

卒業研究報告

題目

フィールドテストの故障検出強化のための疑似乱数生成法の調査と性能評価

指導教官

高橋寛教授

王森レイ講師

報告者

神崎 壽伯

令和 3 年 1 月 20 日提出

愛媛大学工学部情報工学科情報システム工学講座

目次

第 1 章	まえがき	1
第 2 章	準備	3
2.1	諸定義	3
2.2	スキャンベース BIST	4
2.3	マルチサイクルテスト	5
2.4	線形フィードバックレジスタ (LFSR)	5
2.5	メルセンヌ・ツイスタ	6
第 3 章	評価方法	8
3.1	乱数性の評価方法	8
3.1.1	等出現性の評価方法	8
3.1.2	無規則性の評価方法	9
3.2	故障検出率の評価方法	11
第 4 章	評価実験及び考察	12
4.1	乱数性の評価実験	12
4.1.1	等出現性の評価実験	12
4.1.2	無規則性の評価実験	13
4.2	故障検出率の測定及び評価	15
4.2.1	スキャンベース BIST での故障検出率比較	15
4.2.2	マルチサイクルテストでの故障検出率比較	17

4.3	全体の考察	21
第5章	あとがき	22
	謝辞	23
	参考文献	24

第 1 章

まえがき

近年の LSI (Large Scale Integrated circuit) の微細化加工技術の発展に伴い、LSI の高集積化・高速化が進んでいる。そのため、LSI 製造後の出荷テストにおいて高品質かつ低コストのテストを行うことが困難になっている。また LSI の経年劣化を伴う故障が問題となっており、自動車や産業用装置など様々な電子機器に LSI が組み込まれる現在の情報社会において、LSI 運用中の故障を検出することはシステム全体の安全性・信頼性の面で必須である。

経年劣化によって生じる故障の有効な検出方法として、LSI の出荷後に行うフィールドテスト [1] があげられる。フィールドテストとは、システムを実際に運用する中でシステムが立ち上がるまで、もしくは待機中などの間に行うテストである。例えば、自動車の場合はパワーオンセルフテストと呼ばれる、エンジンが起動する際に車載システムに対して故障検査を行うテストが行われている。このフィールドテストを可能にさせる手法として、スキャンベース組込み自己テスト (BIST : Built-In Self-Test) という手法が存在する。スキャンベース BIST とはテストコストを削減する手法の一つで、被検査回路内のフリップフロップ (FF : Flip-Flop) を直列のシフトレジスタ (スキャンチェーン) に置き換え、ランダムパターン発生回路と被検査回路の出力応答を圧縮して、期待値と比較するための署名を生成する回路を組込むことによって、LSI チップ上に簡易的な LSI テスタ機能を有する回路を組み込むという手法である。このような設計を加えた回路は、スキャンチェーンのシフトによるテストパターンの印加、及び結果の出力を行うシフトモードと、印加したテストパターンを用いて回路の応答をスキャンチェーンにキャプチャするキャプチャモードの 2 つのモードでテストを行う。

スキャンベース BIST のテストパターンには擬似乱数が用いられる。擬似乱数を用いる

場合、テストパターン生成器には擬似乱数生成器の一つである線形フィードバックシフトレジスタ(LFSR: Linear Feedback Shift Register)がよく利用される。LFSRを使用すると初期値(シード)を記憶させるだけで疑似ランダムパターンを回路内部に発生させることができるが、検出される故障に重複が多いため、高い故障検出率を得るまでにテストパターン数やテスト時間が膨大になってしまう場合がある。

別の擬似乱数生成法に、メルセンヌ・ツイスタ(MT: Mersenne Twister)アルゴリズムがある。メルセンヌ・ツイスタアルゴリズムの周期は $2^{19937} - 1$ と非常に長く、また出力の連続した値間での相関が無視できる程小さく、乱数性が良いという特徴を持つ[2]。

そこで、本研究では、メルセンヌ・ツイスタをスキャンベース BIST によるマルチサイクルテストにおけるテストパターン発生器として実装し、その有効性を検証する。具体的には、故障検出率だけでなく LFSR とメルセンヌ・ツイスタの乱数性についてもそれぞれ評価する。さらにはキャプチャ動作時に連続して複数回キャプチャを行うマルチサイクルテストにおけるテストパターン生成器として、メルセンヌ・ツイスタを用い、有効性について同様に検証する。

本論文の構成は以下の通りである。第2章では、本論文における用語の定義及び研究背景について述べる。第3章では、各擬似乱数生成器の評価方法について述べる。第4章では、実験結果及び、考察について述べる。第5章では、本研究の全体のまとめと課題について述べる。

第 2 章

準備

本章では，本論文における用語の定義と本論文の研究の背景について述べる．

2.1 諸定義[3]

本節では，本論文で利用する用語の定義を行う．

故障検出

故障の存在する回路と故障の存在しない正常な回路に同じテストパターンを与え，それぞれの出力を比較して故障を検出することを故障検出という．

故障検出率

故障検出率は，どのくらいの故障を検出できるのかをあらわす指標であり，以下の数式で求める．

$$\text{故障検出率(\%)} = \frac{\text{検出できた故障の数}}{\text{仮定された全ての故障の数}}$$

縮退故障

論理回路において，ある信号線の論理値が 0 または 1 に固定される故障である．0 に固定される故障を 0 縮退故障，1 に固定される故障を 1 縮退故障という．

フィールドテスト

LSI が出荷後，運用段階に行う検査をフィールドテストと呼ぶ．システムの待機，又は

立ち上がり時間等にテストを行うため、非常に短い時間で高い故障検出率が求められる。

2.2 スキャンベース BIST[4]

スキャンベース BIST はテスト容易化設計の一つである。図 2.1 はスキャンベース BIST の構造である。スキャン入力にテストパターン発生器、スキャン出力に出力応答圧縮器を LSI 内部に組込むことによって構成される。スキャン設計とは順序回路の FF の値を外部入力で直接制御し、外部入力で直接観測できるようにする設計である。スキャンテスト回路では、対象とする順序回路の FF をスキャンチェーンと呼ばれる直列につないだシフトレジスタに置き換え、スキャン出力まで値をシフトしていき内部状態の制御、観測を行う。このシフトレジスタに観測設定値の送り込み(スキャンイン)、観測結果の送り出し(スキャンアウト)をすることで FF の値の観測が可能になる。また、テストパターン発生回路とテスト応答判定回路を組込むことで簡易的な検査を可能にする。テストパターン発生回路には線形フィードバックシフトレジスタ(LFSR : Linear Feedback Shift Register)がよく用いられる。LFSR を利用すると初期値(シード)を記憶するだけ大量の疑似ランダムパターンを回路内部に発生させることができる。しかしながら、LFSR で発生する疑似ランダムパターンでは、検出される故障に重複が多く、少ないパターン数で高い故障検出率を得ることが困難になっている。

