

SATエンジンを用いた遷移故障テスト生成の高速化

日大生産工 ○藤井 拓磨 日大生産工 山崎 紘史
日本生産工 細川 利典 京都産大 吉村 正義

1. はじめに

近年の半導体微細化技術の進歩に伴って、超大規模集積回路(Very Large Scale Integrated circuits:VLSI)が大規模化・複雑化し、テスト生成が困難になってきている。テスト生成の対象となる回路は、回路の外部出力値が外部入力値みに依存する組合せ回路と、回路の外部出力値が外部入力値と回路内部のフリップフロップ(Flip-Flop:FF)に依存する順序回路の2種類に大きく分けられる[1]。

組合せ回路に対するテスト生成は D アルゴリズム [2], PODEM3 [3], FAN [4], SOCRATES [5] といった経路活性化法を用いることで可能である。しかしながら順序回路の場合、FF によるフィードバックループを考慮する必要があるため、組合せ回路に対するテスト生成と比べて困難とされている [1]。そこで FF を擬似外部入力(出力)と見なし、順序回路を時間展開することで組合せ回路として行う手法 [6] や、制御信号により FF を直列のシフトレジスタとして動作させるスキャン設計を用いた手法 [1] 等が提案されている。

一方、近年では衝突節の学習 [7] や非辞書式バックトラック [8], ブール強制伝搬 [8] により、SAT(Satisfiability problem)-solver の処理速度が急速に進歩している。また SAT を用いたテスト生成も提案されており [9], 近年では組合せ回路に対して既存のテスト生成技術より高速であることが示されている [10]。

本論文では、高速な SAT-solver 及び文献 [10] の高速化技術の対象を組合せ回路から順序回路へ拡張し、遷移故障に対する高速なテスト生成手法を提案する。

2 章では SAT を用いたテスト生成について説明し、

第3章では本論文が扱う遷移故障モデルに対する SAT を用いたテスト生成の提案手法を述べ、第4章に実験結果を示す。

2. SAT を用いたテスト生成

本章では組合せ回路に対する SAT を用いたテスト生成の基本的な手法について説明する。

SAT とは充足可能性問題を示し、これは乗法標準形 (Conjunctive Normal Form: CNF) が与えられたときに、CNF に含まれる全ての変数の値を 1(真)または 0(偽)に定めることで、CNF 全体の値を 1(真)にできる割当てが存在するか否かを判定する問題である。CNF を 1(真)にできる割当てが存在した場合を充足可能といい、存在しない場合を充足不能という。

SAT は CNF を入力として問題を解決するため、SAT を用いてテスト生成を行うためには、対象の論理回路を CNF に変換する必要がある。論理回路を構成する各ゲートの CNF 変換規則を表 1 に示す。各論理ゲートは表 1 の規則に従い、CNF に変換する。それぞれの括弧で括られた論理式を節といい、節を論理積した式が各ゲートの CNF である。回路全体の CNF は各論理

表 1. 論理ゲートの CNF 変換規則

ゲートタイプ	入力	出力	ゲートに対するCNF式
AND	X, Y	Z	$(X + \neg Z) \cdot (Y + \neg Z) \cdot (\neg X + \neg Y + Z)$
OR	X, Y	Z	$(\neg X + Z) \cdot (\neg Y + Z) \cdot (X + Y + \neg Z)$
NOT	X	Y	$(X + Y) \cdot (\neg X + \neg Y)$
BUFF	X	Y	$(\neg X + Y) \cdot (X + \neg Y)$
EXOR	X, Y	Z	$(\neg X + Y + Z) \cdot (X + \neg Y + Z) \cdot (\neg X + \neg Y + Z) \cdot (X + Y + \neg Z)$

On the Acceleration of Test Generation for Transition Faults
Using SAT Solver

Takuma FUJII, Hiroshi YAMAZAKI, Toshinori HOSOKAWA
and Masayoshi YOSHIMURA

ゲートの CNF を論理積することで表現できる。

SAT を用いたテスト生成では、対象回路において正常に動作する正常回路と、故障の影響を仮定した故障回路の 2 つを用意する。この正常回路と故障回路を組み合わせることで、テスト生成モデルとなる合成回路を作成する。図 1 に、合成回路の例を示す。

