster convergence than a direct method.

A sequential circuit can be formally abstracted as an FSM $F = \langle \Omega_I, \Omega_O, \Omega_S, \delta, \rho, s_0 \rangle$, where $\Omega_I, \Omega_O, \Omega_S$ denote the set of possible primary inputs (PI), outputs (PO), and internal states respectively. s_0 is the initial state in Ω_S. δ is the state-transition function:

$$\delta : \Omega_S \times \Omega_I \to \Omega_S,$$

and ρ is the output function:

$$\rho : \Omega_S \times \Omega_I \to \Omega_O.$$

A strike at cycle n can be characterized by a set of parameters p_n that include an indicator variable z_n capturing strike occurrence in cycle n, strike time t_n, location l_n and charge q_n: $p_n = (z_n, t_n, l_n, q_n)$. Possible particle strikes extend the error-free FSM F to a faulty FSM F' with the state-transition function δ' and the output function ρ', which have additional dependence on strike parameters p_n:

$$\delta' : \Omega_S \times \Omega_I \times \Omega_P \to \Omega_S,$$
$$\rho' : \Omega_S \times \Omega_I \times \Omega_P \to \Omega_O.$$

Here Ω_P represents the set of possible particle strikes. A particle strike generated at cycle n may affect the output ports of interest immediately at cycle n or be latched in FFs and affect the outputs later. We use an indicator variable e_{t_n} to denote whether a strike at cycle t_n ever causes an output error. We observe that e_{t_n} is a function of particle strike information p_{t_n}, current state s_{t_n}, and subsequent inputs i_m ($m \geq t_n$):

$$e_{t_n} = f(p_{t_n}, s_{t_n}, \{i_m\}).$$

In the following discussion, we use capital symbols to denote random variables in order to distinguish them from the associated realizations: e.g., E_{T_n} denotes the random variable for the indicator variable e_{t_n}.

The Monte Carlo simulation is outlined in Algorithm 1. The proposed flow is based on nested sampling using conditional expectation but we include direct sampling to highlight the differences.

Algorithm 1 Monte Carlo flow

```
1: procedure SEU ESTIMATION(F)
2:     sample S_0 and {I_n}_{n≥0};
3:     pre-generate states {S_n}_{n≥1};
4:     find N_0: {S_n}_{n≥N_0} are stationary;
5:     i ← 0;
6:     while r̂ or r' has not converged do
Direct sampling strategy:
7:         inject fault at t_i > N_0 on a single gate;
8:         find impacted POs/FFs w. combin. simulator;
9:         find impacted POs w. logic simulator;
10:        update r̂
11:        i ← i + 1;
Nested sampling strategy:
12:        sample cycle t_i > N_0 for error injection;
13:        find error probabilities of POs/FFs w. combin. simulator;
14:        for all FFs with error above threshold do
15:            find impacted POs w. logic simulator
16:        end for
17:        update r'
18:        i ← i + 1;
19:    end while
20: end procedure
```

Let N denote the total number of injected strikes at time t_1, t_2, ..., t_N. Then, an empirical finite-sample estimate of

the error rate is $\hat{r} = \sum_{i=1}^{N} e_{t_i}/N$. An important requirement for the Monte Carlo procedure is the reliable convergence of the empirical estimate. Specifically, the final estimate of $\hat{r}$ should not depend on the simulation trajectory defined by the initial conditions and the input sequence used to drive the simulation. In formal terms this translates to the requirement that the limit $\hat{r}^* = \lim_{N\to\infty} \sum_{i=1}^{N} e_{t_i}/N$ exists.

A state sequences $\{S_n\}$ of a typical sequential circuit can be modeled as an irreducible and aperiodic Markov chain with a stationary distribution π [11]. It can be shown that for these systems $\hat{r}^*$ indeed exists, i.e., is unique, and is equal to the expectation $\mathrm{E}_\pi[E_{T_n}]$ with respect to its stationary distribution [6]:

$$\hat{r}^* = \mathrm{E}_\pi[E_{T_n}].$$

For sequential circuits whose Markov chains lack the above properties the error estimates produced by simulations based on different initial conditions may not converge. Efficient checking of whether a given circuit satisfies these criteria is not dealt with in this paper but is an important challenge for future work.

