

## トランジション故障に対するテストパターンの極小化手法について

梶原誠司<sup>†</sup>

イリス ポメランツ<sup>\*\*</sup>

スダーカ M レディ<sup>\*\*</sup>

<sup>†</sup>九州工業大学情報工学部電子情報工学科  
〒820 福岡県飯塚市川津680-4  
E-mail: kajihara@cse.kyutech.ac.jp

<sup>\*\*</sup>アイオワ大学電気コンピュータ学科  
アイオワシティ, アイオワ州, 米国  
E-mail: {irith, reddy}@eng.uiowa.edu

あらまし 本論文では、組合せ回路のトランジション故障を対象としたテスト生成におけるテストパターン圧縮手法を提案する。トランジション故障の検出には2つのテストパターンが必要であるが、提案する手法は、典型的なテスト圧縮手法の動的圧縮と静的圧縮を2パターンテストに拡張したものである。動的圧縮については、無駄な信号値の割当てをできるだけ少なくする手法である。また、静的圧縮については、不要なテストベクトルが与えられたテスト系列に完全に含まれないようにする新しい手法である。ISCASのベンチマーク回路に対する実験では、動的圧縮と静的圧縮のどちらにおいても提案手法が有効であることを示す。

キーワード テスト生成, テスト圧縮, トランジション故障, 組合せ回路

### A Method of Minimal Test Patterns for Transition Faults

Seiji Kajihara<sup>†</sup>

Irith Pomeranz<sup>\*\*</sup>

Sudhakar M. Reddy<sup>\*\*</sup>

<sup>†</sup>Dept. of Computer Science and Electronics,  
Kyushu Institute of Technology  
680-4 Kawazu, Iizuka 820 Japan  
E-mail: kajihara@cse.kyutech.ac.jp

<sup>\*\*</sup>Electrical and Computer Eng. Dept.,  
University of Iowa  
Iowa City, IA 52242, U.S.A.  
E-mail: {irith, reddy}@eng.uiowa.edu

Abstract This paper proposes a method of test compaction for transition faults in combinational circuits. Since two patterns are required to detect a transition fault, we extend the test compaction techniques for stuck-at faults to two-pattern test. The proposed method includes both a static compaction technique and a dynamic compaction technique which are typical test compaction techniques. The proposed dynamic compaction technique is to avoid assigning unnecessary logic values to primary inputs. And the proposed static compaction technique is to make a test sequence minimal. Experimental results for ISCAS benchmark circuits show the effectiveness of the proposed method.

key words test generation, test compaction, transition fault, combinational circuit

## 1. はじめに

論理回路のテスト生成アルゴリズムは、以下の3つの基本的要求を満たすように開発されてきた。

- ・実用的な時間で計算を終了すること。
- ・論理的な誤りをもたらず縮退故障に対して、高い故障検出率を持つテストパターンを生成すること。
- ・生成されるテストパターンの数をできるだけ少なくすること。

これらの要求に対しては、それぞれ優れた手法が提案されており、その有効性も知られている[1-7]。しかしながら、集積回路の製造技術・設計技術は格段の進歩を遂げており、製造された回路の信頼性を十分に保証するには、テスト生成において、回路の動作速度に影響を与える故障(遅延故障)をも検出するテストパターンを生成することが重要となっている。

遅延故障のテストは信号変化の伝搬時間の増大をテストするため、テストパターンには連続する2つのテストベクトルが必要である。遅延故障のモデルとしては、ゲート遅延故障[8]とパス遅延故障[9]の2つの故障モデルがよく知られているが、2パターンテストが必要な故障モデルという意味では、他にトランジション故障[10]やスタックオープン故障[11]がある。テスト生成を考えると、トランジション故障、スタックオープン故障、ゲート遅延故障は、縮退故障のテスト生成を2パターンテストに拡張することで扱うことができる。その一方で、最も信頼性を高めることのできる故障モデルのパス遅延故障は、故障数の増大などの問題があり[12]、縮退故障のテスト生成を2パターンテストに拡張するだけではうまくいかない。

