

複数故障の同時検出のための SAT ベーステスト生成法

○原 侑也(日大生産工(学部))・鈴木 悠介(日大生産工(院))
山崎 紘史(日大生産工(院))・細川 利典(日大生産工)

1.はじめに

近年、半導体微細化技術の進歩に伴い、大規模集積回路 (Large Scale Integrated circuits LSI) が大規模化・複雑化し、テストコストの増加が問題になっている[1]. テストコストはテストパターン数に比例するため、テストパターン数を削減することにより、テストコストを削減することが期待できる.

テストパターン数を削減する手法として、テスト圧縮手法[2][3]が提案されている. 小規模の回路では、ほぼ最小のテスト集合を得るテスト圧縮法が過去に提案されている[2]. しかしながら、大規模回路において最小のテスト集合を得るためには計算量が多く、現実的な計算時間での適用は困難である.

テスト圧縮には、テスト生成中にテスト圧縮を行う動的圧縮法[2][3]と、テスト生成後にテスト圧縮を行う静的圧縮法がある. 大規模な回路に適用可能な静的圧縮法の一つとして、テストパターンに冗長なテストパターンを 1 個生成し、このテストパターンを追加することにより他のテストパターンを 2 個削除することで、結果的にテストパターン数を削減する手法である冗長テストパターン追加テスト圧縮法(Two By One アルゴリズム)が提案されている[4]. この手法において、故障検出率を落とさずにテストパターンを圧縮するには、一個のテストパターンで削除する二個のテストパターンのみで検出される含まれる複数の故障を検出することが必要である.

Two By One アルゴリズムでは、複数の故障を同時に検出するテストパターンの生成に経路活性化法を用いてテスト生成を行う. しかしながら、経路活性化法による二次故障テスト生成は、一次故障のテスト生成の結果に依存する. そのため、一次故障のテスト生成の値割当てが原因で、複数の故障を検出するテストパターンを生成できない可能性がある.

本論文ではこの問題を解決するために、充足可能性問題 (Satisfiability problem: SAT) を用いて複数の故障を同時に検出可能な

否かを判定するテスト生成法を提案する.

2.1.必須故障

必須故障は、与えられた回路に対して生成されたテスト集合 $T(t \in T)$ において、故障 f がテストパターン t によって検出可能であるが、 t 以外のテストパターンでは検出不可能であるとき、 f を t の必須故障という. 一方、テストパターン t が必須故障を持たないときは、テストパターン t を冗長パターンという. またテスト集合 T が冗長パターンを含まないとき、 T を極小テスト集合という.

2.2.独立故障

与えられた回路の故障集合 $F(f_1, f_2 \in F)$ において、故障 f_1 を検出するために必要な信号線の値割当てを行い、故障 f_2 を検出するために必要な信号線の値割当てを行い、値が衝突するとき、 f_1 と f_2 は同一のテストパターンで検出することが不可能である. f_1 と f_2 の必須割当てにおいて値が衝突するとき、 f_1 と f_2 は独立故障という.

図 1 に故障の必須割当てにおける値の衝突例を示す. 故障 f_1 を検出するためには信号線 c に 0 を、信号線 b, e, g に 1 を割当てる必要があると仮定する. f_2 を検出するためには信号線 a, d, e に 0 を、信号線 b, f に 1 を割当てる必要があると仮定する. それぞれの割当てを比較すると信号線 e において値が衝突している. よって、 f_1, f_2 は独立故障である.

また、同一のテストパターンで検出することができない故障の集合を独立故障集合といい、

	a	b	c	d	e	f	g
f1	X	1	0	X	1	X	1
f2	0	1	X	0	0	1	X

衝突

図 1.故障の値割当てにおける衝突例

A SAT Based Test Generation Method to Simultaneously Detect Several Essential Faults

Yuya HARA, Yusuke SUZUKI, Hiroshi YAMAZAKI and Toshinori HOSOKAWA

集合の要素数が最大である集合を最大独立故障集合という．独立故障集合の要素数が大きいものから順にテスト生成の目標故障を選択することで，より小さいテスト集合が得られると考えられる．

