IPSJ Transactions on System LSI Design Methodology Vol.5 63–70 (Feb. 2012)

[DOI: 10.2197/ipsjtsldm.5.63]

Regular Paper

An Exact Estimation Algorithm of Error Propagation Probability for Sequential Circuits

Masayoshi Yoshimura[1,a)] Yusuke Akamine[2,†1] Yusuke Matsunaga[1]

Received: May 27, 2011, Revised: September 5, 2011,
Accepted: October 19, 2011, Released: February 21, 2012

Abstract: In advanced integrated circuit technology, the soft error tolerance is low. Soft errors ultimately lead to failure in VLSIs. We propose a method for the exact estimation of error propagation probabilities in sequential circuits whose FFs latch failure values. The failure due to soft errors in sequential circuits is defined using the modified product machine. The modified product machine monitors whether failure values appear at any primary output. The behavior of the modified product machine is analyzed with the Markov model. The probabilities that the failure values latched into the flip-flops (FFs) appear at any primary output are calculated from the state transition probabilities of the modified product machine. The time required for solving simultaneous linear equations accounts for a large portion of the execution time. We also propose two acceleration techniques to enable the application of our estimation method to larger scale circuits. These acceleration techniques reduce the number of variables in simultaneous linear equations. We apply the proposed method to ISCAS'89 and MCNC benchmark circuits and estimate error propagation probabilities for sequential circuits. Experimental results show that total execution times for the proposed method with two acceleration techniques are up to 10 times lesser than the total execution times for a naive implementation.

Keywords: soft error, modified product machine, Markov model, absorption state, sequential circuit

1. Introduction

Soft errors are transient errors caused by neutron strikes from cosmic rays in very large-scale integrated circuits (VLSIs). Soft errors might flip the values of memory elements and cause electrical pulses at the outputs of logic gates. Ultimately, soft errors might result failure. In this paper, we refer to the appearance of failure values at the primary outputs of VLSIs as failure. The probability that soft errors result failure is one of the measures of the soft error tolerance of VLSIs.

Soft error occurrences lead to VLSIs failure over several steps. VLSIs consist of logic gates and memory elements; soft errors occur at both these component. The steps from soft error occurrences to the failure of VLSIs are classified into two types. In the case that the soft errors occur at logic gates, failure occurs in three steps —Step-C1 to Step-C3— explained as follows. Neutron strikes might cause electrical pulses at the outputs of logic gates (Step-C1). The electrical pulses might propagate to the inputs of the flip-flops (FFs) and be latched into the FFs (Step-C2). Hence, the FFs that latch these electrical pulses hold failure values. The probability that an FF latches an electrical pulse is calculated from the width of the propagated pulse at the input of the FF and the latch-window size of the FF. Finally, the failure values might propagate to primary outputs (Step-C3). Even if FFs hold the failure values, failure values may not always propagate to the primary outputs. This is because a state transition can change the failure values of the FFs to correct values. In the other case that the soft errors occur at the memory elements (e.g., FFs), failure occurs in two steps —Step-S1 to Step-S2— explained as follows. Neutron strikes might flip the values of the FFs (Step-S1). As a result, the FFs whose values are flipped hold failure values. These failure values might propagate to primary outputs (Step-S2). Step-C3 and Step-S2 are essentially the same steps.

In this study, we focus on Step-C3 and Step-S2. We propose an exact method to estimate error propagation probabilities for sequential circuits whose FFs latch failure values in Step-C3 and Step-S2. In the proposed method, the probabilities that the failure values that are latched into FFs appear at primary outputs are calculated from the state transition probabilities obtained using a modified product machine of the given sequential circuit. The modified product machine is a computational model that consists of a correct circuit, a faulty circuit, and an error detector. The behaviors of the correct and faulty circuits correspond to the behaviors of the original circuit without and with the effects of a soft error, respectively. The error detector compares the output values of the correct circuit with those of the faulty circuit. The output value of ″1″ in the error detector shows the differences between the output values of the correct circuit and those of the faulty circuit. In other words, output value 1 implies that failure values appear at the primary outputs. For each incorrect state that is changed by the effects of soft errors, the probability of the output value 1 in the error detector is calculated; this probability is referred to as the absorption probability. If the input vectors of an

[1] Graduate School of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819–0005, Japan
[2] Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819–0005, Japan
[†1] Presently with Panasonic Corporation
[a)] yosimura@ait.kyushu-u.ac.jp

original circuit are probabilistic, the state transitions of the modified product machine of the original circuit are also probabilistic. Further, if the input vectors and state transitions of the modified product machine are probabilistic, the probabilities of the next states depend only on the probabilities of the current states. Thus, if the input vectors of an original circuit are probabilistic, probability process of states of the modified product machine can be regarded as the Markov process. The behavior of the modified product machine is analyzed using the Markov model to calculate absorption probabilities. Absorption probabilities are obtained by solving simultaneous linear equations where the variables are absorption probabilities. We also propose two acceleration techniques to enable the application of our exact estimation method to larger scale circuits. The time required for solving simultaneous linear equations accounts for a large portion of the execution time. The common aim of the two acceleration techniques is to reduce the number of variables in the simultaneous linear equations. In the first technique, the simultaneous linear equations are divided and each divided simultaneous linear equation is solved. There are certain variables whose values can be obtained without solving the simultaneous linear equations. In the second technique, these variables are identified and removed from the simultaneous linear equations. From the experimental results, the execution time of the proposed method for large scale circuits was up to 10 times faster than that of a naive method for large scale circuits.