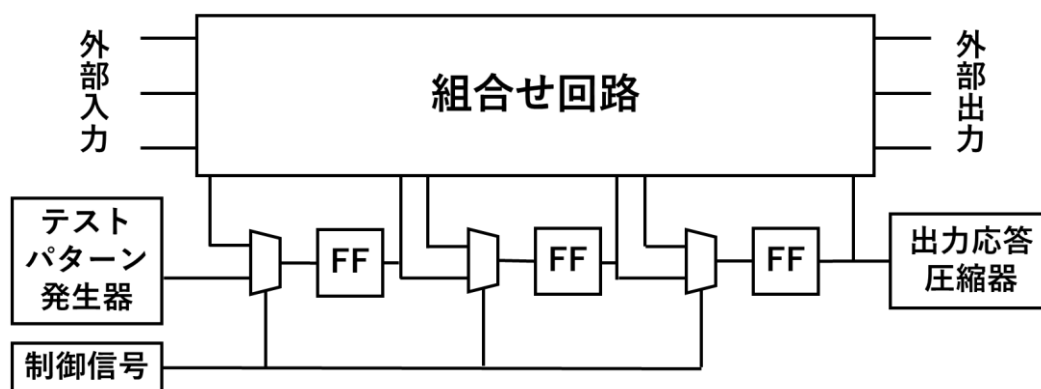


図 2.1 スキャンベース BIST

マルチサイクルテスト[5]

マルチサイクルテストとはスキャンベース BIST におけるテスト手法である。マルチサイクルテストのモデルを図 2.2 に示す。

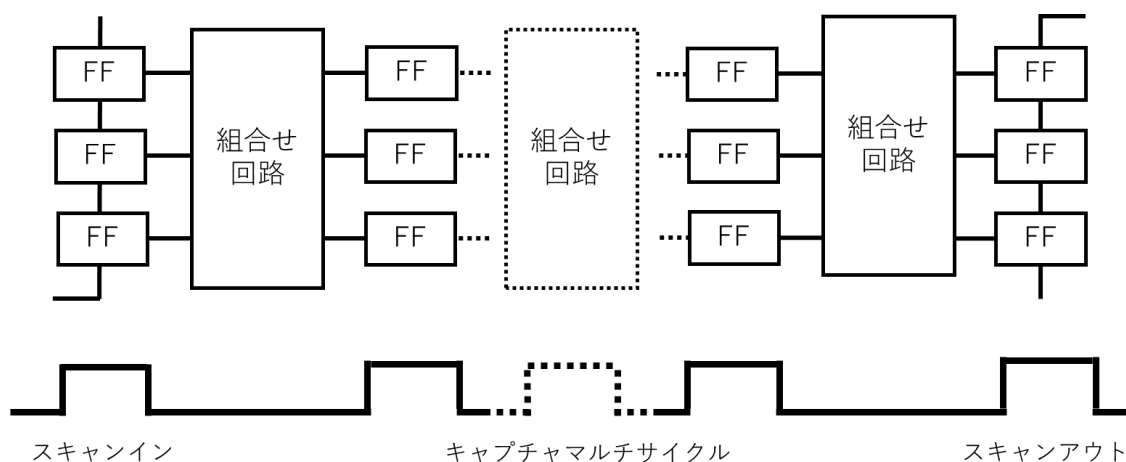


図 2.2 マルチサイクルテスト

マルチサイクルテストはキャプチャ動作時に連続して複数回のキャプチャを行うテストのことである。1つのテストパターンで複数の内部状態についてテストを行うことができる。通常のテストよりも故障検出率及びテストパターン数の観点から、テストパターンを圧縮することで短いテスト時間で多くの故障を検出する可能性があるという利点がある。しかしながら、マルチサイクルテストにおいてキャプチャサイクル数を増加させるとスキャン入力から出力までの信号値の伝搬経路が長くなってしまい、そのためテストパターンによって顕在化した故障の影響が、出力まで伝搬されない可能性がある。これは故障マスク現象と呼ばれ、故障のマスクにより故障検出率が向上しない場合がある。

2.3 線形フィードバックシフトレジスタ(LFSR)[6]

線形フィードバックシフトレジスタとは、その値を構成するビット列の一部の排他的論理和を入力ビットとするシフトレジスタである。レジスタの動作は決定的であるため、レジスタが生成する値の列は初期値(シード)によって完全に決定される。ここでフィボナッチ LFSR の構造について説明する。フィボナッチ LFSR にはタップと呼ばれる次の入力ビ

ットに影響を与えるビットが存在し、それらの位置を列挙したものをタップシーケンスと呼ぶ。図 2.3 に 16 ビットのフィボナッチ LFSR の構造を示す。

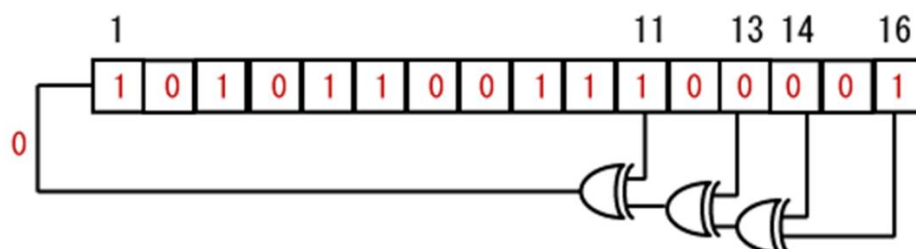


図 2.3 16 ビットのフィボナッチ LFSR

図では、タップシーケンスが $[16, 14, 13, 11, 0]$ である。タップは順次排他的論理和され、その結果が左端にフィードバックされる。このタップシーケンスの取り方によって LFSR の周期が決まる。LFSR は、タップシーケンスが原始多項式であるならば最長 LFSR となる。よって、次の性質を持つ。

- ・ タップ数は必ず偶数。非常に長いレジスタであっても 2 個や 4 個のタップで十分である。
- ・ 最長 LFSR のタップ集合は互いに素でなければならない。

このようにして LFSR は周期の長い乱数生成法を実現している。

一方で LFSR はシフト演算を用いているため、多重スキャンチェーン構造の場合、LFSR から N 本のスキャンチェーンに与えられるテストパターンのビット間に強い相関関係が生じてしまうなどの問題がある。そこで LFSR とスキャンチェーンの間に XOR ゲートからなるフェーズシフタを付加して、LFSR から生成されたテストパターンの複数のビットに対して XOR 演算を行う。その結果、テストパターンの相関関係を弱めることができる。

2.4 メルセンヌ・ツイスタ[2][7]