本研究では、組合せ回路のトランジション故障を対象としたテスト生成におけるテストパターン圧縮手法を提案する。トランジション故障は、最も簡略化された遅延故障のモデルである。故障の顕在化条件を変えれば、スタックオープン故障、ゲート遅延故障が扱えるため、トランジション故障を対象にテスト圧縮技術を開発できれば、スタックオープン故障、ゲート遅延故障、そして、パス遅延故障のテスト圧縮にも利用できる。

縮退故障とトランジション故障の大きな違いは、検出に2パターンテストが必要なことである。そのため、本研究では、典型的なテスト圧縮手法の動的圧縮と静的圧縮[4]を2パターンテストに適用する手法を述べる。動的圧縮については[13]で提案されているが、本研究で提案する手法は、無駄な信号値の割当を

できるだけ少なくする手法である。また、静的圧縮については[14]で提案されているが、本研究で提案する手法は、不要なテストベクトルが与えられたテスト系列に完全に含まれないようにする新しい手法である。ISCASのベンチマーク回路[15]に対する実験では、動的圧縮と静的圧縮のどちらにおいても提案手法が有効であることを示す。

以下、本報告は以下のように構成する。2章では、トランジション故障のテスト生成について述べる。3章では動的圧縮手法と静的圧縮手法を提案する。4章でアルゴリズムと実験結果を示し、5章で本報告をまとめる。

## 2. トランジション故障のテスト生成

### 2.1 トランジション故障

トランジション故障とは、各信号線における信号の遷移時間が増大するような故障である。信号の遷移時間には立ち上がりと立ち下りの2種類あるため、等価故障を考慮しない場合には、故障数は信号線数 $\times$ 2となる。故障の検出には連続する2つのテストベクトルが必要であるが、ここでは、それを初期化ベクトルと検出ベクトルと呼ぶことにする。信号線 $l$ の立ち上がり故障の検出には、初期化ベクトルで信号線 $l$ の値を0にし、検出ベクトルで信号線 $l$ の値を1にしたうえでその値を外部出力に伝搬させることが必要となる。したがって、信号線 $l$ の立ち上がり故障の検出ベクトルは $l$ の0縮退故障のテストベクトルと同じであり、立ち下り故障の検出ベクトルは $l$ の1縮退故障のテストベクトルと同じとなる。

図1は、トランジション故障、スタックオープン故障、ゲート遅延故障の3つの故障モデルに対して、ANDゲートの上側の入力信号線の故障を顕在化する信号値の割当を示す。このように、テスト生成において初期化ベクトルの信号値割当の違いのみで、対象故障モ

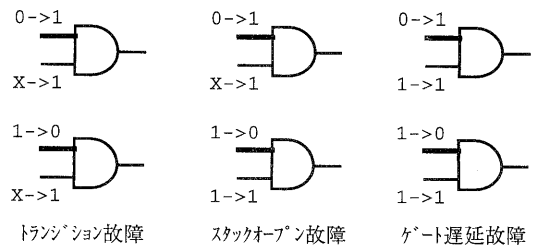


図1 故障顕在化のための信号値割当

デルをトランジション故障からスタックオープン故障またはゲート遅延故障に変えることができる。このため、本研究で提案するテスト圧縮手法は、スタックオープン故障やゲート遅延故障にも適用可能となる。

## 2. 2 関連研究

縮退故障に対するテスト圧縮については数多くの研究が行われている[4-7]。テスト圧縮手法は、テスト生成アルゴリズムの中で用いられる動的圧縮と、生成したテスト集合を小さくする静的圧縮に分けられる。静的圧縮では、冗長なテストベクトルを識別し、不要な(冗長な)テストベクトルをすべてテスト集合から取り除く手法が提案されている[6, 7]。これは、他のテストベクトルで検出されない故障を検出するテストベクトルは必要なテストベクトルであるが、そうでないテストベクトルは不要なテストベクトルと判断する手法である。この手法は2パターンテストにも適用できる[13]。これは、あるテストベクトルが、1度しか検出されない故障(必須故障という[3, 6])の初期化ベクトルでも検出ベクトルでもない場合、不要なテストベクトルと判断できる。例えば、図2(a)のように、 $f_1, f_2, f_3$ の各故障がどのテストベクトルで検出されたかが表された場合、 $v_1$  および  $v_4$  は冗長なテストベクトルとなり、例えば、図2(b)のようなテストパターンにすることができる。本研究では、生成されたテストパターンから、さらに多くのテストベクトルを削除する新しいアイデアを提案する。

