

## メルセンヌ・ツイスタのための最小テスト系列組込み法 Built-in minimal test sequence for Mersenne Twister

渡邊 未来<sup>1)</sup>      山口 賢一<sup>1)</sup>      岩田 大志<sup>1)</sup>

Watanabe Miku Yamaguchi Ken'ichi Iwata Hiroshi

### 1 研究背景

半導体プロセスの微細化に伴い LSI は大規模化・高性能化が進んでいる。その一方で LSI のテストは困難となっている。製造された LSI の中に不良品が含まれることは避けられず、高い品質と信頼性を確保するためには出荷前に LSI のテストが欠かせない。そこで、LSI の故障を効率よく検出するテスト手法が必要となる。

LSI のテスト手法として、乱数発生器に疑似乱数であるメルセンヌ・ツイスタ (MT: Mersenne Twister) をテストパターン発生器として利用する方法がある。MT は非常に長い周期  $2^{19937} - 1$  と高い乱数性を持っているため、乱数間の相関は無視できるほど小さい。そのため、MT は組合せ回路だけでなく順序回路に対して高い故障検出率を得られることが示されている [1]。

MT をテストパターン発生器としてハードウェア化する場合、MT 内部に故障が存在することを想定し、それに対するテストを行わないと信頼性が損なわれる要因となる。MT に基づくテストパターン発生器のためのテスト系列として、長尾ら [2] は LFSR (Linear Feedback Shift Register) が生成するランダムパターンをテスト系列として用いる手法を提案している。この乱数生成によるテスト系列は、LFSR の信頼性を保証する手法 [3] を前提としており、結果として故障検出率 100% となるテスト系列であることが確認された。

しかし、乱数をテスト系列に用いることは、一つの故障に複数のテストパターンが割り当てられる可能性があり、特性多項式や初期値による必要テストパターン数のバラつきも無視できない。本稿では、MT に基づく乱数生成器に対して故障検出率が 100% になる最小テスト系列を提案する。最小テスト系列を印加することで、同じ故障に重複してテストパターンが割り振られるなどのテスト実行時間の無駄を防ぎ、テストコストの削減を目指す。また、テストコストを削減する手法の一つとして、テスト機能をもつ回路を LSI 内部に搭載する組込み自己テスト (Built-In Self Test) がある。BIST の手法の一つに、テストパターン発生器をテスト対象回路に直接接続する方法がある。

本稿で提案する最小テスト系列と LFSR で生成された乱数テスト系列のパターン数を比較評価する。また、それぞれの手法をハードウェアとして実現する場合の面積増加についても評価する。

本稿は、MT に基づく乱数生成器に対して故障検出率が 100% になる最小テスト系列を提案する。また、最小テスト系列を MT に基づく乱数生成器の BIST として実装する。結果として、LFSR で生成された乱数テスト系列と最小テスト系列のテストパターン数と面積を比較評価し、新たな MT 法の乱数生成器の故障検出方法を確立

する。

### 2 諸定義

#### 2.1 故障モデル

生成可能な全ての組合せのテストパターンを印加すれば、全ての故障を検出できる。しかし、テスト実行時間が膨大になるため現実的ではない。そこで、起こりうる欠陥の論理的振舞いをモデル化した故障モデル (Fault Model) によって故障を仮定し、その故障を検出するようなテストを考える必要がある。回路内に存在する故障はただ一つであると仮定する故障モデルを単一故障 (Single Fault)、複数個所での発生も許容する故障モデルを多重故障 (Multiple Fault) といい、他の故障モデルとともに付加的に考慮される。互いに関係のない箇所に同時に存在している複数の故障に対しては、いずれか 1 箇所の単一故障を検出できるテストパターンを用いれば故障は検出されるため、多重故障は単一故障に比べて検出が容易である。以上の理由から、単一故障が用いられることが多い [4]。本稿の故障モデルは単一故障を想定する。

#### 2.2 故障検出率

故障検出率とは、想定した故障モデルのうち、テストパターンの印加によってどれだけの故障を検出できているかという割合を示し、テストパターンの品質の評価指標として用いられる。故障検出率は、式 (1) によって求められる。

$$\text{故障検出率} = \frac{\text{テストによって検出できた故障数}}{\text{想定した全故障数}} \times 100[\%] \quad (1)$$

