

## 順序回路のソフトエラー耐性評価における近似手法 の計算精度および実行時間の評価

赤峰, 悠介  
九州大学大学院システム情報科学府

吉村, 正義  
九州大学大学院システム情報科学研究院

松永, 裕介  
九州大学大学院システム情報科学研究院

<https://hdl.handle.net/2324/16081>

---

出版情報：電子情報通信学会技術研究報告. VLD. 109 (315), pp.55-60, 2009-12. 電子情報通信学会  
バージョン：  
権利関係：

# 順序回路のソフトエラー耐性評価における近似手法の 計算精度および実行時間の評価

赤峰 悠介<sup>†</sup> 吉村 正義<sup>††</sup> 松永 裕介<sup>††</sup>

<sup>†</sup>九州大学 大学院システム情報科学府

<sup>††</sup>九州大学 大学院システム情報科学研究所

〒819-0395 福岡県福岡市西区元岡 744

E-mail: [†akamine@c.csce.kyushu-u.ac.jp](mailto:†akamine@c.csce.kyushu-u.ac.jp), [††{yosimura,matsunaga}@ait.kyushu-u.ac.jp](mailto:††{yosimura,matsunaga}@ait.kyushu-u.ac.jp)

あらまし ソフトエラー耐性を考慮した論理回路の設計ではソフトエラー耐性評価手法が必要となる。順序回路を対象とした評価手法の一つとして、マルコフモデルを用いた状態遷移の振る舞いを厳密に解析するものが提案されているが、この手法は、大規模な回路に対しては現実的な時間で実行できないという問題がある。そのため、大規模な回路に対する評価には何らかの近似手法を用いる必要がある。本稿では、二つの近似評価手法について述べ、その計算精度および実行時間の評価を行う。

キーワード ソフトエラー，順序回路，有限状態機械

## An Evaluation of Approximate Methods for Soft Error Tolerance Evaluation of Sequential Circuits

Yusuke AKAMINE<sup>†</sup>, Masayoshi YOSHIMURA<sup>††</sup>, and Yusuke MATSUNAGA<sup>††</sup>

<sup>†</sup> Graduate School of Information Science and Electrical Engineering, Kyushu University

<sup>††</sup> Faculty of Information Science and Electrical Engineering, Kyushu University

744 Motoooka, Nishiku, Fukuoka 819-0395 JAPAN

E-mail: [†akamine@c.csce.kyushu-u.ac.jp](mailto:†akamine@c.csce.kyushu-u.ac.jp), [††{yosimura,matsunaga}@ait.kyushu-u.ac.jp](mailto:††{yosimura,matsunaga}@ait.kyushu-u.ac.jp)

**Abstract** Soft error tolerance evaluation method is necessary for the soft error aware logic design. An evaluation method with Markov model is one of soft error tolerance evaluation methods for sequential circuits. In this method, we analyze the behavior of the state transition strictly. This method has an issue that it is difficult to apply for large scale circuits, therefore it is necessary some kind of approximate evaluation method. In this paper, we evaluate calculation accuracy and execute time of two approximate methods.

**Key words** soft error, sequential circuit, finite state machine

### 1. はじめに

近年、LSIの信頼性においてソフトエラーが問題として認識されつつある。ソフトエラーとは、トランジスタへの中性子の衝突時に収集される電荷により、記憶素子に保持している値の反転や、論理ゲートの出力値が反転するパルスが生じることである。値の反転に必要な最小の電荷量(臨界電荷量)は、微細化に伴い減少しており、中性子が衝突した際にソフトエラーが発生する確率は増大している。

これまで、メモリ回路におけるソフトエラーに関しては多くの解析がなされ、誤り訂正符号[1]などの比較的lowコストで施せる有効な対策が提案されている。一方、論理回路におけるソフトエラーはメモリ回路と比べて発生頻度が低いことなどからあ

まり重点的に研究されておらず、また、メモリ回路と比較して回路構造に規則性がないことなどから、効果的な対策や解析手法は確立されていない[2]。そのため、論理回路のソフトエラーもメモリ回路同様に無視できない問題となる可能性がある[3]。