3.冗長テストパターンスイカテスト圧縮法アルゴリズム

Two by One アルゴリズムは，極小テスト集合 T に冗長なテストパターンを 1 個追加し， T から 2 個のテストパターンを削除するアルゴリズムである．まず，任意の必須故障 f_1 を 1 個選択し， f_1 を検出するテストパターン t_1 ($t_1, t_2 \in T$) を選択する．次に f_1 が属さない独立故障集合 $F(f_2 \in F)$ から f_2 を選択し， f_2 を検出するテストパターン t_2 を選択する．冗長なテストパターンの生成は，選択された 2 個のテストパターン t_1, t_2 で検出可能な必須故障に着目して行う．その複数の必須故障を同時に検出することが可能なテストパターンが生成可能か否かを判定する．可能であれば生成したテストパターン t_3 を T に追加し， t_1, t_2 を T から削除する．不可能であれば，ほかの 2 個のテストパターンを選択し，同様の処理を実行する．

しかしながら，文献[4]の Two by One で用いられているアルゴリズムでは，複数の故障を同時に検出可能なテスト生成において，もしそれらの故障を検出可能なテストパターンが存在したとしても，必ずしもテスト生成に成功するとは限らない．文献[4]では経路活性化法の動的圧縮のアルゴリズムを用いており，故障 A，故障 B を同時に検出できるテストパターンを生成するときは，故障 A に対して論理値を割当て，テスト生成を実行し，その後 2 次故障として故障 B のテスト生成を実行する．故障 B に対してのテスト生成の結果は，

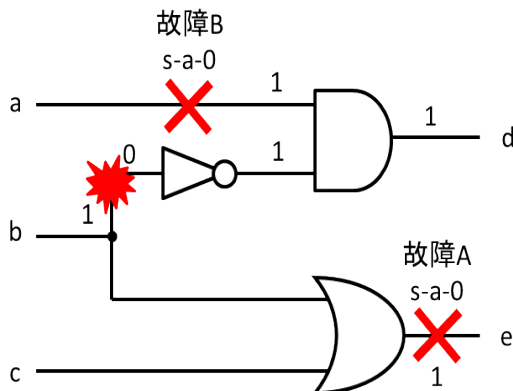


図 2. 複数の故障を同時に検出するテスト生成例

故障 A のテスト生成時の割当ての結果に強く依存する．

図 2 に文献[4]のアルゴリズムを用いた複数の故障を同時に検出するテスト生成例を示す．図 2 では，まず故障 A を検出するため，外部入力 b に 1 を割当てる．次に故障 B を検出するために値割当てを行うが，外部入力 b で衝突を起こすため，故障 B の割当てに失敗する．しかしながら，故障 A を検出するための割当てにおいて，外部入力 c に 1 を割当てたときは，故障 B の検出が可能である．したがって，2 次故障の検出は 1 次故障の割当ての結果に強く依存することがわかる．

SAT を用いてテスト生成を行う場合，故障 A，故障 B を同時に検出可能なテストパターンをどの信号線にも論理値が割当てられていない状態からテスト生成を行う．そのため，故障 A と故障 B が同時に検出可能なか否かを正しく判定することが可能である．

4.充足可能性問題

SAT とは，与えられた乗法標準形 (Conjunctive Normal Form : CNF) の変数の値を 1(真)又は 0(偽)に定め，式全体の値を 1(真)にできる変数の組み合わせが存在するか否かを判定する問題である．

SAT を用いたテスト生成では，回路全体の CNF の値を 1(真)にできる割当てが存在した場合は充足可能といい，その割当てによるテストパターンの生成が可能であることを示す．存在しない場合は充足不可能といい，テストパターンの生成が不可能であることを示す．SAT は CNF を入力とするため，論理回路を CNF に変換する必要がある．

4.1.論理回路の CNF 変換

表 1 に論理ゲートの CNF 変換規則について示す．それぞれの論理ゲートは表 1 の規則に従って CNF に変換される．ゲートタイプは論理ゲートの種類を表し，入力・出力は各論理ゲートの入力・出力を表す．CNF は論理積 ($\cdot$)，論理和 ($+$)，否定 ($\neg$) の演算子で表現される．括弧でくくられた論理式の各項をリテラルといい，括弧でくくられた論理式を節という．回路全体の CNF は各ゲートの CNF の論理積により表現される．例として，2 入力 (X, Y)1 出力 (Z) の AND ゲートについて考える．AND ゲートが正当であるには， $(X, Y, Z) = (1, 1, 1), (1, 0, 0), (0, 1, 0), (0, 0, 0)$ の 4 通りの組み合わせとなる．また，表 1 の AND ゲートの CNF もこの 4 通