故障検出率が高いほどテストパターンの品質は良いといえる。なお、故障検出率は故障シミュレーションによって求められる [4]。

#### 2.3 故障シミュレーション

故障シミュレーションでは、故障回路と正常回路に対して、テストパターンを印加した場合の回路動作のシミュレーションを行い、出力値を比較することで検出できる故障を特定する。本稿では故障シミュレーションは Synopsys 社が提供する TetraMax を用いる。TetraMax は入力したテストパターンに対する故障検出の可否のみを出力する。これより、提案するテスト系列の故障検出率の評価を行う [5]。

#### 2.4 テストパターン

LSI のテストにおいて、回路に入力する入力値をテストパターンという。次にテスト生成について示す。LSI のテスト時に用いるテストパターンを求めることをテスト生成という。テスト生成では、テスト対象となる回路の機能や要素間の接続などの構造情報をもとにテストパターンを生成する。特定の故障を想定し、想定した故障を検出するテストパターンが存在するかを求め、存在するならばそのテストパターンを求めるアルゴリズムをテスト生成アルゴリズムという。テスト生成アルゴリズムは多く考案されており、ATPG (Automatic Test

1) 奈良工業高等専門学校情報工学科. Department of Information Engineering, National Institute of Technology (KOSEN), Nara College.

Pattern Generator) として実現されている。また、テストパターンに疑似乱数を用いる方法もあり、疑似乱数発生機構をハードウェアとして組込んでテストを行う組み込み自己テスト (Built-In-Self-Test) [6] 法も実用化されている。

## 2.5 ATPG

ATPG は対象となる論理回路のゲートレベルネットリスト、および検出したい故障のリストを入力すると、その故障を検出するためのテストパターンを自動的に生成する。ATPG は、故障検出率の高いテストパターン集合を、なるべく少ないテストパターン数で生成できる。現状の ATPG は組合せ回路に対しては効率よく 100% の故障検出率を得るテストパターンを生成できる。

## 2.6 疑似乱数を用いたテストパターン

BIST では、疑似乱数生成器によって発生させた疑似乱数をテストパターンとして用いる手法が一般的である。疑似乱数はハードウェアでも容易に生成できるため、高速なテストが可能である。疑似乱数を用いる手法は、テスト生成により求めた決定的テストパターンをメモリに蓄える手法と比較し、多量のメモリを必要としない。一方で、十分な故障検出率を得るためにはテストパターンの量が多くなるという特徴もある。

## 2.7 メルセンヌ・ツイスタ

メルセンヌ・ツイスタ (Mersenne Twister:MT) とは、松本ら [7] によって開発された疑似乱数生成アルゴリズムである。メルセンヌ・ツイスタは内部状態の大きさや周期が設定可能であり、様々な種類が存在する。本稿では、よく使われている生成法である MT19937 を用いる。MT19937 の特徴を以下に示す。

1. 周期がと非常に長い ( $2^{19937} - 1$ )
2. 高次元 (623 次元) に均等分布する
3. 疑似乱数の生成が高速である
4. 処理単位が 32bit (1 ワード) であり、ソフトウェア実装にも適している
5. 生成メモリ効率が良い

### 2.7.1 MT のハードウェア化

文献 [8] ハードウェア化する際の内部構造を図 1 に示す。

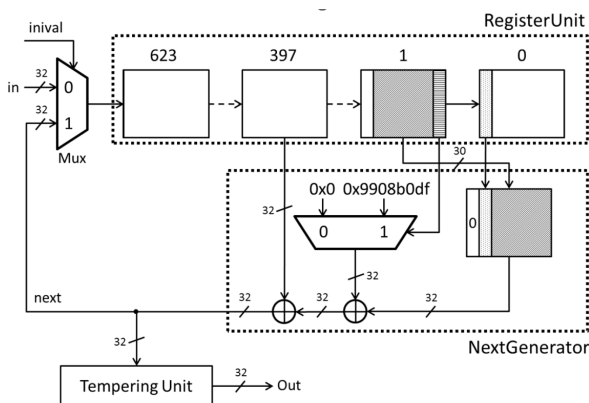


図 1 MT のハードウェア図 [3]

この回路はまず inival を 1 にして in から初期状態をワード 0、ワード 1、ワード 2、…ワード 623 の順で入

力していき clock の立ち上がりのタイミングでシフトが入力される。全ての初期状態の入力が終わったら inival を 0 として clock を印加すると 1 クロックごとに out から疑似乱数が 1 ワード (32bit) 出力される特徴がある。

表 1 MT の入出力一覧

名前	I/O	ビット数	備考
in	入力	32	初期状態入力
inival	入力	1	1 で初期状態入力モード
clock	入力	1	クロック入力
out	出力	32	疑似乱数出力

