ISSN 2186-5647

2-20

k 時間展開モデルを用いたランダムパターンレジスタント故障のシード生成法

日大生産工（院）　○曽根隆暢　日大生産工　細川利典
京産大　　吉村正義　日大生産工　新井雅之

1. まえがき

近年，半導体集積技術の発展に伴い，設計される超大規模集積回路(Very Large Scale Integrated circuits: VLSI)の大規模化，複雑化が急速に進展している．その影響により，多量化したテストパターンをテスタ(Automatic Test Equipment: ATE)に保持することは非現実的であり，製造テストにおけるコスト増大につながる．製造テストコストを削減するための技術として，組込み自己テスト(Built-In Self-Test: BIST)[1,2,3]が広く用いられている．特に，スキャン BIST の代表的なアーキテクチャとして STUMPS 構造[4]が広く知られている．BIST は，被テスト回路(Circuit Under Test: CUT)内部にテストパターン発生器(Test Pattern Generator: TPG)や出力応答圧縮器(Multiple Input Signature Register: MISR)などのテスト用の回路を組込むことで，故障の検出を行う．一般的に，TPG としてはフェーズシフタ[5]付きの線形帰還シフトレジスタ(Linear Feedback Shift Register: LFSR)が用いられる．LFSR に初期値(シード)を与え，その出力をフェーズシフタに入力することで規則性がない擬似ランダムパターンを発生させることが可能となる．しかしながら，擬似ランダムパターンを用いたテストでは，自動テストパターン生成ツール(Automatic Test Pattern Generator: ATPG)で生成した決定的なテストパターンと同等の故障検出率を得ることは困難である．その原因の 1 つとして，ランダムパターンレジスタント(Random Pattern Resistant: RPR)故障の存在が挙げられる．そのため，LFSR を用いた BIST において，高い故障検出率を達成するためには RPR 故障の検出が重要である．RPR 故障を検出するための手法として，テスト中にシードを入替えるリシード技術[6]が提案されている．リシードに用いるシードは，LFSR によって生成されたテストパターンが RPR 故障の検出に有効なものでなければならない．

RPR 故障の検出に有効なシードを生成する手法として，1 パスシード生成法[7]が提案されている．1 パスシード生成法は，LFSR とフェーズシフタ，CUT からシード生成モデルが生成され，ATPG を用いて直接シードを求める．しかしながら，シード生成は単一目標故障に対して行われるため，高い故障検出率を達成するためには多数のシードが必要になる可能性がある．

文献[9]では，複数目標故障のシード生成とそのシードから複数個のテストパターンを生成して故障検出を行った．

本論文では，k サイクル後に複数目標故障を検出するシード生成法を提案する．k 時間展開モデルを用いることにより，1 つのシードからの k サイクルテストで目標故障以外にどの程度，RPR 故障が検出可能か評価する．

本論文は以下のように構成される．第 2 章では 1 パスシード生成モデルについて述べる．第 3 章では複数 RPR 故障のシード生成法ついて述べる，第 4 章では提案手法である k 時間展開モデルを用いたシード生成法について述べる．第 5 章では ISCAS’89 ベンチマーク回路を用いた評価を行う．第 6 章でまとめと今後の課題を述べる．

2. 1 パスシード生成法

本章では，1 パスシード生成法について説明する．文献[7]では，SAT を用いた 1 パスシード生成法が提案されている．SAT[8]とは，与えられた論理積標準形(Conjunctive Normal Form: CNF)を充足する変数割当てが存在するか否かを判定する問題である．対象故障と CUT とフェーズシフタ付き LFSR から構成されるシード生成モデルを CNF 変換し，SAT ソルバーの入力とすることで直接シードを生成することができる．各シード生成では，与えられた故障集合中から 1 つの故障を選択し，目標故障とする．しかしながら，1 つの故障を目標故障としてシード生成を行うため，完全な故障検出率を達成するために多数のシードを必要とする可能性がある．完全な故障検出率とは，テスト可能な故障に対して 100%の故障検出率である．