論理回路の設計時にソフトエラー耐性を考慮する場合、回路が所望のソフトエラー耐性を持つか評価する必要がある。等しい機能を持つ回路であっても、設計の方法によりソフトエラー耐性は異なるためである。よって、ソフトエラー耐性の評価手法が必要となる。ソフトエラー耐性の評価には、ソフトエラーが発生しエラーが外部出力に伝搬する確率を指標として用いるのが一般的である。論理回路に対するソフトエラー耐性評価手法はいくつか提案されているが[4][5][6][7]、それらの手法の多くは、組み合わせ回路を対象としている。すなわち、ソフトエ

ラーが発生したクロックサイクルにおいてエラーが外部出力に伝搬する確率やフリップフロップ (以降, FF と略す) にエラーが取り込まれる確率を計算する手法である。しかしながら, 順序回路においてはソフトエラーにより誤った状態に遷移しても, エラーが外部に顕在化することなく正常な状態に戻る可能性がある。また, あるクロックサイクルにおいてエラーが伝搬するか否かは入力値と FF の値に依存する。そのため, FF の値がどのような確率で分布するのかを考慮する必要がある。文献 [7] の手法は, 順序回路を対象としているが, 組み合わせ回路部分を数時刻分時間展開しただけであり, 本質的には順序回路を扱っているとは言い難い。これに対し著者は, 順序回路におけるソフトエラー発生後の振る舞いを確率的に遷移する状態機械 (マルコフモデル) としてモデル化し, ソフトエラー耐性を評価する手法を提案している [8]。

この手法は, 与えられた入力値の確率分布の仮定のもとでは厳密な解析を行うものであるが, 計算時間の点から実用的とはいえない。文献 [9] の高速化手法により, いくらかの高速化が図られているものの, 大規模な回路に適用することは不可能であると考えられる。したがって, 大規模な回路にも適用可能な何らかの近似手法が必要である。そこで, 本稿では, 順序回路のソフトエラー耐性を近似的に評価する二つの手法について述べ, その計算精度および実行時間を評価する。実験により, 近似評価手法は, 多くのベンチマーク回路に対し精度をあまり落とさずに短時間での実行が可能であり, また, 厳密手法では扱えなかった規模の回路に対して評価が行えることがわかった。

本稿の構成を以下に示す。まず, 2 節でマルコフモデルを用いたソフトエラー耐性評価手法を説明する。次に, 3 節で二つの近似評価手法について説明する。4 節で近似評価手法の計算精度および計算時間の評価実験の結果を示し, 5 節で本稿をまとめる。

## 2. マルコフモデルを用いたソフトエラー耐性評価手法 [8]

本節では, マルコフモデルを用いた順序回路のソフトエラー耐性評価手法を説明する。本手法では, 正常な振る舞いとエラーを含む振る舞いの比較のため, 評価対象の順序回路を複製した回路対を用いる。また, 状態遷移の振る舞いを, 確率的に遷移する状態機械であるマルコフモデルにより表し, ソフトエラーが発生し外部出力に伝搬する確率を計算する。なお, 以降では, 本手法を厳密手法と呼ぶ。

### 2.1 順序回路と回路対

順序回路とは, 有限状態機械を論理素子および FF などの記憶素子を用いて実現したもので, 出力値が現在の入力系列に依存するという特徴を持つ。過去の入力系列に依存した「状態」を記憶素子の値として保持することで, 過去の入力系列を出力に反映させる。順序回路の一般的な構成を図 1 に示す。図 1 では, 内部情報の記憶のために FF を用いている。

順序回路がソフトエラーによって通常とは異なる動作をした結果, 外部出力に誤った値が出力されるかどうかを調べるためには, 正しい動作をしている回路 (以降, 正常回路と呼ぶ) と,

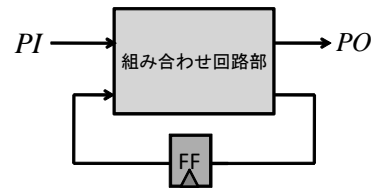


図 1 順序回路の構成例

ソフトエラーの発生を仮定した回路 (以降, 故障回路と呼ぶ) に同一の入力を与え, 出力が異なるかを調べればよい。このような二つの回路を並べた回路を, 本稿では回路対と呼ぶ。回路対の例を, 図 2 に示す。図 2 において, 正常回路は評価対象の回