The contributions of this study are as follows.

(1) The definition of the failure of sequential circuits with the modified product machine
(2) An exact procedure to estimate error propagation probabilities with the Markov model
(3) Two techniques to accelerate one of the processes in the procedure

In this paper, we propose an exact procedure to estimate error propagation probabilities. The exact proposed method can not be applied to very large scale circuits for long execution time. However, the proposed method obtains exact error propagation probabilities of several circuits. The exact error propagation probabilities provide a baseline for estimating approximate methods to calculate error propagation probabilities.

This paper is organized as follows. In Section 2, we present an overview of related work. In Section 3, a procedure to estimate error propagation probabilities is proposed. Section 4 introduces two acceleration techniques that can be used to accelerate the proposed method. The experimental results for ISCAS'89 and MCNC benchmark circuits are shown in Section 5. Finally, Section 6 concludes this paper.

2. Related Work

Here, we discuss the previous soft error tolerance estimation methods for sequential circuits. Several methods for estimating soft error tolerance in Step-C1 have been proposed [1], [5], [8]. The generation probability for each pulse width has been calculated from the results of device simulations [5], [8] and has been evaluated on the basis of the measurement results using TEG [1]. Further, a method for estimating soft error tolerance in Step-C2 [6] and a method for estimating soft error tolerance in Step-S1 [7] has also been proposed.

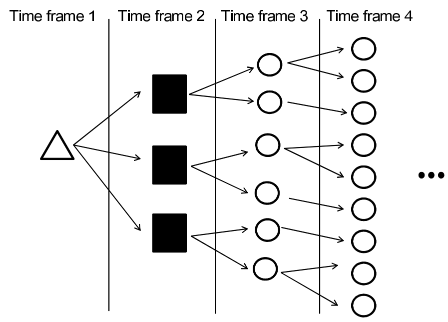

Fig. 1 Example of state transitions.

Estimation methods to calculate error propagation probabilities for sequential circuits whose FFs latch failure values in Step-C3 and Step-S2 have been proposed in several papers [3], [4]. For instance, Hayes et al. proposed an estimation method to calculate the error propagation probabilities only k clock cycles after the soft errors occur [3]. The method is a heuristic method and not an exact method. Even if k is large, error propagation probabilities obtained by this method are not always accurate. This is because there may be cases where the FFs hold flipped values even though failure values do not appear at the primary outputs after k clock cycles. When k is large, the execution time of this method is long. For example, the execution time of this method with $k = 2$ for s382 is longer than the execution time of the proposed method, which is an exact method.

Miskov-Zivanov et al. proposed an estimation method that involves a probabilistic analysis of the behaviors of sequential circuits after the soft errors occur [4]. The probabilities that failure values appear at primary outputs are calculated from the steady-state probabilities of the product machine of sequential circuits after the soft errors occur. The steady-state probabilities are calculated using the Markov model. This Markov model of the product machine of sequential circuits has no absorption state. Thus, the number of states of the product machine is large, the computation time to calculate the steady-state probability is long, and the size of the sequential circuits to be estimated is very small.

Next, we discuss the difference between the probabilities obtained by Miskov-Zivanov et al.'s method [4] and these obtained by our proposed method. **Figure 1** shows an example of the state transitions of the product machine of a sequential circuit after a soft error occurs. In Fig. 1, the triangle, squares, and circles represent the initial state of the product machine, states in which the failure values appear at primary outputs, and states in which the FFs hold no failure values, respectively. In this example, failure values always appear at primary outputs at least once. The time frame in which the failure values appear is only time frame 2. Failure values do not appear at primary outputs on any time frame other than time frame 2. This is because FFs hold no failure values in any the states in time frame 3. Thus, the probability obtained by the previous method [4] is nearly equal to 0. On the other hand, the probability obtained by our proposed method is 1. The method [4] is suitable for use in fields such as picture processing and communication, where it is important to determine the probability that failure values appear at the primary outputs per time. However, the previous method [4] is not suitable for use in other fields involving processors and controllers. In other fields, it is important to determine the probability that failure values appear at the primary outputs.

3. Procedure to Estimate Error Propagation Probabilities

3.1 Modified Product Machine

The modified product machine of an original circuit monitors whether the failure values that are latched into the FFs appear at any primary output of the original circuit. The modified product machine is a computational model. In this study, the output value 1 of the modified product machine indicates that failure values appear at the primary outputs. The modified product machine is illustrated in **Fig. 2**.

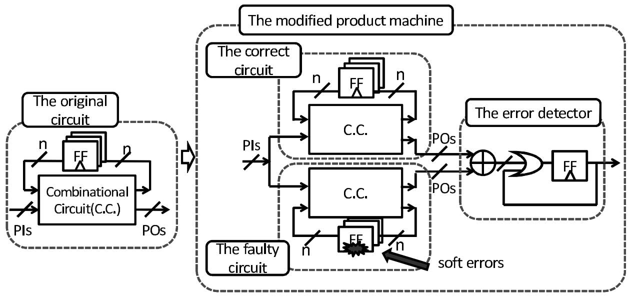

Fig. 2 Modified product machine.