表 1. 論理ゲートの CNF 変換規則

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

りでのみ 1(真)となる．これにより，CNF が実際の論理ゲートの動作を表現していることがわかる．

信号線に値を割り当てる処理では節の少なくとも一つが 0(偽)であった時点で，その割当ての組合せが正当でないことが判定できる．よって，早期に矛盾を発見することができるため，高速な正当化判定処理が期待できる．

4.2.SAT を用いたテスト生成

SAT を用いたテスト生成手順を説明する．まず，テスト対象となる回路に対して，正常に動作する回路と，ある信号線の一つが縮退故障している故障回路を用意する．次に，正常回路と故障回路の入力に共通の外部入力を接続する．外部出力に対し，故障影響を励起・伝搬させるために，正常回路と故障回路で異なる値を出力する必要がある．そのため，外部出力を EXOR ゲートの入力に接続する．故障の影響が伝搬できればいいため，各外部出力を接続した EXOR ゲートの出力を OR ゲートの入力に接続し，OR ゲートの出力が常に

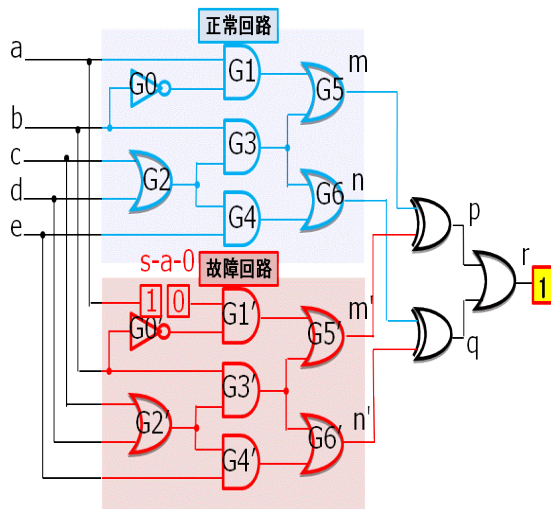


図 3. SAT を用いたテスト生成モデルの回路例

1(真)になる情報を付加する．

図 3 に SAT を用いたテスト生成モデルの例を示す．外部入力 a, b, c, d, e を正常回路と故障回路の共通の入力として与える．また，各回路の出力 m, m' を EXOR ゲートの入力に接続し p とする．同様に各回路の出力 n, n' を EXOR ゲートの入力に接続し q とする．そして p, q を OR ゲートに接続し r とする．

これらの条件を付加した CNF を作成し，SAT-solver を用いて解き，充足可能(SAT)であればテスト生成が可能，充足不可能(UNSAT)であれば，その故障に対するテスト生成が不可能であると判定できる．

5.複数の故障同時検出のためのテスト生成

本提案手法では，選択した 2 個のテストパターンで検出できる必須故障の数だけ故障回路を用意し，次に，表 1 の CNF 変換規則に基づいて CNF 変換を行う．作成した CNF を SAT-solver を用いて解くことで，選択された 2 個のテストパターンで検出される全ての必須故障を検出可能なテストパターンが生成可能か否かの判定を行う．

まず，テスト対象となる回路に対し，正常に動作する回路と，検出する必須故障だけ故障回路を用意する．次に，正常回路と故障回路の入力に共通の外部入力を接続する．外部出力に対し故障影響を励起・伝搬させるために，正常回路とそれぞれの故障回路で異なる値を出力する必要がある．そのため，正常回路の出力と各故障回路の出力を EXOR ゲー