メルセンヌ・ツイスタは松本眞と西村拓士によって考案された擬似乱数列生成器の 1 つである。メルセンヌ・ツイスタは $2^{19937} - 1$ という長い周期を持ち、高次元（623 次元）に均等分布する。このことは出力中の連続する値間の相関性が無視できる程度しかないということを意味し、例えば、32bit 版のメルセンヌ・ツイスタを複数回呼び出して 64bit、

128bit などの疑似乱数として利用しても統計的に安全である。またメルセンヌ・ツイスタは十分に速く、出力の中のすべてのビットが統計的に十分ランダムである。一方で、メルセンヌ・ツイスタは内部に 623 個の 32 ビット長の状態ベクトルを持つため、一般的な疑似乱数生成器と比較して動作に必要なメモリ量が多いなどの短所が挙げられる。

次にメルセンヌ・ツイスタの回路構成について説明する [7]。メルセンヌ・ツイスタの乱数生成回路は、内部状態を記憶する RAM、現在の内部状態から新しい内部状態を作る生成回路 (Generate)、新しい内部状態から乱数を生成する調律回路 (Temper) から構成される。メルセンヌ・ツイスタの処理フローは図 2.4 の通りである。

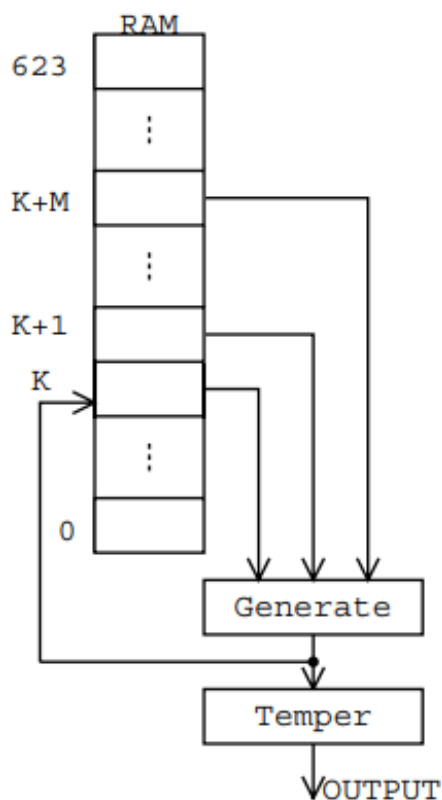


図 2.4 メルセンヌ・ツイスタの処理フロー

図 2.4 では、1 乱数を生成する毎に RAM を 3 回読んで 1 回書き込む。RAM をデュアルポート RAM で実装すると、2 サイクルで 1 乱数、RAM をマルチポート RAM (3R/1W) で実装すると、1 サイクルで 1 乱数生成する回路になる。また RAM をマルチバンク構成にすることにより、1 サイクルに複数乱数を生成することも可能である。

第3章

評価方法

本章では、LFSR とメルセンヌ・ツイスタの評価方法について述べる。先ず、乱数性の評価方法について説明する。次に、故障検出率の測定方法について説明する。

3.1 乱数性の評価方法

乱数性は等出現性と無規則性の二つに分類される。従って評価についても等出現性、無規則性の二つについてそれぞれ評価を行う。

3.1.1 等出現性

等出現性を調べる検定方法にはカイ 2 乗適合度検定や、コルモゴロフ=スミルノフ検定などがある[8]。本研究では、カイ 2 乗適合度検定を用いて等出現性の評価を行う。

ここでカイ 2 乗適合度検定の検定方法について説明する。先ず、生成された N 個の疑似乱数列がとる値の範囲を k 個の等間隔な区間に分類することにより、各区間内の乱数の個数 $\{f_1, f_2, \dots, f_k\}$ を得る。そして下式に従って X^2 を求める。

$$X^2 = \sum_{i=1}^k \frac{(f_i - \frac{N}{k})^2}{\frac{N}{k}}$$

求めた X^2 の値は、カイ 2 分布に基づいて評価する。表 3.1 にカイ 2 乗分布表[9]を示す。

表 3.1 カイ 2 乗分布表

	α (優位水準)							
自由度	0.99	0.975	0.95	0.90	0.10	0.05	0.025	0.01
1	0.00	0.001	0.004	0.016	2.710	3.84	5.02	6.63
2	0.02	0.051	0.103	0.211	4.61	5.99	7.38	9.21
3	0.115	0.216	0.352	0.584	6.25	7.81	9.35	11.3
4	0.297	0.484	0.711	1.06	7.78	9.49	11.1	13.3
5	0.554	0.831	1.15	1.61	9.24	11.07	12.8	15.1
6	0.872	1.24	1.64	2.20	10.64	12.6	14.5	16.8
7	1.24	1.69	2.17	2.83	12.02	14.1	16.0	18.5
8	1.65	2.18	2.73	3.49	13.4	15.5	17.5	20.1

有意水準は、検定において帰無仮説を設定したときにその帰無仮説を棄却する基準となる確率のことである。一般的に 5% (0.05) や 1% (0.01) といった値がよく使われる。また自由度は k から 1 を引いた値となる。計算したカイ 2 乗の値が表 3.1 の値よりも小さかった場合、等出現性は十分であると評価される。

3.1.2 無規則性

無規則性を調べる検定方法には、系列相関、連検定、ギャップ検定、スペクトル検定などがある。本研究では、連検定を用いて無規則性の評価を行う。

同様に連検定の検定方法について説明する。まず上昇連, 下降連の例を図 3.1 に示す。

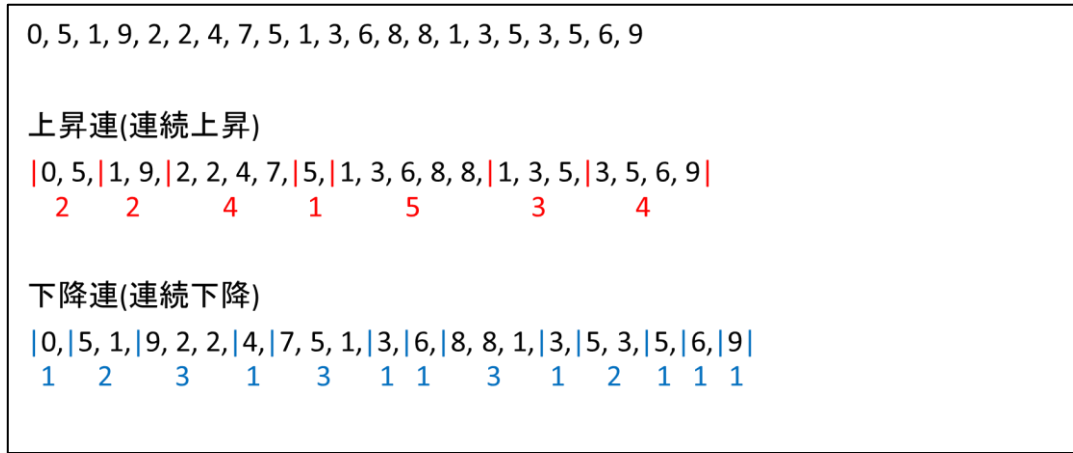


図 3.1 上昇連, 下降連の例

図 3.1 にあるように値の連続上昇部分で区切ったものを上昇連, 連続下降部分で区切ったものを下降連という. 赤と青色で示された数字は各連の長さを示している. 連検定には上昇連(連続上昇), 下降連(連続下降)の 2 種類での検定方法がある. 本研究では, 上昇連での検定を扱う.

