

順序回路のソフトウェア耐性評価手法の高速化

吉村, 正義
九州大学大学院システム情報科学研究院

赤峰, 悠介
九州大学大学院システム情報科学府

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/2324/6794516>

出版情報 : DAシンポジウム2010. 2010 (7), pp.239-244, 2010-09. 情報処理学会
バージョン :
権利関係 :



順序回路のソフトエラー耐性評価手法の高速化

吉村 正義[†] 赤峰悠介[‡] 松永裕介[†]

[†] 九州大学 大学院システム情報科学研究所

[‡] 九州大学 大学院システム情報科学府

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。著者らは、順序回路を対象とした評価手法として、マルコフモデルを用い状態遷移の振る舞いを厳密に解析するものを提案している。この手法は計算時間の点で問題があり、その大部分を占める処理の一つが、状態数を元数とする連立方程式の計算である。そこで、本稿では、連立方程式を解く時間を左右する連立方程式の元数の最大と総和を削減する手法を提案する。

An acceleration of analyzing soft-error tolerance for sequential circuits

Masayoshi YOSHIMURA[†] Yusuke AKAMINE[‡] Yusuke MATSUNAGA[†]

[†] Graduate School of Information Science and Electrical Engineering, Kyushu University

[‡] Faculty of Information Science and Electrical Engineering, Kyushu University

Soft error tolerance estimation method is necessary for the soft error aware logic design. We proposed an estimation method with Markov model for sequential circuits, which can analyze the behavior of the state transition strictly. This method has an issue that it is difficult to apply for large scale circuits, and solving simultaneous equations is one of bottleneck processes. In this paper, we propose acceleration methods by reducing the number of unknowns in simultaneous equations.

1 はじめに

近年 LSI の製造技術の進展に伴い、LSI のソフトエラー耐性が低下してきている。ソフトエラーとは、トランジスタへの中性子の衝突に起因し発生する電荷により、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の一時的な反転が起きることである。値の反転に必要な最小の電荷量（臨界電荷量）は、半導体デバイスの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] などの比較的低コストで施せる有効な対策が提案されている。一方、論理回路に関しては、メモリ回路と比較して回路構造に規則性がないことから、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、今後論理回路のソフトエラーが無視できない問題となる可能性がある [3]。

チップ設計者は論理回路のソフトエラーの影響を評価する必要がある。順序回路のソフトエラー耐性は順序回路の機能だけでなく、順序回路の設計にも依存する。一般にソフトエラーの耐性はソフトエラーレートによって評価される。ソフトエラーレートはソフトエラーが発生し、十分な時間の間少なくとも一度外部出力にソフトエラーの影響が伝搬する確率である。

ソフトエラーによる順序回路の誤動作は十分な時間の間に少なくとも一度いずれかの外部出力にソフ

トエラーの影響が伝搬することと定義される [9]。文献 [7][8] において、順序回路のソフトエラー耐性評価手法が提案されている。文献 [7] では、単一遷移故障モデル (STF:single transient fault) を用いて順序回路の振る舞いを解析し、STF エラー確率を順序回路のソフトエラー耐性の評価尺度としている。STF モデルはある一つのイベントで発生したソフトエラーの影響が複数の FF に伝搬するケースと複数のソフトエラーの影響が複数の外部出力に伝搬するケースを考慮していない。また STF エラー確率はソフトエラーが発生したときに順序回路がどの状態であるかを考慮していない。このため STF エラー確率は順序回路のソフトエラー耐性評価尺度として適切ではない。文献 [8] ではマルコフ解析手法を用いてソフトエラー発生後の順序回路の定常確率を求め、その定常確率に基づいてソフトエラー耐性を評価している。ソフトエラー耐性の尺度はいずれかの外部出力にソフトエラーの影響が伝搬する単位時間当たりの確率に基づいている。初期状態から到達可能な状態でありながら、順序回路の定常状態に含まれない状態が存在する。文献 [8] の確率は十分な時間の間に少なくとも一度いずれかの外部出力にソフトエラーの影響が伝搬する確率とは等しくない。そのため文献 [8] の確率は順序回路のソフトエラー耐性評価尺度として適切ではない。

