

ドントケア抽出を用いた縮退故障テストの 遷移故障検出率向上手法

濱崎 和光[†] 細川 利典[‡]

[†] 日本大学大学院生産工学研究科 〒275-8575 千葉県習志野市泉町 1-2-1

[‡] 日本大学生産工学部 〒275-8575 千葉県習志野市泉町 1-2-1

E-mail: [†] c77130@cit.nihon-u.ac.jp, [‡] t7hosoka@cit.nihon-u.ac.jp

あらまし 近年 VLSI の大規模化、複雑化に伴い、縮退故障テスト以外に遷移故障やブリッジ故障に対するテストの要求が高まってきている。しかしながら、縮退故障以外の故障モデルを検出するために新たにテストパターンを追加すると、テストパターン数に比例してテストコストが増大する。本論文では、ドントケア抽出技術を用いて与えられた縮退故障テストパターンに対して、ドントケアを抽出し、できるだけ多数の遷移故障を検出するようにドントケアに対する値の再割り当て方法を提案する。ITC'99 ベンチマーク回路に対して本提案手法を適用した結果、テストパターン数を増加させることなく、遷移故障検出率を最大 27% 向上させることができた。

キーワード ドントケア抽出、スキュードロード、遷移故障、逆シフト

On Improving Transition Fault Coverage of Stuck-at Fault Tests Using Don't Care Identification Technique

Kazumitsu HAMASAKI[†] Toshinori HOSOKAWA[‡]

[†] Graduate School of Industrial Technology, Nihon University 1-2-1, Izumicho, Narashino, Chiba 275-8575, Japan

[‡] College of Industrial Technology, Nihon University 1-2-1, Izumicho, Narashino, Chiba 275-8575, Japan

E-mail: [†] c77130@cit.nihon-u.ac.jp, [‡] t7hosoka@cit.nihon-u.ac.jp

Abstract In recent year, transition fault testing and/or bridging fault testing for VLSIs are increasingly required in addition to stuck-at fault testing because the number of gates on VLSIs is rapidly increasing and their complexity is growing with advances in semiconductor technology. However, additional test patterns to detect fault models other than stuck-at fault cause the increase of testing cost. In this paper, we propose a method to generate a modified test set that not only guarantees to detect stuck-at faults but also detects as many as possible transition faults by applying don't care identification techniques to a given stuck-at test set. Therefore, there are no negative impacts on testing cost. Experimental results for ITC'99 benchmark circuits show that the modified test sets obtained by the proposed method detect more transition faults from 27% than the test sets initially generated for stuck-at faults.

Keyword don't care, skewed load, transition fault, reverse shifting

1. はじめに

近年、VLSI の大規模化に伴いそれに対するテスト生成時間やそのテストコストが増大している[1]。その増大するテストコストを抑えるために効率のよいテストパターンの生成や回路のテスト容易化設計が行われてきた。テスト容易化設計の代表的なものとしては、回路中に存在するフリップフロップ (FF) に対しスキューチェーンを挿入することによって、順序回路を組合せ回路とみなしテスト生成を実行することができるフルスキュー設計がある。また、VLSI

のテストにおいて縮退故障検出だけでは不十分だとされており、縮退故障の検出に加えて遷移故障[2]や、ブリッジ故障[3][4]の検出が重要となってきている。

縮退故障以外の故障モデルを検出するためのテスト生成は縮退故障と比べて複雑である。また、それらの故障検出により多くのテストパターンが必要となり、結果としてテストコストの増大につながってしまう。そこで、縮退故障用に生成されたテストパターンを基に、そのテストパターン集合内でできるだけ多くの他の故障モデルの検出を行う

ことができれば、テストコストを増大させることなく縮退故障以外の故障に対するテストも行えることが可能になる。

生成されたテストパターンに対し、縮退故障検出率を維持しながらテストパターン中にドントケアを発生させるドントケア抽出技術が提案されている[5]。ドントケア抽出技術を適用することで生成されたドントケアを含んだテストパターンに対して、テストパターン数の削減[5][6]や単一縮退故障以外の故障モデルの検出[5]、消費電力の削減[7]などの新たな特性を持たせることが可能になると考えられる。