2パターンテストの動的圧縮手法については、[14]で提案されている。この手法は、テストベクトルを、一つずつ印加順に生成する。ある故障の検出ベクトル  $v_i$  を生成した後、 $v_i$  のドントケアの入力を用いて、 $v_{i-1}$  が初期化ベクトルになっているできるだけ多くの未検出故障の検出ベクトルになるように  $v_i$  を変更し、

	故障		
	$f_1$	$f_2$	$f_3$
テストベクトル $v_1$	○		
テストベクトル $v_2$		○	
テストベクトル $v_3$			○
テストベクトル $v_4$	○		
テストベクトル $v_5$			○
テストベクトル $v_6$			○

(a) 冗長なテスト

	故障		
	$f_1$	$f_2$	$f_3$
テストベクトル $v_1$	○		
テストベクトル $v_2$		○	
テストベクトル $v_3$			○
テストベクトル $v_5$			○
テストベクトル $v_6$			○

(b)  $v_4$  除去後のテスト

図2 故障顕在化のための信号値割当

さらに、できるだけ多くの未検出故障の初期化ベクトルとなるようにする。この手法では、無駄な信号値の割当が多く発生する可能性がある。本研究では、無駄な信号値割当が少なくなるような動的圧縮手法を提案する。

## 3. テスト圧縮手法

### 3. 1 動的圧縮

[14]のようにテストベクトルを  $v_1, v_2, v_3, \dots$  のように一つずつ生成する場合、 $v_i$  は動的圧縮で複数の故障の初期化ベクトルになるが、それらの故障の検出ベクトルを直後の  $v_{i+1}$  で生成できるとは限らない。このため、 $v_i$  には無駄な信号値の割当が含まれることとなり、動的圧縮の効率を十分にあげることができない。そこで本研究では、 $(v_1, v_2), (v_2, v_3), (v_3, v_4), \dots$  のように2つのテストベクトルごとに生成し、その中で動的圧縮を行うことを提案する。

ある故障  $f_a$  についてテストベクトル  $(v_i, v_{i+1})$  が生成されたとすると、動的圧縮では図3に示すように次の3通りのテスト生成が考えられる。

- ・  $v_i$  が既に初期化ベクトルとなっている故障  $f_b$  に対して、 $v_{i+1}$  を検出ベクトルにする。
- ・  $v_{i+1}$  が既に検出ベクトルとなっている故障  $f_c$  に対して、 $v_i$  を初期化ベクトルにする。
- ・ ある故障  $f_d$  に対して、 $v_i$  を初期化ベクトルに、 $v_{i+1}$  を検出ベクトルにする。

少ない信号値の割当で故障を検出するには、 $f_b$  や  $f_c$  のように既に初期化ベクトルか検出ベクトルが存在する故障を扱うのがよい。このうち、 $v_i$  が初期化ベクトルとなっている故障を調べる処理は高速に実行可能であるが、 $v_{i+1}$  が検出ベクトルとなっている故障を調べるには、故障シミュレーション的な処理が必要となる。計算時間の短縮の観点からは、圧縮途中における故障シミュレーションは避けることが望ましい。そこで、本研究の動的圧縮では、以下のような優先順位で動的圧縮のための故障を選択し、テストパターンを生成する。

	$f_a$	$f_b$	$f_c$	$f_d$
$v_i$	○	○	○	○
$v_{i+1}$	○	○	○	○

図3 動的圧縮のパターン

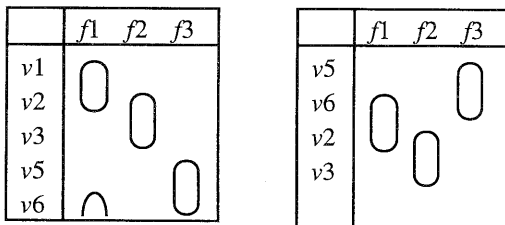
- (1)  $vi$  が既に初期化ベクトルとなっている故障に対して、 $vi+1$  を検出ベクトルにする。
- (2)  $vi$  が初期化ベクトルとなりうる故障に対して、 $vi+1$  を検出ベクトルにし、その生成に成功したら、 $vi$  を初期化ベクトルにする。