連検定では, 生成された N 個の乱数列から長さ $1, \dots, 5$ の連の度数分布 f_1, \dots, f_5 と、長さ 6 以上の連の度数分布 f_6 を求め、統計量 V を計算する。式を以下に示す。

$$V = \frac{1}{N} \sum_{1 \leq i, j \leq 6} (f_i - Nb_i)(f_j - Nb_j) a_{ij}$$

ただし、

$$a_{ij} = \begin{bmatrix} 4529.35 & 9044.9 & 13568 & 18091.3 & 22614.7 & 27892.2 \\ 9044.9 & 18097 & 27139.5 & 36186.7 & 45233.8 & 55788.8 \\ 13568 & 27139.5 & 40721.3 & 54281.3 & 67842 & 83864.6 \\ 18091.3 & 36186.7 & 54281.3 & 72413.6 & 90470.1 & 111580 \\ 22614.7 & 45233.8 & 67852 & 90470.1 & 113262 & 139476 \\ 27892.2 & 55788.8 & 83684.6 & 111580 & 139476 & 172860 \end{bmatrix}$$

$$b_i = \left[\frac{1}{6} \quad \frac{5}{24} \quad \frac{11}{120} \quad \frac{19}{720} \quad \frac{29}{5040} \quad \frac{1}{840} \right]$$

とする. このような条件かつ $N > 10^4$ 程度である場合、統計量 V は自由度 6 の X^2 分布に従うため、表 3.1 のカイ 2 乗分布に基づいて評価を行う。

3.2 故障検出率の測定方法

故障検出率の測定は、スキャンチェーンに各疑似乱数生成器で生成したテストパターンを印加する方法で行う。LFSR についてはフェーズシフトを用いた場合と用いない場合の両方の故障検出率を測定する。またマルチサイクルテストにおいても、キャプチャサイクル数毎の故障検出率を同様の形で測定する。

第 4 章

評価実験及び考察

本章では、本研究で行った実験の内容、及び結果と考察について述べる。先ず乱数性の評価実験の結果を述べ、次に故障検出率の測定及び評価結果について述べる。

本実験に使用した計算機環境を表 4.1 に示す。

表 4.1 計算機環境

CPU	Intel Core i5 7500 3.4GHz
Memory	16.0GB
OS	Linux 3.10.0-1127

4.1 乱数性の評価実験

本節では、乱数性の評価として等出現性、無規則性の評価実験の結果について示す。

4.1.1 等出現性の評価実験

3.1.1 節で説明したカイ 2 乗適合度検定を用いて等出現性の評価実験を行う。生成する乱数列の個数を 10000 から 100000 まで段階的に増やしながら計算したカイ 2 乗の値を表 4.2 にまとめる。ただし、分割の個数 k を 4 とし、各疑似乱数生成器ともに 32 ビット版のものを用いて乱数生成を行ったとする。

表 4.2 カイ 2 乗適合度検定の計算結果

乱数列の 個数	LFSR(フェーズ シフタなし)	LFSR(フェーズ シフタあり)	メルセンヌ・ ツイスタ
10000	2.43	3.98	1.73
20000	1.99	0.68	4.98
30000	5.27	0.97	5.51
40000	5.91	1.49	5.20
50000	2.68	0.57	4.04
60000	4.70	2.90	3.98
70000	3.20	1.70	4.74
80000	4.67	3.14	4.01
90000	6.47	3.33	3.23
100000	9.00	4.06	3.75

$k=4$ より自由度は 3 であるため、有意水準を 5% (0.05) とした場合のカイ 2 乗分布は表 3.1 の分布表より 7.81 となる。表 4.2 の結果より、フェーズシフタなしの LFSR における乱数列の個数が 100000 の場合のみ 9.00 と 7.81 を上回っており、それ以外の場合は全て 7.81 を下回っていることが分かる。また、大数の法則より乱数列の個数が多くなるほど、検定値も収束していくと考え、フェーズシフタなしの LFSR は他の 2 つに比べて等出現性について不十分であることが分かる。

4.1.2 無規則性の評価実験

3.1.2 節で説明した連検定を用いて無規則性の評価実験を行う。同様にこちらも、生成する乱数列の個数を 10000 から 100000 まで段階的に増やしながら計算を行った。統計量 V の計算結果を表 4.2 にまとめる。ただし連の分割は上昇連で行い、値が等しい場合も連が続くものとする。

表 4.3 連検定の統計量 V の計算結果

乱数列の 個数	LFSR(フェーズ シフタなし)	LFSR(フェーズ シフタあり)	メルセンヌ・ ツイスタ
10000	10927.45	14.70	2.13
20000	23987.60	38.97	6.36
30000	39918.82	43.32	6.74
40000	54429.80	49.51	7.99
50000	61881.47	58.03	4.51
60000	78340.89	93.01	3.11
70000	88738.93	100.38	2.59
80000	97723.37	106.53	1.76
90000	109665.67	117.63	1.81
100000	119714.05	119.84	2.04

統計量 V は自由度 6 の X^2 分布に従うため、有意水準を 5% (0.05) とした場合のカイ 2 乗分布は表 3.1 の分布表より 12.6 となる。表 4.3 より、フェーズシフタありの LFSR はフェーズシフタなしのものと比べて値が大幅に小さくなっていることが分かる。しかしながら、フェーズシフタありの LFSR でも 12.6 を上回っており、無規則性について不十分であることが分かる。また、乱数列の個数が 60000 のところまではメルセンヌ・ツイスタの検定値が安定していないが、70000 からは値の振れ幅も 1.0 未満で安定しており、大数の法則よりメルセンヌ・ツイスタは無規則性について十分であると言える。

