

擬似ブール最適化を用いた 複数ランダムレジスタント縮退故障のシード生成法

日大生産工(院) ○三浦 怜 日大生産工 細川 利典
京都産業大 吉村 正義

1. まえがき

近年、半導体技術の進歩により、大規模集積回路 (Large Scale Integrated circuits: LSI) が大規模化している。大規模化の影響により、多量化したテストパターンを外部テスト (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) [1-4] が広く用いられている。BISTは、被テスト回路 (Circuit Under Test: CUT) 内部に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、BISTにおけるテストパターン発生器 (Test Pattern Generator: TPG) としては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 (Seed) を与え、擬似ランダムパターンを発生させることにより、自己テストを行うことができる。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器 (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。この原因の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 [5] の存在があげられる。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れ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あることが重要である。

RPR故障の検出に有効なシードを生成する手法として、2パスシード生成法 [7,8] と1パスシード生成法 [9-11] が挙げられる。2パスシード生成法は2プロセスからなる。最初のプロセスでは、RPR故障を検出するテストパターンをATPGにより生成する。次のプロセスでは、LFSRに適用するために、生成したテストパターンをシードに変換する。しかしながら、テストパターンの生成とシード変換は独立して実行されるため、テストパターンはシードへの変換を前提として生成されていない。そのため、必ずしも生成されたテストパターンがシードへ変換できるとは限らず、故障検出率の損失が発生する。

2パスシード生成における故障検出率の損失を解決

するために、テストパターン生成とシード変換を1つの問題として解く1パスシード生成法 [10] が提案されている。CUTをシード生成モデルに変換し、商用のATPGを用いて直接シードを求める。シード生成モデルは、CUTとフェーズシフト付きLFSRで構成される。これにより、効率的にシードを生成することができる。しかしながら、一般的に使用される商用のATPGは経路活性化ベースのATPG [12][13] であり、テスト可能な故障であるか否かを判定するために長い時間を要する。一定時間を越えてもその判定ができない場合は、打ち切り (Abort) 故障となり故障検出率の損失が発生する。

この問題を解決するために、文献 [11] ではSATを用いた単一縮退故障BIST向けシード生成法が提案されている。従来の経路活性化ベースのATPGに代わり、打ち切り故障の検出に有効である充足可能性問題 (Satisfiability problem: SAT) [14] ベースのATPGを用いる。これにより、打ち切り故障に対してもシードを生成することが可能となり完全な故障検出率を可能とした。一方で、単一故障を目標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くシード数が必要とされる可能性がある。

本論文では、擬似ブール最適化 (Pseudo-Boolean Optimization: PBO) [14] を用いた複数ランダムレジスタント縮退故障のシード生成法を提案する。各シード生成時に目標とする故障を複数設定し、その検出数を最大化することで、少数のシードで高い故障検出率を達成することを目的とする。

本論文は以下のように構成される。第2章では、従来手法の課題を示し、3章ではPBOベースの複数目標故障テスト生成について説明し、4章で擬似ブール最適化を用いた複数ランダムレジスタント縮退故障のシード生成法を提案する。5章では、ITC99ベンチマーク回路を用いた評価を行う。6章でまとめと今後の課題を述べる。

2. 従来手法

文献 [11] ではSATベースのATPGを用いた1パスシード生成法が提案されている。SAT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。CUTとフェーズシフト付きLFSRから構成されるシード生成モデルをCNF変換し、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

A Seed Generation Method for Multiple Random Resistant Stuck-at Faults
Using Pseudo Boolean Optimization

Rei MIURA, Toshinori HOSOKAWA and Masayoshi YOSHOMURA

を選択し、目標とする。しかしながら、1つの故障を目標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くのシードを必要とする可能性がある。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あげられる。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、完全な両立可能故障集合[15]を目標故障集合として設定しなければならないため効率的ではない。そのため、提案手法ではPBOを用いて可能な限り多数の故障を検出するシード生成を行うことで、効率的にシード数を削減する。

3. PBO ベース複数目標故障テスト生成

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。すべてのブール制約を充足するブール変数割当てが存在する場合は充足可能(Satisfiability: SAT)であり、否である場合は充足不可能(Unsatisfiable: UNSAT)と呼ぶ。

一般的に、PBOベースのATPGではCUTにおいて、正常回路と故障回路からなるテスト生成モデルを構成する。故障回路は目標故障数用意する。故障検出の判定は、正常回路と故障回路の出力値を比較することで可能である。これを表現するために、故障回路に含まれる外部出力(Primary Output: PO)または擬似外部出力(Pseudo Primary Output: PPO)とそれに対応した正常回路のPO, PPOを入力としてXOR演算を行うモデルを構築する。例として、目標故障数2のテスト生成モデルをFig1に示す。故障f1の故障回路にはPO1, PPOの2つの出力があるため、各出力に対してXOR演算を行う。また、故障はいずれか1つの出力に伝搬ができれば十分である。そのため、2つのXOR演算出力をOR演算し、その出力Xが”1”になるような外部入力(Primary Input: PI)および擬似外部入力(Pseudo Primary Input: PPI)を決定することでテストパターンを定めることができる。故障f2では、故障回路に含まれる出力はPO2の単一であるため、PO2とPO2'をXOR演算し、その出力Yを”1”とする。

