

MAX-SAT を用いた M バイ N によるテスト圧縮法

日大生産工(学部) ○浅見 竜輝 日大生産工 細川 利典 日大生産工 山崎 紘史
日大生産工 新井 雅之 京産大 吉村 正義

1. はじめに

近年, 半導体微細化技術の進歩に伴い, 超大規模集積回路(Very Large Scale Integrated circuits: VLSI)が大規模化・複雑化し, テストコストの増大が問題となっている[1]. テストコストはテストパターン数の増加に比例するため, テストパターン数を削減することにより, テストコストの削減が期待できる. したがって, テストパターン数の削減のために様々なテスト圧縮手法が提案されている[2-9]. テスト圧縮手法として静的圧縮手法[2,3]や, 動的圧縮手法[4,5]が提案されている. 静的圧縮は生成したテスト集合に対して行う手法で, 冗長なテストパターンを削減する圧縮技術である. 動的圧縮はテスト生成の過程で行う圧縮技術で, 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.

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[10]に関する制約等の付与を与えてもテスト生成が可能であり, テストパターン数の削減に効果的であることが報告されている[6,7]. 文献[5]の SAT による多重目標故障テスト生成(Multiple Target Test Generation: MTTG) を用いたテスト圧縮法では, テストパターン数の削減に対し効果的であることが示された. しかしながら, 文献[5]では目標とする故障の選択方法は回路構造に関する情報を用いていること以外特に述べられていない. さらに SAT では命題論理式が必ず真にならないといけないため, MTTG においてテスト生成モデルの再生成回数が増大する可能性があり, テスト生成時間の増大が問題となる可能性がある. 文献[11]では最大充足化問題(Maximum Satisfiability Problem: MAX-SAT)を解くためのアルゴリズムが提案されている.

MAX-SAT は重み付けを考慮した SAT であり, 最大の重みになる解を求める問題である. そのため与えられた問題に対して最適解を求めることが可能である. したがって, MTTG において各故障の検出条件に対して重み付

けが考慮可能な MAX-SAT を用いることで, テスト生成モデルを再生成することなく問題を解くことができるため, テスト生成時間の高速化が可能であると考えられる.

本論文では, MTTG の高速化, テストパターン数の削減を目的として, MAX-SAT による多重目標故障テスト生成手法を提案する. その MAX-SAT による MTTG 手法をテスト圧縮の妨げとなる M 個のテストパターンを削除し, その M 個のテストパターンのみで検出可能な故障に対して N 個のテストパターンを再生成する M バイ N 法を提案する本論文の構成は, 2 章で MAX-SAT について説明し, 3 章で MTTG と提案手法を説明する. 4 章で提案手法に対する実験結果を示す. 最後に 5 章でまとめと今後の課題について述べる.

2. 最大充足化問題

MAX-SAT とは, 論理積標準形(Conjunctive Normal Form: CNF)で与えられた命題論理式が真となる論理変数の組合せが存在するか否かに加えて, 各 CNF の重みの総和を最大化する問題である.

CNF を構成する各論理積または単項を節, 必ず真とする節をハード節, 偽になった際に設定した重みが発生する節をソフト節とする. ソフト節は必ず真とする必要がなく, 任意の値に重みを設定できる. ハード節は, 必ず真となるため重みは無限大である. また, CNF で与えられたハード節の命題論理式が真となる論理変数の組合せが存在しない場合, 充足不能(Unsatisfiability: UNSAT)と呼ぶ.

次に MAX-SAT 判定の具体例について 4 つの論理変数(a, b, c, d)によって構成された CNF: $(a + \neg b + c)(b + \neg c)(a + \neg b)(d)$ を基に説明する. 例では $(a + \neg b + c)$, (d) をハード節とし, $(a + \neg b)$ を重み 3, $(b + \neg c)$ を重み 2 のソフト節とする. ハード節が真となるための割当ては $(a,b,c,d) = (0,0,0,1), (0,0,1,1), (0,1,1,1), (1,0,0,1), (1,0,1,1), (1,1,0,1), (1,1,1,1)$ である. 表 1 に, ハード節とソフト節が充足可能となる変数割当てと, そ