### 2.7.2 MT の回路構成

回路構成としては、主に Register Unit, NextGenerator Unit, Tempering Unit の 3 つの回路ユニットにより構成されている。表 3 に各ユニットの説明を示す。

表 2 MT の回路構成説明

<b>Register Unit</b>
内部状態を保存しておくレジスタ群である 全部で 624 ワード分のシフトレジスタが存在するが 最下位の 31bit は、疑似乱数の生成に関与していない
<b>NextGenerator Unit</b>
疑似乱数生成を行う主要部であり、(ワード 0,1,397) の 64bit の入力を受け取り、シフト演算や排他的論理和 をとり 32bit の出力を行う回路となっている。図 1 のように疑似乱数生成時には全ワード中特定の 3 ワード (0,1,397) の一部に依存している
<b>Tempering Unit</b>
出力疑似乱数列の分散をよりよくするために設け られたものである。32bit 入力に対してシフト演算と 排他的論理和を数回繰り返しビット列をかき混ぜる 組合せ回路になっている

最小テスト系列の提案では、これらの各 Unit 毎に着目してテスト系列の提案を行う。

## 2.8 LFSR

線形帰還シフトレジスタ (Linear Feedback Shift Register: LFSR) とは疑似乱数生成手法の 1 つである。4bit の LFSR 回路を図 2 に示す。

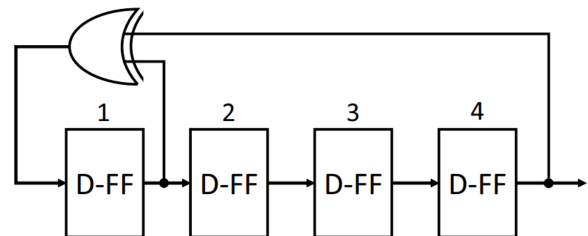


図 2 LFSR のハードウェア化

LFSR は D フリップフロップ (D-FF) と XOR のみで構成される。各 D-FF が保持している値が LFSR の出力値となる。出力ビットの一部の XOR を取った値を入力値としてフィードバックしているため、出力値は周期性を持つ。また、最適なビット列をフィードバックに用いる

ことにより、全ビットが 0 の状態以外の全ての状態を擬似乱数的に生成できる。XOR を取るビットが既約多項式であるならば、LFSR の周期は出力  $n$  ビットの場合  $2^n - 1$  となり、最大周期となる。しかしこの既約多項式は、最大 168bit までしか解明しておらず、MT 以上の周期を実現することはできない。

LFSR は BIST で用いられることが多いため、LFSR をハードウェア化する必要があるが、LFSR は D-FF と XOR のみで構成できるため、構造が簡単で回路面積が小さい特徴がある。

### 2.9 MT の BIST 評価

吉田ら [1] によって、MT によるテストパターン生成器を用いた出荷テストの評価のために、LFSR と比較実験が行われた。比較したポイントは信頼性(故障検出率)と実装コスト(面積増加)の 2 点である。信頼性に関する実験では、ITC' 99 ベンチマーク回路 8 つに対し、全ての回路において MT は LFSR より高い故障検出率が得られた。実装コストに関する実験では、MT を STUMPS アーキテクチャにより実装する場合でも、既存の LFSR の BIST アーキテクチャに比べて 1.9 倍程度の面積増加を許容すれば、TT800 を用いた STUMPS を実装できることが示された。STUMPS アーキテクチャは、BIST を実現するアーキテクチャの 1 つである。

### 3 提案する最小テスト系列

本稿で提案する最小テスト系列は MT が構成されている回路ごとに最小テストパターンを導きだし、最終的に 1 つの最小テスト系列集合としてまとめる。以下のステップによって MT に対する最小テスト系列を求める。

- ステップ 1: MT を各ユニットごとに分割
- ステップ 2: 組合せ回路部分に対する最小テストパターンを生成
- ステップ 3: テストパターン数の dont care 部に対する 0,1 割り当て処理
- ステップ 4: 入力マルチプレクサに対する処理の割り当て