The modified product machine $\mathcal{M}_p$ is a sequential circuit defined by the 6-tuple $(S_p, S0_p, I_p, O_p, \delta_p, \lambda_p)$ where S_p is the set of states; $S0_p$, the set of initial states; I_p, the set of input vectors; O_p, the set of output vectors; δ_p, the state transition function $S_p \times I_p \rightarrow S_p$; and λ_p, the output function $S_p \times I_p \rightarrow O_p$. The modified product machine $\mathcal{M}_p = (\mathcal{M}_c, \mathcal{M}_f, \mathcal{M}_e)$ consists of three sequential circuits: the correct circuit $\mathcal{M}_c = (S_c, S0_c, I_c, O_c, \delta_c, \lambda_c)$, the faulty circuit $\mathcal{M}_f = (S_f, S0_f, I_f, O_f, \delta_f, \lambda_f)$, and the error detector $\mathcal{M}_e = (S_e, S0_e, I_e, O_e, \delta_e, \lambda_e)$. The original circuit $\mathcal{M}_o = (S_o, S0_o, I_o, O_o, \delta_o, \lambda_o)$ is a given sequential circuit to be estimated.

The behaviors of the correct and faulty circuits indicate the behaviors of the original circuit without and with the effects of the soft error, respectively. Tuples of the correct circuit and faulty circuit are shown using tuples of the original circuit as follows.

$$\begin{array}{lll} S_o = S_c = S_f, & S0_o = S0_c \neq S0_f, & I_o = I_c = I_f, \\ O_o = O_c, & \delta_o = \delta_c = \delta_f, & \lambda_o = \lambda_c = \lambda_f. \end{array}$$

An initial state of the faulty circuit $S0_f$ is changed by the effects of a soft error. These effects of the soft error are only in the faulty circuit.

The error detector compares the output vectors of the correct circuit O_c with those of the faulty circuit O_f. The output value of the error detector $O_e = \{0, 1\}$ shows whether there is a difference between O_c and O_f at least once. In this paper, $O_e = 1$ implies that there is a difference between O_c and O_f at least once. In other words, $O_e = 1$ implies that failure values appear at primary outputs at least once. As there is a difference between O_c and O_f, the output value O_e and the state S_e of the error detector are always 1. Tuples of the error detector are defined using tuples of the correct circuit and faulty circuit as follows.

$$\begin{array}{ll} S_e = \{0, 1\}, & S0_e = 0, \\ I_e = (O_c, O_f), & O_e = \{0, 1\}, \\ \delta_e = \begin{cases} 0 & (O_c = O_f \wedge S_e = 0) \\ 1 & \text{otherwise} \end{cases}, & \lambda_e = \delta_e. \end{array}$$

Now, we discuss the state transitions from current states to next states; which occur with the soft error. The next states are the initial states of the faulty circuit in the modified product machine. It should be noted that soft errors occur at random. An occurrence of a soft error does not depend on the current state of the original circuit or the input values of the original circuit. The probabilities of the initial states of the correct circuit in the modified product machine are equal to the probabilities of the next states when soft errors do not occur. However, the next states depend on the current states of the original circuit and the input values of the original circuit. The probabilities of the next states can be calculated using a timing fault simulation.

3.2 Markov Model with Absorption States

The behavior of the modified product machine is analyzed using the Markov model to calculate the absorption probabilities. In this study, we assume that the probabilities for each input vector of the original circuit are given. If the input vectors of an original circuit are probabilistic, the state transitions of the modified product machine of the original circuit are also probabilistic. Further, if the input vectors and state transitions of the modified product machine are probabilistic, the probabilities of the next states depend only on the probabilities of the current states. Thus, if the input vectors of an original circuit are probabilistic, probability process of states of the modified product machine can be regarded as the Markov process.

The set of all the states of the modified product machine Π is separated into three sets: the failure set Π_f, the masked set Π_m, and the transition set Π_t. The state of the modified product machine S_p consists of the state of the correct circuit S_c, the state of the faulty circuit S_f, and the state of the error detector S_e. S_p is called an MPM (modified product machine) state; $S_p = (S_c, S_f, S_e)$. Once S_e is transferred from 0 to 1, it cannot be transferred from 1 to 0. $\Pi_f = \{S_p | S_e = 1\}$ is called the failure set Π_f. Once S_f is transferred to the same state of S_c, then S_f is always equal to S_c. $\Pi_m = \{S_p | S_e = 0 \wedge S_c = S_f\}$ is called the masked set Π_m. The transition set Π_t is the other state set. $\Pi_t = \{S_p | S_e = 0 \wedge S_c \neq S_f\}$ is called the transition set Π_t. The set of the MPM states satisfies the following two conditions.

- $\Pi = \Pi_t \cup \Pi_f \cup \Pi_m$.
- $\Pi_t \cap \Pi_m = \Pi_m \cap \Pi_f = \Pi_f \cap \Pi_t = \phi$.

Each of the MPM states in the failure set could be considered as **the failure state**. The failure state can be transferred to itself by any input vector. Each of the MPM states in the masked set could be considered as **the masked state**. The masked state can also be transferred to itself by any input vector. A state which is always transferred to itself is called the absorption state. Hence, both the failure state and masked state are the absorption state. **Figure 3** shows an example of the Markov model with absorption states.