3.1. ゲート制約

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、論理回路を含むテスト生成モデルをブール制約で表現する必要がある。基本ゲートにおけるブ

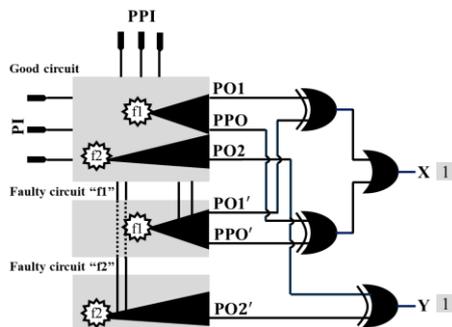


Fig1. テスト生成モデル(故障数2の場合)

Table1. 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	$\bar{x} + z = 1, x + \bar{z} = 1$
INV	x		z	$x + z = 1, \bar{x} + \bar{z} = 1$
XOR	x	y	z	$\bar{x} + \bar{y} + \bar{z} = 1$ $x + y + \bar{z} = 1$ $x + \bar{y} + z = 1$ $\bar{x} + y + z = 1$

ール制約表現をTable1に示す。表1に基づいて、正常回路および故障回路を制約へと変換する。

3.2. 故障検出制約

テスト生成モデルにおいて故障の検出を行うためには、故障励起条件と故障検出条件の2つが必要となる。この2つの条件を合わせて故障検出制約とする。信号線Aの0縮退故障を例とした場合では、正常値は”1”となるため、正常回路においてはA=1は必須割当てとなる。また故障値は”0”であるため、故障回路ではA=0となる。これらを制約で表現すると式(3.1)となる。Gは正常回路中の論理値を示し、Fは故障回路中の論理値を示す。同様に、1縮退故障の故障励起条件は式(3.2)で表現することができる。

$$G = 1, \bar{F} = 1 \quad (3.1)$$

$$\bar{G} = 1, F = 1 \quad (3.2)$$

$$G, F \in \{0,1\}$$

また、Fig1で示したように故障の検出を行うためにはXOR演算出力を”1”にすればよい。これを式(3.3)のように表現する。

$$XOR_{out} = 1 \quad (3.3)$$

$$XOR_{out} \in \{0,1\}$$

しかしながら、複数目標テスト生成(Multiple Target Test Generation: MTTG)において、複数存在する故障の故障検出制約を式(3.1), (3.2)および(3.3)で定式化した場合ではすべての故障を必ず検出しなければならない。そのため、目標故障集合中に両立不可能な故障が含まれていた場合には、すべて制約を充足する入力が存在しないため、”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をもつ故障についての故障検出制約は式(3.4), (3.5)および(3.6)で定式化する。Relaxは緩和変数を表す。式(3.1), (3.2)および(3.3)

部分が”UNSAT”である場合に、Relax = 1を割当ること、”SAT”となる。

$$\text{Relax} + G = 1, \text{Relax} + \bar{F} = 1 \quad (3.4)$$

$$\text{Relax} + \bar{G} = 1, \text{Relax} + F = 1 \quad (3.5)$$

$$\text{Relax} + \text{XORout} = 1 \quad (3.6)$$

$$G, F, \text{Relax} = \{0,1\}$$

3.3. 検出故障最大化を目的とした最適化関数

前節で説明した通り、目標故障集合中に両立不可能な故障が含まれていた場合は式(3.4)、(3.5)および(3.6)を用いることで解決することができる。しかしながら、ソルバーが優先的にRelax = 1を割当てた場合、故障検出要求が無視される。これを避けるために、最適化関数として式(3.7)を設定する。式(3.7)により、各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ことができる。

$$\text{Minimize} : \sum_{i=1}^N \text{Relax}_i \quad (3.7)$$

3.4. 目標故障集合の設定

本手法ではMTTGを行うため、目標故障集合の設定が重要である。SATを用いたMTTGを行う場合では、”SAT”にするために完全な両立可能故障集合を目標故障集合とする必要がある。一方で、PBOでは3.2節で説明した故障検出制約を用いることで、目標故障集合の設定が容易になる。本手法では、入力として与えられた故障集合中からランダムに1故障を選択し、絶対に検出すべき故障とする。残りの故障は、検出数を最大化すべき故障とする。すなわち、可能な限り検出すべき故障とする。

