

ツーバイワンによるテスト圧縮のためのテストパターン選択法

日大生産工(院) ○原 侑也 日大生産工(院) 平井 敦士
日大生産工(院) 山崎 紘史 日大生産工 細川 利典 日大生産工 山内 ゆかり

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適用は困難である.

テスト圧縮手法には, テスト生成中にテスト圧縮を行う動的圧縮法[2]と, テスト生成後にテスト圧縮を行う静的圧縮法[3]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ツーバイワンアルゴリズム[4]が提案されている. この手法において, 故障検出率[5]を低下させずにテストパターンを圧縮するには, 1個のテストパターンにおいて, 削除する2個のテストパターンのみで検出される複数の故障を全て検出することが必要条件である.

ツーバイワンアルゴリズムでは, 削除するテストパターンを2個選択する. そして選択したテストパターンでのみ検出できる故障をすべて検出できる1個のテストパターンを生成し, 選択した2個のテストパターンを削除する. すなわち, 再生成するテストパターンが生成可能か否かは, 選択した2個のテストパターンに強く依存する. そのため, 選択する2個のテストパターンの組み合わせによって, 最終的なテストパターン数が異なる. 文

献[4]では, 必須故障[4]の数に着目しているが, 削除するテストパターンの選択法については述べていない. 本論文では, 削除するテストパターンの選択法について提案する.

本論文の構成は, 2章でツーバイワンアルゴリズムに必要な予備知識について説明し, 3章でツーバイワンアルゴリズムについて説明する. 4章では提案手法であるテストパターンに含まれる必須故障と必須故障の必須割当てに着目したテストパターン選択法を説明し, 5章で予備実験結果を示し, 6章でまとめて今後の課題について述べる.

2.予備知識

本章ではツーバイワンアルゴリズムに必要な予備知識である必須故障, 三重検出法における故障辞書・故障リストについて説明し, 必須割り当てについて説明する.

2.1.必須故障

必須故障[4]とは, 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, テストパターン t を冗長パターンという. 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い, ツーバイワンアルゴリズムでは初期テスト集合として極小テスト集合を用いる.

2.2.三重検出法における故障辞書と故障リスト

三重検出法[4]とは, 各テストパターンの必須故障の情報を基にしたテストパターン圧縮法の1つである. 三重検出法では, テスト集合で検出される各故障が, テスト集合で1回のみ検出できるか, 2回のみ検出できるか, 3回以上検出できるか否かを求める. テスト集合において1回もしくは2回のみ検出される

A Test Pattern Selection Method for Test Compaction

by Two By One Algorithm

Yuya HARA, Astushi HIRAI, Hiroshi YAMAZAKI,
Toshinori HOSOKAWA and Yukari YAMAUCHI

故障を検出できないテストパターンを削除するテスト圧縮法である。そのため、三重検出法では必須故障であるか否かだけではなく、テスト集合において 2 個のテストパターンのみで検出可能である故障まで判定する。

本提案手法でのツープイワンアルゴリズムでは、三重検出法を用いてテスト圧縮するのではなく、故障が必須故障であるか、テスト集合で 2 回のみ検出できる故障であるか否かの情報をもつ故障リストと、各テストパターンでどの故障が検出されるか否かの情報を持つ故障辞書を利用してテストパターンを削減する。故障辞書の例を表 1 に示し、故障リストの例を表 2 に示す。

表 1.故障辞書

test_vector	one_check	two_check	detected_faults
t1	1	1	f1, f3, f5
t2	1	0	f2, f5
t3	1	1	f3, f4, f5
t4	1	0	f6

表 2.故障リスト

fault	check	test_vector1	test_vector2
f1	1	t1	-
f2	1	t2	-
f3	2	t1	t3
f4	1	t4	-
f5	3	t1	t2
f6	1	t4	-

表 1 において test_vector はテストパターン名であり、one_check はテストパターンに含まれる必須故障を示す。two_check は全てのテストパターンにおいて 2 個のテストパターンのみで検出可能な故障数を示す。detected_faults はテストパターンで検出可能な故障である。また、表 2 において fault は故障名であり、check は全てのテストパターンにおいて、その故障が検出される回数を表す。0 ならば全てのテストパターンにおいて検出不可能であることを示し、1 ならば 1 回のみ検出可能な必須故障であることを示し、2 ならば 2 回検出可能であり、3 ならば 3 回以上検出可能な故障であることを示す。test_vector 1 は故障を最初に検出するテストパターン名を示し、test_vector2 は故障を 2 番目検出するテストパターン名を示す。

2.3. 必須割当て

必須割当てとは故障を検出するために必要不可欠な信号線の論理値割当てであり、必須割当ては '0', '1', 'X' の 3 値で表現される。

図 1 の故障 f1 は信号線 e の 0 縮退故障、故障 f2 は信号線 e の 1 縮退故障を示している。これらの故障を検出するために、必要不可欠な論理値を各信号線に割当てていく。その結果、故障 f1 の必須割当ては、信号線 a, b, e, g が 1 であり、信号線 f が 0 である。故障 f2 の必須割当ては、信号線 e, f, g が 0 である。