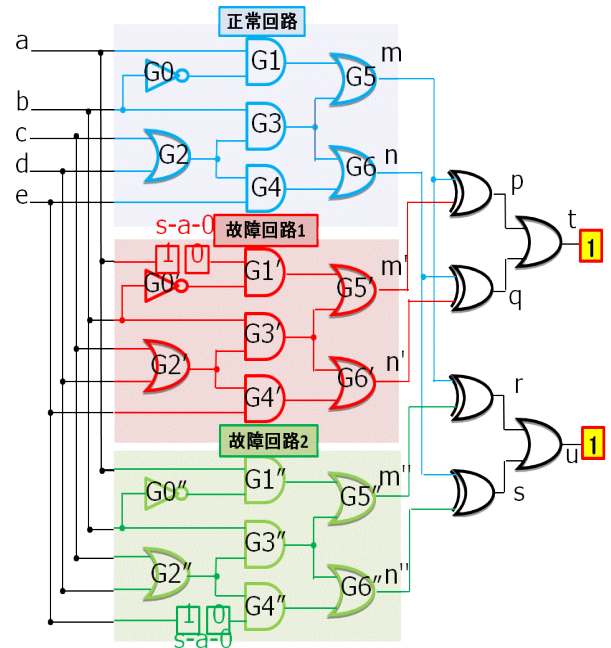


図 4. 2 個の故障を検出する SAT を用いたテスト生成モデルの回路例

トに接続する. 故障を検出するためには, 故障の影響が少なくとも一つ以上の外部出力に伝搬すればいいため, それぞれの EXOR ゲートの出力を OR ゲートに接続する. その各 OR ゲートの出力が常に 1(真)になる情報を付加する.

図 4 に必須故障が 2 個の場合の SAT を用いたテスト生成モデルの回路例について示す. 外部入力 a, b, c, d, e を正常回路と各故障回路の共通の入力回路として与える. 正常回路と故障回路 1 の出力 m, m' を EXOR ゲートの入力に接続し p とする. また, 正常回路と故障回路 2 の出力 m, m' を EXOR ゲートに接続し, r とする. 同様に正常回路と故障回路 1 の n, n' を EXOR ゲートの入力に接続し q とする. また, 正常回路と故障回路 2 の出力 n, n' を EXOR ゲートに接続し s とする. そして p, q を OR ゲートの入力に接続し, t とする. 同様に r, s を OR ゲートの入力に接続し u とする.

これらの条件を付加した CNF を生成し, SAT ソルバーを解くことにより充足可能であれば故障を同時に検出できるテストパターンが生成でき, 充足不可能であればその故障に対するテストパターンの生成が不可能であると判定できる.

6. おわりに

本論文では SAT を用いた複数故障の同時検出したテスト生成法を提案した. 今後の目標としては Two by One アルゴリズムに組み込みテスト圧縮効率を評価することが挙げられる.

参考文献

[1] Y.Sato, T.Ikeya, M.Nakao, and T.Nagumo, "A bist approach for Very Deep Sub-Micron (VDSM) Defects", Proc. International Test Conference, pp.283-291, 2000.

[2] Y.Matsunaga, "MINT-An exact algorithm for finding minimum test set", IEICE trans. Fundamentals vol.E76-A, pp1652-1658, 1993.

[3] Seiji Kajihara, Irith Pomeranz, Kozo Kinoshita and Sudhakar M.Reddy "Cost-Effective Generation of Minimal Test Sets for Stuck-at Faults in Combinational Logic Circuits", 30th ACM/IEEE Design Automation Conference, pp102-106, 1993.

[4] D.Brelaz, "New methods to color the vertices of a graph", Communications of the ACM, 22, pp-251-256, 1979.

[5] Seiji Kajihara, Irith Pomeranz, Kozo Kinoshita and Sudhakar M.Reddy "On Compaction Test Sets by Addition and Removal of Test Vectors", 12th IEEE, pp.202-207, 1994.

[6] Tracy Larrabee, "Test Pattern Generation Using Boolean Satisfiability", IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN, VOL.11, No.1, pp4-15, 1992.

[7] Paul Stephan, Robert K.Brayton, Alberto.L.Sangiovanni-Vincentelli, "Combinational Test Generation Using Satisfiability" IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN, VOL.15, No.9 pp1167-1176, 1996.

[8] Paul Tafertshofer, Andreas Ganz, "SAT Based ATPG Using Fast Justification and Propagation in the Implication Graph", IEEE/ACM international conference on Computer-aided design, pp139-146, 1999.