4. 提案手法

4.1 シード生成モデル

本手法で用いるシード生成モデルの例をFig2に示

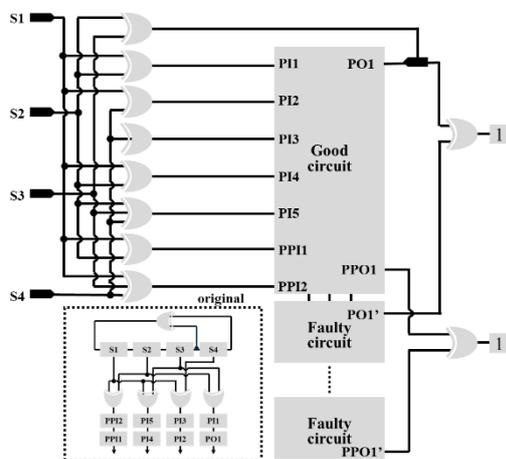


Fig2. シード生成モデル

す。テスト生成モデルの入力にフェーズシフタ付きLFSRを接続したモデルをシード生成モデルとする。そのため、テスト生成モデルの制約に加えて、PI, PPI とフェーズシフタから出力される論理値の関係性を示した制約が必要になる。フェーズシフタ付きLFSR から展開されるテスト系列は制約として表現することが可能であり、各PI, PPI との関係性を連立方程式で表す。Fig2におけるPPI1についての例を示す。PPI1に設定される論理値は1サイクル目におけるフェーズシフタ1の出力値であり、式(4.1)で表現できる。同様に、各PI, PPIは式(4.2)から式(4.7)の方程式が成り立つ。

$$S1 \oplus S2 = \text{PPI1} \quad (4.1)$$

$$S1 \oplus S3 \oplus S4 = \text{PPI2} \quad (4.2)$$

$$S1 \oplus S3 = \text{PI4} \quad (4.3)$$

$$S2 \oplus S3 \oplus S4 = \text{PI5} \quad (4.4)$$

$$S1 \oplus S4 = \text{PI2} \quad (4.5)$$

$$S4 = \text{PI3} \quad (4.6)$$

$$S1 \oplus S2 = \text{PI1} \quad (4.7)$$

4.2 全体のアルゴリズム

擬似ブール最適化を用いた複数ランダムレジスタント縮退故障のシード生成法のアルゴリズムをFig3に示す。

入力はネットリストC, 故障集合 F_{set} , スキャンチェーンSC, フェーズシフタPSとする。出力はシード集合 $Seed_{set}$ とする。 $Seed_{set}$ を \emptyset に初期化する(1行目)。PSおよびSCを用いて、フェーズシフタとPI, PPIに関する連立方程式 Φ_{PS} を生成する(2行目)。 F_{set} が \emptyset になるまで、4行目から8行目の処理を行う(3行目)。Cおよび F_{set} を用いて、テスト生成モデル Φ_{TPG} を生成する(4行目)。ソルバーに Φ_{PS} , Φ_{TPG} を入力として与え、変数の割当て結果 $solution$ を取得する(5行目)。 $solution$ を取得できたか否か、すなわち”SAT”, ”UNSAT”の判定を行う(6行目)。C, F_{set} および $solution$ を用いて故障シミュレーションを実行する。検出故障をドロップし、 F_{set} を更新する(7行目)。 $solution$ を $Seed_{set}$ に追加する(8行目)。最後に、 $Seed_{set}$ をリターンする(11行目)。

Algorithm Seed Generation for Multiple Random Resistant Stuck-at Faults Using Pseudo Boolean Optimization

Input : Circuit C, Fault set F_{set} , Scan chain SC, Phase-shifter PS

Output : Seed set $Seed_{set}$

1. $Seed_{set} = \emptyset$;
2. $\Phi_{PS} = \text{Create_Constraint_Phase_Shifter}(PS, SC)$;
3. while ($F_{set} \neq \emptyset$) then
4. $\Phi_{TPG} = \text{Create_Test_Pattern_Generation_Model}(C, F_{set})$;
5. $solution = \text{Slover}(\Phi_{PS}, \Phi_{TPG})$;
6. if ($solution == \text{”SAT”}$) then
7. $F_{set} = \text{Fault_Simulation}(C, F_{set}, solution)$;
8. $Seed_{set} = Seed_{set} \cup solution$;
9. endif
10. endwhile
11. return $Seed_{set}$;
12. End