The well-behaved convergence of $\hat{r}^*$ to $\mathrm{E}_\pi[E_{T_n}]$ enables the nested sampling strategy. According to the law of total expectation:

$$\mathrm{E}_\pi[E_{T_n}] = \mathrm{E}_{I,S}[\mathrm{E}_P[E_{T_n}|I,S]].$$

where $\mathrm{E}_P[E_{T_n}|I,S]$ is the conditional expectation of error rate evaluated over all possible strike injection sites for a given input and state. The nested sampling strategy uses this conditional probabilty in place of E_{T_i} defined for a single injection site. Now, the empirical error rate under the nested sampling strategy $\hat{r}'$ is:

$$\hat{r}' = \frac{\sum_{i=1}^{M} \mathrm{E}_P[E_{T_i}|I,S]}{M}.$$

where M is the number of cycles for which injections are evaluated. Critically, in terms of empirical convergence, the above analysis establishes that $\hat{r}^* = \lim_{N\to\infty} \hat{r} = \lim_{M\to\infty} \hat{r}'$.

The evaluation of $\mathrm{E}_P[E_{T_n}|I,S]$ needs to further comprehend two cases. A strike can affect primary outputs immediately at the injection cycle: we represent this possibility via an indicator variable $F_\emptyset$. Alternatively, a strike can first cause errors on one or more internal flip-flops. We capture this by identifying for each injection cycle the set of affected flip-flops (FF sets) and the associated error probability. We use the indicator variable F_i to represent the errors latched on the i-th set of FFs. Because $F_\emptyset$ and F_i's are exclusive, $\mathrm{E}[E_{T_n}|I,S]$ can be represented as:

$$\begin{aligned}\mathrm{E}_P[E_{T_n}|I,S] = Pr(F_\emptyset = 1|I,S) + \\ \sum_i \mathrm{E}_P[E_{T_n}|I,S,F_i=1]Pr(F_i=1|I,S).\end{aligned} \quad (1)$$

This decomposition is useful since $\mathrm{E}_P[E_{T_n}|I,S,F_i=1]$ can be evaluated efficiently through fast logic-level simulation. The computation of $Pr(F_\emptyset = 1|I,S)$ and $Pr(F_i = 1|I,S)$ is the focus of Section 3.

We now analyze the convergence behavior and prove that the procedure that relies on nested sampling and the use of conditional expectation has faster convergence than the direct sampling method. Since E_{t_n} and $\mathrm{E}_P[E_{T_n}|I,S]$ are functions of the Markov chain of the state sequence S, we rely on the weak Law of Large Numbers (LLN) for stationary Markov chains [11] to analyze the convergence rate of direct and nested sampling:

$$Pr(|\frac{\sum_{n=1}^{N} E_{T_n}}{N} - \mathrm{E}_\pi[E_{T_n}]| \geq \epsilon) \leq \frac{\sigma^2_{E_{T_n}}}{N\epsilon^2},$$

where σ^2 denotes the steady-state variance constant (SSVC), given by

$$\sigma^2_{E_{T_n}} = \mathrm{Var}(E_{T_n}) + 2\sum_{i=1}^{\infty} \mathrm{Cov}(E_{T_0}, E_{T_i}).$$

Intuitively, SSVC acts as the measure of total covariance for a Markov chain. Because an empirical average of conditional expectation $\mathrm{E}_P[E_{T_n}|I,S]$ is used in the nested sampling strategy, the corresponding SSVC becomes

$$\begin{aligned}\sigma^2_{\mathrm{E}_P[E_{T_n}|I,S]} =& \mathrm{Var}(\mathrm{E}_P[E_{T_n}|I,S]) + \\ & 2\sum_{i=1}^{\infty} \mathrm{Cov}(\mathrm{E}_P[\mathrm{E}_{T_0}|I,S], \mathrm{E}_P[E_{T_i}|I,S]).\end{aligned}$$

According to LLN, smaller SSVC implies faster convergence (i.e., smaller N for same ϵ). Now we show that nested sampling reduces SSVC enabling faster convergence. In order to demonstrate that conditioning improves convergence rate, we need the following Theorem regarding SSVC of conditional expectations [12]:

Theorem. *For a random process $\{X_n\}$ with a finite steady-state variance constant σ^2_X, the SSVC $\sigma^2_{\mathrm{E}[X|Y]}$ of its conditional expectation $\mathrm{E}[X_n|Y]$ on a random variable Y is no larger than σ^2_X,*

$$\sigma^2_{\mathrm{E}[X|Y]} \leq \sigma^2_X.$$

Thus, compared to direct sampling, sample SSVC is reduced in nested sampling, leading to faster convergence. Section 4 confirms this significant improvement in rate of convergence.