合成回路の外部入力数は対象回路の外部入力数と同一であり、それぞれが対応する正常回路と故障回路の外部入力と接続している。また、合成回路の外部出力数は 1 である。合成回路の外部出力は、正常回路と故障回路の対応する各外部出力を EXOR 演算し、その各 EXOR 演算の出力を OR 演算したものとなっている。このため、故障の影響が外部出力に伝搬され故障を検出できる時、合成回路の外部出力の値は 1 となる。SAT を用いたテスト生成は、この合成回路の出力が 1 となるような外部入力の組合せを求める。この時、充足可能であればテスト生成が可能で、充足不能であればテスト生成は不可能、つまり冗長故障[1]と判定される。

また、図 1 において故障回路における三角部分である①は故障の影響が伝播する可能性がある部分回路、①を含んだ台形部分である②は故障の影響を伝搬させるために必要な部分回路としている。

故障回路において故障の影響は①のみ考えられる。また、故障箇所に至る経路及び故障の伝搬は正常時と故障時で同じ振る舞いをしており、故障回路において正常回路と同じ CNF 変換を行う部分が混在する。さらに故障の影響及び伝搬に関係の無い部分は、テスト生成に必要ない。

以上を考慮したテスト生成用の合成回路を図 2 に示す。

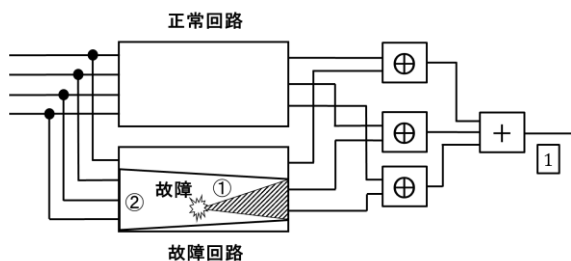


図 1. テスト生成用の合成回路

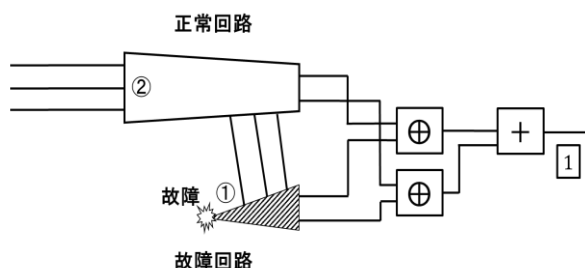


図 2. テスト生成対象の合成回路

図 2 に示すように故障回路を①に限定し、①の正常時も含めた②を正常回路とすることで、テスト生成において故障の影響に関係のない部分、つまりテスト生成において不必要な部分及び重複する部分を削除できるので、回路規模が縮小されて CNF データ量の削減ができる[9][10]。また、CNF データ量の削減により CNF 変換に要する時間も短縮できる。ただし、故障回路において故障を伝搬させるための入力は正常回路から入力する。

また、テスト生成を効率良く行うために、故障が伝搬されたことを示す変数の導入[9]、複数個所の故障を 1 つの CNF にまとめ、段階的に回路を CNF に変換するテスト生成法[10]が提案されている。

3. 遷移故障に対する SAT によるテスト生成

遷移故障[1]のテスト生成手法として、2 パターンテストが提案されている。また 2 パターンテストを用いた実速度スキャンテスト法として、スキュードロード方式[11][12]とブロードサイド方式[13]が提案されている。本論文ではフルスキャン設計された回路を対象に、SAT を用いたブロードサイド方式による遷移故障テスト生成を対象とする。