Fig3. 全体アルゴリズム

5. 実験結果

ITC99ベンチマーク回路を用いた本手法の実験結果を示す。使用したソルバーはSCIP[16]である。

まず、Table2に実験に用いたベンチマーク回路の回路特性を示す。CUTは回路名を示す。#PIsは外部入力数、#POsは外部出力数、#FFsはフリップフロップ数を示す。#SCsはスキャンチェーン数、SLは最大スキャンチェーン長を示す。実験に用いたフェーズシフタ付きLFSRは文献[17]を基に設計されたものである。各回路で用いるLFSRは#SCsのビットで設計される。また、フェーズシフタの最低タップ数は2に設定した。Table3に実験結果を示す。#TPsはRPR故障を求める際に用いた擬似ランダムパターン数である。#RPRsは擬似ランダムパターンによって未検だった故障数を示す。#Seedsは生成されたシード数を示す。#DFsは#RPRsに対する検出故障数である。CPUはシード生成時間である。

シード数を平均30%削減することができた。また、シード生成時間は、b05回路において最大で7倍増加した。

6. むすび

本論文では、擬似ブル最適化を用いた複数ランダムレジスタ縮退故障のシード生成を提案した。文献[11]と比較して、故障検出率を損失することなく、シード数を平均30%削減した。

今後の課題として、さらなるシード数の削減を目的とし、シードから生成される1番目のテストパターンのみではなく、2番目以降にテストパターンを考慮することが挙げられる。

参考文献

- [1] 藤原秀雄, デジタルシステムの設計とテスト, 工学 Fig 書株式会社
- [2] T. Hiraide, K.O. Boateng, H. Konishi, K. Itaya, M. Emori, H.Yamanaka, and T. Mochiyama, "BIST-aided scan test-A new method for test cost reduction," Proc. VTS, pp. 359-364, 2003..
- [3] P. H. Bardell, W. H. McAnney and J. Savir, Built-In Pseudo-Random Testing of Digital Circuits, Chapter 8. John Wiley & Sons, New York, 1987.
- [4] R. Chandramouli, S. Pateras, "Testing Systems on a Chip," IEEE Spectrum, Vol. 33, No. 11, pp. 42-47, November 1996.

Table2. 回路特性

CUT	#PIs	#POs	#FFs	#SCs	SL
b01	2	2	5	4	2
b02	1	1	4	4	2
b05	1	26	34	4	9

Table3. 実験結果

CUT	#TPs	#RPRs	Conventional[11]			Proposed		
			#Seeds	#DFs	CPU[s]	#Seeds	#DFs	CPU[s]
b01	10	11	4	7	0.13	4	7	0.15
b02	10	6	4	6	0.02	2	6	0.10
b05	10,000	102	5	5	5.02	2	5	30.59

- [5] M. Abramovici, M. A. Breuer, and A. D. Friedman, Digital Systems Testing and Testable Design. IEEE Press, 1990.
- [6] S.Hellebrand, B.Reed, S.tarnic, and H.-J.Wunderloch, "Pattern generation for a deterministic BIST scheme"Proc.IEEE International Conference Computer Aided Design, pp.146-149, 1984.
- [7] B. Konemann, "LFSR-coded test patterns for scan designs," in Proc. of European Test Conference, pp.237-242, 1991.
- [8] S. Venkataraman, J. Rajski, S. Hellebrand, and S. Tarnick, "An efficient BIST scheme based on reseeding of multiple polynomial linear feedback shift registers," in Proc. of International Conference on Computer-Aided Design, pp.572-577, 1993.
- [9] 佐脇光亮, 大竹哲史, "階層 BIST 向け LFSR シード生成法," 電子情報通信学会技術報告 (DC2014-85), Vol.114, No.446, pp.43-48, Feb. 2015.
- [10] T. Honda and S. Ohtake, "A method of LFSR seed generation for delay fault BIST using constrained ATPG," in Digest of Papers of 15th IEEE Workshop on RTL and High Level Testing, pp.20-25, 2014.
- [11] 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.
- [12] Roth, J. P. :Programmed Algorithms to Compute Tests to Detect and Distinguish between Failures in Logic Circuits, IEEE Trans. 1967.
- [13] M. henfiling, H. C. Wittmann, K. J. Antrich, "A Single-Path-Oriented Fault-Effect Propagation in Digital Circuits", IEEE/ACM International Conference, Nov, 1995, pp.304-309.
- [14] V.Manquinho, R.Martins, and I.Lynce, "Theory and Applisations of Satisfiability Testing -SAT 2010" in Proc.13thInternational Cnference., pp. 181-193, 2010.
- [15] 松永裕介, "SAT ソルバによる両立故障集合検査を用いたテストバタン圧縮手法について", Design Automation Symposium, 2015
- [16] Zuse Institute Berlin: SCIP (Version 8.0.0). Zuse Institute Berlin, 2022. <https://www.scipopt.org/doc/html/>
- [17] Janusz Rajski, Associate Member, IEEE, Nagesh Tamarapalli, and Jerzy Tyszer, Senior Member, IEEE, "Automated Synthesis of Phase Shifters for Built-In Self-Test Applications", IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, VOL. 19, NO. 10, OCTOBER 2000.