An Evaluation Method for Soft Error Tolerance of Sequential Circuits with Markov Model

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

出版情報：DA Symposium. 2009, pp.121-126, 2009-08
バージョン：
権利関係：

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

赤峰　悠介 †　　吉村　正義 ‡　　松永　裕介 ‡

† 九州大学 大学院システム情報科学府
‡ 九州大学 大学院システム情報科学研究院

概要 LSI のソフトエラーに対する耐性の低下が問題となっている．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が，順序回路を対象とした評価手法は確立されていない．本稿では，マルコフモデルを用い順序回路のソフトエラー耐性を評価する手法を提案する．提案手法は，ソフトエラー発生後の状態遷移の振る舞いを解析し，エラーが外部出力に伝搬する確率を計算するものである．また，提案手法を用いいくつかのベンチマーク回路のソフトエラー耐性の評価を行う．

An Evaluation Method for Soft Error Tolerance of Sequential Circuits with Markov Model

Yusuke Akamine†, Masayoshi Yoshimura‡, and Yusuke Matsunaga‡

†Graduate School of Information Science and Electrical Engineering, Kyushu University
‡Faculty of Information Science and Electrical Engineering, Kyushu University

Abstract A lowering of tolerance for the soft error of the LSI becomes the problem. Soft error tolerance evaluation method is necessary by the logic design that considered soft error tolerance, but the evaluation method for sequential circuits is not established. This paper proposes an evaluation method for soft error tolerance of sequential circuits. In our approach, we analyzes the behavior of the state transition after a soft error occurs, and calculates the probability that the error propagates to external outputs. In addition, we evaluate the soft error tolerance of some benchmark circuits with proposal method.

1　はじめに

近年，トランジスタの微細化に伴い，LSI のソフトエラーに対する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より回路の一時的な誤動作が引き起こされることである．メモリ回路においては，中性子の衝突によりメモリセルに保持している値が反転する現象が起こる．一方，論理回路では，FF(フリップフロップ)，もしくは論理ゲートにおいてソフトエラーが発生する．FF では中性子の衝突により保持している値の反転が起き，また，論理ゲートにおいては出力値が反転するパルスが発生する．メモリ回路においては，ECC(誤り訂正符号) [9] などを用いることで比較的低コストで有効な対策を施せるのに対し，論理回路のソフトエラーは十分な対策が確立されていない [8]．今後のさらなる微細化に伴う臨界電荷量の低下により，論理回路のソフトエラーがより深刻な問題となる可能性がある [6].

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用いるのが一般的である．組み合わせ回路を対象とした評価手法はいくつか提案されている [2][3]．これらの手法は，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 FF にエラーが取り込まれる確率を計算するものである．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．し

たがって，既存の評価手法を順序回路に対し用いるのは適切でなく，順序回路に対する評価手法は確立されていないといえる．

そこで，本稿では，順序回路のソフトエラー耐性を評価する手法を提案する．提案手法は，ソフトエラー発生後の回路の振る舞いをマルコフモデルを用いモデル化し，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ものである．また，提案手法を用い，有限状態機械から合成した回路のソフトエラー耐性を評価する実験を行った．実験の際，論理合成の処理がソフトエラー耐性に与える影響を調べるため，二通りの方法で合成を行った回路を用いた．実験結果より，ベンチマークによりソフトエラー耐性に差があることがわかった．また，同じ有限状態機械から合成された回路でも，合成の方法によりソフトエラー耐性に違いがあることがわかった．

本稿の構成を以下に示す．まず，2 節で順序回路のソフトエラーについて述べる．次に 3 節で提案手法について述べる．また，4 節でソフトエラー耐性の評価実験について述べ，5 節で本稿をまとめる．

2　順序回路におけるソフトエラー

順序回路とは，出力が現在の入力値だけでは定まらず，過去の入力値に依存する論理回路のことである．すなわち，過去の入力値に依存する情報を内部に記憶しており，出力値は現在の入力値と記憶している情報によって定まる．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．