If an MPM state $\pi \in \Pi$ is the state immediately after a soft

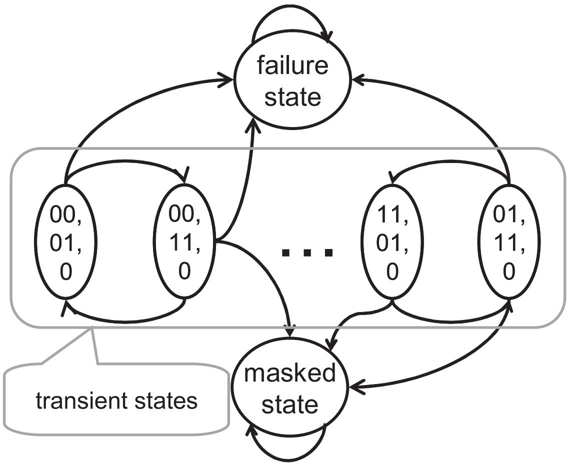

Fig. 3 Markov Model with absorption states.

error occurrence, then the state π is called the initial state. The probability that an initial state is absorbed to the failure state is called the absorption probability.

3.3 Calculation Method for Absorption Probability

In this section, we show a method for calculating the absorption probability. The probability that the failure values that are latched into FFs appear at the primary outputs is called the probability P_{fail}. Let Π be the set of all the MPM states; $\Pi_t \subseteq \Pi$, the set of all the transient states; π_f, the failure state; and Π_{init}, the set of all the initial states. $P_{tr}(\pi_i, \pi_j)$ is the probability of the transfer of an MPM state from π_i to π_j in one clock cycle. The probability of the model in the MPM state π_f to be absorbed to the failure state is $P_{abs}(\pi_i)$. P_{fail} is shown in Eq. (1).

$$P_{fail} = \sum_{\pi \in \Pi_{init}} P_{init}(\pi) \cdot P_{abs}(\pi) \tag{1}$$

The probability $P_{abs}(\pi_i)$ is the probability that π_i finally transfers to the failure state π_f. Paths of state transition from an MPM state to the failure state are classified into two groups: (a) a direct path of state transition from the MPM state to the failure state and (b) state transition from the MPM state to the failure state via transient states. The absorption probability $P_{abs}(\pi_i)$ is shown in Eq. (2).

$$P_{abs}(\pi_i) = P_{tr}(\pi_i, \pi_f) + \sum_{\pi_j \in \Pi_t} P_{tr}(\pi_i, \pi_j) \cdot P_{abs}(\pi_j) \tag{2}$$

The probability $P_{init}(\pi_i)$ is the probability that the MPM state is π_i when a soft error occurs. S_o denotes the set of all the states of the original circuit. e_{s,π_i} denotes the probability that the MPM state π_i is an initial state under the condition that the state of the original circuit is $s \in S_o$ when a soft error occurs. The initial state probability of $\pi_i \in \Pi_{init}$ is shown in Eq. (3).

$$P_{init}(\pi_i) = \sum_{s \in S_o} P_{steady}(s) \cdot e_{s,\pi_i} \tag{3}$$

$P_{steady}(s)$ denotes the steady-state probability of a state $s \in S_o$, i.e., the probability that the state of the original circuit is s. $P_{tr}(s_i, s_j)$ is the probability that a state $s_i \in S_o$ transfers to a state $s_j \in S_o$ at one clock cycle. The steady-state probability of s_j is shown in Eq. (4).

$$P_{steady}(s_j) = \sum_{s_i \in S_o} P_{steady}(s_i) \cdot P_{tr}(s_i, s_j) \tag{4}$$

The sum of the steady-state probabilities for all states in the original circuit is 1.

$$\sum_{s \in S_o} P_{steady}(s) = 1 \tag{5}$$

If the state transition probability for each ordered pair of states of the original circuit is known, simultaneous linear equations whose variables can be steady-state probabilities are formulated using Eqs. (4), (5). In general, $e_{s,\pi}$ is obtained based on probabilities such as the probability that a soft error occurs at an FF.

If the state transition probability for each ordered pair of MPM states is known, the absorption probabilities $P_{abs}(\pi_i)$ are obtained by solving the simultaneous linear equations whose variables are $P_{abs}(\pi_i)$.

3.4 Procedure of the Proposed Calculation Method

A procedure to calculate the probability P_{fail} discussed in Section 3.3 is as follows:

(1) Enumeration of reachable states of the original circuits
(2) Calculation of state transition probabilities $P_{tr}(s_i, s_j)$
(3) Calculation of initial state probabilities $P_{init}(\pi_i)$
(4) Enumeration of reachable MPM states
(5) Calculation of state transition probabilities $P_{tr}(\pi_i, \pi_j)$
(6) Calculation of absorption probabilities $P_{abs}(\pi_i)$
(7) Calculation of the probability P_{fail}

First, the state transition probabilities $P_{tr}(s_i, s_j)$ between each ordered pair of all reachable states in the original circuit are obtained. A logic simulation or an implicit enumeration using binary decision diagrams [2] are used to enumerate all reachable states and calculate state transition probabilities for each ordered pair of all reachable states. Steady-state probabilities $P_{steady}(s)$ are obtained using state transition probabilities $P_{tr}(s_i, s_j)$. However, state transition probabilities from unreachable states to reachable states are unnecessary to calculate steady-state probabilities. Steady-state probabilities are obtained by solving simultaneous linear equations. Since $e_{s,\pi}$ is given in this paper, the probability of an initial state $P_{init}(\pi_i)$ is obtained using Eq. (3). State transition probabilities of the modified product machine $P_{tr}(\pi_i, \pi_j)$ are obtained as well as state transition probabilities of the original circuits. Absorption probabilities $P_{abs}(\pi_i)$ are obtained by solving simultaneous linear equations where variables are $P_{abs}(\pi_i)$. Finally, P_{fail} is calculated using Eq. (1).