表 1.割当て表

割当て (a,b,c,d)	偽になる節	重みの総和
(0,0,0,1)	なし	5
(0,0,1,1)	$(b + \neg c)$	3
(0,1,1,1)	$(a + \neg b)$	2
(1,0,0,1)	なし	5
(1,0,1,1)	$(b + \neg c)$	3
(1,1,0,1)	なし	5
(1,1,1,1)	なし	5

の時の重みの総和の一覧を示す。表 1 より割当ての結果 $(a,b,c,d) = (0,0,0,1), (1,0,0,1), (1,1,0,1), (1,1,1,1)$ が重みの総和が最大となる割当てとなる。

3-1.SAT を用いた多重目標故障テスト生成

テスト生成とは、故障を仮定した論理回路に対して、その故障影響を外部出力で観測可能な外部入力値（テストパターン）を生成することである。MTTG とは、テスト生成において複数個の故障を仮定し、仮定した故障すべてを検出するためのテストパターンを生成する動的圧縮手法である。ただし、目標とした複数の故障は同時発生する多重故障ではなく、それぞれが個別の故障が単一で発生する故障の同時検出を考慮してテスト生成を行う。MTTG では対象回路に対してテスト生成に必要な回路範囲を特定するために図 1 に示すような範囲の特定を行う。対象回路の例として外部入力数 4、外部出力数 3 の組合せ回路とし、対象故障数は 3 とする。また、本論文では単一縮退故障モデルを対象とする。範囲の特定はテスト生成対象の故障箇所から出力方向に到達可能な範囲 (Transitive Fan-Out: TFO) で行う。故障回路は故障箇所に対する TFO とするが、MTTG では複数の故障を対象としているため、図 1 に示すように各故障に対して故障回路の範囲を特定する。この範囲特定を行うことにより、故障回路の規模を削減することができ、テスト生成時間を短縮することができる。

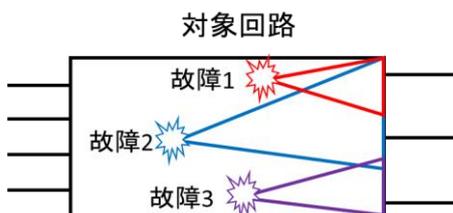


図 1. 対象故障に対する故障ごとの範囲特定

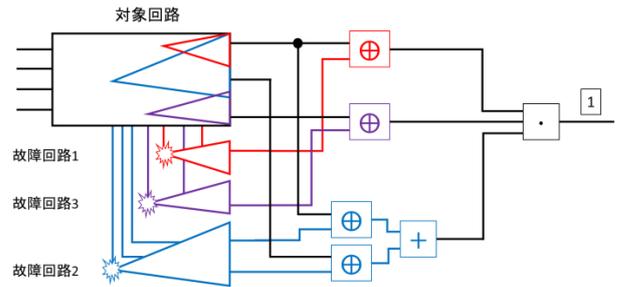


図 2. 対象回路に対する MTTG モデル例

テスト生成モデルの正常回路は全故障を検出できるようにするため、回路全体がテスト生成モデルの対象となる。図 2 のように 1 つの正常回路と対象故障数の故障回路を用意し、各故障回路の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 演算したものに対して OR 演算を行う。また、MTTG は対象とした故障すべ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で行った EXOR 演算（もしくは OR 演算）結果を AND 演算して 1 つの外部出力としている。このテスト生成モデルの各 EXOR 演算（もしくは OR 演算）を入力とする AND 演算の出力値が 1 の時、対象とする 3 つの縮退故障を検出することができる。

MTTG において対象故障数すべてが同時検出不可能であった場合、テスト生成モデルの再生成を行う。しかしながら、対象故障数が増加するとテスト生成時間が増大する。その解決策として SAT ではなく MAX-SAT を用いて MTTG を実行する。

3-2.MAX-SAT を用いた MTTG

本節では提案手法である MAX-SAT を用いた多重目標故障テスト生成法について説明する。各故障に対する故障回路の範囲特定の手順は前節と同じである。

MAX-SAT を用いた MTTG でのテスト生成モデルを図 3 で表す。対象回路及び対象故障モデルは 3 章と同様の例を用いて対象故障数は 3 とする。SAT を用いた MTTG では定義したすべての故障を検出するテストパターンを生成する。そのため、正常回路と各故障回路で行った EXOR 演算（もしくは OR 演算）結果を AND 演算して 1 つの外部出力としている。しかしながら、MAX-SAT を用いた MTTG では各故障回路の外部出力信号線をソフト