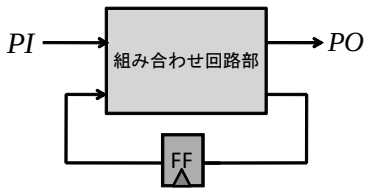

図 1: 順序回路の構成例

順序回路におけるソフトエラーとは，中性子の衝突に起因する論理ゲートでのパルスの発生，または，FF に保持されている値の反転のことを指す．しかし，発生した全てのソフトエラーが，回路としてのエラーにつながるわけではない．ソフトエラーは，その影響が回路の外部出力に伝搬して初めて回路としてのエラーとなる．そのため，論理回路のソフトエラー耐性は，ソフトエラーが発生し外部出力に伝搬する確率を指標として評価されるのが一般的である．既存の論理回路のソフトエラー耐性評価手法 [2][3] は，対象の回路を組み合わせ回路とみなし評価を行う．すなわち，FF を回路の外部であるとみなし，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 FF にエラーが取り込まれる確率を計算するものである．しかしながら，順序回路においては，論理ゲートからのパルスが FF に伝搬する，または，FF で値の反転が起き，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．したがって，既存の評価手法を順序回路に対し用いるのは適切ではない．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ためには，エラーが発生したクロックサイクルのみに着目するのではなく，エラー発生以後の状態遷移の振る舞いを調べる必要があると考えられる．

3　提案手法

本節では，順序回路のソフトエラー耐性評価手法を提案する．提案手法は，ソフトエラー発生後の回路の振る舞いをマルコフモデルを用いモデル化し，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ものである．

3.1　計算に用いるモデル

図 1 で示した順序回路のソフトエラー耐性を評価するとする．本手法では，回路のモデルとして，評価対象の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マルコフモデルにより表す．

3.1.1　回路モデル

本稿では，図 2 のような回路を回路対と呼ぶ．図 2 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の直前までは正常回路と等しい振る舞いをする．このような回路対を用いる理由は，正常な振る舞いとエラーを含んだ振る舞いの比較を行うためである．正常回路と故障回路の出

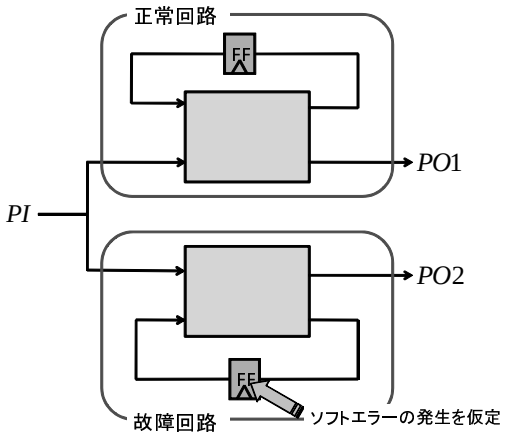

図 2: 回路対

力値が異なるとき，エラーは外部出力に伝搬したといえる．

3.1.2 マルコフモデル

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 T，ソフトエラー発生後の状態の集合を Π とする．T の状態から Π の状態への遷移は，ソフトエラーの発生によってのみ起こる．

本手法では，ソフトエラー発生後の状態遷移の振る舞いを，吸収的マルコフモデルにより表現する．確率 1 で自分自身へ遷移する状態を吸収状態と呼ぶ．どの状態から出発してもいつかはいずれかの吸収状態に到達するマルコフモデルを吸収的マルコフモデルと呼ぶ [10]．ソフトエラー発生後の回路は，一定時刻経過後に,「エラーが外部出力に伝搬する」もしくは,「正常な状態に戻る」かのいずれかである．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以後の状態

correct 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

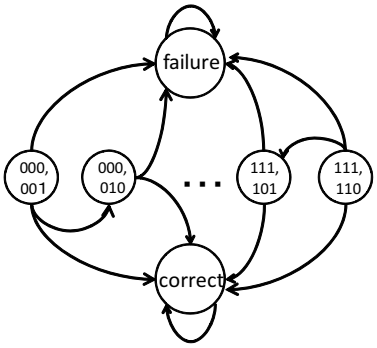

図 3: 吸収的マルコフモデル

図 3 は，ソフトエラー発生後の状態遷移を吸収的マルコフモデルで表わした例である．図 3 において,「failure」,「correct」は，failure 状態，correct 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吸収的マルコフモデルで表わしたとき，エラーが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．