文献 [9] では、順序回路のソフトエラー耐性を評価するために、修正回路対を用いている。修正回路対はソフトエラーが発生した後に十分な時間の間に

少なくとも一度順序回路の外部出力にソフトエラーの影響が現れるかどうかを測るために用いられる。順序回路の状態遷移を吸収状態を持つマルコフモデルを用いて厳密に解析する。少なくとも一度いずれかの外部出力にソフトエラーの影響が伝搬する確率は、修正回路対の吸収状態への遷移確率（吸収確率）から求めることができる。この手法は与えられた入力値の確率分布の基では厳密なソフトエラー耐性を求めることができる。しかしこの手法は大規模回路に適用することが困難である。吸収確率を求める際の連立方程式を解く時間は実行時間の大半を占める処理の一つである。

そこで本論文では吸収確率を求める際の連立方程式を解く時間を短縮するための手法を提案する。この手法は、拡大 pre-failure 状態の特定と強連結成分に基づく状態集合の分割から構成されている。これらの手法により、吸収確率を求める連立方程式において、一度に解く元の数の最大数と、すべての連立方程式の元の数の総和の削減をはかる。

本論文の構成を以下に記す。2 節では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 節で提案する高速化手法を説明し、4 節で高速化手法の評価実験結果を示す。5 節で本論文をまとめる。

2 マルコフモデルを用いたソフトエラー耐性評価手法

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。なお、以降では、本手法を厳密手法と呼ぶ。

2.1 順序回路と修正回路対

順序回路とは、有限状態機械を論理素子およびフリップフリップ（以降、FF と略す）などの記憶素子を用いて実現したもので、出力値が過去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、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、出力が少なくとも一度以上異なるかを調べればよい。

本論文では正常回路と故障回路およびソフトエラーの影響が少なくとも一度外部出力に伝搬したかを判定する回路をあわせて修正回路対と呼ぶ。修正回路対の例を図 1 に示す。図 1 において、正常回路は評価対象の回路である。故障回路は、回路構成は

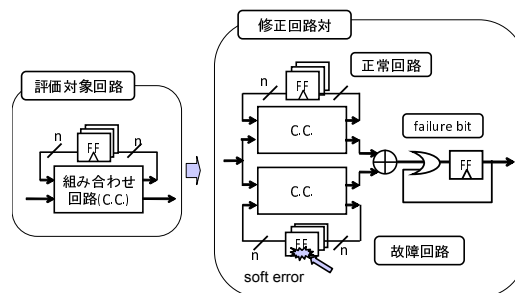


図 1: 評価対象回路と修正回路対

正常回路と同様であるが、ソフトエラーが 1 度発生することを仮定した回路である。また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。failure bit を用いて、正常回路と故障回路の出力値が少なくとも 1 度異なるかを観測する。failure bit の値はソフトエラー発生前は 0 とする。もしソフトエラー発生後十分な時間後に failure bit の値が 1 になっていれば、ソフトエラーの影響が外部出力に伝搬したことを示す。

2.2 マルコフモデル

修正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、ソフトエラー発生の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みなすことができる。そこで、修正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ることができる。

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を求めるためには、外部出力へのエラーの伝搬を考慮しつつ状態遷移の振る舞いを解析する必要がある。そこで、文献 [9] の手法では、修正回路対の状態遷移を表わすマルコフモデル上に、外部出力にエラーが伝搬したことを表わす仮想的な状態を設け、状態遷移の振る舞いを解析する。このように状態遷移の振る舞いを表したとき、ソフトエラーが発生し外部出力に伝搬する確率は、この仮想的な状態に到達する確率を求めることで計算できる。

外部出力にエラーが伝搬した以降の状態を、failure 状態と呼ぶ。本手法の目的はソフトエラーが発生し少なくとも一度ソフトエラーの影響が外部出力に伝搬する確率を求めることであるため、エラーの影響が外部出力に伝搬した以降の振る舞いに関して解析

する必要はない．つまり failure 状態は自分自身に確率 1 で遷移する状態とすることができる．このような状態に到達すると，以降は永久にその状態にとどまり続けることになる．ここで，確率 1 で自分自身に遷移する状態を吸収状態と呼び，吸収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．また，一旦正常回路と故障回路の FF の値が等しい状態に遷移すると，それ以降の時刻で外部出力の値が誤る可能性はない．よって，正常回路と故障回路の値が等しい状態に関しても，以降の振る舞いを解析する必要はなく，吸収状態とみなせる．正常回路と故障回路の FF の値が等しい状態を masked 状態と呼ぶ．