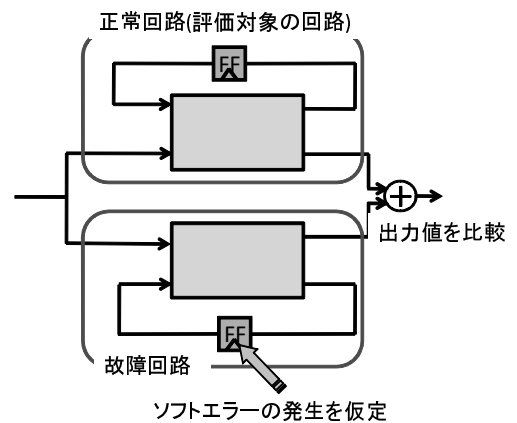


図 2 回路対

路である。故障回路は, 回路構成は正常回路と同様であるが, FF でのソフトエラーの発生を仮定した回路である。故障回路はソフトエラー発生直前までは正常回路と等しい振る舞いをする。正常回路と故障回路の出力値が異なる時, エラーは外部出力に伝搬したといえる。

### 2.2 マルコフモデル

以上のように, 回路対を用いることで, 正常な回路の振る舞いとエラーを含む振る舞いの比較を行うことができる。本来, 順序回路の振る舞いは入力と状態によって一意に定まる, 決定的なものである。ただし, 回路の動作と中性子が衝突するタイミングには関連がないため, 順序回路の動作から見ればソフトエラーはランダムに起こるものとみなすことができる。また, 与えられる入力に関しても, ソフトエラー発生タイミングとは無関係に決まるので, なんらかの確率分布に従ったランダムなものともみなすことができる。そこで, 回路対の状態遷移の振る舞いを, 確率的に状態遷移する状態機械であるマルコフモデルであると考えられる。

回路対の状態は, ソフトエラー発生前の状態と, ソフトエラー発生後の状態に大別できる。ここで, ソフトエラー発生前の状態の集合を  $S$ , ソフトエラー発生後の状態の集合を  $\Pi$  とする。 $S$  の状態から  $\Pi$  の状態への遷移は, ソフトエラーの発生によるのみ起こる。

ソフトエラー発生後の回路の振る舞いにおいて, 一時刻でも誤った値を出力すれば, その後どのような動作をしようとエ

ラーは外部出力に伝搬したことになる．また，一旦正常な状態に遷移してしまえば，次にエラーが起きるまでは正常な動作をする．よって，状態を以下のように三つに分類することで，ソフトエラー発生後の振る舞いをモデル化できる．

#### failure 状態

エラーが外部出力まで伝搬した後の状態

#### masked 状態

正常回路と故障回路の FF の値が等しい状態

#### 一時状態

正常回路と故障回路の FF の値は異なるが，エラーは顕在化していない状態

failure 状態および masked 状態は，確率 1 で自分自身へ遷移する状態であり，一旦 failure 状態や masked 状態に到達すると，以降は永久にその状態にとどまり続けることになる．以降，これらの状態に到達することを，その状態に吸収されると表現する．図 3 は，ソフトエラー発生後の状態遷移をマルコフモデルで表わした例である．図 3 において，「failure」，「masked」は，

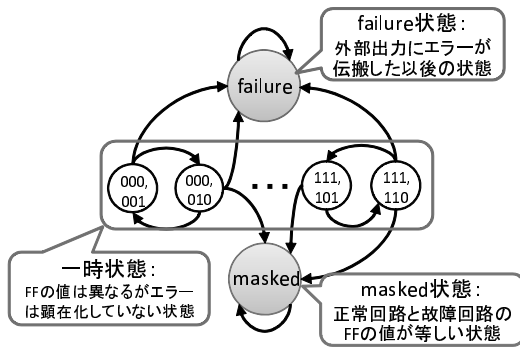


図 3 マルコフモデルと状態の分類

failure 状態，masked 状態をそれぞれ表す．また，一時状態は正常回路と故障回路の FF の値のペアで表わしている．

ソフトエラー発生後の振る舞いを上記のマルコフモデルで表わしたとき，エラーが発生し外部出力に伝搬する確率を求めることは，failure 状態に吸収される確率を求めることと等しい．