動的圧縮のための故障シミュレーションは実行しないため、 $vi+1$  が既に検出ベクトルとなっている故障が選択される場合もある。

### 3. 2 静的圧縮

次に与えられたテストパターンをより小さくする新しい手法を述べる。2パターンテストでは、テストパターンがテスト系列として扱われるため、テストベクトルの印加順序を変えことはできない。例えば、図2(b)では、 $v1, v2$  が故障  $f1$  を検出し、 $v2, v3$  が故障  $f2$  を、 $v5, v6$  が故障  $f3$  を検出するため、 $v1, v2, v3$  と  $v5, v6$  の印加順序を変えることはできない。しかしながら、 $v1, v2, v3$  を  $v5, v6$  の前に印加する必然性はなく、 $v5, v6, v1, v2, v3$  の順序にすることは可能である。このとき、 $v1$  は故障  $f1$  の初期化のために必要なテストベクトルであるが、 $v6$  が  $f1$  の初期化ベクトルの役割を果たすなら、 $v5, v6, v2, v3$  にできる(図4)。つまり、初期化ベクトルとしてのみ使われているテストベクトルは、検出ベクトルとしてのみに使われている他のテストベクトルでその役割を果たすことができるなら、削除できる可能性がある。

冗長ベクトルの判定には、必須故障を対象に初期化ベクトルか検出ベクトルになっているテストベクトルを調べればよい。しかし、上述のテスト系列の並べ替えによるテストベクトルの削減では、必須故障を対象にただけでは故障の見逃しが生じることがある。例えば、図4(a)のテスト系列  $v1, v2, v3, v5, v6$  において、 $v1, v2$  と  $v3, v5$  でのみ検出されるような故障  $f4$  があつたと仮定する。その故障は必須故障ではないが、テ



(a) 元のテスト系列 (b) 並び替え後のテスト系列  
図4 テスト系列の入れ替えによる静的圧縮

ト系列  $v5, v6, v2, v3$  では検出できない。このような検出可能故障の見逃しを防ぐには、新しく得られたテスト系列で故障シミュレーションを行って見逃しがないことを確認するか、または、必須故障の計算のみでなく2回検出される故障も計算する必要がある。

### 3. 3 必須故障の削減

静的圧縮によってテストベクトルを削減するには、テストベクトルが持つ必須故障の数を0にする必要がある。静的圧縮におけるパターン削減の効果を上げるには、動的圧縮において必須故障を検出するようにテストパターンを生成すればよい[6], [13]。

例えば、図4(b)のように各故障を検出するパターンが存在する場合、故障  $f1$  が他のテストベクトルで検出されるようになって、圧縮は不可能であるが、 $f1$  以外の故障が他のテストベクトルで検出されるなら、圧縮が可能となる。

## 4. 実験結果

### 4. 1 テスト生成アルゴリズム

本研究において作成したトランジション故障用のテスト圧縮アルゴリズムは、以下のような手順のものである。

[テスト圧縮アルゴリズム]

- step 1: 初期化ベクトルと検出ベクトルのいずれかが生成できない故障を取り除いて、故障リストを作成する。
- step 2: 未検出故障に対する動的圧縮を伴ったテスト生成により、回路に対するテストパターンを生成する。
- step 3: 必須故障が0のテストベクトルを削除する静的圧縮により、生成したパターンを圧縮する。
- step 4: テストベクトルの入れ替えによる静的圧縮により、生成したパターンを圧縮する。
- step 5: 得られたパターンに対して、必須故障に対する動的圧縮を実行する。
- step 6: 必須故障が0のテストベクトルを削除する静的圧縮により、生成したパターンを圧縮する。
- step 7: テストベクトルの入れ替えによる静的圧縮により、生成したパターンを圧縮する。

