

ランダムパターン耐性故障集合生成のための一考察

日大生産工（院） ○小泉優哉 日大生産工 細川利典 京産大 吉村正義

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、複雑化により、多量化したテストパターンをテスト(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)[1,2,3]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み込み、故障の検出を行う。一般的に、TPGとしてはフェーズシフト[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トに入力することで規則性を緩和した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、ランダムパターン耐性(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。しかしながら、RPR故障の判定には複数の判定基準が使用されており[6, 7]、その基準によつてのRPR故障の集合が異なる。本論文では、RPR故障を客観的な尺度に基づいて定義することが目的である。ある故障を検出するすべてのテストパターン数を分子とし、回路に入力可能なすべてのテストパターン数を分母とすることで、ある故障の故障検出確率を計算する。故障検出確率が低い故障をRPR故障と定義する。また、完全に一様分布するランダムパターンが印加された際のテスト対象回路全体の故障検出確率についての評価指標と、テスト対象回路に印加される擬似ランダムパターンのランダム性についての考察を述べる。

本論文は以下のように構成される。第2章では回路構造解析について述べる。第3章ではBDDを用いた真理値表密度計算について述べる。第4章では故障検出確率と擬似ランダムパターン評価指標について述べる。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. 回路構造解析

本章では、RPR故障判定の前処理として、故障と外部入出力との依存関係を解析するためにトランジティブファンアウト[8](Transitive Fan-Out: TFO)および

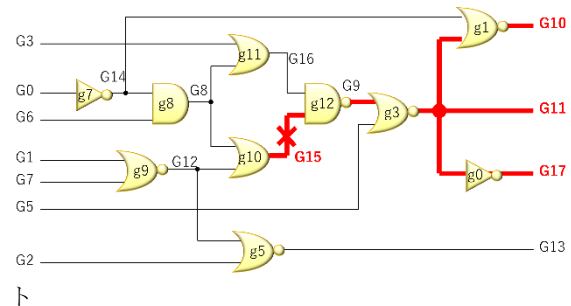


図1. 故障信号線 G15 の TFO

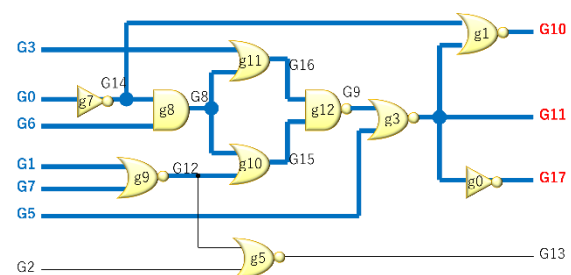


図2. 故障信号線 G15 の TFO に属する各外部出力の TFI

ランジティブファンイン(Transitive Fan-In: TFI)[8]を用いてテスト生成モデルに含まれる外部入力数の算出方法を述べる。故障の故障箇所から出力方向に故障観測箇所、つまり外部出力へ到達する範囲であるTFOを計算し、そのTFOに属する各外部出力から外部入力方向へ到達する範囲、すなわちTFIを計算して、その故障のテストに関係する外部入力数の割合を求める。回路全体の外部入力数のうち、一定の割合以下のものは、テストパターン探索の際に考慮すべき入力変数が少ないということであり、ランダムに生成されるパターンでも故障を検出する確率が相対的に高くなるため、RPR故障判定対象としない。

図1は故障箇所であるG15におけるTFOの範囲を示している。故障信号線G15は外部出力G10, G11, G17の3つの外部出力に影響を与えることを示している。図2は図1で求められたTFOに属する各外部出力から入力方向に到達する範囲を示している。TFOに属する外部出力であるG10, G11, G17から入力方向に到達する外部入力はG0, G1, G3, G5, G6, G7の6つであることを示している。この例題回路は外部入力数7本の回路であるが、そのうち故障信号線G15のテストに関係する外部入力数は6本であり、故障信号線G15のテストに関係のない外部入力数は1本となる。つまり約86%の外部入力が、この故障をテストする際に関係する可能性があるということがわかる。この割合が一定以下の場合、その故障を検出するテストパターン数の割合は多いと考えられ、RPR故障判定対象外とする。例えば回路の外部入力数のうち、ある故障のテストに関係する外部入力数が1%未満である