### 2.3 表 記

- $S$  : ソフトエラー発生前の状態の集合
- $\Pi$  : ソフトエラー発生後の状態の集合
- $\Pi_{tmp}$  : 一時状態の集合
- $\Pi_{init}$  : ソフトエラー発生直後の状態 (初期状態) の集合
- $\pi_f$  : failure 状態
- $r_{s_i s_j}$  :  $s_i \in S$  から  $s_j \in S$  への遷移確率
- $e_{s\pi}$  : ソフトエラーにより  $s \in S$  から  $\pi \in \Pi$  に遷移する確率

確率

- $p_{\pi_i \pi_j}$  :  $\pi_i \in \Pi$  から  $\pi_j \in \Pi$  への遷移確率
- $q_{\pi \pi_f}$  :  $\pi \in \Pi$  を出発し  $\pi_f$  に吸収される確率
- $P_{abs}(\pi_f)$  :  $\pi_f$  に吸収される確率
- $P_{init}(\pi)$  :  $\pi \in \Pi$  の初期状態確率

- $P_{steady}(s)$  :  $s \in S$  の定常確率

### 2.4 確率の計算

ソフトエラーが発生し外部出力に伝搬する確率を求めることは，failure 状態  $\pi_f$  に吸収される確率を求めることと等しい．ソフトエラーが発生し外部出力に伝搬する確率  $P_{abs}(\pi_f)$  は，以下の式で表わされる．

$$P_{abs}(\pi_f) = \sum_{\pi \in \Pi} P_{init}(\pi) \cdot q_{\pi \pi_f} \quad (1)$$

式 (1) において， $P_{init}(\pi)$  は状態  $\pi$  の初期状態確率であり，ソフトエラー発生直後の状態が  $\pi$  である確率である． $q_{\pi \pi_f}$  は，状態  $\pi$  を出発して  $\pi_f$  に吸収される確率であり，吸収確率と呼ぶ．以下に，初期状態確率及び，吸収確率の計算方法を示す．

#### 2.4.1 初期状態確率の計算

状態  $\pi$  の初期状態確率  $P_{init}(\pi)$  は，以下の式で表わされる．

$$P_{init}(\pi) = \sum_{s \in S} P_{steady}(s) \cdot e_{s\pi} \quad (2)$$

$s \in S$  の定常確率  $P_{steady}(s)$  は，ソフトエラー発生前のある時刻の状態が  $s$  である確率である．また， $e_{s\pi}$  は，ソフトエラーにより状態  $s$  から状態  $\pi \in \Pi$  に遷移する確率を表す．

ソフトエラー発生以前の状態  $s$  の定常確率  $P_{steady}(s)$  は以下のように計算できる．まず，ソフトエラー発生以前において，状態  $s_i$  から  $s_j$  に遷移する確率を  $r_{s_i s_j}$  とする．このとき， $s_j$  の定常確率は以下の式で表わされる．

$$P_{steady}(s_j) = \sum_{s_i \in S} P_{steady}(s_i) \cdot r_{s_i s_j} \quad (3)$$

また，全状態の定常確率の総和は 1 であるので，以下の式が成り立つ．

$$\sum_{s \in S} P(s) = 1 \quad (4)$$

式 (3),(4) により，各状態間の遷移確率が既知であれば，定常確率を未知数とする連立方程式を解くことにより，定常確率は計算可能である．

一方， $e_{s\pi}$  は，FF においてソフトエラーが発生する確率や，論理素子におけるソフトエラーの影響が FF に到達する確率などから計算することができる．

#### 2.4.2 吸収確率の計算

$\pi_i \in \Pi$  を出発し failure 状態  $\pi_f$  へ吸収される確率  $q_{\pi_i \pi_f}$  は以下のように表される．

$$q_{\pi_i \pi_f} = \sum_{\pi_j \in \Pi_{tmp}} p_{\pi_i \pi_j} q_{\pi_j \pi_f} + p_{\pi_i \pi_f} \quad (5)$$

各状態間の遷移確率が既知であれば， $q_{\pi_i \pi_f}$  を未知数とする連立方程式となるため，各状態間の遷移確率を求めればよい．

### 2.5 実装および問題点

上記の手法により，順序回路においてエラーが外部出力に伝搬する確率を計算することが可能であるが，このままでは計算時間のうえで問題がある．回路対がソフトエラー発生後に取り