まず step 1 では、縮退故障用の ATPG により検出できない故障を取り除いて、故障リストを生成する。代表故障の選択には信号値の支配関係も利用したの

で、通常の等価故障解析による故障の選択より故障数を少なくできる。step 2 では、テスト系列を生成する。選択する故障は、縮退故障の独立故障集合 [6] に基づいて決定している。また、このテスト生成では、巡回後方追跡 [5] も導入している。ここで生成するテストベクトルに残っているドントケアには論理値を割り当てず、step 5 での動的圧縮に利用する。step 3 は、従来型の必須故障の有無による静的圧縮を実行する。続く step 4 では、3.2 節で提案したテスト系列の並び替えによる静的圧縮を実行する。その後、step 5 でテストパターン中のドントケアを利用して、必須故障に対する動的圧縮による入力値の割当を行う。最後に、再び静的圧縮を適用し、最終的なテスト系列を得る。

このアルゴリズムの中で用いられる新しいテスト圧縮手法の特徴としては、

- ・故障リストをより圧縮したものを使用。
- ・テストパターンを 2 つずつ生成 (3.1 節)。
- ・すべての外部入力に値を割当てる前に静的圧縮。
- ・テスト系列の並び替えによる静的圧縮 (3.2 節)。

の 4 点が上げられる。

#### 4. 2 実験結果

提案手法を SUN Ultra1 Model 140 のワークステーション上にプログラム化し、ISCAS'85 のベンチマーク回路に適用することで、有効性を検証した。表 1 にその結果を示す。表中の 2 列目から 4 列目は故障数を表す。"Total faults" の欄は、通常の等価故障に基づいたトランジション故障の代表故障数を示し、そのうち検出可能な故障数は "Det. faults" の欄に示す。本手法で用いた圧縮故障リストでは、"Det. faults" の故障を

"Com. faults" の欄に示す値にまで削減した。約 20% の削減である。

テスト生成結果は、表中の 5 列目から 9 列目までに示す。"DC1" の欄は、未検出故障にのみ動的圧縮を適用して (step 2 で) 得られたテストベクトル数である。"DC1-ST1" と "DC1-ST2" の各項目は、"DC1" のテスト系列に、それぞれ必須故障に基づく静的圧縮 (step 3) と更にテスト系列の並び替えによる静的圧縮 (step 4) を適用して得られたテストベクトル数である。"DC2-ST1" と "DC2-ST2" の各項目は、必須故障に動的圧縮を適用した後、それぞれ必須故障に基づく静的圧縮 (step 6) と更にテスト系列の並び替えによる静的圧縮 (step 7) を適用して得られたテストベクトル数である。回路によって有効な圧縮手法は異なるが、平均的には、どのステップにおいても同程度の圧縮率が得られており、提案手法が有効であったといえる。なお、すべての処理に要した計算時間を、表 1 の最も右の欄に示す。

表 2 には、提案手法と従来手法 [13], [14] とを比較した結果を示す。"DC" の欄には、動的圧縮のみの結果を、"DC-ST" の欄には、動的圧縮と静的圧縮の両方を適用した結果を示す。[13] では動的圧縮に特別な手法が取られておらず、最初に得られたテストベクトル数は大きい。また、静的圧縮も必須故障に基づく圧縮のみのため、十分に小さなテスト系列になっていない。しかしながら、一旦生成したテストベクトルを変更する後処理を導入しており、最終的には "Final" に示す値が得られている。

#### 5. まとめ

本報告では、組合せ回路のトランジション故障を対

表 1 実験結果

circuit	Total faults	Det. faults	Com. faults	DC1	DC1	DC1	DC2	DC2	Time (sec.)
				-ST1	-ST2	-ST1	-ST2		
c880	1582	1582	1277	43	43	43	42	42	3.7
c1355	2566	2558	1980	203	187	175	174	174	38
c1908	2938	2929	2476	219	206	192	188	178	63
c2670	4306	4151	3461	118	117	112	112	107	60
c3540	5654	5459	4513	233	233	228	215	204	143
c5315	8842	8780	7324	96	96	95	95	94	88
c6288	12512	12427	9580	58	54	54	54	53	108
c7552	12284	12141	10004	151	151	149	146	146	587

