

多重目標故障テスト生成の一考察

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) の高集積化・複雑化が進んでいる。これに伴い、設計された VLSI に対するテストパターン数及びその生成時間も増大し、テスト生成[1]は益々困難な課題となってきた。特にテストパターン数の増加はテストコストの増加に直結するため、テストパターン数の削減のために様々な手法が提案されている[2-5][6,7][14,15]。テスト圧縮技術は大きく分けて静的圧縮、動的圧縮の2種類に分類される。静的圧縮[2,3]は生成したテスト集合に対して冗長なテストパターンを削減する技術で、動的圧縮[4,5]はテスト生成の過程で行う技術で、より多くの故障を検出するテストパターンを生成することでテストパターンの圧縮を行う。このため動的圧縮問題は一般的に複雑で計算量が多いが、静的圧縮と比較するとより圧縮されたテスト集合が生成される。

また、動的圧縮に対して SAT (Satisfiability problem)-solver を用いることでテスト生成に関して多くの付加条件を与えても対処が可能となり、より圧縮されたテスト集合の生成が報告されている[6][7]。近年の SAT-solver の処理速度は衝突節の学習[8]や非辞書式バックトラック[9]、ブール強制伝搬[10]により、急速に進歩している。SAT を用いたテスト生成も提案されており、PODEM[11]や FAN[12]などの既存のテスト生成アルゴリズムと比較して高い故障検出率を得ることができ、テスト生成困難故障に対して有効とされる[13]。さらに、複数の故障を同時に検出するテスト生成 (Multiple Target Test Generation: MTTG) と組み合わせることで、さらに効率的にテスト圧縮さ

れた結果が報告されている[14][15]。

一方、文献[16]では RTL (Resister Transfer Level) 演算器を並列にテストすることで、テストパターン数を削減する手法が提案されている。文献[16]はコンローラ拡大を用いて各演算器を並列にテスト可能にするようにテストレジスタを割当て、理論上、回路のテストパターンは演算器の最大テストパターン数に等しくなると考えられる。しかしながら文献[16]ではこの手法で設計した回路情報を利用したテスト生成でなければテストパターン数を削減できない問題がある。

本論文ではテストパターン数の削減を目標に文献[16]において設計された回路情報 (以下、テストスケジューリング情報と呼ぶ) を利用する SAT を用いた MTTG 手法を提案する。本論文は次のように構成される。第2章で SAT を用いたテスト生成手法について説明し、第3章で SAT を用いた MTTG について説明する。第4章で提案手法のフローを説明する。

2. SAT を用いたテスト生成

本章ではまず、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を説明する。

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。まず、論理回路を構

Consideration On Multiple Target Test Generation

Takuma FUHII, Hiroshi YAMAZAKI, Toshinori HOSOKAWA
and Masayoshi YOSHIMURA

成する各ゲートを CNF に変換する．次に，変換した各 CNF を論理積することで，論理回路全体を CNF として表現することができる．本論文が扱う故障モデルは縮退故障モデルとする．

SAT を用いたテスト生成では，対象回路に対する故障回路と正常回路の 2 つを用いて図 1 に示すようなテスト生成モデルとなる合成回路を作成する．この時，故障回路はテスト生成対象の故障箇所から到達可能な範囲(Transitive Fan-Out: TFO)とし，正常回路は故障回路へ到達可能な範囲(Transitive Fan-In: TFI)とする．また，故障回路において故障箇所から斜線で示した外部出力までの範囲を仮定する故障の影響とし，正常回路から故障回路への接続は故障の影響を伝搬するための入力とする．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，各外部入力に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．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，各 EXOR 演算の出力を OR 演算したものである．したがって，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とは，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問題である．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であり，テスト生成対象の故障は冗長故障[1]と判定される．

3. SAT を用いた MTTG

本章では SAT を用いた MTTG の基本的な手法について説明する．SAT を用いたテスト生成と基本的な流れは同じだが，複数の故障を対象とする点で異なる．

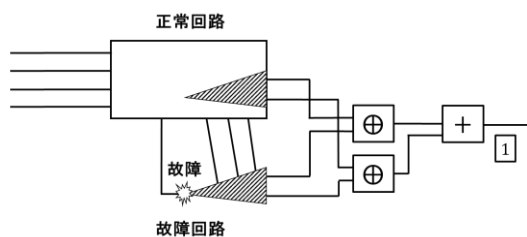


図 1. テスト生成モデル例

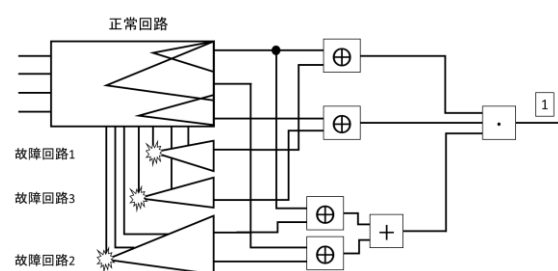


図 2. テスト生成モデル例(対象故障数 3)