4.2. 故障検出率の測定及び評価

本節ではスキャンベース BIST とマルチサイクルテストのそれぞれの故障検出率の測定結果を示す。

故障シミュレーションに用いた回路の回路名と故障数を表 4.4 にまとめる。

表 4.4 回路名と故障数

回路名	故障数
s9234	6927 個
s15850	11725 個
s38417	31180 個

4.2.1. スキャンベース BIST での故障検出率比較

スキャンテストを用いて各回路の故障検出率を測定した結果を以下の図 4.1～4.3 に示す。

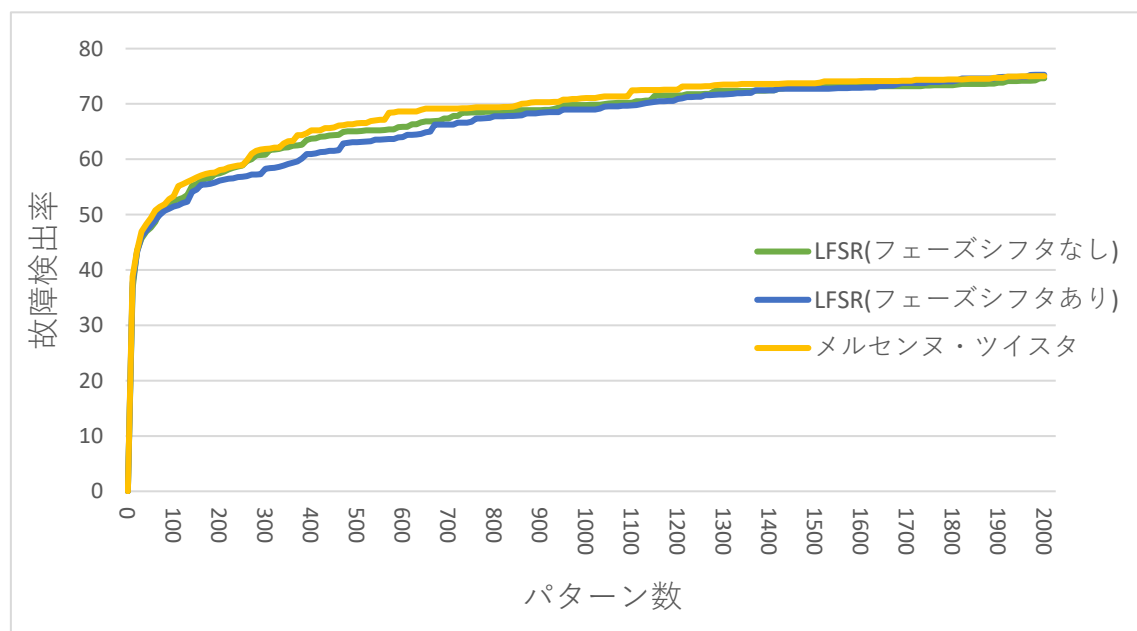


図 4.1 s9234 回路でのスキャンテスト結果

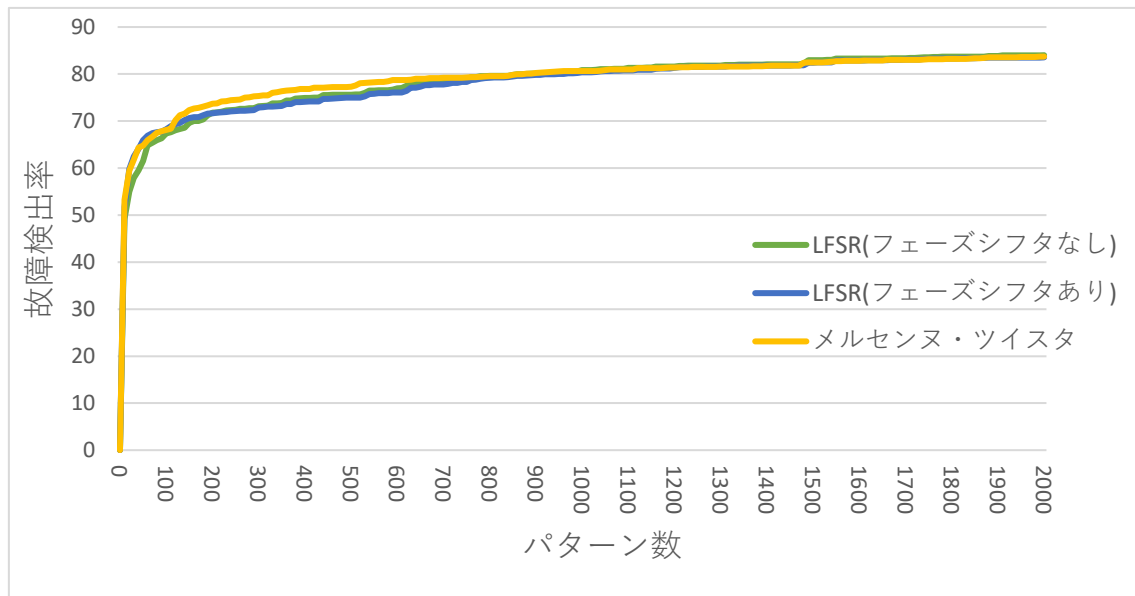


図 4.2 s15850 回路でのスキャンテスト結果

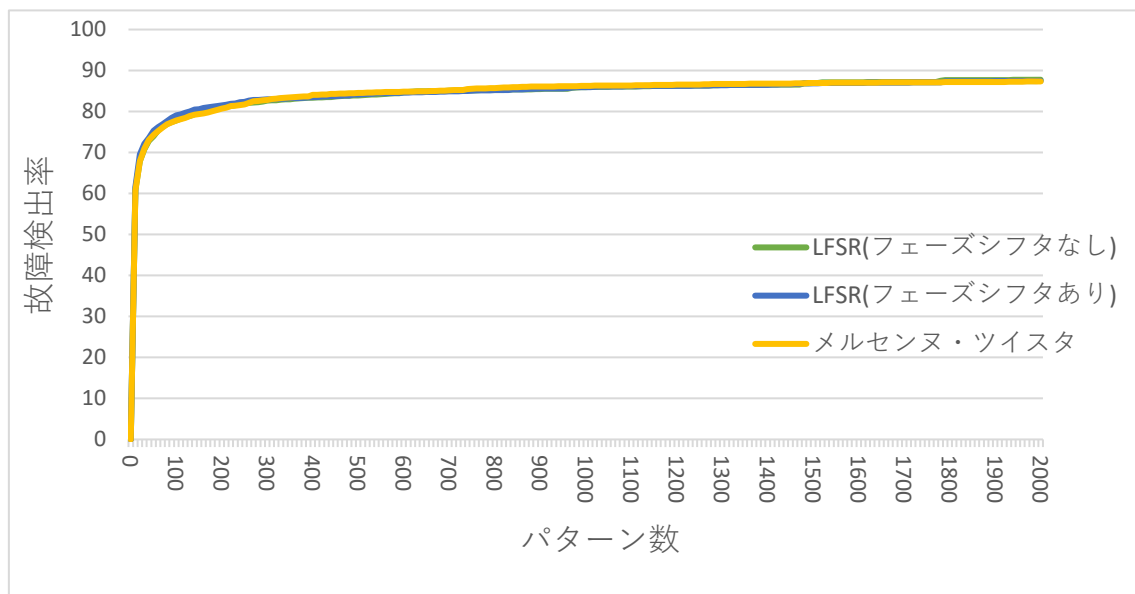


図 4.3 s38417 回路でのスキャンテスト結果

図 4.1 より s9234 回路で測定した場合，パターン数が 400 個から 700 個の間においてメルセンヌ・ツイスタが最も故障検出率が高くなっていることが分かる．同様に図 4.2 より s15850 回路で測定した場合も，パターン数が 200 個から 700 個の間でメルセンヌ・ツイ