The above convergence analysis is based on the important assumption that we sample from the stationary distribution of FSM states. If we sample FSM states that do not represent a stationary distribution, the convergence is slower. To overcome this, we introduce a diagnostic that monitors a sequence of FSM states to determine the on-set of stationarity. The monitoring involves only fast logic-level simulation. We use a monitoring method based on the potential scale reduction factor (PSRF) to infer stationarity [10]. The method generates m independent sequences of length n that originate from different random seeds and convergence to stationarity is established when the parallel trajectories begin to have similar distributional properties as measured by the PSRF, defined as:

$$PSRF = \frac{n}{n-1} + \frac{m+1}{m}\lambda_{max}(\frac{W^{-1}B}{n}),$$

in which W is the average covariance matrix of each state trajectory and B is the covariance matrix between trajectories. $\lambda_{max}(\cdot)$ denotes the largest eigenvalue of a matrix. In the experiments, the matrices B and W are observed to be sparse. Therefore, the efficient power method [13] can be employed to find the largest eigenvalue, as we show in Section 4.

3. COMBINATIONAL ERROR ANALYSIS

This section presents an efficient algorithm to evaluate the conditional error probabilities defined in Equation 1 at the combinational circuit level. For compactness, we refer to both $Pr(F_\emptyset = 1|I,S)$ and $Pr(F_i = 1|I,S)$ as P_e. We use a static

algorithm that combines electrical cell pre-characterization and a new timing-window analysis to capture electrical and timing-window masking. Consider a combinational circuit with N_g gates each of area A_i such that the the total circuit area is A_t. Then, the error probability P_e at the output can be calculated as [1]:

$$P_e = \frac{1}{A_t} \sum_{i=1}^{N_g} \sum_{m,n} A_i R_i(q_m) \Delta q \Delta t P_i(q_m, t_n)$$

where $R(q)$ is the effective charge-dependent strike rate, Δq and Δt are the discretization steps for charge distribution and time, $q_m = m\Delta q$, and $t_n = n\Delta t$. $P_i(q,t)$ is the probability for a strike at gate i with charge q and at a time t to be latched.

The challenge of combinational analysis is an efficient computation of $P_i(q,t)$. We use the cell-based technique of [1] to capture the evolution of a pulse, i.e., its electrical masking. The novel contribution is in customizing the analysis to the Monte Carlo setting in which the combinational input is fixed at the time of analysis: this allows us to improve the accuracy of latching window modeling by incorporating signal controllability. A soft error glitch is only captured at a latch if it is present at the D-input of a flip-flop (for DFF) in the interval $[-T_s, T_h]$. (This is because it needs to reach D-input T_s before the clock edge t_{clk} and also to remain steady for T_h following the clock edge.) Note that an error pulse is latched only if covers the entire interval. A timing window is the time such that a glitch that covers it will propagate to a latch, and a glitch outside it will not be latched. A latching window for a gate g is the set of time intervals $LW(g)$ in which a pulse generated at g can propagate to some set of latches *if the pulse contains at least one of intervals*. Each interval is described by start and end times: $LW_i(g) = [S_i^w, E_i^w]$, such that the overall latching window $LW(g)$ for a gate g is given by

$$LW(g) = \{[S_0^w, E_0^w], [S_1^w, E_1^w], \ldots\}.$$

Since the pulse needs to cover at least one interval to be latched, the latching condition for a pulse of width PW generated at time t can be formally stated as below, where we designate the pulse start and end times as $S_i^p = t$ and $E_i^p = t + PW$:

$$t = S_i^p \leq S_i^w,$$
$$t + PW = E_i^p \geq E_i^w.$$

Starting with the latching window at the D-input to a latch, the LWs of gates are derived by a reverse-topological order traversal of the circuit graph. We distinguish two scenarios in the backtracking procedure: non-reconvergent paths and reconvergent paths. For the scenario of a non-reconvergent path from gate g to gate h, the error pulse generated at (the output of) gate g at time t with pulse width PW is delayed and shifted in time by $d(h)$ once it propagates to gate h. In order to be latched, the pulse at the output of gate h needs to cover at least one latch window $LW_i(h)$:

$$S_i^p(g) + d(h) \leq S_i^w(h),$$
$$E_i^p(g) + d(h) \geq E_i^w(h).$$

Solving the inequality above, we get

$$S_i^p(g) \leq S_i^w(h) - d(h),$$
$$E_i^p(g) \geq E_i^w(h) - d(h).$$

Thus, $LW(g)$ is just a translation of $LW(h)$ by the propagation delay $d(h)$ from its subsequent gate h:

$$\begin{aligned} LW(g) &= LW(h) - d(h) \\ &= \{[S_0^w(h) - d(h), E_0^w(h) - d(h)], \ldots\}. \end{aligned}$$

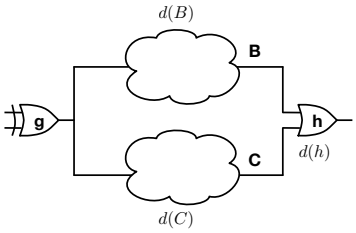

Figure 1: Reconvergent paths g-B-h and g-C-h.

We now consider the case in which reconvergent paths are present. Figure 1 illustrates the case of two reconvergent paths g-B-h and g-C-h. We assume that $d(B) \leq d(C)$. It turns out that the resulting LW depends on the gate type at the reconvergent node, instantaneous signal values, the propagation delays and the error pulse width.

An error pulse propagating along two paths creates two pulses at the inputs of a reconvergent gate: $[t + d(B), t + PW + d(B)]$ and $[t + d(C), t + PW + d(C)]$, see Figure 2.

In contrast to the conventional method taking union of all LWs [14], the main intuition for our model is that a gate will not produce an error when there is a correct *controlling* signal at one of its inputs. We capture this via the notion of the error inhibition window as the set of time intervals over which at least one controlling input is correct: $t_{inh} = \{t : \exists j, in_{h,j}(t)$ is controlling$\}$, where $in_{h,j}(t)$ refers to the signal values of the j-th input of gate h at time t. Next, we introduce the notion of an effective pulse $EP(g,h)$ as the union of input pulses minus the error inhibition window:

$$EP(g,h) = \cup[t + d(i), t + PW + d(i)] - t_{inh}.$$

Thus, an effective pulse represents all time intervals of faulty pulses at the inputs of gates that produce error at the outputs. Excluding the inhibition time is critical for the overall accuracy, as the results from experiments show that t_{inh} and PWs are of the same order of magnitude (the typical value of t_{inh}/PW is about 40%). Generally, the effective pulses $EP(g,h)$ can be represented as a set of intervals:

$$EP(g,h) = \{[S_0^{ep}, E_0^{ep}], \ldots\},$$

in which S_i^p and E_i^p are the *pulse* start and end times.

We now consider two examples to illustrate the use of the model. First, assume B and C have high controlling values, e.g. for an OR gate, in Figure 2. If $d(C) - d(B) \leq PW$, before $t + d(C)$ and after $t + d(B) + PW$, at least one correct signal is controlling, which inhibits an error at the output of h. Thus, the effective error pulse is $[t + d(C), t + d(B) + PW]$, rather than the conservative $[t + d(B), t + d(C) + PW]$ (that would be obtained by the simple union of shifted pulses in [14]). The effect of error suppression is even more pronounced when $d(C) - d(B) \geq PW$. As shown in Figure 2, at all times there is at least one correct input eliminating the error pulse, i.e. $EP(g,h) = \emptyset$.