本論文では、フルスキャン設計された順序回路に対し、縮退故障用テストパターンを用いてスキュードロード方式[8][9]で遷移故障の検出を行う。さらに縮退故障用テストパターンに対してドントケア抽出を行い、抽出されたドントケアを再割り当てすることにより効率的に遷移故障検出率を改善する方法を提案する。

第2章でドントケア抽出技術について説明し、第3章でスキュードロード方式による遷移故障検出技術について説明する。第4章で遷移故障の検出を指向したドントケアの再割り当て法を提案し、第5章で縮退故障用テストパターンによる遷移故障検出率の評価を行い、第6章で結論と今後の課題について述べる。

2. ドントケア抽出

2.1. ドントケア

ドントケアとは、テストパターンにおいて対象となる故障を検出するときに必要のない論理値のことである。その値は"0"もしくは"1"のどちらでもよい。ドントケアはテストパターン中に"X"または"x"と表記される。

2.2. ドントケア抽出問題の定式化

本論文では、与えられたテストパターンから、潜在的に存在するドントケアを抽出する問題[3]を以下のように定式化する。与えられたテストパターン集合 T に対して次の(1)~(3)のような特性を持ったドントケアを含むテストパターン集合 T' を導出する。

- (1) T は T' を被覆する。
- (2) T と T' の縮退故障検出率は等しい。
- (3) T' はできるだけ多くのドントケアを含む。

図1の回路に対してテストパターン集合 T が与えられたとき、テストパターン集合 T' はドントケア抽出問題の解の1つである。テストパターン $t1$ は、故障 $a/0$, $b/0$, そして $c/1$ を検出する。ここで s/v は信号線 s の v 縮退故障を

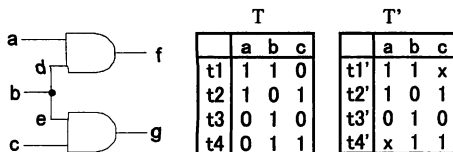


図1: テスト対象回路図とそのテストパターン集合

意味する。 $a/0$ は $t1$ 以外で検出することができないため、 $a/0$ は必須故障[10]である。しかしながら、 $c/1$ は $t3$ でも検出可能であるため、 $t1$ で必ずしも検出される必要はない。そのために、 $t1$ の入力 c の値"0"はドントケアにすることができる。同様にして $t4$ における外部入力 a の値もドントケアにすることができる。このようにして図1に示すテストパターン集合 T' を求めることができる。

3. 遷移故障検出

3.1. スキュードロード方式

遷移故障は2パターンで検出できる。第1パターンを初期パターン、第2パターンを遷移パターンと呼ぶ。本論文では遷移故障検出の方法としてスキュードロード方式を用いている。2パターンテストによる遷移故障の検出方法はスキュードロード方式の他にブロードサイド方式[11]が存在するが、今回は縮退故障用のテストパターン集合を用いて遷移故障の向上を目指すため、スキュードロード方式を用いた。スキュードロード方式は大きく分けて2つの動作からなるテスト方式である。まずシフト動作により初期パターンをスキャンフリップフロップ(SFF)に設定する。次に1ビットシフト動作を行って遷移パターンを発生させ、故障の影響をキャプチャ動作でSFFに取り込んで検出する。図2はスキュードロード方式における信号線 a の立ち上がり遷移故障の検出を行う例である。まずシフト動作によりスキャンチェーンに1時刻目のテストパターンである初期パターンを印加する。この場合ではスキャンインに(1,0,0)の順で入力する。この状態で1ビットのみシフトを行う。2時刻目には1ビットシフト後の値が各SFFに印加されており、スキャンインに近いSFFに印加される値を"1"とするとSFFの値は左からスキャンイン印加順に(0,0,1)と表すことができる。1時刻目で信号線 a の値が"0"になり、かつ2時刻目に信号線 a の値が"1"に遷移すると仮定する。1時刻目で信号線 a の値が"0"となり2時刻目で信号線 a の0縮退故障が検出できれば、信号線 a の立ち上がり遷移故障が検出できる。