スタの故障検出率が最も高くなっていることが分かる。一方で、図 4.3 より故障数を約 30000 個含む s38417 回路で測定した場合、LFSR とメルセンヌ・ツイスタの故障検出率の差はほとんどないことが分かる。

4.2.2. マルチサイクルテストでの故障検出率比較

マルチサイクルテストを用いて s9234 回路の故障検出率を測定した結果を以下の図 4.4～4.6 に示す。

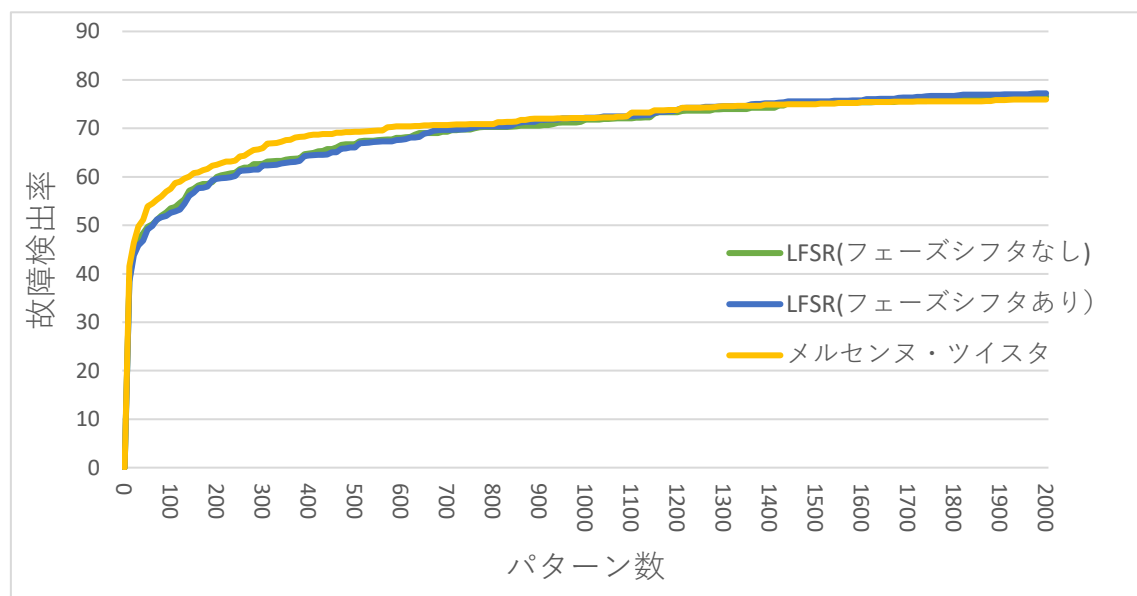


図 4.4 s9234 回路でのマルチサイクルテスト結果（サイクル数：2 回）

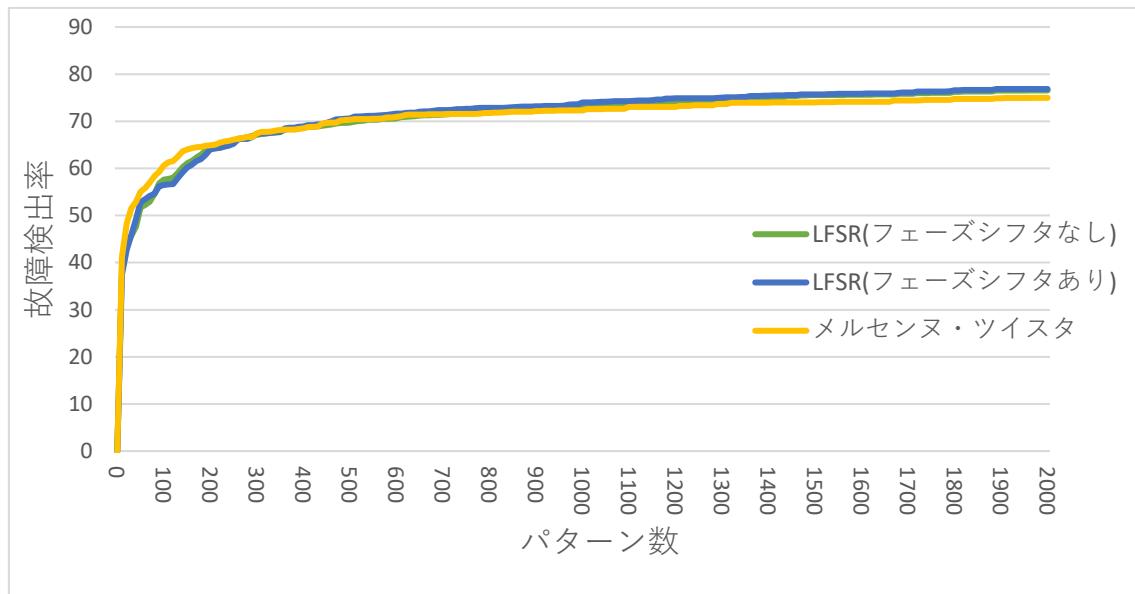


図 4.5 s9234 回路でのマルチサイクルテスト結果（サイクル数：5 回）

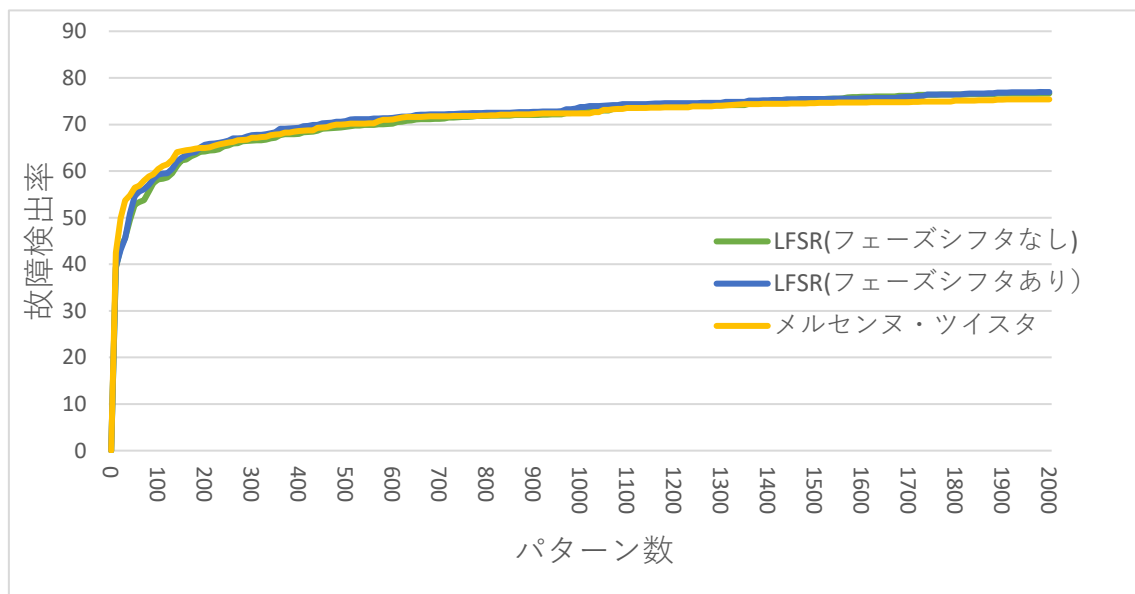


図 4.6 s9234 回路でのマルチサイクルテスト結果（サイクル数：10 回）

図 4.4 よりキャプチャサイクル数を 2 回とした場合，パターン数が 100 個の時点でメルセンヌ・ツイスタを用いての故障検出率が，LFSR よりも大きく上回っていることが分かる．同様に図 4.5 より，キャプチャサイクル数を 5 回とした場合も，立ち上

がりはメルセンヌ・ツイスタの方が良いことが分かる。しかしながら、サイクル数を2回とした場合ほどの差はなく、パターン数が300個になったあたりでLFSRに追いつかれていることが分かる。さらに図4.6よりサイクル数を10回とした場合、立ち上がりの差も縮まり、パターン数200個のあたりで同じになっていることがわかる。