場合は、その故障は RPR 故障判定対象外となるという閾値を設定した場合、故障信号線 G15 の場合はテストに關係する外部入力数が約 86%なので、この故障は RPR 故障判定対象となる。

3. BDD を用いた真理値表密度計算

本章では、提案手法で用いる BDD を用いた真理値表密度計算について述べる。3.1 節では BDD について述べ、3.2 節で BDD を用いた充足解探索と真理値表密度計算について述べる。

3.1. 二分決定グラフ(BDD)

BDD は論理関数の正規表現をグラフで表現したものである。例えば、論理関数 $f = A \vee B\bar{C}$ において $A = 1$ とおくと、 $f = 1$ となる。 $A = 0$ の場合には、 $B = 0$ とすると、 $f = 0$ となる。また、 $A = 0$ かつ $B = 1$ の場合には、 $C = 0$ の時、 $f = 1$ となり、 $C = 0$ の時 $f = 0$ となる。図 3 は、この手続きを決定グラフで表したものである。このとき、A, B, C は非終端節点で、入力変数に対応する。また、0, 1 は終端節点で、論理関数値に対応する。このような決定グラフを BDD という。

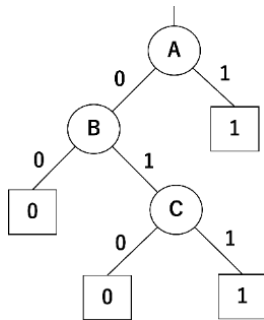


図 3. $A \vee B\bar{C}$ の BDD

表 1. $f = A \vee B\bar{C}$ の真理値表

A	B	C	$f = A \vee B\bar{C}$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

図 3 に示した論理関数 $f = A \vee B\bar{C}$ に対して表 1 に心理表を示す。各入力変数 A, B, C のすべての組合せに対して関数値を列挙した結果、出力が 1 となる入力の組合せは 5 通りである。これらは、図 3 に示す BDD において根から終端節点 1 に到達するすべての経路に対応している。すなわち、BDD は真理値表における各入力割当てに対する論理関数値を等価に表現しており、真理値表の各行を経路としてグラフを表現している。

3.2. 充足解探索と真理値表密度

本節では、BDD を用いることによる充足解探索と真

理値表密度について述べる。ある論理関数を表現する BDD を構成したとき、その論理関数が充足可能(1 である)であるか否かの判定が可能である。これは BDD のインデックスが 1 定数節点を持つ場合に、その論理関数は充足可能であることを表現するためである。そして BDD が充足可能である場合に、根から 1 終端節点までの経路をたどることで、容易に充足解を求めることができる。さらに充足解の個数のカウントを行うことで真理値表密度を計算することもできる。図 3 の BDD を例に考える。真理値表密度は、各ノードにおいて、そのノード以下で関数が 1 となる確率を下位ノードから順に計算し、最終的に根ノードの値として求められる。この根ノードの値が、論理関数の真理値表密度となる。ノード C は左右の子が終端節点 0 と 1 であるため、ノード C が 1 となる確率は 0.5 である。次にノード B では、0 枝が終端節点 0、1 枝がノード C に接続しているため、ノード B が 1 となる確率は 0.25 となる。最後に根ノード A では、0 枝がノード B、1 枝が終端節点 1 に接続しているので、根ノード A が 1 となる確率は $0.625(0.125+0.5)$ となる。したがって、関数 $f = A \vee B\bar{C}$ の真理値表密度は 0.625 であり、全入力パターンの 8 通りのうち、5 通りの入力で出力が 1 になることを意味する。このように BDD を用いることで充足可能性の判定だけでなく、真理値表密度を求めることができる。

4. 故障検出確率計算と擬似ランダムパターン評価指標

本章では、故障検出確率計算と擬似ランダムパターン評価指標について述べる。4.1 節で故障検出確率計算法について述べ、4.2 節で故障検出確率計算アルゴリズムについて述べる。4.3 節で擬似ランダムパターンの評価指標について述べる。