表2 結果の比較

circuit	DC			DC-ST		Final [13]
	ours	[13]	[14]	ours	[13]	
c880	43	70	62	42	70	38
c1355	203	425	233	174	261	156
c1908	219	287	239	178	242	144
c2670	118	151	147	107	145	72
c3540	233	257	135	204	251	150
c5315	96	168	107	94	163	72
c6288	58	113	52	53	75	44
c7552	151	220	152	146	219	100

象としたテスト生成におけるテストパターン圧縮手法を提案した。典型的なテスト圧縮手法の動的圧縮と静的圧縮をトランジション故障の検出のための2パターンテストに適用する新しい手法を述べた。ISCASのベンチマーク回路に対する実験では、動的圧縮と静的圧縮のどちらの提案手法もテスト圧縮に有効であることを示した。本報告では、テスト圧縮の要素技術を提案しただけであるが、テスト最小化において、これらの技術をどのような順序で適用するのがよいか検討することや、テストベクトルの変更による圧縮技術を開発することが今後の課題である。

#### 参考文献

- [1] M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design*, COMPUTER SCIENCE PRESS, 1990.
- [2] M. Schulz, and E. Auth, "Advanced Automatic Test Pattern Generation and Redundancy Identification Techniques," FTCS-18, pp. 30-35, June, 1988.
- [3] J. -S. Chang, C. -S. Lin, "Test Set Compaction for Combinational Circuits," First Asian Test Symposium, pp. 20-25, Nov. 1992.
- [4] P. Goel, and B. C. Rosales, "Test Generation and Dynamic Compaction of Tests," Digest of Papers 1979 Test Conf., pp. 189-192, Oct. 1979.
- [5] I. Pomeranz, L. N. Reddy, and S. M. Reddy, "COMPACTEST: A Method to Generate Compact Test Sets for Combinational Circuits," International Test Conf., pp. 194-203, Oct. 1991.
- [6] S. Kajihara, I. Pomeranz, K. Kinoshita and S. M. Reddy, "Cost-Effective Generation of Minimal Test Sets for Stuck-at Faults in Combinational Logic Circuits," IEEE Trans. CAD., Vol. 14, No. 12, pp.1496-1504, Dec. 1995.
- [7] I. Hamzaoglu, and J. H. Patel, "Test Set Compaction Algorithms for Combinational Circuits," International Conf. on Computer-Aided Design, pp. 283-289, Nov. 1998.
- [8] G. L. Smith, "Model for Delay Faults Based upon Paths," International Test Conf., pp. 342-349, 1985.
- [9] Z. Barzilai and B. K. Rosen, "Comparison of AC Self-testing Procedures," International Test Conf., pp. 89-91, 1983.
- [10] J. A. Waicukauski, E. Lindbloom, B. K. Rosen, and V. S. Iyengar, "Transition Fault Simulation," IEEE Design & Test of Computers, 4, pp. 32-38, April 1987.
- [11] R. L. Wadsack, "Fault Modeling and Logic Simulation of CMOS and MOS Integrated Circuits", Bell System Technical Journal, 57, pp. 1449-1473, May-June 1978.
- [12] I. Pomeranz and S. M. Reddy, "An Efficient Non-Enumerative Method to Estimate Path Delay Fault Coverage," International Conf. on Computer-Aided Design, pp. 560-567, Nov. 1992.
- [13] I. Hamzaoglu, and J. H. Patel, "Compact Two-Pattern Test Set Generation for Combinational and Full-Scan Circuits," International Test Conf., pp. 944-953, Nov. 1998.
- [14] L. N. Reddy, I. Pomeranz, and S. M. Reddy, "COMPACTEST-II: A Method to Generate Compact Two-Pattern Test Sets for Combinational Logic Circuits," International Conf. on Computer-Aided Design, pp. 568-574, Nov. 1992.
- [15] F. Brglez, and H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Design and a Special Translator in Fortran," ISCAS'85, June 1985.