以上より，ソフトウェア発生後の状態を以下のように三つに分類することができる．

- failure 状態 failure bit が 1 である状態
- masked 状態 failure bit が 0 であり，かつ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 failure bit が 0 であり，状態正常回路と故障回路の FF の値は異なる状態

図 2 は，ソフトウェア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．ソフトウェア発生前の状態は正常回路の FF の値の組で表し，ソフトウェア発生後の状態を正常回路と故障回路の FF の値と failure bit の値の組で表している．ソフトウェアのみによって，ソフトウェア発生前の状態からソフトウェア発生後の状態に遷移する．

図 2 における failure 状態と masked 状態をひとつの吸収状態としたマルコフモデルを図 3 に示す．順序回路においてソフトウェアが発生し外部出力に伝搬する確率を求めることは，上記の吸収状態をもつマルコフモデル上で，failure 状態に吸収される確率を求めることに他ならない．

2.3 ソフトエラーが発生し外部出力に伝搬する確率の計算

ソフトウェア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 π_f に吸収される確率を求めることと等しい．ソフトウェア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．

$$P_{abs}(\pi_f) = \sum_{\pi_i \in \Pi_{init}} P_{init}(\pi_i) \cdot P_{tr_abs}(\pi_i, \pi_f) \quad (1)$$

式 1 において， $P_{init}(\pi)$ は状態 π の初期状態確率であり，ソフトウェア発生直後の状態 (初期状態と呼ぶ) が π である確率である． Π_{init} は初期状態の

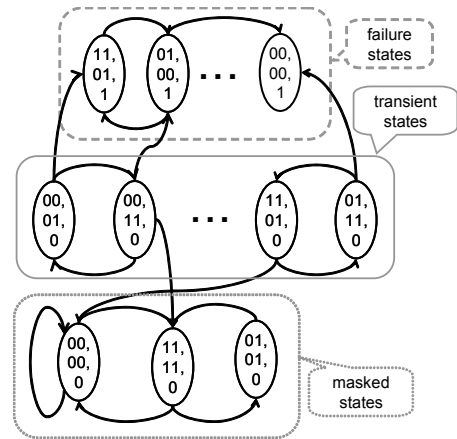


図 2: マルコフモデル

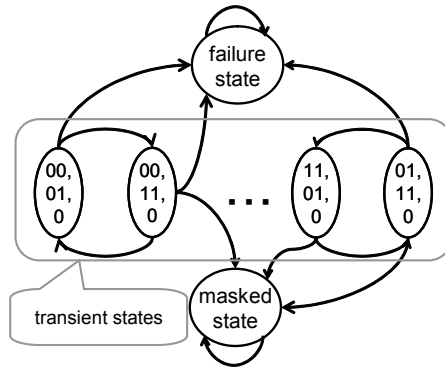


図 3: 吸収状態をもつマルコフモデル

集合である．また， $P_{tr_abs}(\pi, \pi_f)$ は状態 π を出発して π_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 初期状態確率の計算

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

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

ここで，正常回路の状態の集合を S とする． $s \in S$ の定常確率 $P_{steady}(s)$ は，正常回路のある時刻の状態が s である確率である．また， $e_{s\pi}$ は，ソフトウェアにより正常回路の状態 s から吸収状態を持つマルコフモデルの状態 $\pi \in \Pi$ に遷移する確率を表す．

正常回路の状態 s の定常確率 $P_{steady}(s)$ は以下のように計算できる．まず，正常回路において，状態 s_i から s_j に遷移する確率を $P_{tr}(s_i, s_j)$ 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．

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

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．

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

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．

2.4.1 吸収確率の計算

修正回路対の状態の集合を Π とする． $\pi_i \in \Pi$ を出発し failure 状態 π_f へ吸収される確率 $P_{tr_abs}(\pi_i, \pi_f)$ は以下のように表される．

$$P_{tr_abs}(\pi_i, \pi_f) = P_{tr}(\pi_i, \pi_f) + \sum_{\pi_j \in \Pi_{tmp}} P_{tr}(\pi_i, \pi_j) \cdot P_{tr_abs}(\pi_j, \pi_f) \quad (5)$$

Π_{tmp} は一時状態の集合である．各状態間の遷移確率が既知であれば， $P_{tr_abs}(\pi_i, \pi_f)$ を未知数とする連立方程式を解くことで吸収確率を計算できる．