3.2 表記

- T：ソフトエラー発生前の状態の集合
- Π：ソフトエラー発生後の状態の集合
- Π_{abs}：吸収状態の集合
- Π_{tmp}：一時状態の集合
- π_f：failure 状態
- $r_{\tau_i\tau_j}$：$\tau_i \in T$ から $\tau_j \in T$ への遷移確率
- $e_{\tau\pi}$：ソフトエラーにより $\tau \in T$ から $\pi \in \Pi$ に遷移する確率
- $p_{\pi_i\pi_j}$：$\pi_i \in \Pi$ から $\pi_j \in \Pi$ への遷移確率
- $q_{\pi\pi_a}$：$\pi \in \Pi$ を出発し $\pi_a \in \Pi_{abs}$ に吸収される確率
- $P_{abs}(\pi_a)$：$\pi_a \in \Pi_{abs}$ に吸収される確率
- $P_{init}(\pi)$：$\pi \in \Pi$ の初期状態確率
- $P_{steady}(\tau)$：$\tau \in T$ の定常確率

3.3 確率の計算

ソフトエラーが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．π_f の吸収確率とは，いずれかの状態から出発して π_f に吸収される確率である．吸収確率 $P_{abs}(\pi_f)$ は，以下の式で表わされる．

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

式 (1) において，$P_{init}(\pi)$ は状態 π の初期状態確率であり，ソフトエラー発生直後の状態が π である確率である．$q_{\pi\pi_f}$ は，状態 π を出発して π_f に吸収される確率である．また，本稿では $q_{\pi_i\pi_j}$ を (i,j) 要素とする行列 Q を，吸収確率行列と呼ぶ．以下に，初期状態確率及び，吸収確率行列の計算方法について述べる．

3.3.1 初期状態確率の計算

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

$$P_{init}(\pi) = \sum_{\forall \tau \in T} P_{steady}(\tau) \cdot e_{\tau\pi} \tag{2}$$

$\tau \in T$ の定常確率 $P_{steady}(\tau)$ は，ソフトエラー発生前のある時刻の状態が τ である確率である．また，$e_{\tau\pi}$ は，ソフトエラーにより状態 τ から状態 $\pi \in \Pi$ に遷移する確率を表す．$e_{\tau\pi}$ は，ソフトエラーが発生する確率から計算可能である．

以下に，$P_{steady}(\tau)$ の計算方法を述べる．まず，ソフトエラー発生前の遷移確率 $r_{\tau_i\tau_j}$ を (i,j) 要素とする行列 R を遷移確率行列と呼ぶ．また，ソフトエラー発生前の状態の集合 T を，$T = \{\tau_1, \tau_2, \cdots, \tau_n\}$ とする．このとき，定常確率と遷移確率行列の間には以下の式が成り立つ．

$$\begin{bmatrix} P_{steady}(\tau_1) \\ P_{steady}(\tau_2) \\ \vdots \\ P_{steady}(\tau_n) \end{bmatrix} = R^T \begin{bmatrix} P_{steady}(\tau_1) \\ P_{steady}(\tau_2) \\ \vdots \\ P_{steady}(\tau_n) \end{bmatrix} \tag{3}$$

式 (3) において，R^T は，R の転置行列を表す．また，定常確率は以下の式を満たす必要がある．

$$\sum_{\forall \tau \in T} P(\tau) = 1 \tag{4}$$

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立てられる．よって，定常確率は，遷移確率を求めることにより計算できる．遷移確率の算出は，ランダムサンプリングによる論理シミュレーションにより行う．

3.3.2 吸収確率行列の計算

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

$$q_{\pi_i\pi_j} = \sum_{\forall \pi_k \in \Pi_{tmp}} p_{\pi_i\pi_k} q_{\pi_k\pi_j} + p_{\pi_i\pi j} \tag{5}$$

ここで，ソフトエラー発生後の遷移確率行列 M を考える．M は，以下のように小行列 G, H, O, I を用いて表される．

$$M = \begin{pmatrix} G & H \\ O & I \end{pmatrix} \tag{6}$$

小行列 G は Π_{tmp} の状態から Π_{tmp} の状態への遷移確率行列，小行列 H は Π_{tmp} の状態から Π_{abs} の状態への遷移確率行列を表す．また，I は単位行列，O はゼロ行列である．

式 (5) と式 (6) の小行列より，吸収確率行列 Q は，式 (7) のように表され，変形により式 (8) が得られる．

$$Q = GQ + H \tag{7}$$

$$Q = (I - G)^{-1} H \tag{8}$$

式 (8) より，G, H を求めることで，Q は計算可能であることがわかる．つまり，各状態間の遷移確率を求めればよい．初期状態確率の計算と同様に論理シミュレーションを用いて遷移確率を計算する．論理シミュレーションの際，正常回路と故障回路の出力値が異なる場合 failure 状態への遷移とみなす．また，正常回路と故障回路の FF の値が等しい場合は correct 状態への遷移とみなす．ただし，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 k とすると，2^{2k} となる．そのため，FF 数の多い回路に対して遷移確率行列を求めるのは難しい．しかし，ソフトエラー発生後の状態のうち到達可能であるのはごく一部である．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．