次に s15850 回路での測定した結果を以下の図4.7～4.9に示す。

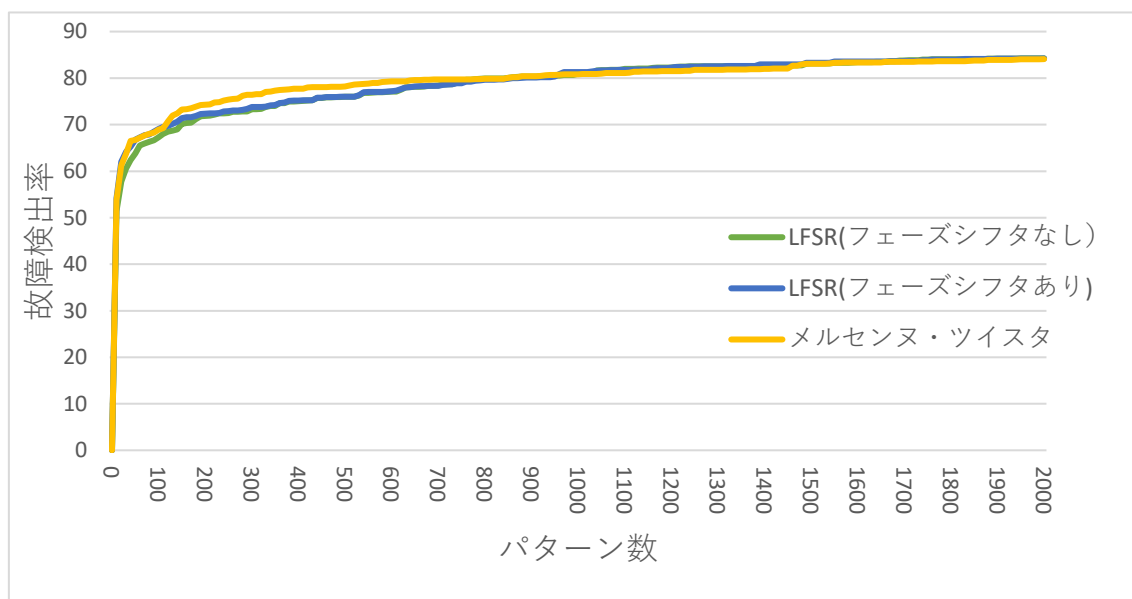


図4.7 s15850 回路でのマルチサイクルテスト結果（サイクル数：2回）

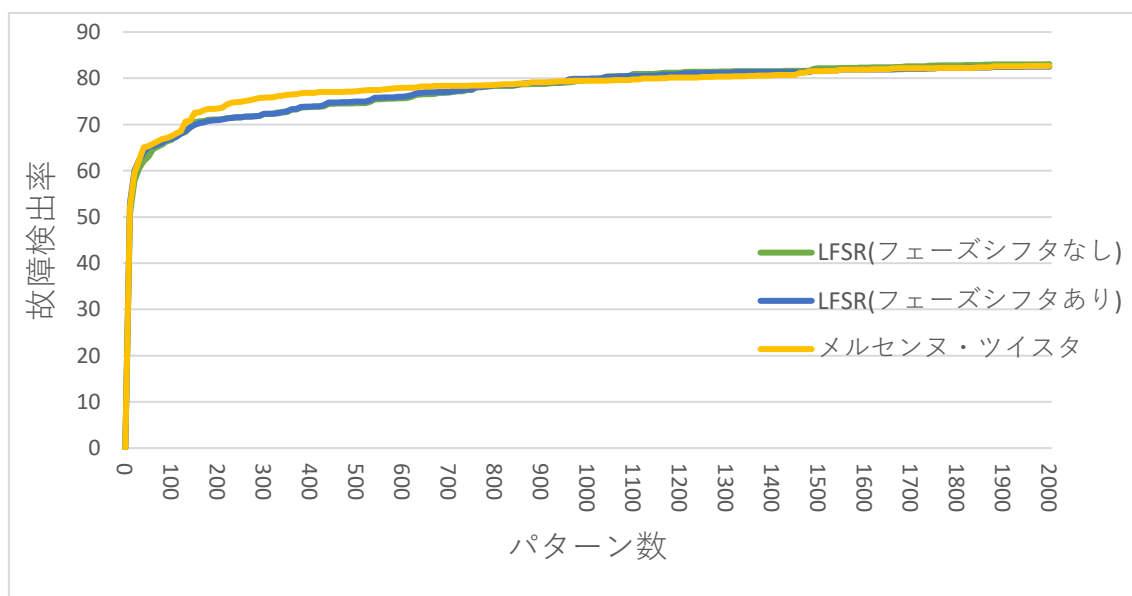


図 4.8 s15850 回路でのマルチサイクルテスト結果（サイクル数：5 回）

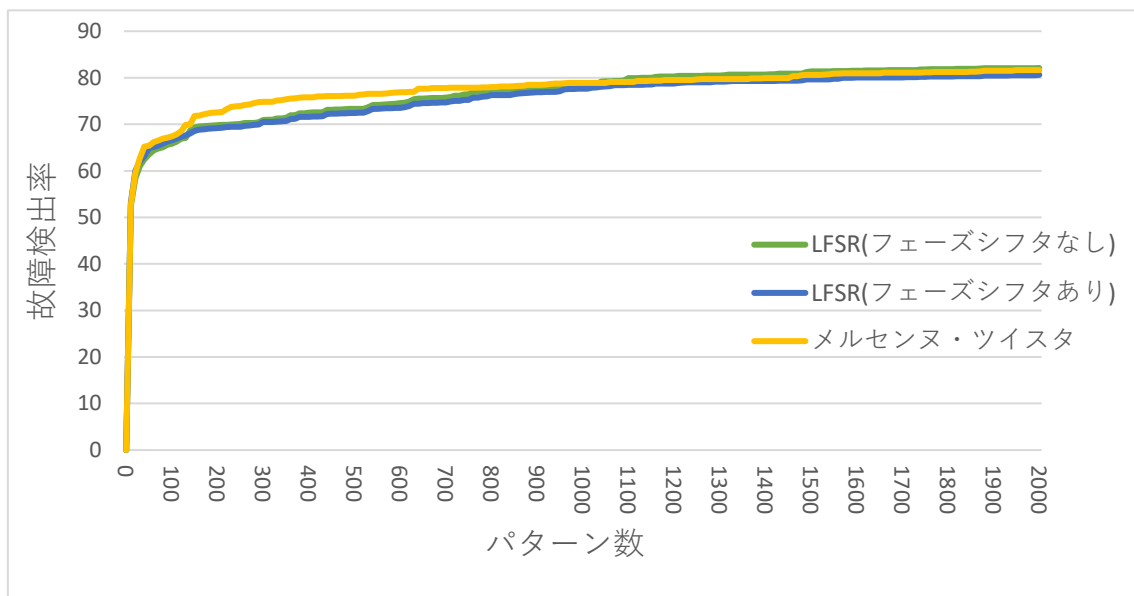


図 4.9 s15850 回路でのマルチサイクルテスト結果（サイクル数：10 回）

図 4.7 より s15850 回路においても，メルセンヌ・ツイスタが少ないパターン数で高い故障検出率を得ていることが分かる．また図 4.8，4.9 よりサイクル数を増やしても，s9234 回路とは異なって，LFSR とメルセンヌ・ツイスタの故障検出率の差が縮まっていないことが分かる．

4.3. 全体の考察

4.1 節では乱数性の評価として、等出現性と無規則性の評価を行った。実験結果より LFSR はフェーズシフタを用いたとしても無規則性が不十分であり、等出現性と無規則性の両方をまとめた乱数性として評価すると、やはりメルセンヌ・ツイスタが優れているということが改めて分かった。

4.2 節では故障検出率の評価として、スキャンテストとマルチサイクルテストの2つで故障検出率の測定を行った。スキャンテスト結果の図 4.1, 4.2 より、少ないパターン数ではメルセンヌ・ツイスタが LFSR よりも高い故障検出率を得ており、立ち上がりはメルセンヌ・ツイスタの方が優れていることが分かった。しかしながら、図 4.3 より、s38417 回路の場合では、立ち上がりの段階で既に同じ故障検出率になっており、必ずしもメルセンヌ・ツイスタが優れているわけではないということが分かった。また図 4.4 から、メルセンヌ・ツイスタと LFSR の立ち上がりの差がスキャンテストの時と比べて大きくなっており、マルチサイクルテストを用いると、メルセンヌ・ツイスタによる故障検出率増加をさらに見込めることが分かる。一方で図 4.5, 4.6 から、サイクル数を増やすと、LFSR とメルセンヌ・ツイスタの差が縮まっており、サイクル数の増やしすぎるとメルセンヌ・ツイスタによる故障検出率強化を受けられないことが分かった。

第5章

あとがき

本研究では、メルセンヌ・ツイスタの乱数性の評価とメルセンヌ・ツイスタをテストパターン生成器として実装し、スキャンベース BIST、マルチサイクルテストを用いてその有効性を検証した。乱数性の評価では、メルセンヌ・ツイスタが LFSR よりも無規則性という面で優れているということが分かった。また、スキャンベース BIST の実験では、メルセンヌ・ツイスタを用いると、少ないパターン数で高い故障検出率が得られることが分かった。さらには、マルチサイクルテストの実験結果より、マルチサイクルテストにおいてもメルセンヌ・ツイスタが有効であることが示された。