シード数を削減する解決策として，複数の故障を目標故障とすることが挙げられる．しかしながら，SAT を用いて，複数の故障を目標としてシード生成を行う場合は，すべての目標故障を同時に検出できるシードが存在することを保証しなければならないため効率的ではない．

3. 複数 RPR 故障のシード生成法

本章では，複数 RPR 故障の 1 パスシード生成法[9]について説明する．3.1 節で擬似ブール最適化について説明し，3.2 節でシード生成モデルについて説明し，3.3 節でシード生成モデルの定式化について説明する．

3.1. 擬似ブール最適化

擬似ブール最適化(Pseudo Boolean Optimization: PBO)は，与えられたすべてのブール制約を充足し，最適化関数を最小化するブール変数割当てを求める最適化問題である．割当てが存在する場合は充足可能(Satisfiable: SAT)となり，そうでない場合は充足不可能(Unsatisfiable: UNSAT)となる．

3.2. シード生成モデル

本節では，PBO を用いた複数 RPR 故障の 1 パスシード生成モデルについて説明する．

A Multiple Target Seed Generation Method for Random Pattern Resistant Faults Using K-Time Expansion Models

Takanobu SONE, Toshinori HOSOKAWA,
Masayoshi YOSHIMURA and Masayuki ARAI

目標故障数 n におけるシード生成モデルを図 1 に示す. シード生成モデルは, フェーズシフタ付き LFSR, 正常回路，故障回路，故障検出判定用回路で構成される．故障回路は目標故障数分用意する．故障検出の判定は，正常回路と故障回路の外部出力値を比較することで判定される．対象故障に対応する XOR ゲートの出力値を 1 に設定するようにシード変数を決定することでシードが得られる．複数の外部出力に故障が伝搬する場合は，対象故障に対応する XOR ゲートの出力を入力とする OR ゲートの出力値を 1 に設定することで，いずれかの外部出力で故障が観測可能なシードが生成できる．そのため，図 1 ではdet_1およびdet_nを 1 に設定する．

3.3. シード生成モデルの定式化

本節では，シード生成モデルの定式化について説明する. PBO では最適化関数と制約を入力として問題を解決するため，シード生成モデルを回路動作制約と故障検出制約で表現する．

回路動作制約は，正常回路，故障回路，故障検出回路の回路動作条件を制約式で表現したものである．基本ゲートにおけるブール制約表現を表 1 に示す. また, フェーズシフタ付き LFSR から生成されるテスト系列は, XOR ネットワークとして表現することが可能であり，擬似外部入力(Pseudo Primary Input: PPI)に設定される値は XOR 演算の結果として表現される．4bit のフェーズシフタ付き LFSR および 8 つの PPI が接続されているスキャンチェインに対応する論理回路を例として図 2 に示す．PPI8 に設定される論理値は, S1 と S2 の XOR 演算の結果であるフェーズシフタの最初のサイクルの出力 PS1 の出力値である．S1 から

$$S1 \oplus S2 = PPI8 \quad (1)$$
$$S1 \oplus S3 = PPI7 \quad (2)$$
$$S1 \oplus S4 = PPI6 \quad (3)$$
$$S2 \oplus S3 = PPI5 \quad (4)$$
$$S1 \oplus S3 \oplus S4 = PPI4 \quad (5)$$
$$S2 \oplus S3 \oplus S4 = PPI3 \quad (6)$$
$$S4 = PPI2 \quad (7)$$
$$S1 \oplus S2 = PPI1 \quad (8)$$