4.1. 故障検出確率計算法

本節では、故障検出確率計算法について述べる。

テスト対象となる回路の対象となる故障モデルの各故障の故障検出確率に基づいて、回路全体の故障集合の故障検出確率を計算する。ここで各故障の検出が独立であると仮定すると、完全に一様分布するランダムパターンをパターン数 N_p 分だけ印加された時のテスト対象回路全体の故障検出確率 F は、各故障の故障検出確率から求められる。算出式を式(1)に示す。ここで式(1)において、テスト対象回路の故障数を N_f 、ある故障 f_i の故障検出確率を $D(f_i)$ とする。この故障検出確率 $D(f_i)$ は、故障 f_i を検出するすべてのテストパターン数を分子とし、回路に入力可能なすべてのテストパターンを分母とする。テスト時に入力可能な外部入力数を N_I とすると、分母の値は 2^{N_I} となる。分子の値は故障 f_i を検出するすべてのテストパターン数である。

式(1)の具体的な計算例として、テスト対象回路に含まれる故障数を 3 とし、それぞれの故障検出確率を 0.10, 0.30, 0.50 と仮定する。印加するランダムパターン数を 10 個とした場合、各故障の検出確率に基づいて求められる回路全体の検出確率は、およそ 0.87 となる。印加する N_p の数を増加させることで、各故障検出確率は単調に増加する。十分な数のランダムパターンを印加すれば、テスト対象回路全体の故障検出確

率 F が1に近づく。

$$F = \frac{1}{N_f} \sum_{i=1}^{N_f} (1 - ((1 - D(f_i))^{N_p})) \quad (1)$$

4.2. 故障検出確率計算アルゴリズム

本節では、各故障の故障検出確率計算アルゴリズムについて述べる。

図4にある故障を検出するすべてのテストパターン数を求める手順を示す。はじめにテスト生成の制約を空にする(step1)。制約下で、ある故障を検出するテストパターンを生成する。もしテストパターン生成不能の場合はstep6へ進む(step2)。テストパターン中のケアビットに対して、ドントケア判定を行い、テストキューブを求める(step3)。テストキューブから同一のテストパターンを生成しないための禁止制約を生成する(step4)。step4で生成した禁止制約をテストパターンの制約に追加し、step2へ戻る(step5)。テストキューブの集合から、故障を検出するすべてのテストパターン数を求める(step6)。ここでstep2は、既存のSATソルバーを用いたテストパターン生成で実装する。SATソルバーとして使用するのはclasp3.3.8[6]を用いる。SATソルバーを用いるため、生成されるテストパターンに様々な制約を追加することは容易である。またstep6のテストキューブ集合からテストパターン数を算出するために、BDDを用いた真理値表密度の計算を行う。

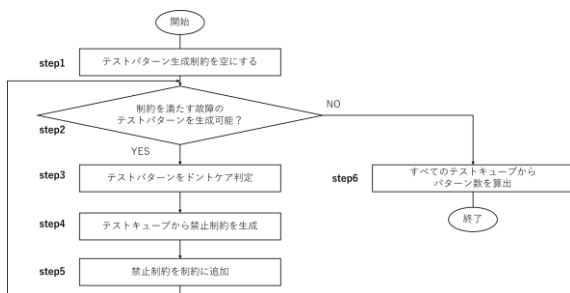


図4. 故障検出確率計算フロー

4.3. 擬似ランダムパターン評価指標

本節では、擬似ランダムパターン評価指標について説明する。

擬似ランダムパターン集合から完全に一様分布するランダムパターンとの差異を示すためのランダム性に関する指標について述べる。この指標は、ランダム性が高い場合は、その擬似ランダムパターン集合に対する実際の故障検出確率と、式(1)によるテスト対象回路全体の故障検出確率と近い値になる確率が高くなり、ランダム性が低い場合には、近い値になる確率が低くなる性質を持つ指標となる。この指標によって、テスト対象回路に印加される擬似ランダムパターンのランダム性の質を示すことができる。この指標はテスト対象回路の特性に左右されないため、擬似ランダムパターン生成器で生成されたテスト対象回路に印加されるランダムパターン集合を評価することが可能である。