4 実験

本節では，提案手法を用いたソフトエラー耐性の評価実験について述べる．実験の目的は，ソフトエラー耐性が回路によってどの程度違うのかを調べることである．加えて，本実験では，論理合成の処理がソフトエラー耐性に与える影響の評価を行う．今回は，論理合成の処理の一部である状態符号化に着目する．二通りの状態符号化手法を用いて合成した回路のソフトエラー耐性を評価する．

4.1 準備

C++言語により提案手法を実装し，実験を行った．実験に用いたマシンの CPU は Intel Xeon 3.3GHz である．ベンチマークとして，MCNC ベンチマークセット [7] の有限状態機械を用いた．各ベンチマークを，状態符号化ツール JEDI および one-hot encoding により符号化した．なお，符号化後の有限状態機械には，状態遷移の定義されていない状態が存在する．通常の動作ではこれらの状態に到達する可能性はないが，エラーを含む遷移ではこれらの状態に到達する可能性がある．提案手法を用いて評価を行うためには，それらの状態にも何らかの状態遷移が定義されている必要がある．そのため，符号化後の有限状態機械にではなく，合成後の回路に対する評価を行った．符号化後の合成の処理には，論理合成ツール SIS[1] を用いた．

状態符号化ツール JEDI では，回路面積を考慮した最短符号化が用いられている [4]．最短符号化とは，できるだけ短い符号長で符号を割り当てることである．有限状態機械の状態数を m とすると，JEDI を用い符号化された回路の FF 数は $\lceil \log m \rceil$ となる．また，one-hot encoding は，各符号の 1 ビットだけが 1 になるように符号を割り当てる手法である．one-hot encoding を用いて合成された回路の FF 数は m である．

本実験では，ソフトエラーは FF においてのみ起き複数の FF で同時には起きないとする．また，ソフトエラーが発生したタイミングによらず，論理的に値が伝搬しさえすればエラーは伝搬するとする．各 FF のソフトエラー発生確率を occur rate と呼び，occur rate $= 1.0 \times 10^{-10}$ とする．ソフトエラーが発生し外部出力に伝搬する確率を failure rate と呼ぶ．また，エラーが外部出力に伝搬する確率を propagation rate と呼ぶ．FF 数を k とすると，propagation rate = failure rate/(occur rate $\cdot k$) である．

4.2 実験結果

実験結果を表 1 に示す．表 1 において，#state は有限状態機械の状態数，#FF は FF 数を示す．failure rate は回路のエラー率を示す．また，propagation rate は，エラーが外部出力に伝搬する確率である．time は実行時間である．JEDI および one-hot は符号化手法を表す．

まず，failure rate の項目について，JEDI の結果に着目する．値が最小のもので約 2×10^{-11}, 最大のものでは約 6×10^{-10} である．ベンチマークにより，結果に大きな違いがあることがわかる．また，JEDI と one-hot の結果を比較すると，JEDI は全てのベンチマークにおいて one-hot よりも小さい値となっており，10 倍以上の大きな差がみられるものもある．これは，符号化手法の違いにより FF 数が異なることの影響が顕著に現れたためであると考えられる．

次に，propagation rate の項目に着目する．JEDI では，一部のベンチマークを除き，0.3〜0.9 程度に分布していることがわかる．一方，one-hot は，ほとんどのベンチマークで JEDI より大きな値となっており，propagation rate がほぼ 1 となっているベンチマークも多い．

最後に，提案手法の実行時間について述べる．規模の大きなベンチマークに対しては，数千秒から 1 万秒程度の時間がかかっている．この実行時間の大部分を占めるのは回路対の到達可能状態の列挙及び遷移確率の計算の処理であり，より大規模な回路に対して提案手法を用いるためには，これらの処理の高速化が必要であると考えられる．この高速化に関しては，BDD(Binary Decision Diagram: 二分決定グラフ)[5] を用いた効率的な処理が効果的であると考えられる．

実験結果より，ベンチマークにより，ソフトエラー耐性に大きな差が生じることがわかった．また符号化手法の違いによる影響も大きいことがわかった．符号化手法の違いは，FF 数の違いによる影響だけでなく，propagation rate にも影響を与えている．