図 3 に、ブロードサイド方式による立下り遷移故障の検出例を示す。まず順序回路に対して 2 時間展開を行い、各時刻の FF の出力を回路の擬似外部入力、FF の入力を回路の擬似外部出力とし、FF を介した 2 つの組合せ回路としている。次に図 3 のような立下り遷移故障を検出する場合、1 時刻目で故障個所に 1 を設定し、2 時刻目では故障個所に 1 縮退故障[1]を設定する。このように、縮退故障モデルを用いることで、

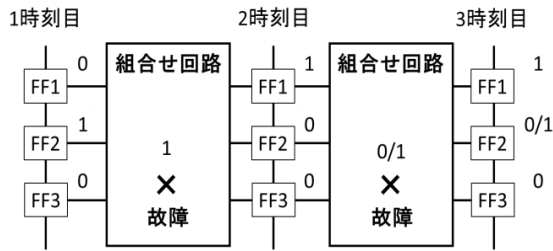


図 3. ブロードサイド方式による故障検出例

ブロードサイド方式の順序回路に対してテスト生成を行うことができる。

SAT を用いたブロードサイド方式による遷移故障テスト生成の提案手法として、図 3 に示す 2 時間展開した組合せ回路のモデルに対して CNF 変換を行い、文献[10]の高速化技術を適用する手法を提案する。

2 時刻目の組合せ回路は 1 縮退故障に対するテスト生成と同等なので、2 章で述べた合成回路を作成する。ただし順序回路であるため、擬似外部入力及び擬似外部出力の FF が組合せ回路に付与されることを考慮して合成回路の作成し、テスト生成を行う。

1 時刻目の組合せ回路も故障を仮定しているが、この故障は遷移故障のための固定値(図 3 では立下り遷移故障のために故障箇所を 1 とする)なので、テスト生成の対象ではない。これにより 1 時刻目の組合せ回路に対しては、固定値の制約を付与した SAT による正当化操作となる。

以上のように 2 時間展開した組合せ回路を CNF 式に変換し、SAT を用いたブロードサイド方式による遷移故障テスト生成を行う。

提案手法の全体アルゴリズムを図 4 に示す。

(Step1)

テスト生成を行う対象回路のデータを読み込む。

(Step2)

Step1 で読み込んだ回路に対し等価故障解析を実行し、代表故障[1]のリストを作成する。

(Step3)

代表故障リスト内の全故障に対してテスト生成を行ったかを判断する。生成し終わった場合は終了し、それ以外のときは Step4 へ進む。

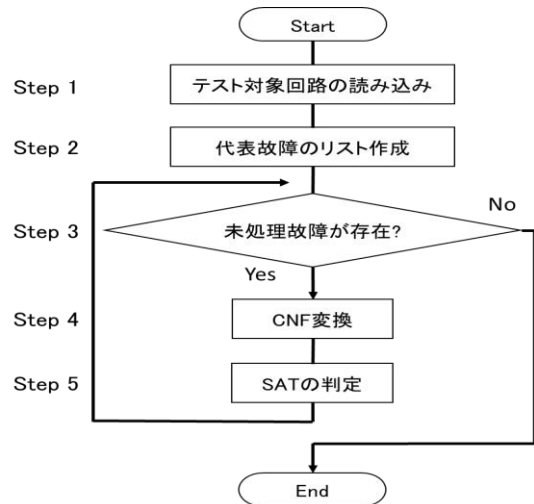


図 4. 提案手法の全体アルゴリズム

(Step4)

対象回路を 2 時間展開して CNF 変換を行う。ただし 2 時刻目の組合せ回路は、2 章で述べた合成回路に FF を考慮した合成回路とする。

(Step5)

Step4 で生成した CNF に対して文献[10]の高速化技術を用いて、SAT の判定を行う。充足可能なら検出可能な故障、充足不能なら冗長故障と判断する。

4. 実験結果

本章では、3 章で提案した手法に対する評価に関する実験について述べる。実験として、組合せ回路に対して 2 章で述べた SAT を用いたテスト生成に文献[10]の高速化技術を加えた SAT を用いたテスト生成と、既存の SAT を用いたテスト生成[14]とテスト生成時間を比較した。対象回路は ISCAS' 85 ベンチマーク回路、使用計算機は Intel Core i7-4770(3.40GHz)、対象故障モデルは単一縮退故障としている。