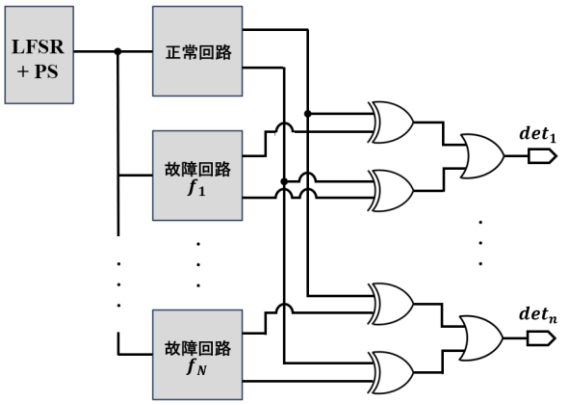

図 1. 複数 RPR 故障のシード生成モデル

表 1. 基本ゲート制約

Gate	Input		Output	Constraint
AND	x	y	z	$x+\bar{z}\geq 1$ $y+\bar{z}\geq 1$ $\bar{x}+\bar{y}+z\geq 1$
NAND	x	y	z	$x+z\geq 1$ $y+z\geq 1$ $\bar{x}+\bar{y}+\bar{z}\geq 1$
OR	x	y	z	$\bar{x}+z\geq 1$ $\bar{y}+z\geq 1$ $x+y+\bar{z}\geq 1$
NOR	x	y	z	$\bar{x}+\bar{z}\geq 1$ $\bar{y}+\bar{z}\geq 1$ $x+y+z\geq 1$
BUF	x		z	$\bar{x}+z\geq 1$ $x+\bar{z}\geq 1$
INV	x		z	$x+z\geq 1$ $\bar{x}+\bar{z}\geq 1$
XOR	x	y	z	$\bar{x}+\bar{y}+\bar{z}\geq 1$ $x+y+\bar{z}\geq 1$ $x+\bar{y}+z\geq 1$ $\bar{x}+y+z\geq 1$

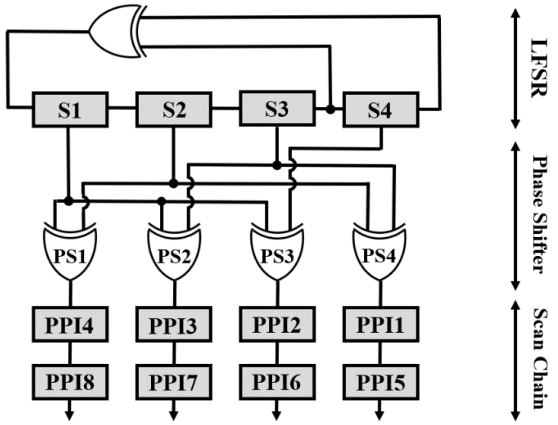

図 2. フェーズシフタ付き LFSR

S4 は 4 ビット LFSR の初期値である．したがって, PPI8 の値は式(1)で表現できる．同様に，PPI7 から PPI1 に設定される値は式(2)から式(8)の制約式が与えられる．

故障検出制約は，故障の検出を行うために必要な故障励起条件と故障検出条件を制約式で表現したものである．信号線 A の 0 縮退故障を例とした場合では，故障を励起するために, 正常回路では A に論理値"1"を割当て，故障回路では A に論理値"0"を割当てる．0 縮退故障の故障励起条件の一般的な制約を式(9)に示す．G は正常回路における目標故障信号線の論理値を表し，F は故障回路における目標故障信号線の論理値を表す. 同様に 1 縮退故障の故障励起条件は式(10)で表現する．

$$G = 1, \bar{F} = 1 \quad (9)$$
$$\bar{G} = 1, F = 1 \quad (10)$$
$$G, F \in \{0,1\}$$

また，故障の検出を行うためには故障検出判定回路で用いられる XOR または OR 演算出力信号線 OUT を"1"に設定する必要がある．故障検出条件の制約を式(11)に示す.