得る状態数は、評価対象の回路の FF 数を  $N$  とすると、 $2^{2N}$  となる。しかしながら、その全ての状態が初期状態から到達可能であるとは限らず、多くの場合到達可能であるのはごく一部の状態のみである。よって、到達可能状態を列挙し、遷移確率の計算はそれらに対してのみ行う。到達可能状態の列挙を単純に行った場合、多大な実行時間を要する。そこで著者らは、BDD(Binary Decision Diagram) を用いた状態の非明示的列挙による高速化を提案した [9]。しかしながら、より規模の大きな回路に対しては、本手法をそのまま適用することは困難であり、大規模な回路にも適用可能な近似手法が必要となると考えられる。

### 3. 近似評価手法

前節で示した厳密手法では、実行時間が長く、規模の大きな回路には適用できない。そこで、本節では、厳密手法の計算の一部を近似的な処理に置き換えた、近似評価手法について説明する。近似評価手法においては、計算を高速化しつつも、できるだけ精度を落とさないようにする必要がある。本節で示す近似手法は二つあり、一つ目は、FF を外部出力とみなして解析する手法 (以下、PPO 法と呼ぶ) である。また、二つ目は、PPO 法からさらに処理を省略したもので、ソフトエラー発生以前の各状態の定常確率が一樣であると仮定して解析する手法 (以下、PPO+PPI 法と呼ぶ) である。

#### 3.1 PPO 法

PPO 法は、FF を外部出力とみなして解析する評価手法である。図 4 に、PPO 法で解析に用いる回路モデルを示す。PPO

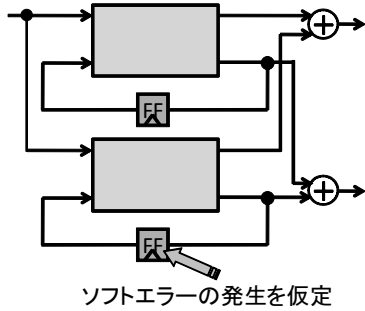


図 4 PPO 法に用いる回路モデル

法においても、正常な振る舞いをする正常回路と、ソフトエラーの発生を仮定した故障回路の振る舞いを比較するというのが基本的な考え方である。厳密手法との違いは、外部出力だけでなく、FF の値の比較を行い、異なった場合 failure 状態への遷移とみなすという点である。これはすなわち、一時状態に遷移した後に masked 状態へ遷移する可能性を無視することを意味する。このような近似を行うことで、回路対の到達可能状態の列挙、および吸収確率の計算における連立方程式を解く処理の省略が行える。以下に、PPO 法における処理を説明する。

求めるべき値である、failure 状態に吸収される確率は、式 (1) と同様である。また、初期状態確率の計算も式 (2) と同様であり、厳密手法と同じ処理で求める。PPO 法では、外部出

力だけでなく、FF の値が異なる場合も failure 状態への遷移とみなす。よってある状態  $\pi_i$  から failure 状態  $\pi_f$  への吸収確率は、以下のように表わされる。

$$q_{\pi_i \pi_f} = \sum_{\pi_j \in \Pi_{tmp}} p_{\pi_i \pi_j} + p_{\pi_i \pi_f} \quad (6)$$

以上より、PPO 法では、厳密手法のように吸収確率を未知数とする連立方程式を解く必要はない。また、初期状態以外からの遷移確率を求める必要がなく、回路対の到達可能状態の列挙も不要である。これらの省略された処理は、厳密手法において計算時間の大部分を占めている処理であることから、PPO 法では計算時間の短縮が見込まれる。しかしながら、前述したように、PPO 法では、一時状態に遷移した後に masked 状態に戻る可能性を無視しているため、エラーが伝搬する確率を過大に見積もる可能性がある。

#### 3.2 PPO+PPI 法

PPO+PPI 法は、PPO 法からさらに処理を省略し、ソフトエラー発生以前の各状態の定常確率を一樣とみなして解析する手法である。PPO+PPI 法におけるソフトエラー発生以前の状態  $s \in S$  の定常確率は以下の式で得られる。

$$P_{steady}(s) = 1/|S| \quad (7)$$

定常確率の計算以外の処理は、PPO 法と同様である。

PPO+PPI 法では、PPO 法からさらに、連立方程式で定常確率を計算する処理が省略される。PPO 法では規模の大きい回路に対し、実行時間の大部分が連立方程式による定常確率の計算で占められる可能性もあるため、PPO+PPI 法はさらなる実行時間の短縮が見込める。一方で、PPO 法同様に FF の値が異なるとき failure とみなす点、また、定常確率の偏りを考慮していない点から、計算精度に問題がある可能性がある。