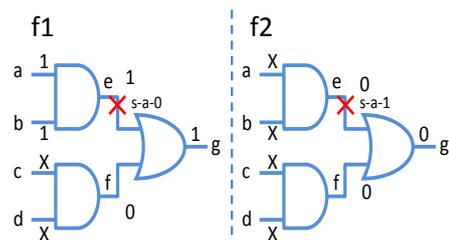


図 1.必須割り当ての例

3. ツープイワンアルゴリズム

本章ではツープイワンアルゴリズムについて説明する。ツープイワンアルゴリズム[4]は、極小テスト集合 T に追加しても故障検出率が変化しない冗長なテストパターンを 1 個追加し、テスト集合 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。まず削除するテストパターン t1 と t2 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 t1 と t2 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。しかしながら、t1 と t2 で検出される必須故障のみを同時に検出できるテストパターン t3 を生成できるとしても、t1 と t2 で検出できる故障の中で必須故障でなく、t1 と t2 以外のテストパターンで検出できない故障 f が存在する場合、再生成したテストパターン t3 においては故障 f を検出できるか否かについて考慮していない。したがって、t1 と t2 を削除し t3 を追加した場合、再生成された t3 において故障 f が検出されない場合、テスト集合 T の故障検出率が低下する。そのために三重検出法で求めた故障辞書、故障リストを用いる。t1 と t2 の故障辞書の two_check が 1 以上の場合、t1 と t2 で検出される故障の故障リストを参照し、各故障の test_vector1, test_vector2 が t1 と t2 であるか否かの判定を行う。test_vector1, test_vector2 が t1 と t2 だった場合

は、その故障を再生成するテストパターン t_3 の検出する故障集合にその故障を追加する。 t_1 と t_2 で検出可能な故障全てに対して判定を行った後に必須故障と追加した故障全てを検出するテストパターン t_3 が生成可能か否かの判定を行う。可能であれば生成したテストパターン t_3 をテスト集合 T に追加し、 t_1 と t_2 をテスト集合 T から削除する。テスト生成が不可能であると判定された場合は、テストパターンを選択し直し、同様の処理を実行する。図 2 にツーバイワンアルゴリズムの基本的なフローチャートを示す。ツーバイワンアルゴリズムの入力は回路情報、テストパターン集合、故障辞書、故障リストである。

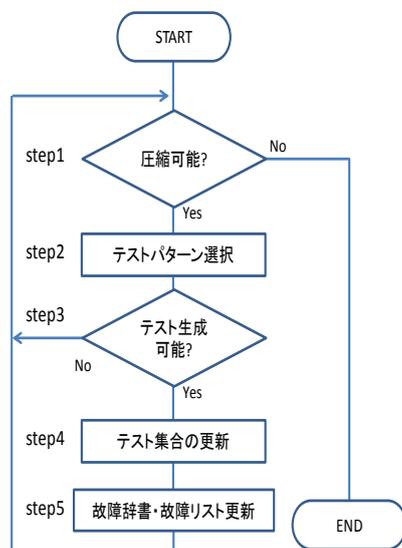


図 2. ツーバイワンアルゴリズムフローチャート

(step1)

テストパターン集合でまだ圧縮可能かどうかの判定を行う、2 個のテストパターンでのみ検出される故障を同時に検出可能なテストパターンが生成できる可能性があれば圧縮可能であると判定され step2 へ進む。選択する 2 個のテストパターンがなければ圧縮不可能と判断されツーバイワンアルゴリズムの処理を終了する。

(step2)

削除する 2 個のテストパターンの選択を行う。

(step3)

選択された 2 個のテストパターンの必須故障とその 2 個のテストパターンでのみ検出可能な故障を同時に検出可能なテストパターンを生成可能か否かの判定を行う。生成可能な場合はテストパターンを生成し、step4 へ進

み生成不可能な場合は step1 へ進む。

(step4)

生成されたテストパターンをテスト集合に追加し、step2 で選択した 2 個のテストパターンをテスト集合から削除し、テスト集合を更新する。

(step5)

更新されたテスト集合から故障辞書と故障リストを更新する。

4. テストパターン選択法

本章では、ツーバイワンアルゴリズムにおけるテストパターンの選択法を説明する。提案手法では、必須故障数がある閾値以下の各テストパターンから同時検出可能グラフを作成する。そして各テストパターンペアの必須故障数を基に削除するテストパターンを選択する。