また，MTTG は対象とした複数の故障全てを検出するテスト生成である．

本研究ではまず各故障に対して故障回路の範囲を特定し，図 2 に示すように対象故障数の故障回路と 1 つの正常回路(各故障回路に対する全ての TFI)を用意し各故障回路に含まれる外部出力に対応する正常回路の外部出力と EXOR 演算を行う(1 つの故障回路に複数の外部出力が含まれる場合は EXOR 演算したのに対して OR 演算を行う)．また，MTTG は対象とした故障全てを検出することを目標としているため，正常回路と各故障回路の外部出力の EXOR 演算(もしくは OR 演算)結果を論理積して 1 つの外部出力としている．これにより，SAT を用いたテスト生成と同様にテスト生成モデルの外部出力の値が 1 の時，対象とする故障全てを検出することができる．この時，充足可能であれば MTTG が可能で対象とした全ての故障を同時検出するテスト生成が可能と判定できる．しかしながら充足不能の場合，対象とした全ての故障が冗長故障とは限らない．つまり各故障に対してテスト生成が可能でも同時のテスト生成が不可能である場合がある．それゆえ本論文では充足不能になった場合，MTTG における同時検出故障数を 1 つ減らして図 2 のテスト生成モデルを図 3 のように再構築する．

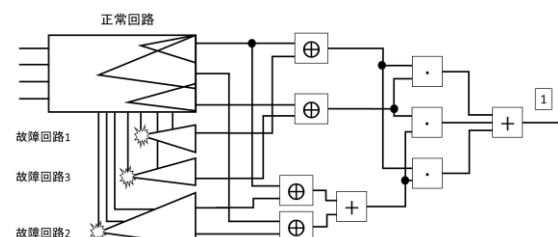


図 3. テスト生成モデルの再構築例(対象故障数 3)

テスト生成モデルの再構築は図 3 に示すように各 EXOR 演算(または OR 演算)を組合せに則って論理積を行い、その各出力を OR 演算する。図例では同時検出故障数を 3 から 2 へ削減して再度テスト生成を実行する。これにより、再構築したテスト生成モデルの外部出力の値が 1 になる(充足可能)時、選択した 3 つの故障の中から 2 つの故障を検出可能と判定する。充足不能の場合、さらに同時検出故障数を 1 つ削減してテスト生成モデルを再構築するが、同時検出故障数が 1 の時に充足不能の場合、選択している故障全てを冗長故障と判定する。

対象故障において検出判定すべき故障が残っている場合、その故障集合から故障を選択して再び SAT を用いた MTTG を実行するが、テスト生成モデルを再構築した際に充足可能となり検出可能と判定されなかった故障を本論文では引き続き選択する。例えば選択した 3 つの故障から 2 つの故障を検出可能と判定した場合、残る 1 つの故障は引き続き選択され、検出判定すべき故障の中から新たに 2 つの故障を選択する。

4. SAT を用いた MTTG フロー

本章では提案手法であるテストスケジューリング情報に基づいて故障を選択する SAT を用いた MTTG のフローについて説明する。図 5 に提案手法のフローチャートを示す。

(Step 1)

テスト生成対象の回路データを読み込む。この時、テストスケジューリング情報も読み込む。

(Step 2)

Step 1 で読み込んだ回路データを基に故障リストを作成する。

(Step 3)

Step 1 で読み込んだテストスケジューリング情報から未実行の同時検出可能箇所が存在するか判断する。存在する場合は Step 4 へ進み、存在しない場合は Step 6 へ進む。

(Step 4)

Step 1 で読み込んだテストスケジューリング情報

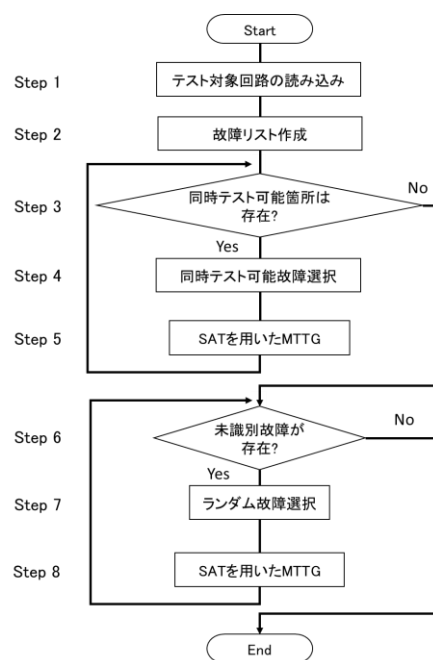


図 4. 提案手法フローチャート

における同時テスト可能箇所を基に故障を選択する。

(Step 5)