If the above procedure is naively implemented, the total execution time might be very long. The execution time for the calculation of state transition probabilities of the modified product machine $P_{tr}(\pi_i, \pi_j)$ and the calculation of absorption probabilities $P_{abs}(\pi_i)$ accounts for a very large portion of the total execution time. If the number of FFs in the original circuit is k, the number of states in the modified product machine is 2^{2k} in the worst case. Let $|I_p|$ be the number of input vectors of the modified product machine. If the state transition probabilities $P_{tr}(\pi_i, \pi_j)$ are calculated using logic simulation, then logic simulation is executed $2^{2k} \times |I_p|$ times. If the simultaneous linear equations are solved using a Gaussian elimination method, the execution time is proportional to the number of variables to the power of three. Because the number of variables is the number of states of the modified product machine, the execution time to obtain absorption probabilities $P_{abs}(\pi_i)$ is proportional to $(2^{2k})^3 = 2^{6k}$ in the worst case.

4. Two Acceleration Techniques

The aim of the two proposed acceleration techniques is a reduction in the execution time when the proposed procedure is applied to larger scale circuits. The calculation of the absorption probabilities accounts for a large portion of the execution time. The two acceleration techniques reduce the number of variables in the simultaneous linear equations. The first technique involves dividing the simultaneous linear equations and solving the divided equations several times. Hence, the first technique reduces the number of variables in each divided equation. Next, there are certain variables whose values can be obtained without solving the simultaneous linear equations. In the second technique, these variables are identified and removed from the simultaneous linear equations. Hence, the second technique reduces the number of variables in the overall simultaneous linear equations.

4.1 Dividing the Simultaneous Linear Equations

When the absorption probability $P_{abs}(\pi_i)$ of an MPM state π_i is obtained, it is unnecessary to solve the simultaneous linear equations whose variables correspond to $P_{abs}(\pi_j)$ for all $\pi_j \in \Pi$. Let $R(\pi_i)$ be the set of all MPM states that are reachable from the MPM state π_i. $P_{abs}(\pi_i)$ can be obtained by solving the simultaneous linear equations whose variables correspond to $P_{abs}(\pi_j)$ for only all $\pi_j \in R(\pi_i)$. The number of states in $R(\pi_i)$ for each initial state $\pi_i \in \Pi_{init}$ is equal to or less than the number of all the MPM states.

In general, the total execution time to solve all divided simultaneous linear equations decreases. Let $N_{v_{org}}$ be the number of variables of an original simultaneous linear equation; N_d, the number by which the original simultaneous linear equations are divided; and $N_{v_{\max}}$, the maximum number of variables for all the divided simultaneous linear equations. The computational complex to solve the original simultaneous linear equations is $O({N_{v_{org}}}^3)$, and the computational complex to solve all the divided simultaneous linear equations is $O(N_d \times {N_{v_{\max}}}^3)$. If $N_d \times {N_{v_{\max}}}^3$ is smaller than ${N_{v_{org}}}^3$, the total execution time to solve all the divided simultaneous linear equations decreases.

Figure 4 shows an example of the divided MPM states. In Fig. 4, π_1 and π_2 are the initial states, $\pi_i (3 \leq i \leq 8)$ are the transient states, and π_f is the failure state. When absorption probabilities of both π_1 and π_2 are obtained by solving the simultaneous linear equations without dividing the simultaneous linear equations, the number of variables is 8. To obtain the absorption probability of π_1, it is necessary to solve the simultaneous linear equations whose variables are $P_{abs}(\pi)$ for all $\pi \in R(\pi_1)$. The number of elements in $R(\pi_1) = \{\pi_1, \pi_3, \pi_4, \pi_6, \pi_7, \pi_8\}$ is 6. The number of elements in $R(\pi_2) = \{\pi_2, \pi_4, \pi_5, \pi_7, \pi_8\}$ is 5. Since 2×6^3 is smaller than 8^3, dividing the simultaneous linear equations reduces the total execution time of solving the simultaneous linear equations.

In this acceleration technique, there might be some MPM states

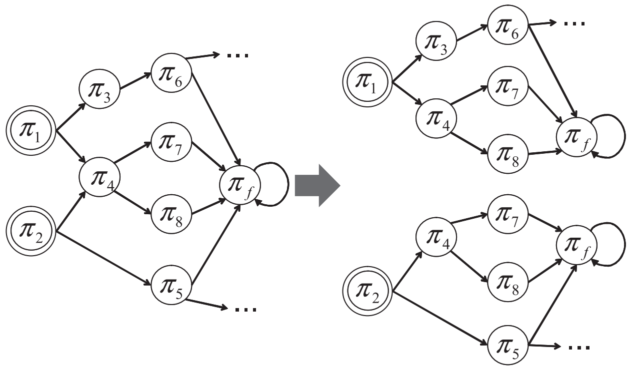

Fig. 4 Example of dividing MPM states.

that are reachable from different initial states. For such states, it is unnecessary to calculate the absorption probability multiple times.