In deriving the corresponding latching windows, we repeat the earlier procedure but replace the raw (physical) pulse start/end times $t, t + PW$ with the effective pulse start/end

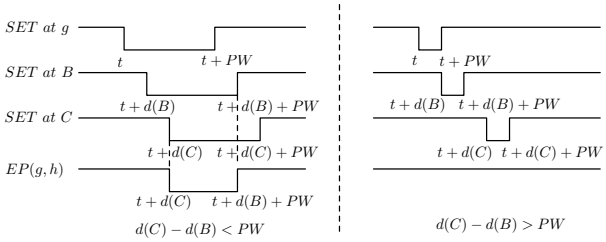

Figure 2: Both B and C are controlling signals.

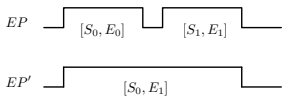

Figure 3: Effective pulse approximation strategy.

times at the input to h: $S_i^{ep}(h), E_i^{ep}(h)$:

$$S_i^{ep}(h) + d(h) \leq S_j(h),$$
$$E_i^{ep}(h) + d(h) \geq E_j(h).$$

The overall LW at gate g is the union of the sub-intervals $LW_{ij}(g)$ found above:

$$LW(g) = \cup LW_{ij}(g).$$

We continue using Figure 1 and 2 as an example. Suppose $LW(h)$ contains only one interval $[S_0(h), E_0(h)]$ and that $d(C) - d(B) \leq PW$ and both B and C are controlling. This generates the effective pulse $[t + d(C), t + d(B) + PW]$. The constraints for t and PW at gate g are solved through the inequalities:

$$t + d(C) + d(h) \leq S_0(h),$$
$$t + PW + d(B) + d(h) \geq E_0(h),$$

which can be re-written as

$$t \leq S_0(h) - d(C) - d(h),$$
$$t + PW \geq E_0(h) - d(B) - d(h).$$

Thus, the latch window of g becomes $[S_0(h) - d(C) - d(h), E_0(h) - d(B) - d(h)]$ rather than $[S_0(h) - d(B) - d(h), E_0(h) - d(C) - d(h)]$.

Reconvergent paths can lead to an exponential number of intervals in the representations of LWs for some gates. This is because a raw SET pulse will generate two effective pulses under the scenario that both B,C are non-controlling and $d(C) - d(B) > PW$, illustrated in Figure 2. In order to achieve scalability, we introduce the approximation of interval filling to make $EP(g, h)$ contain only one interval, i.e. $EP'(g, h) = [S_0^{ep}(g, h), E_1^{ep}(g, h)]$. Technology scaling reduces gate delay while PW of SETs increases. For the 16nm technology node used in the experiments, the gate delay is about 10ps while PW is about 1ns. Thus, the fraction of reconvergent paths that require interval filling is relatively small: less than 0.002% in the experiments. Among the cases that required interval filling, the interval gap ϵ is negligible compared to the total $PW = PW1 + PW2$. We also find that the average relative gap $\epsilon/(PW1 + PW2)$ is less than 3%, indicating that the inaccuracy introduced by this approximation is small.

4. EXPERIMENTAL RESULTS

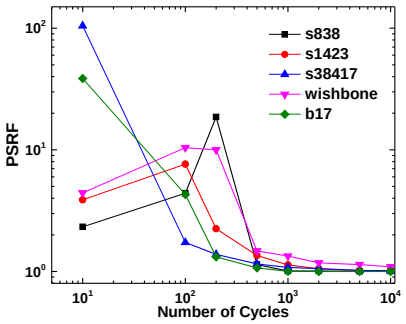

Figure 4: Stationarity diagnostics using PSRF.

Cases	HSPICE		W/ controllability		W/o controllability	
	Time (s)	SER	Time (s)	SER	Time (s)	SER
4b-MUL	2703	2.42	0.003	2.44	0.003	2.86
4b-ALU	2127	2.12	0.003	2.13	0.003	2.28
c432	14060	0.52	0.004	0.52	0.004	0.56
c499	5973	2.16	0.01	2.14	0.01	2.51
wishbone	n/a	n/a	352.0	0.14	349.5	0.33
b17	n/a	n/a	1012.6	0.014	861.2	0.026

Table 1: Runtime and accuracy analysis for circuit-level simulators.

In this section, we report on the extensive experiments to demonstrate the effectiveness of the proposed Monte Carlo framework together with the circuit-level simulation algorithm.