2.5 処理の流れと問題点

本節では，厳密手法の処理の流れを説明する．厳密手法の処理の流れを，図 4 に示す．まず，正常回路の定常確率を求めるために，正常回路の遷移確率を求める必要がある．遷移確率を計算する方法としては論理シミュレーションなどが考えられる．到達不可能な状態の定常確率を求める必要はないため，遷移確率の計算は到達可能な状態に対してのみ行う．次に，正常回路の遷移確率から，定常確率を計算する．定常確率は，連立方程式を解くことで求める．次に，求めた定常確率から，初期状態確率を求める．本手法では， $e_{s\pi}$ は与えられるものとし，正常回路の各状態の定常確率との積をとることで初期状態確率を求める．次に，吸収確率計算のために，修正回路対の遷移確率を求める．正常回路の場合と同様，論理シミュレーションなどの方法が考えられる．次に，修正回路対の遷移確率から，吸収確率を求める．吸収確率は，連立方程式を解くことで求める．最後に，各初期状態の初期状態確率と吸収確率の積の総和を計算することで，ソフトエラーが発生し外部出

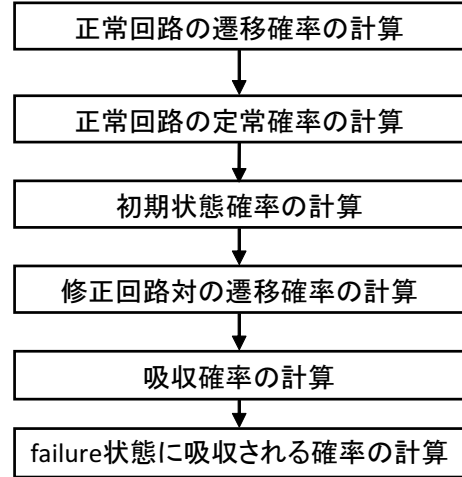


図 4: 厳密手法の処理の流れ

力に伝搬する確率を求める．本手法の問題点は，多大な実行時間を要するという点であり，特に，修正回路対の遷移確率の計算と，吸収確率の計算が実行時間の大部分を占める処理である．FF 数を k とすると，最悪の場合修正回路対の状態数は 2^{2k} である．修正回路対の遷移確率を論理シミュレーションにより求める場合，入力パターン数を I とすると，論理シミュレーションの実行回数は $2^{2k} \times I$ 回である．また，吸収確率の計算においては連立方程式を解く．連立方程式の解法の一つであるガウスの消去法を用いる場合，実行時間は元数の 3 乗に比例することが知られている．元数は修正回路対の状態数であるから，吸収確率の計算は，最悪の場合 $(2^{2k})^3 = 2^{6k}$ に比例した時間を要する．

3 高速化手法

前節で述べたように厳密手法においては，吸収確率を求めるための連立方程式を解く計算が実行時間の大部分を占める処理の一つである．文献 [10] において，連立方程式を解く計算を高速化する手法が提案されている．文献 [10] では，連立方程式の元数を削減する二つの手法が提案されている．一つは一度に解く連立方程式の元数を削減する手法，もう一つは連立方程式を解かずに元の値を求める手法である．しかし文献 [10] の手法では，同じ元の値を複数回の連立方程式で求めることがあるため，処理全体における元数が必ずしも削減できていない．また連立方程式を解かずに元の値を求める処理においても，可能なすべての元に対して処理を行っていない．そのため連立方程式を解く計算を高速化する余地がある．

本論文では，二つの連立方程式の元数を削減する手法を提案する．本論文で提案する手法は文献 [10] による手法より多くの連立方程式の元数の削減することができる．ここで本論文で削減をはかる連立方

程式の元数は二つの値をさす．一つは処理中複数回解かれる連立方程式の元数の合計である．もう一つは一度に解く連立方程式の最大の元数である．

3.1 拡大 pre-failure 状態の削除

文献 [10] で、pre-failure 状態とは、一時状態のうち、確率 1 で failure 状態へ遷移する状態と定義されている．pre-failure 状態の次状態は failure 状態のみであり、pre-failure 状態から一時状態への遷移確率は 0 であることと、式 (5) から、ある pre-failure 状態 π_i からの吸収確率 $P_{tr_abs}(\pi_i, \pi_f)$ は 1 である．このため吸収確率を求める連立方程式において、既知である pre-failure 状態からの吸収確率は元として持つ必要はない．そのため連立方程式の元数を削減することができる．