今後の課題としては、メルセンヌ・ツイスタを実際にフィールドテストのテストパターン生成器として用いる場合に、メルセンヌ・ツイスタの軽量化をどのように図るかである。

謝辞

本研究を進めるにあたり，懇篤な御指導，御鞭撻を賜りました本学高橋寛教授に深く御礼申し上げます。

本論文の作成に関し，詳細なるご検討，貴重な御教示を頂きました本学甲斐博准教授ならびに王森レイ講師に深く御礼申し上げます。

最後に，多大な御協力と貴重な御助言を頂いた本学工学部情報工学科情報システム工学講座高橋研究室の諸氏に厚く御礼申し上げます。

参考文献

- [1] 山口久登, 松蘭誠, 佐藤康夫, 梶原誠司, “スキャン BIST におけるマルチサイクルテストと部分観測方式の提案と評価“, 信学技, DC2010-2028, pp.31-36, Nov.2010

- [2] “メルセンヌ・ツイスタ - Wikipedia”
<https://ja.wikipedia.org/wiki/メルセンヌ・ツイスタ> (参照 2021-1017)

- [3] 藤原 秀雄(1990)『コンピュータの設計とテスト』工学図書 p.221

- [4] 藤原 昌宏(2006)『システム LSI 設計工学』オーム社 p.231

- [5] 門田 一樹, 濱田 宗, 王 森レイ, 樋上 喜信, 高橋 寛, 岩田 浩幸, 松嶋 潤 “マルチサイクルテストのための FF の構造的評価”, 電子情報通信学会総合大会 2016(発表予定)

- [6] “線形帰還シフトレジスタ - Wikipedia”
<https://ja.wikipedia.org/wiki/線形帰還シフトレジスタ> (参照 2021-01-17)

- [7] “疑似乱数生成回路 MT19937 の設計と評価”
<http://www.ccs.ee.tut.ac.jp/ich/thesis/papers/ugrad/2004/konuma2004.pdf>
(参照 2021-02-04)

- [8] “統計の分析と利用 Contents”
http://www.bunkyo.ac.jp/~hotta/lab/courses/2010/2010dist/10dist_6.pdf
(参照 2021-01-19)

[9] “カイ 2 乗分布表”

<http://www3.u-toyama.ac.jp/kkarato/2016/statistics/handout/chisqdist.pdf>

(参照 2021-01-19)