## 4. 実験

近似評価手法の計算精度および実行時間を評価するために実験を行った。各評価手法は、C++ 言語で実装した。実験に用いたマシンの CPU は Intel Xeon 3.3GHz、メモリは 16GB である。ベンチマーク回路として、ISCAS'89 ベンチマークセットおよび MCNC ベンチマークセット [10] の順序回路の一部を用いた。

本実験では、入力値の確率分布は一樣であると仮定し、遷移確率の計算には全入力パターンに対する論理シミュレーションを用いた。また、ソフトエラーは FF においてのみ起き、複数の FF では同時におきないと仮定する。計算精度の評価には、順序回路においてソフトエラーが発生し外部出力に伝搬する確率  $P_{abs}(\pi_f)$  ではなく、各 FF のエラーが外部出力に伝搬する確率 (以降、サセプタビリティと呼ぶ) を用いる。サセプタビリティは、個々の FF のソフトエラー耐性を評価する指標であり、回路の評価指標と同様に、ソフトエラー耐性を考慮した設計では重要なものである。 $P_{abs}(\pi_f)$  はサセプタビリティとソフトエラーの発生確率の積の総和であるため、サセプタビリティの精度は、 $P_{abs}(\pi_f)$  の精度に影響を与える。FF  $k$  のサセプタ

表 1 実験結果 (計算精度)

circuit	method	susceptibility													
		FF1	FF2	FF3	FF4	FF5	FF6	FF7	FF8	FF9	FF10	FF11	FF12	FF13	FF14
s27	base	0.2283	0.2262	0.1454											
	PPO	0.2283	0.2262	0.3086											
	PPO+PPI	0.2708	0.2917	0.3125											
s820	base	0.9264	0.9995	0.7594	0.7593	0.7707									
	PPO	0.9448	0.9996	0.7594	0.7594	0.7994									
	PPO+PPI	0.6103	0.9500	0.8450	0.6525	0.6025									
s386	base	0.6319	0.6166	0.7970	0.8785	0.6423	0.6327								
	PPO	0.6789	0.6634	0.8031	0.8846	0.6938	0.6540								
	PPO+PPI	0.7115	0.3750	0.9423	0.8269	0.4904	0.4808								
s510	base	0.9900	0.8976	0.9397	0.9921	0.9809	1.0000								
	PPO	1.0000	0.9575	1.0000	1.0000	0.9858	1.0000								
	PPO+PPI	1.0000	0.9255	1.0000	1.0000	0.9894	1.0000								
s1488	base	0.9616	0.9894	0.9972	0.9998	0.9612	0.7958								
	PPO	0.9616	0.9894	0.9973	0.9999	0.9617	0.7958								
	PPO+PPI	0.9323	0.8724	0.9154	0.9010	0.9375	0.9323								
s208.1	base	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000							
	PPO	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000							
	PPO+PPI	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000							
mm4a	base	0.2500	0.2500	0.2500	0.2500	0.0115	0.0218	0.0364	0.0655	0.0037	0.0150	0.0297	0.0589		
	PPO	0.2500	0.2500	0.2500	0.2500	0.0150	0.0223	0.0369	0.0660	0.0150	0.0223	0.0369	0.0660		
	PPO+PPI	0.2500	0.2500	0.2500	0.2500	0.0898	0.0928	0.0967	0.0964	0.0898	0.0928	0.0967	0.0964		
s298	base	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
	PPO	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
	PPO+PPI	0.7500	0.5938	0.6240	0.7021	0.7314	0.9189	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.5313	0.5000

ピリティの計算式を以下に示す．ソフトエラー発生以前の状態  $s \in S$  において，FF  $k$  でエラーが起きたときの遷移先を  $s^k$  とすると， $s^k$  の初期状態確率は  $k$  でソフトエラーが発生する確率  $P_{error}(k)$  を用いて，以下のように表わせる．

$$P_{init}(s^k) = P_{steady}(s) \cdot P_{error}(k) \quad (8)$$

式 (1)，(8) より，FF  $k$  でエラーが発生し外部出力に伝搬する確率  $P_{failure}(k)$  は，以下の式で表わされる．

$$P_{failure}(k) = P_{error}(k) \sum_{s \in S} P_{steady}(s) \cdot q_{s^k \pi_f} \quad (9)$$