5. 実験結果

本章では、本論文で示した故障検出確率を計算する、

前の予備実験である回路構造解析結果を示す。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。

表2に回路構造解析の実験結果を示す。回路名はテスト対象回路名である。外部入力数は各テスト対象回路の総外部入力数である。総故障数は全故障集合から等価故障と冗長故障を除いた故障数である。RPR判定対象故障数は回路構造解析においてテストに関係する外部入力数が設定した割合を超える故障の数である。本実験で設定した割合は総外部入力数の1%である。非RPR故障数は回路構造解析においてテストに関係する外部入力数が設定した割合を下回る故障の数を示す。実行時間は回路構造解析の総実行時間を示す。

表2の結果を棒グラフとして可視化したものを図5から図10に示す。各図では、各故障に対してテストに関係する外部入力数を縦軸に示しており、この数を基準に降順ソートしている。

実験結果から、ISCAS'89ベンチマーク回路における大規模回路において、削減効果が顕著であることがわかる。特にs35932では約60.2%の故障が非RPR故障判定され、s38584においても約43.4%が非RPR故障判定となった。これは、大規模回路において局所的なTFO、TFIを持つ故障が多数存在することを意味する。つまり、回路規模が増大するにつれて、この回路構造解析によるRPR故障判定対象削減効果が高いと考えられる。これらの故障は、ごく少数の外部入力変数にのみ依存するため、ランダムパターンによる検出が容易であり、RPR故障となる確率が低いと推定される。これらの故障を事前解析により除外することで、RPR故障判定が高速化される。

以上の結果から、提案手法に用いる回路構造解析は、RPR故障検出確率計算における前処理として有効であり、BDDおよびSATソルバーを用いた確率解析の効率化に寄与することができる。

表2. 回路構造解析実験結果

回路名	外部入力数	総故障数	RPR判定対象故障数	非RPR故障数	実行時間(s)
s5378	214	4511	4391	120	0.04
s9234	247	6475	6309	166	0.18
s13207	700	9512	6923	2589	8.27
s15850	611	11308	9351	1957	14.18
s35932	1763	34534	13732	20802	680.77
s38417	1664	30579	21882	8697	592.40
s38584	1464	34489	19506	14983	633.30