4.2 Removing Pre-failure States

The absorption probability $P_{abs}(\pi_{pf})$ of a specific MPM state π_{pf} is obtained without solving the simultaneous linear equations. The specific MPM state π_{pf} is a transient state whose next state is the failure state by any input vector and is hence called the pre-failure state. The absorption probability of a pre-failure state π_{pf} is shown in Eq. (6).

$$\begin{aligned} P_{abs}(\pi_{pf}) &= P_{tr}(\pi_{pf}, \pi_f) + \sum_{\pi_j \in \Pi_t} P_{tr}(\pi_{pf}, \pi_j) \cdot P_{abs}(\pi_j) \\ &= P_{tr}(\pi_{pf}, \pi_f) \\ &= 1 \end{aligned} \tag{6}$$

According to the definition of pre-failure state, all the state transition probabilities $P_{tr}(\pi_{pf}, \pi_j)$ are 0 and both the state transition probability $P_{tr}(\pi_{pf}, \pi_f)$ and absorption probability $P_{abs}(\pi_{pf})$ are 1. The variables of pre-failure states can be removed from the simultaneous linear equations to obtain absorption probabilities. Therefore, the number of variables of the simultaneous linear equations and the execution time to solve these equations are reduced by removing the variables of the pre-failure states. Let $\Pi_{pf} \subseteq \Pi_t$ be the set of pre-failure states. The absorption probability $P_{abs}(\pi_i)$ of a state $\pi_i \in \Pi$ is shown in Eq. (7).

$$\begin{aligned} P_{abs}(\pi_i) &= P_{tr}(\pi_i, \pi_f) + \sum_{\pi_k \in \Pi_{pf}} P_{tr}(\pi_i, \pi_k) \\ &\quad + \sum_{\pi_j \in \{\Pi_t - \Pi_{pf}\}} P_{tr}(\pi_i, \pi_j) P_{abs}(\pi_j) \end{aligned} \tag{7}$$

5. Experimental Results

In this section, we discuss the experimental assessment of the proposed procedure and two acceleration techniques. The procedure is implemented on C++. The machine for the experiments is equipped with Intel Xeon 3.3 GHz and 32 GB memory. The benchmark circuits are chosen from the ISCAS'89 and MCNC suite. Logic simulation is employed to obtain state transition probabilities. In these experiments, four assumptions are made: soft errors occur only at FFs, a soft error occurs only once, the occurrence probability of each input vector is uniform, and the probability that the value of an FF is flipped is uniform. The susceptibility of an FF is a measure of the soft error tolerance for sequential circuits and is defined as the probability that a failure value that is flipped on the FF appears at the primary outputs. The susceptibilities of FFs are calculated from the absorption probabilities. In these experiments, we use the susceptibilities of FFs as the soft error tolerance of the sequential circuits. The susceptibilities of FFs are calculated using four implementation methods.

- **naive:** a naive implementation
- **partition:** an implementation with divided states
- **pre-failure:** an implementation with reduced pre-failure states
- **partition+pre-failure:** an implementation with divided states and reduced pre-failure states

Table 1 lists the susceptibilities of FFs. In Table 1, "bench," "sum," "max," and "min" denote the circuit name, the sum of the susceptibilities for all the FFs, the maximum susceptibilities for all the FFs, and the minimum susceptibilities for all the FFs, respectively. Please note that the "min" value for s382, i.e., "<1.00E-16" indicates that susceptibility was less than 1.00×10^{-16}. If the probabilities that the values of FFs are flipped are uniform, then "sum" shows a soft error tolerance for all FFs in the sequential circuits. Both maximum and minimum susceptibilities of s27 are small and those of s208.1 are 1. The difference between the maximum susceptibilities of s382 and the minimum susceptibilities of s382 is very large.

Table 1 Susceptibilities of FFs.

bench	sum	max	min
s27	0.59992	0.22829	0.14544
s820	4.21529	0.99954	0.75932
s386	4.19913	0.87852	0.61663
s510	5.87054	1.00000	0.90094
s1488	5.70505	0.99984	0.79583
s208.1	8.00000	1.00000	1.00000
mm4a	1.24260	0.25000	0.00367
s298	9.74507	1.00000	0.10728
s344	11.23206	1.00000	0.15137
s1196	3.36120	1.00000	0.00014
s382	7.50005	1.00000	<1.00E-16

Table 2 shows the accuracy of the estimation method using $k-$time expansion model. In the estimation method using $k-$time expansion model [3], the failure of the sequential circuits is the case that the failure values appear at any primary output or any input of FF on $k-$time frame. This table lists the susceptibility of each FF. In this table, "method," "exact," "1−time," and "2−time" denote the evaluation methods, the exact proposed method for sequential circuits, the estimation method using 1−time expansion model and the estimation method using 2−time expansion model, respectively. The difference between the probability of the exact method and the probability of the estimation method using $k-$time expansion model gives the probability that failure values propagate to any FF on $k-$time frame and that the failure values do not appear at any primary output. The difference between the probability of the exact method and the probability of the estimation method using 2−time expansion model is small for all the FFs of s820, s386, s510, s1488, s208.1, and mm4a and large for FF3 of s27 and some of the FFs of s298. The susceptibilities of all the FFs in all the benchmark circuits are overestimated by the method using $k-$time expansion model.