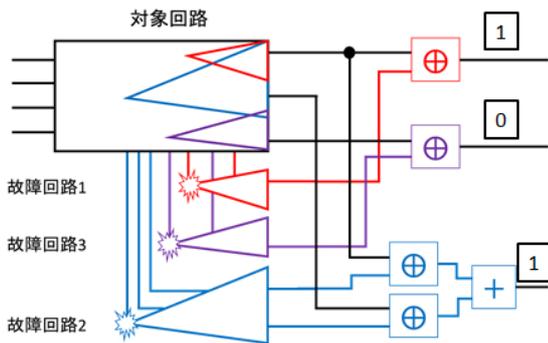


図 3. MAX-SAT における MTTG モデル例

節とする。そのため各故障回路の外部出力と正常回路の外部出力に EXOR 演算(もしくは OR 演算)を行い、結果を AND 演算せず外部出力とする。

MAX-SAT を用いた MTTG では図 3 に示すようなテスト生成モデルを CNF に変換し、MAX-SAT 判定を行うことで MTTG を実現している。CNF の変換はテスト生成モデルに対して表 2 の規則に基づいて各ゲートが CNF へ変換し、変換した各ゲートの CNF 式を AND 演算することでテスト生成モデルの変換は可能である。

本手法である MAX-SAT を MTTG に用いることにより、1 つのテストパターンで設定した全故障が検出できない場合でも、なるべく多くの故障が検出できるテストパターンを生成することができる。生成したテストパターンで故障シミュレーションを実行し、検出された故障は故障集合から削除する。故障集合が空になった場合、本手法は終了する。

M 個のテストパターンで検出していた故障を新たに生成した $N(<M)$ 個のテストパターンで検出できるようにすることを M パイ N [12][13] という。 $N \ll M$ の場合テストパターン数を削減効果が高く効率の良いテスト圧縮となる。

本論文では MAX-SAT を用いた MTTG により M パイ N テスト圧縮を実行する手法を提案する。テストパターンの生成後に未検出である故障を目標故障集合とし、新たにテスト生成モデルの作成を行う。作成の手順は 3 章の手順と同じである。CNF 作成時の重み割当ては 2 章で

表 2.論理ゲートに対する CNF 変換規則

ゲートタイプ	入力	出力	CNF
AND	X Y	Z	$(X + \neg Z) \cdot (Y + \neg Z) \cdot (\neg X + \neg Y + Z)$
NAND	X Y	Z	$(X + Z) \cdot (Y + Z) \cdot (\neg X + \neg Y + \neg Z)$
OR	X Y	Z	$(\neg X + Z) \cdot (\neg Y + Z) \cdot (X + Y + \neg Z)$
NOR	X Y	Z	$(\neg X + \neg Z) \cdot (\neg Y + \neg Z) \cdot (X + Y + Z)$
NOT	X	Y	$(X + Y) \cdot (\neg X + \neg Y)$
BUFF	X	Y	$(\neg X + Y) \cdot (X + \neg Y)$
EXOR	X Y	Z	$(\neg X + Y + Z) \cdot (X + \neg Y + Z) \cdot (\neg X + \neg Y + \neg Z) \cdot (X + Y + \neg Z)$

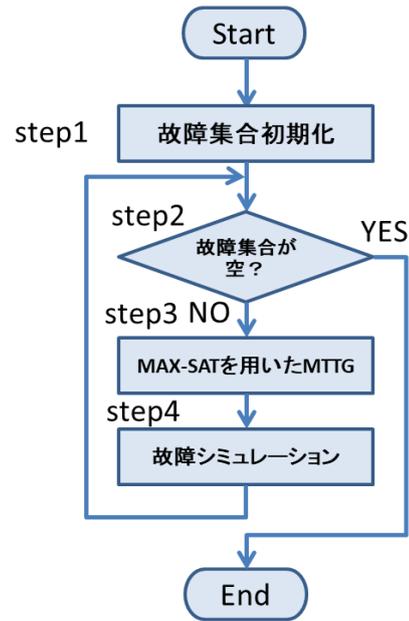


図 4.提案手法フローチャート

説明した外部出力信号線に該当する節に重み 1 を与える。その他の節はハード節なので重みは考慮しない。

図 4 で提案手法のフローを示す。

(step 1)