$$\mathrm{OUT} = 1 \quad (11)$$
$$\mathrm{OUT} \in \{0,1\}$$

しかしながら，複数目標故障を検出するためのシード生成モデルにおいて，目標故障の故障検出制約を式(9)，(10)，および(11)で定式化すると，すべての目標故障を検出するためのシードを生成する必要がある．そのため，目標故障集合中に同じシードで検出できない両立不可能な故障が含まれていた場合，すべての制約を充足するシードが存在しないため，"UNSAT"となる．この問題を解決するため，両立不可能である可能性がある複数目標故障に対する故障検出制約は式(12)，(13)および(14)のような制約で定式化する．Relax は緩和変数を表す．

$$\mathrm{Relax} + \mathrm{G} \geq 1, \mathrm{Relax} + \bar{\mathrm{F}} \geq 1 \quad (12)$$
$$Relax + \bar{\mathrm{G}} \geq 1, \mathrm{Relax} + \mathrm{F} \geq 1 \quad (13)$$
$$\mathrm{Relax} + \mathrm{OUT} \geq 1 \quad (14)$$
$$\mathrm{G}, \mathrm{F}, \mathrm{OUT}, \mathrm{Relax} \in \{0,1\}$$

式(9)，(10)，および(11)を充足できない場合には，式(12)，(13)および(14)を用いて，Relax に"1"を割当てることで制約を充足することができる．目標故障集合に両立不可能な故障が含まれている場合でも，式(12)，(13)および(14)を用いることでシード生成モデルに対応する制約を充足することができる．しかしながら，PBO ソルバーが Relax に"1"を優先的に割当てた場合，故障検出条件が無視される．これを避けるために，最適化関数として式(15)を設定する．N は緩和変数の個数すなわち目標故障数を示す．式(15)により，Relax に可能な限り"0"が割当てられるようになり，検出故障数を最大化するシードを生成することができる．

$$\text{Minimize}: \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{Relax_i} \quad (15)$$

PBO ソルバーに与える式を式(16)に示す．式(16)は全体の定式化を示す．式(15)は最適化関数である．

$$\phi_{seed} \equiv \phi_{LFSR} \cdot \phi_c \cdot \bigwedge_{i=1}^{N} (\phi_{f_i} \cdot \phi_{act_i} \cdot \phi_{det_i}) \quad (16)$$

式(16)においてϕ_{seed}はシード生成モデルの制約を表し，ϕ_{LFSR}はフェーズシフタ付き LFSR の動作制約を表し，ϕ_cは正常回路の動作制約を表し，ϕ_fは目標故障に対する故障回路の動作制約を表し，ϕ_{act}は目標故障に対する故障励起制約を表し，ϕ_{det}は目標故障に対する故障検出制約を表す．

4. k 時間展開モデルを用いたシード生成

本章では，提案手法である k 時間展開モデルを用いたシード生成法について説明する．4.1 節で k 時間展開シード生成モデルについて説明し，4.2 節でシード生成アルゴリズムについて説明する．

4.1. k 時間展開シード生成モデル

本節では，提案手法で用いる k 時間展開シード生成モデルについて説明する．目標故障数 N，時間展開数 K のシード生成モデルを図 3 に示す．k 時間展開シード生成モデルは，図 1 と同様にフェーズシフタ付き LFSR，正常回路，故障回路，故障検出判定用回路で構成されるが，正常回路は時間展開数分，故障回路は目標故障数×時間展開数分用意する．故障検出の判定は，最終時刻 K における正常回路と故障回路の外部出力値を比較することで判定される．k 時間展開モデルを用いたシード生成における PBO ソルバーに与える式を式(17)に示す．

$$\phi_{seed} \equiv \phi_{LFSR} \cdot \bigwedge_{t=1}^{K} \left(\phi_{c_t} \cdot \bigwedge_{i=1}^{N} (\phi_{f_{i,t}}) \right) \cdot \bigwedge_{i=1}^{N} (\phi_{act_i} \cdot \phi_{det_i}) \quad (17)$$