Tables 3 and **4** show the execution time and number of variables, respectively, for various circuits. In Table 3, "#input" and "#FF" denote the number of inputs and number of FFs in the benchmark circuits, respectively. In Table 3, "total" and "abs_prob" denote the total execution time of the entire proposed procedure and the execution time for only the calculation of the absorption probabilities in the proposed procedure, respectively. "< 0.01" and "—" indicate that execution time was less than 0.01 seconds and execution time was more than 3 days, respectively. In Table 4, "#variables," "max," and "sum" denote the number of variables, the maximum number of variables for all the divided state sets, and the total number of variables for all the divided state sets, respectively.

Table 2 Comparison of proposed method with estimation method using $k-$time expansion model.

circuit	method	susceptibility													
		FF1	FF2	FF3	FF4	FF5	FF6	FF7	FF8	FF9	FF10	FF11	FF12	FF13	FF14
	exact	0.2283	0.2262	0.1454											
s27	1−time	0.2283	0.2262	0.3086											
	2−time	0.2283	0.2262	0.1765											
	exact	0.9264	0.9995	0.7594	0.7593	0.7707									
s820	1−time	0.9448	0.9996	0.7594	0.7594	0.7994									
	2−time	0.9310	0.9996	0.7594	0.7593	0.7742									
	exact	0.6319	0.6166	0.7970	0.8785	0.6423	0.6327								
s386	1−time	0.6789	0.6634	0.8031	0.8846	0.6938	0.6540								
	2−time	0.6352	0.6167	0.7972	0.8786	0.6469	0.6328								
	exact	0.9900	0.8976	0.9397	0.9921	0.9809	1.0000								
s510	1−time	1.0000	0.9575	1.0000	1.0000	0.9858	1.0000								
	2−time	0.9988	0.9198	1.0000	0.9965	0.9823	1.0000								
	exact	0.9616	0.9894	0.9972	0.9998	0.9612	0.7958								
s1488	1−time	0.9616	0.9894	0.9973	0.9999	0.9617	0.7958								
	2−time	0.9616	0.9894	0.9972	0.9999	0.9617	0.7958								
	exact	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000						
s208.1	1−time	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000						
	2−time	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000						
	exact	0.2500	0.2500	0.2500	0.2500	0.0115	0.0218	0.0364	0.0655	0.0037	0.0150	0.0297	0.0589		
mm4a	1−time	0.2500	0.2500	0.2500	0.2500	0.0150	0.0223	0.0369	0.0660	0.0150	0.0223	0.0369	0.0660		
	2−time	0.2500	0.2500	0.2500	0.2500	0.0117	0.0219	0.0365	0.0656	0.0040	0.0151	0.0298	0.0591		
	exact	0.1073	0.1957	0.3331	0.9179	0.7951	0.9991	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.1136	0.2832
s298	1−time	0.5007	0.5620	0.5625	0.9381	0.8814	0.9991	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.5007	0.5000
	2−time	0.2901	0.3547	0.3905	0.9228	0.8265	0.9991	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.2516	0.3812

Table 3 Execution time.

bench	#input	#FF	time[sec]									
			naive		partition		pre-failure		partition+pre-failure		$k-$time expansion	
			total	abs_prob	total	abs_prob	total	abs_prob	total	abs_prob	$1-$time	$2-$time
s27	4	3	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01	0.01	0.01
s820	18	5	93.44	<0.01	92.16	<0.01	93.27	<0.0	92.37	<0.01	59.62	—
s386	7	5	0.03	<0.01	0.02	<0.01	0.02	<0.01	0.03	<0.01	0.01	2.13
s510	19	6	416.80	<0.01	416.93	<0.01	414.34	<0.01	410.24	<0.01	168.52	—
s1488	8	6	0.28	<0.01	0.28	<0.01	0.33	<0.01	0.34	<0.01	0.25	126.99
s208.1	10	8	7.98	0.03	7.87	<0.01	7.98	0.03	7.85	<0.01	1.14	1959.30
mm4a	7	12	4.77	0.30	4.60	0.06	4.80	0.30	4.59	0.06	1.30	132.56
s298	3	14	0.49	0.11	0.46	0.03	0.44	0.03	0.45	0.02	0.13	0.34
s344	9	15	12,646.09	11,522.14	5,865.96	4,750.53	1,111.54	3.79	1,109.52	2.16	39.98	13448.40
s1196	14	18	4,076.59	372.44	3,863.48	0.91	3,873.84	19.64	3,865.78	0.86	1912.65	—
s382	3	21	8,9398.83	8,7834.66	6,5208.42	6,4923.48	6,3091.25	61,536.03	42,788.79	41,180.84	809.39	818.87

Table 4 Number of variables.

bench	#input	#FF	#unknown					
			naive	partition		pre-failure	partition+pre-failure	
				max	sum		max	sum
s27	4	3	18	6	18	18	6	18
s820	18	5	186	94	280	108	79	196
s386	7	5	91	30	144	77	25	126
s510	19	6	452	97	486	97	24	100
s1488	8	6	350	72	374	63	16	71
s208.1	10	8	3,840	512	3,840	3,840	512	3,840
mm4a	7	12	10,577	1,328	12,661	10,577	1,328	12,661
s298	3	14	6,332	1,645	7,730	3,192	1,102	4,144
s344	9	15	678,160	266,901	806,399	33,145	14,125	41,590
s1196	14	18	47,656	4,422	51,681	42,346	4,422	45,958
s382	3	21	1,502,857	777,881	2,217,277	1,150,087	642,249	1,738,535

The execution time of the proposed method for s344 is 1,100 seconds; which is up to 10 times lesser than that of a naive method. In all the circuits expect s382, the execution times of proposed method are lesser than the execution time of the estimation method of $2-$time expansion model. This is because the execution time for the estimation method of $k-$time expansion model exponentially increases with k. In all the circuits except s382, the abs_prob time is very short. The execution time exponentially increases with the number of FFs. This is because the execution time for logic simulation in order to obtain the state transition probabilities of the original circuit and the modified product machine is very long.