ここで pre-failure 状態が既知である場合を考える．failure 状態と一つもしくは複数の pre-failure 状態のみに遷移する一時状態があったとする．この一時状態は確率 1 では次の時刻に failure 状態に遷移しないため、pre-failure 状態ではない．しかし一時状態から、1 時刻後には failure 状態と pre-failure 状態に遷移し、2 時刻後には、pre-failure 状態から必ず failure 状態に遷移する．つまりこの一時状態も 2 時刻後に確率 1 で failure 状態に遷移する．式 (5) から、この一時状態 π_{epf} からの吸収確率は以下の式で表わされる．

$$P_{tr_abs}(\pi_{epf}, \pi_f) = P_{tr}(\pi_{epf}, \pi_f) + \sum_{\pi_j \in \Pi_{pf}} P_{tr}(\pi_{epf}, \pi_j) \cdot P_{tr_abs}(\pi_j, \pi_f) \quad (6)$$

ここで $\Pi_{pf} \subseteq \Pi_{tmp}$ は pre-failure 状態の集合である．pre-failure 状態の定義より、pre-failure 状態から failure 状態への吸収確率 $P_{tr}(\pi_{epf}, \pi_j)$, $\pi_j \in \Pi_{pf}$ は 1 と既知である．このため式 (6) の右辺には未知数が存在しない．また一時状態 π_{epf} からの遷移確率の総和は 1 である．よって $P_{tr_abs}(\pi_{epf}, \pi_f)$ は、連立方程式を解くことなく、1 と求めることができる．

今回連立方程式を解くことなく一時状態の吸収確率 1 が求められた理由について考える．この一時状態は、failure 状態、pre-failure 状態にのみ遷移する状態、つまり、すべての遷移先の状態の吸収確率が既知である状態である．ある一時状態からのすべての遷移先の状態の吸収確率が既知であれば、式 (5) より、その状態の吸収確率は遷移確率とその吸収確率の積の総和と求めることができる．この一時状態を拡大 pre-failure 状態と定義する．

また拡大 pre-failure 状態は再帰的に定義される．状態の吸収確率が既知である状態は、failure 状態、pre-failure 状態および拡大 pre-failure 状態である．ある一時状態 π_a と π_b があり、 π_a は failure 状態と

pre-failure 状態にのみ遷移する状態、 π_b は π_a にのみ遷移する状態であるとする．一時状態 π_b は一時状態 π_a の吸収確率が既知でないため、この段階では拡大 pre-failure 状態ではない．しかし π_a は吸収確率が既知である状態、failure 状態と pre-failure 状態にのみ遷移する状態であるため、拡大 pre-failure 状態であることがわかり、吸収確率も連立方程式を解くことなく求めることができる． π_a が拡大 pre-failure 状態であることがわかった後に、一時状態 π_b について考える．唯一の遷移先の一時状態 π_a は拡大 pre-failure 状態と判明しているため、一時状態 π_b も拡大 pre-failure 状態であることがわかり、吸収確率が連立方程式を解くことなく求めることができる．

拡大 pre-failure 状態とはすべての遷移先の吸収確率が既知である一時状態である．このためすべての pre-failure 状態は拡大 pre-failure 状態に含まれる．よって、拡大 pre-failure 上体の数は pre-failure 状態の数以上である．拡大 pre-failure 状態を求めることによって、吸収確率を求める連立方程式の元数をより多く削減することができる．

3.2 強連結成分を用いた状態集合の分割

前節で示したように、ソフトウェアが発生し外部出力に伝搬する確率は、ある初期状態の初期状態確率と、その初期状態からの吸収確率の積の総和をとることにより求められる．初期状態からの吸収確率は、連立方程式を解くことで求められるが、この際に、取りうる全ての状態からの吸収確率を連立方程式の元として持つ必要はない．ある状態 π_i からの吸収確率を求めるためには、ある状態 π_i からの到達可能状態のみを元とする連立方程式を解けばよい．また既にその状態からの吸収確率が既知である場合は、その状態からの吸収確率をあらかじめ定数とした連立方程式を解けばよい．

ここで文献 [10] では、一度に解く連立方程式の元数を削減するために、初期状態からの到達可能状態の吸収確率のみを元としている．この文献 [10] の手法では、複数の初期状態から到達可能な一時状態の吸収確率は複数の連立方程式の元とされている．これは同じ一時状態の吸収確率を複数回求めており、連立方程式の元数を不必要に増加させている．また必ずしも初期状態から到達可能な状態の吸収確率を元とする連立方程式とすることが最小の状態集合の分割とはならない．