よって，FF  $k$  のサセプタビリティ  $P_{prop}(k)$  は，以下の式で表わされる．

$$P_{prop}(k) = \sum_{s \in S} P_{steady}(s) \cdot q_{s^k \pi_f} \quad (10)$$

#### 4.1 計算精度の評価

実験結果を表 1 に示す．表 1 において，method の列は，評価手法を表し，base は厳密手法，PPO は PPO 法，PPO+PPI は PPO+PPI 法をそれぞれ表わす．また，susceptibility の列において，回路中の各 FF のサセプタビリティを示す．

まず，厳密手法と PPO 法の比較を行う．厳密手法に対し，PPO 法で求めたサセプタビリティは，多くの FF で近い値になっていることがわかる．ただし，s27 の FF3 や，mm4a，s298 などの一部の FF においては誤差が大きい．s298 の FF1 では，厳密手法と比較して 5 倍程度の値となっている．また，全ての FF において，過大見積もりとなっていることがわかる．これは，本来の出力だけでなく FF の値が異なる場合も failure とみなしたことによるものである．以上より，一部の例外はあるものの，PPO 法は比較的よい近似となっているといえる．

次に，PPO+PPI 法を厳密手法と比較する．比較的誤差の小さいベンチマークはあるものの，s820 や s386 などのような，誤差の大きなベンチマークがあることがわかる．これらのベンチマークにおいて，PPO 法での誤差は小さい．よって，

PPO+PPI 法の誤差が大きい原因は，定常確率の偏りを考慮していないことであると考えられる．また，PPO 法で誤差の大きい FF は，PPO+PPI 法でも誤差が大きく，s298 の FF1 では厳密手法と比較して 7 倍程度の値となっている．また，PPO + PPI 法の結果は，PPO 法のように常に過大見積もりとなるわけではなく，大小にばらついていることがわかる．

結果より，PPO 法は，一部の例外はあるものの，多くのベンチマーク回路で厳密手法と比較してあまり誤差がなく，PPO+PPI 法は比較的誤差が大きいといえる．PPO 法と PPO+PPI 法の違いは，定常確率の偏りを考慮しているか否かである．ソフトエラー耐性の正確な評価には，FF の値の定常確率を考慮することが重要であるといえる．

#### 4.2 実行時間の評価

表 2 実験結果 (実行時間)

circuit	#input	#FF	time[s]		
			base	PPO	PPO+PPI
s27	4	3	0.01	0.01	0.01
s820	18	5	84.99	22.50	19.74
s386	7	6	0.06	0.01	0.01
s510	19	6	382.91	63.66	55.65
s1488	8	6	0.55	0.43	0.32
s208.1	10	8	129.06	6.27	4.45
mm4a	7	12	2388.28	1.60	0.46
s298	3	14	516.82	0.40	0.28
s344	9	15		63.84	5.21
s1196	14	18		661.11	627.22
s382	3	21		1411.93	6.65

表 2 に実行時間の比較を示す．表 2 において，#input，#FF は，入力数，FF 数をそれぞれ表わす．time の列は各手法の実行時間である．ここで，base の列における「」は，数日間の実行で終了しなかったことを表す．

まず，s510 の結果に着目する．各手法の実行時間は，厳密手法では 380 秒程度，二つの近似手法では 60 秒程度である．FF 数は多くないため連立方程式を解くことによる吸収確率の計算にはさほど時間がかからないが，厳密手法では到達可能状態の列挙および遷移確率の計算に多くの時間を要している．

次に，mm4a の結果に着目する．厳密手法では，2400 秒ほ

どの時間がかかっているのに対し、近似手法では1秒程で計算を終了した。mm4a に対しての厳密手法の実行時間のほとんどは、連立方程式を解くことによる吸収確率の計算で占められる。そのため、吸収確率の計算に連立方程式を用いない近似手法では、実行時間が大幅に短縮したと考えられる。