表 2 に上記の実験結果を示す。"回路名"は実験対象とした回路名、"文献[10]"は文献[10]の高速化技術によるテスト生成時間、"SAT-ATPG"は既存の SAT を用いたテスト生成によるテスト生成時間をそれぞれ示している。各テスト生成時間の単位は秒(s)としている。

いずれのテスト生成時間も文献[10]の高速化技術を備えた結果の方が高速であることがいえる。

表 2. 実験結果

回路名	文献[10](s)	SAT-ATPG(s)
C1355	0.05	1.74
C1908	0.10	5.07
C2670	0.18	6.15
C3540	0.43	1.17
C5315	0.46	17.32
C6288	0.39	61.19
C7552	0.72	26.69

実験結果より, 既存の SAT によるテスト生成よりも高速である文献[10]の高速化技術を用いれば, 順序回路に対しても高速にテスト生成が可能と考えられる。

5. 終わりに

本論文では, 組合せ回路に対する SAT を用いたテスト生成について述べ, 順序回路における遷移故障に対してブロードサイド方式による SAT を用いたテスト生成の手法を提案した。実験結果より, [10]の高速化技術を用いた SAT によるテスト生成ならば, 順序回路に対して高速にテスト生成が行える可能性を示した。

今後は, 提案したテスト生成手法を[10]の高速化技術を備えた SAT-Solver を用いて実装し, 実験・評価を行う。

参考文献

- [1]藤原秀雄, “デジタルシステムの設計とテスト” (2004)
- [2] H. Fujiwara and T. Shiono, “On the acceleration of test generation algorithms,” IEEE Trans. Comput., vol. C-31, pp. 1137-1144, 1983.
- [3] P. Goel, “An implicit enumeration algorithm to generate tests for combinational logic circuits,” IEEE Trans. Comput., vol. C-31, pp.215-222. 1981.
- [4] J. P. Roth, “Diagnosis of automata failures: A calculus and a method,” IBM J. Res. Develop., vol. 10, pp. 278-291, 1966.
- [5]M. H. Schulz, E. Trischler, and T. M. Sarfert, “Socrates: A highly efficient automatic test pattern generation system,” IEEE Trans. Computer-Aided Design, vol. 7, pp. 126-137, Jan. 1988.
- [6]T.Inoue, T.Hosokawa, T.Mihara and H.Fujiwara, “An Optimal Time Expansion Model Base on Combinational ATPG for TR Level Circuits,”IEEE Asian Test Symp., Vol.39,No4,pp.190-197, Apr.1998.
- [7]Matthew W. Moskewicz, “Chaff: Engineering an Efficient SAT Solver”, IEEE xplore (2001)
- [8]Joao P. Marques-Silva, “GRASP: A Search Algorithm for Propositional Satisfiability”, IEEE TRANSACTIONS ON COMPUTERS (1999)
- [9]T. Larrabee. Test pattern generation using Boolean satisfiability. IEEE Trnsactions on Computer-Aided Design of Integrated Circuits, Vol. 11, No.1, pp.4-15, 1992.
- [10] 松永祐介 “SAT ソルバを用いた高速なテスト生成手法の開発” (2014)
- [11] J.Savir. “Skewd-Load Transition Test: Part1: Cakulus.” Proceedings of IEEE International Test Conference, pp 705-713 Oct.1992.
- [12] J.Savir. “Skewd-Load Transition Test: Part2: Cakulus.” Proceedings of IEEE International Test Conference, pp 714-722 Oct.1992.
- [13] J.Savir and S.Patil “On Borad-Side Delay Test” VLSI Test Symposium, pp.284-290 Sept.1994
- [14] 山崎紘史, “充足可能性問題の解法を用いた縮退故障モデルのテスト生成に関する研究” (2009)