#### 3.1 MT の分割

MT の回路部分は組合せ回路部分と順序回路部分に回路を分けて考えられる。

組合せ回路部分は NextGenerator Unit ・ Tempering Unit, 順序回路部分は Register Unit に分けられる。

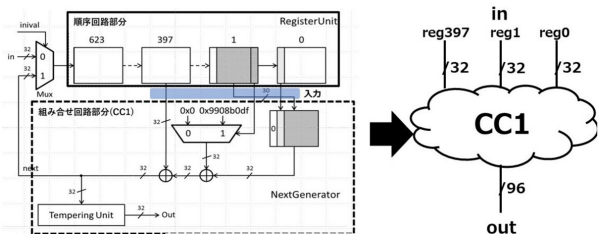


図 3 CC1 の回路図

#### 3.2 組合せ回路部分に対する最小テストパターン

組合せ回路部分は、ATPG を用いて最小テストパターンを生成するため、組合せ回路部分のみを図 3 のように抽出する。CC1 に ATPG を用いた出力結果を図 4 に示す。図 4 より、CC1 の故障を見つけるために、入力すべ

きテストパターンが 9 パターン生成された。左からワード 397,1,0 の順番に羅列されており、合計で 96bit 生成されている。

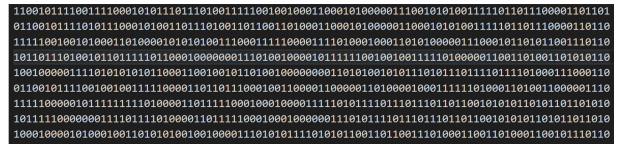


図 4 ATPG の出力結果

実際に in より入力する bit 数は 32bit のため図 5 のようにワード 0,1,397 ずつ抽出したテスト系列を作成し、最小テスト系列集合とする。

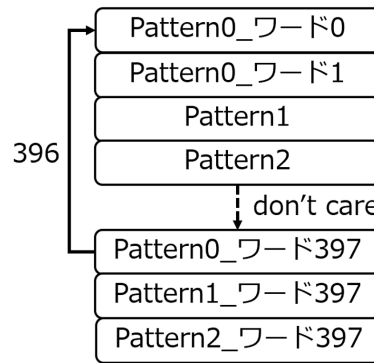


図 5 最小テスト系列集合

#### 3.3 dont care 部に対する 0,1 割り当て処理

図 5 の don't care の部分に以下で述べるテスト系列を加える。Register Unit 部分は 623 個のシフトレジスタで構成されているため、all0 のパターンと all1 のパターンを最小テスト系列に加える。また、主要回路部分以外の next 線についてもテストパターンを生成した。

next 線の場合、0 縮退故障・1 縮退故障が起こる可能性がある。それらを活性化させるため、以下のテストパターンを最小テスト系列に加える。

表 3 0,1 縮退故障のパターン生成

0 縮退故障
ワード 0,1,397 全て all0 のパターン生成する
1 縮退故障
表 5 のパターンを生成する

表 4 1 縮退故障時の next 線のパターン

ワード 0	[1xx... xxxx]
ワード 1	[x1X... XX10]
ワード 397	[100... XXX0]

この段階で故障シミュレーションを行った結果、故障検出率 100%にはならなかった。そのため、検出できなかった故障を活性化させるパターンを最小テスト系列に加える。検出できなかった故障場所は Mux の inival 線であった。以下で inival 線の故障の活性化パターンについて述べる。

### 3.4 入力マルチプレクサに対する処理

Mux の故障検出には図 6,7 の場合が考えられる. 図 6 では in が 1 で next が 0 であるテストパターンを生成する. 図 7 は in が 0 で next が 1 であるテストパターンを最小テスト系列集合に加える. 故障シミュレーションでは, (a)(b) の故障は片方が活性化されれば, もう片方の故障も検出できる故障であるため, どちらかをテストパターンとして加えればよい.

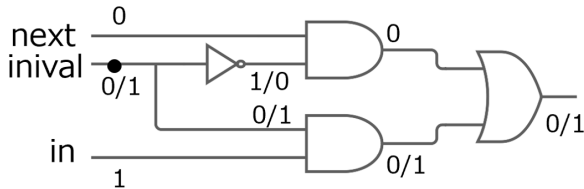


図 6 inival 線の故障 (a)

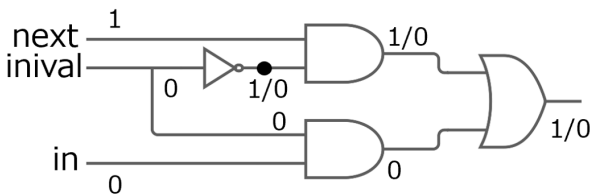


図 7 inival 線の故障 (b)

最終的に組合せ回路部分の入力を出力する必要があるため, 組合せ回路部分の 9 パターンを出力させるために, ワード 1 まで初期状態が入力された後に, 9 パターン出力パターンを入力する. 出力パターンはどのようなパターンでも良い. 作成したテスト系列のテスト数と入力制御信号を表 5 に示す. 生成したテストパターン数の合計は 632 パターンになる.