The Monte Carlo framework and the circuit-level static SEU modeling algorithm were implemented in C++ and run on an Intel Xeon 2.93GHz workstation with 74G bytes of memory. The circuits are chosen from the ISCAS and ITC benchmarks [15, 16] as well as the functional units from ARM Amber 25 and synthesized based on the 16nm Predictive Technology Model [17]. For each testbench, the technology-mapped netlist is taken as an input. The output of our tool is a list of sets of impacted primary outputs and their error probabilities.

We first demonstrate the accuracy and efficiency of the proposed circuit-level simulator as a crucial building block of the framework. Table 1 shows the accuracy and runtime comparison of the proposed simulator considering signal controllability in LW modeling against the strategy ignoring signal controllability and HSPICE. The unit for the SER in the table is 10^{-10}. Both circuit-level simulators, with and without considering controllability, achieve a 10^6X speed-up compared to the HSPICE simulation. HSPICE cannot handle the larger circuits (Wishbone and b17), which we designate by n/a. The effective pulse and inhibition time are not negligible compared to the pulse width: the average t_{inh}/PW is 40%. The ability to accurately model LW by considering signal controllability translates into higher accuracy. Considering controllability allows matching HSPICE to within 0.6%, on average, while the error is 12.6% when signal controllability is ignored.

Next we investigate the effectiveness of monitoring stationarity through the PSRF method. We set $m = 5$ and choose 1.1 as the threshold for diagnostics. Figure 4 plots PSRF for several benchmark circuits. Each state trajectory approaches the stationary distribution with the increase of the number of cycles. The ability to sample from a stationary distribution results in a large reduction of the number of samples needed, as shown in Table 2. The runtime needed for the diagnostics, including running multiple parallel logic simulations and evaluating the PSRF metric, is relatively small compared to the runtime savings in needed circuit-level simulation.

Cases	W/ diagnostics		W/o diagnostics	
	# samples	Time (s)	# samples	Time (s)
s838	2200	16.1	4500	32.4
s1423	400	7.7	600	8.0
wishbone	400	352.0	600	430.5
s38417	1700	271.2	2000	322.4
b17	850	1012.6	1000	1200.8

Table 2: Impact of stationary diagnostics on sample number and runtime.

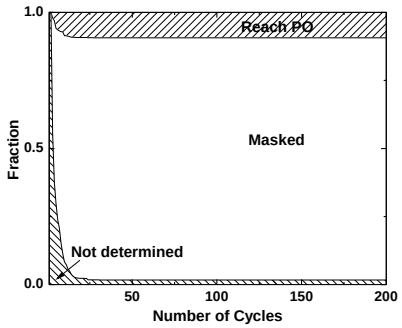

Figure 5: Evolution of errors at logic level.

During the experiments, the empirical convergence analysis on the results of the Monte Carlo runs is done using the non-overlapping batch means method [18]. The number of samples and total runtimes are listed in Table 3. As expected, compared to direct sampling, nested sampling reduces the number of samples by up to 1500X and runtime by up to 25X.

After injecting faults into the FF sets, we observed the evolution of status of different errors (propagated to primary outputs, masked, and still residing on FFs) in Figure 5. Most errors are either masked or reach the primary outputs in a relatively small number of cycles. There is little change in the distribution of errors after about 20 cycles (for benchmark b17). This means that a small number of cycles may be sufficient to accurately compute the overall error.

To understand the effect of the error injection threshold, we run experiments on one of the benchmarks (b17) and plot the runtime and the relative error (with respect to an evaluation with zero threshold) in Figure 6. Increasing the error injection threshold increases the inaccuracy but reduces runtime as fewer errors are captured. This can be used to find the optimum setting for the threshold.

5. CONCLUSIONS

In this paper we describe a Monte Carlo simulation flow for SEU estimation in sequential circuits. We propose a nested sampling strategy that allows faster statistical convergence and introduce a check on the stationarity of machine states for further convergence efficiency. Experiments show that nested

Cases	Nested Sampling		Direct Sampling	
	# samples	Time (s)	# samples	Time (s)
s298	50	0.3	2.79×10^4	3.3
s344	400	1.9	3×10^5	31.4
s641	190	1.7	1.44×10^5	25.2
s1238	340	3.7	2.466×10^5	72.9
s1423	400	6.0	9.74×10^5	153.4
b17	850	1012.6	1.279×10^6	7140.2

Table 3: Nested vs. direct sampling: sample numbers for convergence and runtime.