式(17)においてϕ_{seed}はシード生成モデルの制約を表し，ϕ_{LFSR}はフェーズシフタ付き LFSR の動作制約を表し，ϕ_cは正常回路の動作制約を表し，ϕ_fは目標故障に対する故障回路の動作制約を表し，ϕ_{act}は目標故障に対する故障励起制約を表し，ϕ_{det}は目標故障に対する故障検出制約を表す．式(16)と比較して時間展開数 K 個分の正常回路と故障回路を追加することで，1 つのシードで多くの故障を検出することを目的とする．

4.2. シード生成アルゴリズム

k 時間展開モデルを用いた複数 RPR 故障のシード生成法のアルゴリズムを図 4 に示す．入力は RPR 故障集合F，スキャン設計回路C，フェーズシフタ付き LFSR$LFSR$，時間展開数Kである．出力はシード集合Sである．まず，Sを空集合に初期化する(行 4)．次にフェーズシフタ付き LFSR の制約式およびCのスキャンチェイン内の各 PPI の動作制約式ϕ_{LFSR}を生成する(行 5)．正常回路の動作制約ϕ_{GC}を K 時間分生成する(行 6)．Fが空になるまで行 8 から行 12 までの処理を繰り返す(行 7)．故障回路の動作制約ϕ_{FC}を K 時間分生成する(行 8)．ϕ_{LFSR}，ϕ_{GC}，ϕ_{FC}を用いてシード生成モデルの制約式ϕ_{seed}を生成する(行 9)．ϕ_{seed}を PBO ソルバーの入力として与え，Fの検出故障数を最大化するシードsが生成され，sによって検出される故障集合F_Dを得る(行 10)．Sとsの和集合を計算し，Sを更新する(行 11)．F_D内の故障をFから削除し，Fを更新する(行 12)．最後にSを返す(行 14)．

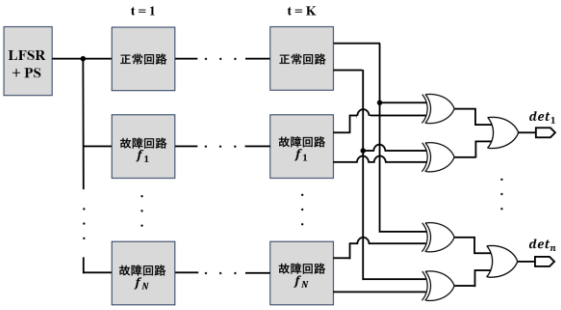

図 3. k 時間展開シード生成モデル

```
1.  Input: RPRF set F ,Circuit C , LFSR with phase shifter LFSR , Time Expansion K
2.  Output: Seed set S
3.  Seed_Generation(F, C, LFSR, K){
4.      S = φ ;
5.      Φ_LFSR = Generate_Constraint_LFSR(LFSR);
6.      Φ_GC = Generate_Constraint_Good_Circuit(C, K);
7.      while (F ≠ φ) {
8.          Φ_FC = Generate_Constraint_Faulty_Circuit(F, K);
9.          Φ_SEED = Generate_Seed_Generation_Model (Φ_LFSR, Φ_GC, Φ_FC);
10.         (s , F_D) = PBO_Solver (Φ_SEED);
11.         S = S ∪ {s};
12.         F = F − F_D;
13.     }
14.     return S;
15. }
```

図 4. k 時間展開シード生成アルゴリズム

5. 実験結果

本章では，本手法の実験結果を示す．提案手法は C 言語で実装し，Core i7-13700(3.40GHz)およびメモリ 32.0GB を搭載したコンピュータを用いて実験を行った．対象回路は ISCAS’89 ベンチマーク回路である．使用した PBO ソルバーは CLASP[10]である．