The total number of variables involved in the partition implementation is close to the total number of variables involved in the naive implementation. The number of transient states that are reachable from different initial states is small. On the other hand, the total execution time for small benchmark circuits does not decrease because the number of FFs in these circuits and the number of all the MPM states are small.

The number of variables involved in the acceleration method does not exponentially increase with the number of FFs in all the benchmark circuits. Because the number of FFs in the original circuit increases, the initial states and the number of divisions also increases exponentially.

6. Conclusion

In this paper, we proposed a procedure for the exact estimation of error propagation probability of sequential circuits and also proposed two acceleration techniques that involve dividing simultaneous linear equations into small problems and removing the variables of pre-failure states. Experimental results show the susceptibilities of FFs in various sequential circuits. The total execution time for the proposed method is found to be up to 10 times lesser than that for a naive implementation. However, there are still a few processes in this method, which are proving to be bottlenecks, for instance, the calculation of the state transition probabilities of the modified product machine. One topic for research is the acceleration of the calculation of state transition probabilities in order to apply the exact estimation method to larger scale circuits. Another topic for future research is the development of a heuristic estimation method for sequential circuits with accuracy assurance.

Acknowledgments A part of this work has been supported by CREST-DVLSI of JST.

Reference

[1] Furuta, J., Kobayashi, K. and Onodera, H.: Measurement Results of Multiple Cell Upsets on a 65nm Tapless Flip-Flop Array, *2010 IEEE Workshop on Silicon Errors in Logic-System Effects* (2010).

[2] Hachtel, G.D. and Somenzi, F.: *Logic synthesis and verification algorithms*, pp.219–244, Kluwer Academic Publishers (1996).

[3] Hayes, J.P., Polian, I. and Becker, B.: An Analysis Framework for Transient-Error Tolerance, *Proc. VLSI Test Symposium 2007*, pp.249–255 (2007).

[4] Miskov-Zivanov, N. and Marculescu, D.: Modeling and Optimization for Soft-Error Reliability of Sequential Circuits, *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, Vol.27, No.5, pp.803–816 (2008).

[5] Nicolaidis, M. and Perez, R.: Measuring the Width of Transient Pulses Induced by Ionising Radiation, *Proc. 2003 IEEE International Reliability Physics Symposium*, pp.56–59 (2003).

[6] Shivakumar, P., Kistler, M., Keckler, S.W., Burger, D. and Alvisi, L.: Modeling the Effect of Technology Trends on the Soft Error Rate of Combinational Logic, *International Conference on Dependable Systems and Networks* (*DSN'02*), Los Alamitos, CA, USA, pp.389–398, IEEE Computer Society (online), DOI: http://doi.ieeecomputersociety.org/10.1109/DSN.2002.1028924

(2002).
[7] Wang, F. and Agrawal, V.: Single Event Upset: An Embedded Tutorial, *Proc. 21st International Conference on VLSI Design*, pp.389–398 (online), DOI: http://dx.doi.org/10.1109/DSN.2002.1028924 (2008).
[8] Wirth, G.I., Vieira, M.G., Neto, E.H. and Kastensmidt, F.G.L.: Single Event Transients in Combinatorial Circuits, *Proc. 18th Annual Symposium on Integrated Circuits and System Design*, pp.121–126 (online), DOI: http://dx.doi.org/10.1145/1081081.1081115 (2005).

Masayoshi Yoshimura received his B.E. and M.E. degrees from Osaka University, Osaka, Japan in 1996 and 1998, respectively. He joined Matsushita Electrical Industrial Corporation in Osaka, Japan, in 1998 and began working in computer-aided design. He received his Ph.D. from Osaka University in 2003. He became an assistant professor at Kyushu University, Fukuoka, Japan in 2007. His current research interests include reliability-aware design methodology, design for testability, automatic test pattern generation, secure-aware design methodology. He is a member of IPSJ, IEEE and IEICE.

Yusuke Akamine received his B.E. and M.E. degrees in computer science from Kyushu University, Japan in 2008 and 2010 respectively. He currently works for Panasonic Co. Ltd. His research interests include design methodology for VLSI. He is a member of IPSJ.

Yusuke Matsunaga received his B.E., M.E. and Ph.D. degrees in electronics and communications engineering from Waseda University, Tokyo, Japan, in 1985, 1987 and 1997, respectively. He joined Fujitsu Laboratories in Kawasaki, Japan, in 1987 and he has been involved in research and development of CAD for digital systems. From October 1991 to November 1992, he has been a visiting Industrial Fellow at the University of California, Berkeley, in the department of Electrical Engineering and Computer Sciences. In 2001, he joined the faculty at Kyushu University. He is currently an associate professor of Department of Computer Science and Communication Engineering. His research interest includes logic synthesis, formal verification, high-level synthesis and automatic test patterns generation. He is a member of IEICE, IEEE, ACM and IPSJ.

(Recommended by Associate Editor: *Masanori Hashimoto*)