4.1. 同時検出可能グラフ

同時検出可能グラフとは、各テストパターンを頂点とし、テストパターン間での必須故障の割当てが衝突しないテストパターンのペア間を辺で接続したものである。接続された 2 個の頂点は、2 個の頂点に対応するテストパターンのすべての必須故障を検出する、テストパターンを生成できる可能性があることを示す。例えば、信号線を 6 本持つ回路のテストパターン t_1 とテストパターン t_2 について考える。 t_1 で検出可能な必須故障が f_1, f_2, f_3 で、それぞれの必須割当てが $01XXX, X10XXX, 01XX1X$ であり、 t_2 の必須故障が f_4, f_5 で、それぞれの必須割当てが $0X0XXX, X1XX11$ とする。 t_1 と t_2 のそれぞれの必須故障の必須割り当てを併合し、 $tp1:010X1X, tp2:010X11$ を生成する。そして $tp1$ と $tp2$ の必須故障ではなく 2 つのテストパターンでしか検出できない故障が存在するか否かの検証をする。存在しなければ $tp1'$ と $tp2'$ の値が衝突しないか否かの判定を行い、衝突しない場合は頂点 $tp1$ と $tp2$ の間に辺を接続する。必須故障ではなく 2 個のテストパターンでしか検出できない故障が存在する場合はその故障の必須割当てと $tp1'$ と $tp2'$ の値が衝突するか否かの判定を行う。値が衝突しない場合は $tp1$ と $tp2$ の間に辺を接続する。この処理を全ての頂点に対して行い、同時検出可能グラフを作成する。

図 3 に同時検出可能グラフの例を示す。例として、あるテスト対象回路のテスト集合のテストパターン数は 8 個であり、各テストパ

ターンの必須故障の必須割当てが tp1 は 010X1X, tp2 は 010X11, tp3 は 01XXX1, tp4 は 110X11, tp5 は X10X01, tp6 は 110XX1 とする. 各頂点はテストパターンであり, 各辺は各頂点の必須故障と必須故障以外であり 2 個のテストパターンでしか検出できない故障をすべて検出できるテストパターンが生成可能である可能性があることを示している.

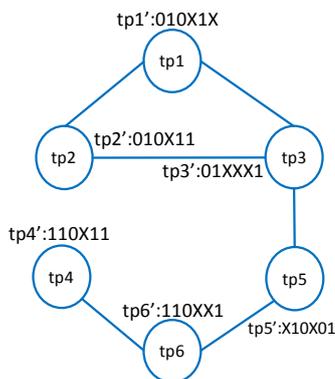


図 3.同時検出可能グラフ

4.2.テストパターン選択

本提案手法のテストパターンの選択は同時検出可能グラフと各テストパターンの必須故障数に着目して行う. 新たなテストパターンを生成できるか否かは必須故障数によって異なる. 必須故障数が大きくなると生成できる可能性は低くなり, 必須故障数が少ないと生成できる可能性は高くなる. そのため, 同時検出可能グラフの頂点の間に辺が接続されている組み合わせの中で 2 個の頂点の必須故障数の和が小さい組み合わせを選択する. 図 3 の同時検出可能グラフで, 各テストパターンの必須故障数が, tp1 が 3 個, tp2 が 2 個, tp3 が 3 個, tp4 が 3 個, tp5 が 1 個, tp6 が 3 個の場合について考えると tp3 と tp5 の必須故障の和が 4 となり, 最小となる. そのため, テストパターンは tp3 と tp5 が選択される.

5. 予備実験

本章では, 提案手法の評価のための予備実験について示す. 対象回路は ISCAS89 の順序回路を組合せ回路に変更した回路を対象とし実験を行った. テスト集合は TetraMax で生成した極小テスト集合を用いた. 必須故障の閾値を 10 に設定し, 必須故障数が 10 以下のテストパターンのペアを, 重複しないようすべて選択し, テストパターンが生成できるか否か検証した. それぞれのペアでしか検出できない故障すべてを検出するテストパターンを生成できた場合は, ペアをテスト集合か

ら削除し, 生成したテストパターンテスト集合に追加した. 表 3 に予備実験の結果を示す.

表 3.予備実験

故障数	信号線数	テストパターン数	
		TMAX	閾値(10)
s27_C	26	7	7
s298_C	298	28	28
s382_C	382	28	28
s400_C	400	28	27
s420_C	420	45	45
s444_C	444	28	27
s526_C	526	55	55
s713_C	713	32	30
s832_C	832	100	100
s838_C	838	80	79
s953_C	930	81	81
s1196_C	1195	127	127
s1238_C	1237	138	137
s1423_C	1423	37	36
s1488_C	1488	112	112
s5378_C	5269	124	123
s9234_C	9234	147	144

6.おわりに

本論文では, ツーバイワンアルゴリズムにおけるテストパターン選択法を提案した. 今後の課題としては, ツーバイワンアルゴリズムにおける更に効率のいいテストパターン選択法の提案や大規模回路による実験評価, N By M アルゴリズムへの拡張などが挙げられる.

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]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.
- [5]M. Abramovici., M. A. Breuer and A. D. Friedman., "Digital systems testing and testable design"., IEEE Press., 1995.