実験に用いた RPR 故障集合の特定のための実験結果を表 2 に示す．実験では，擬似ランダムパターンテストによる故障検出回数が閾値を下回る場合，その故障を RPR 故障と判定した. LFSR の各シードから生成された 10000 個の擬似ランダムパターンを使用して故障シミュレーションを行い，RPR 故障集合を生成した．シードはランダムな 10 個のシードを用いた．表 2 では，”CUT”は回路名を示し，”#Input”は外部入力数と擬似外部入力数の総和を示し，”#SC”はスキャンチェイン数を示し，”#SLen”は最大スキャンチェイン長を示し，”#TP”は RPR 故障を判定するために用いた擬似ランダムパターン数を示し，”#DTth”は RPR 故障判定に用いた故障検出回数の閾値を示し，”FLT”は検出可能な故障の総数を示し，”#RPRF”は RPR 故障数を示す．

シード生成の実験結果を表 3 に示す．”CUT”は回路名を示し，”#TIME”は時間展開数を示し，”#RPRF”は RPR 故障数を示し，”#DT”は検出故障数を示し，”#UD”は未検出故障数を示し，”#SEED”は生成されたシード数を示し，”CPU”はシード生成時間を示す．時間展開をしない場合と比較して，シード数を最大 48.84%削減することができた．一方で，シード生成時間は時間展開数に応じて増加した．また，時間展開を行うことで検出できない RPR 故障が確認された．

6. まとめと今後の課題

本論文では，k 時間展開モデル用いたランダムパターンレジスタント故障のシード生成法を提案した．

実験結果では，時間展開を行わない場合と比較して，シード数を最大 48.84%削減したことを示した．

今後の課題として，シード生成時間の削減，完全な故障検出率の達成などが挙げられる．

表 2. 実験環境

CUT	#Input	#SC	#SLen	#TP	#DTth	#FLT	#RPRF
s5378	214	32	7	100,000	10	4,511	51

表 3. シード数

CUT	#TIME	#RPRF	#DT	#UD	#SEED	CPU[s]
s5378	1	51	51	0	43	5.82
	2		48	3	22	54.10
	3		48	3	23	200.29
	4		48	3	24	244.58
	5		48	3	25	275.58

参考文献

1） H. Fujiwara, Logic Testing and Design for Testability, MIT Press, 1985

2） M. Abramovici, M. A. Breuer, and A. D. Friedman, Digital Systems Testing and Testable Design, IEEE Press, 1990.

3） P. Bardell, W.H. McAnney and J. Savir: “Built-In Test for VLSI”, New York: Wiley-Interscience, 1987.

4） P. H. Bardell and W. H. McAnney, “Self-testing of multi-chip logic modules,” in Proc. IEEE Int. Test Conf., October 1982, pp. 200–204.

5） J. Rajski, N. Tamarapalli, and J. Tyszer, “Automated Synthesis of Phase shifters for Built-In Self-Test Applications,” IEEE Trans. Comput. Aided Design Int. Circuits & Syst., vol. 19, no.10, pp. 1175–1188, 2000.

6） S. Hellebrand, S. Tarnick, J. Rajski, and B. Courtois,” Generation of Vector Patterns Through Reseeding of Multiple-Polynomial Linear Feedback Shift Registers”, Proc. IEEE Int. Test Conf., Baltimore 1992, pp. 120-129.

7） T. Moriyasu and S. Ohtake, “A method of one-pass seed generation for LFSR-based deterministic/pseudo-random testing of static faults,” in Proceedings of Latin American Test Symposium, pp.1-6, 2015.

8） T. Larrabee, “Test Pattern Generation Using Boolean Satisfiability,” IEEE Trans. Comput. Aided Design Int. Circuits & Syst., vol. 11, no. 1, pp. 4-15, 1992.

9） 曽根隆暢，細川利典，吉村正義，新井雅之“組込み自己テストにおける両立可能故障集合を用いた複数ランダムパターンレジスタント故障のシード生成法”信学技報，vol123，no.304，DC2023-88，pp.7-12，2023.

10） M. Gebser, B. Kaufmann, A. Neumann, and T. Schaub, “Conflict Driven answer set solving,” Artificial Intelligence，vol.187-188， pp.52-89, Aug. 2012.