表 1: 実験結果

benchmark	# state	# FF		failure rate[$\times 10^{-10}$]		propagation rate		time[s]	
		JEDI	one-hot	JEDI	one-hot	JEDI	one-hot	JEDI	one-hot
lion	4	2	4	1.625	3.750	0.813	0.938	0.01	0.01
mc	4	2	4	2.000	4.000	1.000	1.000	0.02	0.03
tav	4	2	4	2.000	4.000	1.000	1.000	0.07	0.11
train4	4	2	4	1.200	3.333	0.600	0.833	0.01	0.01
dk15	4	2	4	1.469	4.000	0.734	1.000	0.03	0.03
bbtas	6	3	6	1.999	6.000	0.666	1.000	0.02	0.03
beecount	7	3	7	0.704	6.375	0.235	0.911	0.08	0.15
dk27	7	3	7	2.619	7.000	0.873	1.000	0.04	0.01
ex6	8	3	8	2.226	7.750	0.742	0.969	0.56	1.21
dk17	8	3	8	2.712	8.000	0.904	1.000	0.04	0.07
ex5	9	4	9	1.146	4.232	0.287	0.470	0.07	0.10
ex7	10	4	10	1.053	2.561	0.263	0.256	0.08	0.13
bbara	10	4	10	2.224	10.000	0.556	1.000	0.29	0.57
ex3	10	4	10	0.216	2.109	0.054	0.211	0.08	0.12
opus	10	4	10	1.631	4.914	0.408	0.491	0.80	1.17
ex4	14	4	14	1.896	5.625	0.474	0.402	2.69	6.07
dk512	15	4	15	3.583	15.000	0.896	1.000	0.06	0.19
bbsse	16	4	16	2.864	14.500	0.716	0.906	6.99	25.46
cse	16	4	16	1.367	15.712	0.342	0.982	14.43	30.11
keyb	19	5	19	1.988	19.000	0.398	1.000	63.53	57.46
s1	20	5	20	4.022	20.000	0.804	1.000	84.74	190.10
ex1	20	5	20	1.500	14.625	0.300	0.731	551.86	965.11
styr	30	5	30	1.925	29.341	0.385	0.978	620.43	1625.17
sand	32	5	32	4.883	31.252	0.977	0.977	3408.02	10704.10
planet	48	6	48	5.971	48.000	0.995	1.000	673.74	2608.48

5 おわりに

本稿では，順序回路のソフトエラー耐性評価手法を提案した．また，提案手法を用い，二通りの状態符号化手法により合成されたベンチマークのソフトエラー耐性の評価を行った．実験結果より，ベンチマークによりソフトエラー耐性に大きな差があること，および，状態符号化手法がソフトエラー耐性に影響を与えることを明らかにした．

今後の課題を以下に述べる．提案手法の問題点は実行時間の大きさである．実行時間の大部分を占めるのは，回路対の到達可能状態の列挙及び遷移確率の計算であるため，これらの処理の高速化が課題となる．また，実験結果より，論理合成の処理がソフトエラー耐性に影響を与えることがわかった．今後は，ソフトエラー耐性を考慮した論理合成手法の検討を行っていく予定である．今回の実験では，全ての状態に対し状態遷移を定義する必要性から，既存の合成ツールによって回路の合成を行った．しかし，それらの状態に対しどのような状態遷移を割り当てるかは任意である．そのため，それらの状態にソフトエラー耐性を考慮した状態遷移を割り当てることで，合成後の回路のソフトエラー耐性を高めることができる可能性があると考えられる．

参考文献

[1] E. M. Sentovich et al. “SIS: A system for Sequential Circuit Synthesis”. *Memorandum No. UCB/ERL M92/41*, 1992.

[2] 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.

[3] Yan Lin and Lei He. “Device and Architecture Concurrent Optimization for FPGA Transient Soft Error Rate”. In *Proc. of ICCAD*, pp. 194–198, 2007.

[4] S. Devadas P. Ashar and A.R. Newton. *“Sequential Logic Synthesis”*. Kluwer Academic Publishers, 1992.

[5] R.E.Bryant. “Graph-Based Algorithm for boolean function manipulation”. *IEEE Trans. on Compute*, Vol. C-35, No. 8, pp. 677–691, 1986.

[6] 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.

[7] S Yang. “Logic Synthesis and Optimization Benchmarks User Guide: Version 3.0”. 1991.

[8] 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.

[9] 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.

[10] 森村英典, 高橋幸雄. “マルコフ解析”. 日科技連, 1979.