Step 4 で選択した故障に対して第 3 章で説明した SAT を用いた MTTG を実行する。

(Step 6)

Step 2 で作成した故障リストにおいて検出判定していない故障(未識別故障[1])が存在するか否かを判断する。存在する場合は Step 7 へ進み、存在しない場合は提案手法を終了する。

(Step 7)

代表故障リストに存在する未識別故障からランダムに故障を選択する。選択する故障数は外部から指定する。

(Step 8)

Step 7 で選択した故障に対して第 3 章で説明した SAT を用いた MTTG を実行する。

5. おわりに

本論文ではテストスケジューリング情報を用いて故障を選択する SAT を用いた MTTG 手法を提案した。

今後は提案した手法の実装に取り組む予定である。

参考文献

- [1] 藤原秀雄, “デジタルシステムの設計とテスト” (2004)
- [2] L. N. Reddy, I. Pomeranz, and S. M. Reddy, “ROTCO: A reverse order test compaction technique,” in Proc. Euro ASIC Conf., Paris, France, 1992, pp. 189–194.
- [3] M. S. Hsiao, E. M. Rudnick, and J. H. Patel, “Fast static compaction algorithms for sequential circuit test vectors,” IEEE Trans. Comput., vol. 48, no. 3, pp. 311–322, Mar. 1999.
- [4] P. Goel and B. C. Rosales, “Test generation and dynamic compaction of tests,” in Proc. Int. Test Conf., 1979, pp. 189–192.
- [5] G.-J. Tromp, “Minimal test sets for combinational circuits,” in Proc. Int. Test Conf., Nashville, TN, USA, 1991, pp. 204–209.
- [6] Kareem Habib, Mona Safer, Mohamed Dessouky and Ashraf Salem “Dynamic Compaction using Multi-Valued Encoding in SAT-based ATPG”, Engineering and Technology (ICET), 19-20 April 2014
- [7] Alexander Czutro, Sudhakar M. Redddy, Ilia Polian and Bernd Becker “SAT-Based Test Pattern Generation with Improved Dynamic Compaction”, International Conference on Embedded Systems 5-9 Jan. 2014
- [8] Matthew W. Moskewicz, “Chaff: Engineering an Efficient SAT Solver”, IEEE xplore (2001)
- [9] Joao P. Marques-Silva, “GRASP: A Search Algorithm for Propositional Satisfiability”, IEEE TRANSACTIONS ON COMPUTERS (1999)
- [10] T. Larrabee, “Test pattern generation using Boolean satisfiability”, IEEE Transactions on Computer-Aided Design of Integrated Circuits, Vol. 11, No.1, pp.4-15, 1992.
- [11] P. Goel, “An implicit enumeration algorithm to generate tests for combinational logic circuits,” IEEE Trans. Comput., vol. C-3 1, pp.215-222. 1981.
- [12] J. P. Roth, “Diagnosis of automata failures: A calculus and a method,” IBM J. Res. Develop., vol. 10, pp. 278-291, 1966.
- [13] R. Drechsler et al., “On acceleration of SAT-based ATPG for industrial designs,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 27, no. 7, pp. 1329–1333, Jul. 2008.
- [14] Stephan Eggersgluß, Rene Krenz-Baath, Andreas Glowatz, Friedrich Hapke and Rolf Drechsler, “A New SAT-based ATPG for Generating Highly Compacted Test Sets”, Design and Diagnostics of Electronic Circuits & Systems (DDECS), 18-20 April 2012
- [15] Stephan Eggersgluß, Kenneth Schmitz, René Krenz-Baath, and Rolf Drechsler, Fellow, “On Optimization-Based ATPG and Its Application for Highly Compacted Test Sets”, IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, VOL. 35, NO. 12, DECEMBER 2016
- [16] T. Hosokawa, S. Takeda, H. Yamazaki, M. Yoshimura, “Controller Augmentation and Test Point Insertion at RTL for Concurrent Operational Unit Testing”, IEEE IOLTS'17, vol.23, pp.17-20, Thessaloniki, Greece, July. 2017.