また、FF 数が 15 以上のベンチマークに対しては、厳密手法は、数日かけても処理を終了しなかった。これは、吸収確率の計算に多大な時間を要しているためである。一方、二つの近似手法では、FF 数が 21 の s382 において、PPO 法で約 1400 秒、PPO+PPI 法では 6 秒程度である。同様に s344 でも二つの近似手法間で実行時間の差が大きい。これらの回路で、PPO+PPI 法が PPO 法と比較しても高速であるのは、定常確率の計算を省略しているためである。

## 5. おわりに

本稿では、順序回路のソフトエラー耐性の近似評価手法について述べ、その計算精度および実行時間の評価を行った。実験結果より、近似評価手法である PPO 法は、多くのベンチマーク回路で計算精度をあまり落とすことなく、厳密手法と比較して短時間で実行できることがわかった。また、PPO + PPI 法は、PPO 法よりさらに短時間で実行できるものの、誤差の大きいベンチマークが多いことがわかった。

しかしながら、PPO 法においても、mm4a や s298 のような一部のベンチマークでは誤差の大きい FF があり、精度向上に関する検討が必要である。精度向上に向けたアプローチの一つとして、PPO 法と時間展開を組み合わせたの解析が考えられる。PPO 法の誤差の原因は、一時状態への遷移を failure 状態への遷移とみなすことである。PPO 法に時間展開を組み合わせると、数時刻先までの状態遷移を解析できるため、計算精度が向上する可能性がある。また、PPO 法は、FF 数が 20 程度の回路に対して 1400 秒程度の実行時間を要しており、実行時間の観点からも根本的な解決とはなっていない。そのため、さらなる高速化が必要である。s382 や s344 などの回路に対する処理においては、定常確率の計算が PPO 法の実行時間の大部分を占めた。よって、定常確率の計算の高速化が今後の課題の一つである。また、本実験では、遷移確率の計算の際、入力値の確率分布は一様であるという仮定のもと、全入力パターンを用いた論理シミュレーションを用いた。規模の大きな回路に対しては全入力パターンに対する論理シミュレーションを行うのは難しく、また、BDD を用いた遷移確率の計算処理 [9] を用いることも困難である。そこで、今後は、入力パターンは与えられた確率分布に従いランダムサンプリングすることで、処理の高速化を検討する予定である。

## 謝 辞

本研究の一部は、科学技術振興機構 (JST) の戦略的創造研究推進事業 (CREST) 「統合的高信頼化設計のためのモデル化と検出・訂正・回復技術」の支援によるものである。

## 文 献

- [1] 情報理論とその応用学会編. “符号理論とその応用”. 培風館,

2003.

- [2] 上村大樹, 戸坂清春, 芹沢芳夫, 岡秀樹, 佐藤成生. “中性子ソフトエラーシミュレーションの新展開”. 信学技報, ICD, Vol. 105, No. 2, pp. 37–42, 2005.
- [3] P. Shivakumar, M. Kistler, S. W. Keckler, D. Burger, L. Alvisi. “Modeling the Effect of Technology Trends on the Soft Error Rate of Combinational Logic”. In *Proc. of DSN*, 2002.
- [4] Smita Krishnaswamy, Stephen M. Plaza, Igor L. Markov, John P. Hayes. “Enhancing design robustness with reliability-aware resynthesis and logic simulation”. In *Proc. of ICCAD*, pp. 149–154, 2007.
- [5] Yan Lin and Lei He. “Device and Architecture Concurrent Optimization for FPGA Transient Soft Error Rate”. In *Proc. of ICCAD*, pp. 194–198, 2007.
- [6] N. Miskov-Zivanov and D. Marculescu. “MARS-C: Modeling and Reduction of Soft errors in Combinational Circuits”. In *Proc. of DAC*, pp. 767–772, 2006.
- [7] N. Miskov-Zivanov and D. Marculescu. “MARS-S: Modeling and Reduction of Soft Errors in Sequential Circuits”. In *Proc. of ISQED*, pp. 755–760, 2007.
- [8] 赤峰悠介, 吉村正義, 松永裕介. “マルコフモデルを用いた順序回路のソフトエラー耐性評価手法”. DA シンポジウム 2009 論文集, 2009.
- [9] 松永裕介, 赤峰悠介. “順序回路のソフトエラー率解析手法の非明示的列挙による高速化について”. 信学技報, VLD, Vol. 109, No. 201, pp. 31–36, 2009.
- [10] S Yang. “Logic Synthesis and Optimization Benchmarks User Guide: Version 3.0”. 1991.