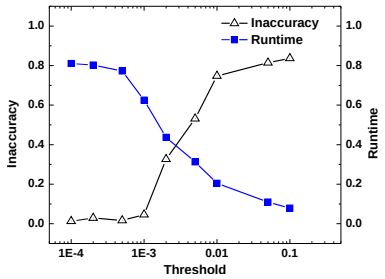

Figure 6: Impact of error-injection threshold on runtime and accuracy.

sampling reduces the number of samples by up to 1500X and runtime by up to 25X compared to direct sampling. Stationarity checking allows reducing sampling number by 25%, on average.

6. ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation under grant CCF-1255757 and by the Semiconductor Research Corporation under grant 2416.001.

7. REFERENCES

[1] B. Zhang, W.-S. Wang, and M. Orshansky, "Faser: Fast analysis of soft error susceptibility for cell-based designs," in *ISQED*, 2006.

[2] N. Miskov-Zivanov and D. Marculescu, "Mars-c: modeling and reduction of soft errors in combinational circuits," in *Proceedings of the 43rd annual Design Automation Conference*, pp. 767–772, ACM, 2006.

[3] S. Krishnaswamy, G. F. Viamontes, I. L. Markov, and J. P. Hayes, "Accurate reliability evaluation and enhancement via probabilistic transfer matrices," in *Design, Automation and Test in Europe, 2005. Proceedings*, pp. 282–287, IEEE, 2005.

[4] G. Asadi and M. B. Tahoori, "An accurate ser estimation method based on propagation probability," in *DATE*, 2005.

[5] C.-C. Yu *et al.*, "Scalable and accurate estimation of probabilistic behavior in sequential circuits," in *VTS*, 2010.

[6] N. Miskov-Zivanov and D. Marculescu, "Soft error rate analysis for sequential circuits," in *DATE*, 2007.

[7] R. Burch, F. N. Najm, P. Yang, and T. N. Trick, "A monte carlo approach for power estimation," *TVLSI*, vol. 1, no. 1, pp. 63–71, 1993.

[8] C.-Y. Tsui, M. Pedram, and A. M. Despain, "Exact and approximate methods for calculating signal and transition probabilities in fsms," in *Design Automation, 1994. 31st Conference on*, pp. 18–23, IEEE, 1994.

[9] D. Holcomb, W. Li, and S. A. Seshia, "Design as you see fit: System-level soft error analysis of sequential circuits," in *DATE*, 2009.

[10] S. P. Brooks and A. Gelman, "General methods for monitoring convergence of iterative simulations," *Journal of computational and graphical statistics*, vol. 7, no. 4, pp. 434–455, 1998.

[11] P. Brémaud, *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*, vol. 31. Springer Science & Business Media, 2013.

[12] C. J. Geyer, "Conditioning in markov chain monte carlo," *Journal of Computational and Graphical Statistics*, vol. 4, no. 2, pp. 148–154, 1995.

[13] G. H. Golub and C. F. Van Loan, *Matrix computations*, vol. 3. JHU Press, 2012.

[14] S. Krishnaswamy, I. L. Markov, and J. P. Hayes, "On the role of timing masking in reliable logic circuit design," in *DAC*, 2008.

[15] F. Brglez, D. Bryan, and K. Koźmiński, "Combinational profiles of sequential benchmark circuits," in *ISCAS*, 1989.

[16] F. Corno, M. S. Reorda, and G. Squillero, "Rt-level itc'99 benchmarks and first atpg results," *DTC*, 2000.

[17] S. Sinha, G. Yeric, V. Chandra, B. Cline, and Y. Cao, "Exploring sub-20nm finfet design with predictive technology models," in *DAC*, 2012.

[18] S. P. Brooks, "Quantitative convergence assessment for markov chain monte carlo via cusums," *Statistics and Computing*, vol. 8, no. 3, pp. 267–274, 1998.