表 5 提案するテスト系列

reset	inival	テストパターン数	備考
0	1	1	リセット
1	1	410	初期状態入力
1	0	213	帰還入力
1	0	9	出力パターン

## 4 実験結果

生成したテスト系列で MT に基づく乱数発生器の故障シミュレーション用いた結果 632 パターンで故障検出率 100%を達成した. 比較対象として先行研究 [2] のランダムテスト系列を用いる. 表 6 より, ランダムテスト系列よりもテストパターン数を約 70%削減できた.

表 6 パターン数比較

ランダムテストパターン	生成テストパターン
1480	632

LFSR 回路と生成したテスト系列を格納した ROM を MT に組込んだ面積の比較を行う. 各回路を Synopsys Design Compiler で論理合成を行い, 面積を算出した.

Synopsys Design Compiler の面積評価は 1 つの NAND で 1area 換算で表示されている. 表 7 より, ROM 組み込んだ面積は LFSR 組み込み面積より, 約 1.04 倍増加したがほぼ同等であった.

表 7 total area

LFSR 組み込み面積	ROM 組み込み面積
180,863area	187,976area

## 5 結論

本稿では, MT をテストパターン発生器としてハードウェア化する場合, MT 自身に故障が存在することを想定したテストを行った. 提案手法のアイデアとしてあらかじめ MT に対する最小テストパターン集合を求めておき, テスト系列のドントケア部分に他のユニットのテスト系列を挿入する最小テスト系列生成手法を提案した. 結果として, LFSR が生成するランダムパターンをテスト系列として用いる手法よりもテストパターン数を 70%削減した 632 パターン数で故障検出率 100%を達成するテスト系列であることが確認された.

回路面積の比較評価では, 生成したテスト系列を格納した ROM を MT に組込んだ面積は, LFSR 回路を MT に組込んだ面積とほぼ同等であった. MT の回路面積が約 1.04 倍増加することが許容できるのなら, MT 自身の故障テスト方法として, あらかじめ生成したテスト系列を格納した ROM を BIST により実装することも有用であることを示した.

### 謝辞

本研究は, 東京大学 V D E C 活動を通して, 日本シノプシス合同会社の協力で行われたものである.

### 参考文献

- [1] 吉田拓弥, 里中沙矢香, 山口賢一, 岩田大志, “BIST 環境下におけるメルセンヌ・ツイスタアルゴリズムの評価”, 情報科学技術フォーラム一般講演論文集, Vol.14, No.1, C-018, pp.269-270, Sep. (2015).
- [2] Takuma Nagao, Ken' ichi Yamaguchi, and Hiroshi Iwata, “Test Plan For Detecting Mersenne Twister Faults In BIST”, The 23rd Workshop on Synthesis And System Integration of Mixed Information Technologies, R3-5, pp.148-149, May 2021.
- [3] Hideo Fujiwara, Katsuya Fujiwara, and Toshinori Hosokawa, “Universal Testing for Linear Feed-Forward/Feedback Shift Registers,” IEICE Trans. on Inf. and Syst., Vol. E103-D, No. 5, pp. 1023-1030, May 2020
- [4] 大和勇太, “VLSI 回路の高歩留り化のための誤テスト回避と高精度故障診断に関する研究”, 九州工業大学博士学位論文, 情工博甲第 242 号, 2017.
- [5] Hiroshi Iwata and Ken' ichi Yamaguchi, “A BIST Evaluation System using Automatic Test Pattern Generator”, 奈良工業高等専門学校 研究紀要 43 号, pp.45-46, 2007.
- [6] Miron Abramovici, Melvin A. Breuer, and Arthur D. Friedman, “Digital Systems Testing and Testable Design”, Wiley-IEEE Press, 1994.
- [7] M.Matsumoto and T.Nishimura, “Mersenne twister, A 623-dimensionally equidistributed uniform pseudorandom number generator”, ACM Trans. on Modeling and Computer Simulations, 1998.
- [8] 渡部信吾, 阿部公輝 “疑似乱数生成器 Mersenne Twister の VLSI 設計”, 社団法人 情報処理学会 研究報告, Vol.2005, No.41(CSEC-29), pp.13-18, 2005.