削除する N 個のテストパターンでのみ検出される故障集合を作成し初期化する。

(step 2)

故障集合が空であるか否かを判断する。故障集合が空でない場合は step3 に進む。故障集合が空である場合は、終了する。

(step 3)

故障集合中の全故障を対象とし、MAX-SAT を用いた MTTG を実行する。

(step 4)

step3 で生成されたテストパターンで故障シミュレーションを実行し、検出された故障を故障集合から削除する。

4.実験結果

本実験では、市販のテスト生成ツール TetraMAX を用いてテスト集合を生成し、テスト集合中の各テストパターンに対して X 判定を行い、各テストパターン間の圧縮可能性を解析し、他のテストパターンとの圧縮可能性が低いテストパターンを削除の対象とした。実験結果では、M パイ N テスト圧縮の適用によるテストパターン数の変化を示す。対象回路は ISCAS'89 の s5378,s9234 を対象

に行った。使用計算機は Intel Xeon で OS は Red Hat Enterprise Linux Workstation 7.3, 実装言語はC言語で, MAX-SAT-solver は Dist[14]を使用した。Dist の実行時間は 3000 秒で打ち切るよう設定した。そのためすべての対象故障を検出するような解が存在しない場合は, 実行時間が 3000 秒の時点での解を出力する。

本手法の実行結果を図 5 に示す。図 5 はテストパターン圧縮前と圧縮後の結果を示している。s5378 では圧縮前, 圧縮後のテストパターン数は同じであったが, s9234 ではでテストパターン数を 4 つ削減することができた。

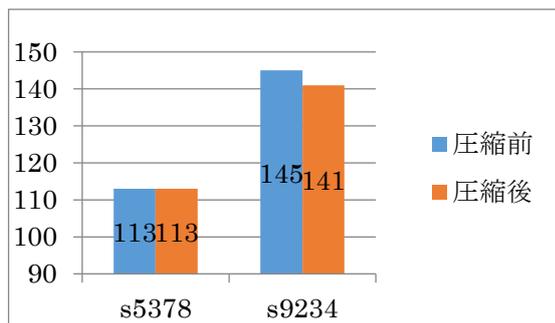


図 5.実験結果

5.まとめ

本論文では MAX-SAT を用いた MTTG による M バイ N テスト圧縮手法を提案した。本実験では効果のない回路が存在した。これは Dist の実行時間の判断や削除する M 個のテストパターンの選択法に課題があると考えられる。対象故障数が多いと, CNF 数が増大し, 最適解を導き出す時間が増大する。

今後の課題として故障集合中の部分集合を選択して MTTG の対象にする方法や削除ウする M 個のテストパターンを必須故障の少ないものにする方法などを検討する。

参考文献

- 1) Y.Sato, T.Ikeya, M.Nakao, and T.Nagumo, "A bist approach for Very Deep Sub-Micron
- 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, Iliia Polian and Bernd Becker "SAT-Based Test Pattern Generation with Improved Dynamic Compaction", International Conference on Embedded Systems 5-9 Jan. 2014
- 8) Stephan Eggersglüß, Kenneth Schmitz, René Krenz-Bååth, 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
- 9) 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.
- 10) 藤原秀雄, "デジタルシステムの設計とテスト", 2004
- 11) Roberto Battiti, Macro Protasi, "Approximate Algorithms and Heuristics for MAX-SAT", pp77-138 1998
- 12) S Kajihara, I. Pomeranz, K. Kinoshita and S. M.Reddy "On Compaction Test Sets by Addition and Removal of Test Vectors", 12th IEEE, 1994.
- 13) Hamzaoglu, I. and J. H. Patel, "Test Set Compaction Algorithms for Combinational Circuits," IEEE Trans. on CAD, vol. 19, no. 8, pp. 957-963, Aug. 2000.
- 14) Shaowei Cai, Local Search for Maximum Satisfiability ,Shaowei Cai, <http://lcs.ios.ac.cn/~caisw/MaxSAT.html>