そこで本論文では一度に解く連立方程式の元数を最小にすることを考える．修正回路対中の強連結成分を構成する状態の吸収確率は連立方程式を解かなければ求めることができない．つまりある状態から到達可能な状態の集合に強連結成分が含まれないのであれば、その集合に含まれる状態はすべて拡

表 1: 連立方程式の元数

bench	#input	#FF	base	文献 [10]		proposed	
				max	sum	max	sum
s298	3	14	6,322	1,102	4,144	160	1,962
s344	9	15	678,160	14,125	41,590	1	33,145
s382	3	21	1,502,857	642,249	1,738,535	54,580	620,650

張 pre-failure 状態となり，連立方程式を解くことなく，その状態からの吸収確率を求めることができる．よって修正回路対の各状態の failure 状態への吸収確率を効率よく求めるためには，failure 状態に近い状態を含む強連結成分の状態集合の吸収確率を元とする連立方程式から順に解いていけばよい．連立方程式を解くことによって得られた，状態の吸収確率を代入することによって，新たに拡大 pre-failure 状態となった状態の吸収確率を求める．さらに failure 状態に近い状態を含む強連結成分の状態集合を求める．これを繰り返す．この手法により，一度に解く連立方程式の元の数は最も大きい強連結成分に含まれる状態数となり，連立方程式を解く状態数の総和は，すべての強連結成分に含まれる状態数となる．

4 実験

本節で示したの高速化手法の効果を確認するための実験を行った．高速化手法を C++ 言語により実装した．実験に用いた計算機は，CPU が Intel Xeon 3.3GHz，メモリが 32GB 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の順序回路の一部を用いた．なお，ソフトエラーは FF でのみ発生し，また，failure 状態，masked 状態に吸収されるまでの間に再び発生することはないと仮定して実験を行った．なお，遷移確率の計算は全入力パターンを用いた論理シミュレーションにより行い，各入力パターンは一様に現れると仮定した．

表 1 に文献 [10] との連立方程式の元の数を比較した結果を示す．表 1 では，#input は外部入力数，#FF は FF 数，base はすべての高速化手法を用いなかった場合の結果を，文献 [10] は文献 [10] の結果を，proposed は本論文の提案手法を用いた結果を，max は分割した連立方程式の最大の元数を，sum は連立方程式の元数の総和をそれぞれ示す．表 1 に示すとおり，本手法によって連立方程式の最大の元数を少なくとも 1/6 以下に削減することができた．特に s344 では最大数が 1 であるため，連立方程式を解くことなく，吸収確率を求めることができた．また連立方程式の減数の総和も最大 2.8 倍削減することができた．

5 まとめ

本稿では，順序回路のソフトエラー耐性評価の高速化手法として，拡大 pre-failure 状態と強連結成分

の特定による連立方程式の減数の削減手法を提案した．拡大 pre-failure 状態は，連立方程式を解く際に必要のない状態を削除する手法である．強連結成分に基づく状態集合の分割は，連立方程式の計算を複数回に分けて行うことで一度に扱う状態数を削減する手法である．従来手法と比較して元数を少なくとも 1/6 に削減した．特に s344 では最大の元数を 1 にすることができた．しかしながら，吸収確率の計算以外の処理がボトルネックとなっている回路に対しては，全体の実行時間はさほど削減できない．よって，実行時間のボトルネックの一つである遷移確率の計算の高速化が今後の課題として挙げられる．

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．

参考文献

- [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] J.P. Hayes, I. Polian, and B. Becker. “An Analysis Framework for Transient-Error Tolerance”. In Proc. of the VLSI Test Symposium 2007 (2007).
- [8] N. Miskov-Zivanov and D. Marculescu. “Modeling and Optimization for Soft-error Reliability of Sequential Circuits”. IEEE Trans. Computer-Aided Design., Vol. 27, No. 5, pp. 803.816, May 2008.
- [9] 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，pp. 121.126，2009.
- [10] 赤峰悠介，吉村正義，松永裕介．“順序回路のソフトエラー耐性評価手法の状態数削減による高速化”．電子情報通信学会技術研究報告，VLD2009-126，Vol.109，No.462，pp.163-168，Mar. 2010.