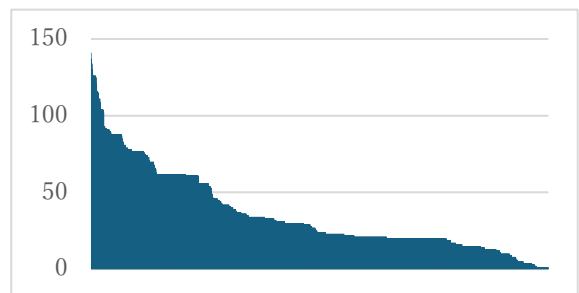


図5. s5378実験結果

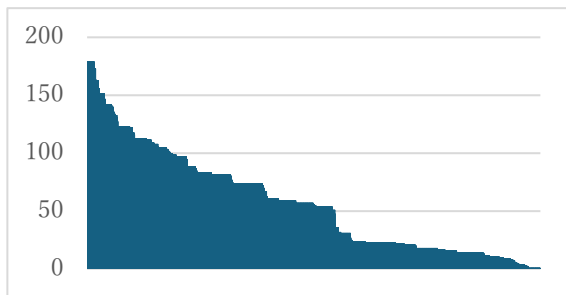


図 6. s9234 実験結果

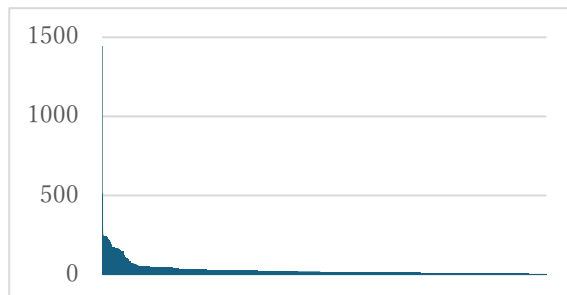


図 10. s35932 実験結果

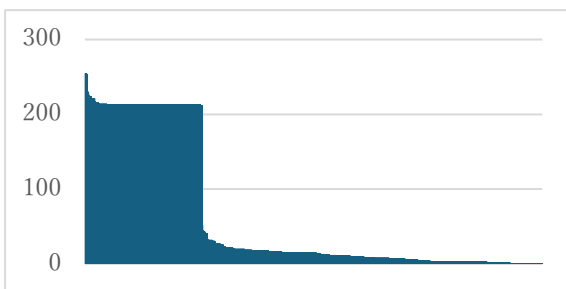


図 6. s13207 実験結果

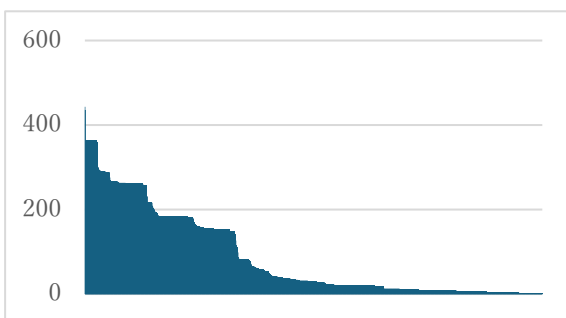


図 7. s15850 実験結果

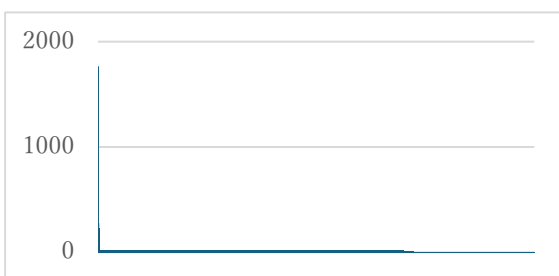


図 8. s35932 実験結果

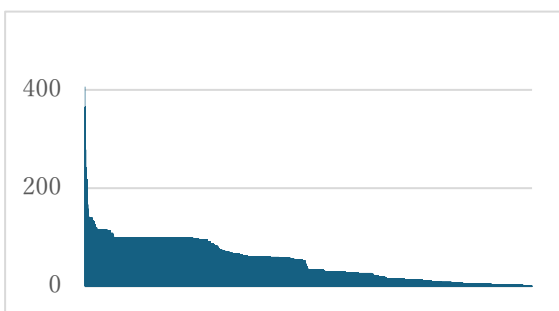


図 9. s38417 実験結果

6. まとめと今後の課題

本論文では、SAT ソルバーによるパターン列挙および BDD を用いた真理値表密度計算を組み合わせ、RPR 故障を確率的に定義する新たな評価指標を提案した。

今後の課題として、SAT ソルバーを用いたテスト生成において制約の追加回数が膨大になってしまうことや、大規模回路に対して論理関数を BDD で表現可能であるかという課題などが挙げられる。また、これらのモデル確立後に、ランダム性の指標が低い擬似ランダムパターン生成回路に対して、故障検出率を高める手法についての検討も課題として挙げられる。

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. Rajsiki, 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] T. Sone, M. Yoshimura, R. Miura, M. Arai, and T. Hosokawa, “A Multiple Target Seed Generation Method for Random Pattern Resistant Faults on Built-In Self-Test,” Proc. IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS), pp. 1–6, 2024.
- [7] J. Savir, S. Patil, and P. H. Bardell, “Random Pattern Testability,” IEEE Transactions on Computers, vol. C-33, no. 1, pp. 79–90, Jan. 1984.
- [8] Y. Matsunaga, “An Accelerating Technique for SAT-based ATPG,” IPSJ Transactions on System LSI Design Methodology, vol. 10, pp. 39–44, Feb. 2017.
- [9] M. Gebser, R. Kaminski, B. Kaufmann, J. Romero, and T. Schaub, “Progress in clasp series 3,” Proc. of LPNMR 2015, pp. 368–383, 2015.