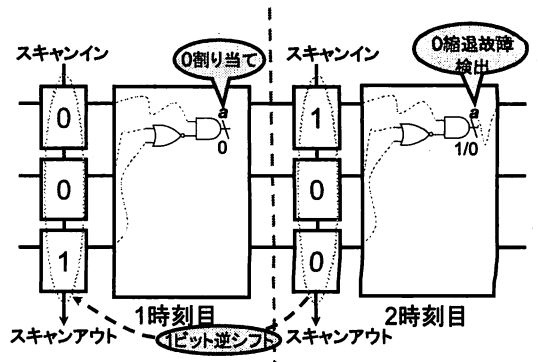


図2: スキュードロード方式

3. 2. 2パターンテスト変換

通常、遷移故障は1パターンテストでなく2パターンテストで検出されるため ATPG(Automatic Test Pattern Generation)で縮退故障検出用に生成されたテストパターンを2パターンテスト用に変換する必要がある。ATPGにより生成された縮退故障検出用テストパターン集合を $T=\{t1,t2,t3,t4\}$ とする。以下に2パターンテストを行うテストパターンに変換する例を示す。

例として、外部入力数が3、スキャンチェーン長が5であるフルスキャン設計された順序回路における縮退故障用テストパターンを変換する。

ATPGより表1のようなテストパターンが生成されたものとする。左3ビット(p1~p3)が外部入力から入力される値、右5ビット(s1~s5)がシフト動作で設定される擬似外部入力(SFF)への入力値である。スキャンチェーンはATPGより表1のようなテストパターンが生成されたものとする。左3ビット(p1~p3)が外部入力から入力される値、右5ビット(s1~s5)がシフト動作で設定される擬似外部入力(SFF)への入力値である。スキャンチェーンは(スキャンイン→s1→s2→s3→s4→s5→スキャンアウト)の順に接続されていると仮定する。

ここでt1~t4までのテストパターンをそれぞれ1ビット逆シフトしたものをt1'~t4'すると、スキュードロード方式のテストパターンは表2ようになる。逆シフト[12]とは、仮想的にスキャンチェーン方向(スキャンインからスキャンアウトへの方向)の反対方向に値をシフトすることである。t1'~t4'は初期テストパターンでありt1~t4は遷移パターンである。

表1：縮退故障テストパターン集合 T1

	p1	p2	p3	s1	s2	s3	s4	s5
t1	0	X	0	1	1	X	0	1
t2	0	0	X	1	X	0	X	X
t3	X	1	0	1	1	1	X	1
t4	0	1	X	0	X	0	1	X

表2：スキュードロード方式による
2パターンテスト集合 T2

	p1	p2	p3	s1	s2	s3	s4	s5
t1'	X	X	X	1	X	0	1	X
t1	0	X	0	1	1	X	0	1
t2'	X	X	X	X	0	X	X	X
t2	0	0	X	1	X	0	X	X
t3'	X	X	X	1	1	X	1	X
t3	X	1	0	1	1	1	X	1
t4'	X	X	X	X	0	1	X	X
t4	0	1	X	0	X	0	1	X

縮退故障に関しては各 t1~t4 で検出されることが ATPG によって保証されているので、SFF に関係ない t1' ~t4' の外部入力値は"0"か"1"のどちらでもよい。よってドントケアとして表記することができる。

擬似外部入力の値はスキャンチェーンを通り各 SFF へと伝搬されていく。よって、スキュードロード方式を適用するために1ビットのみ逆シフトを行う。ここで逆シフトを行った後に発生するスキャンアウトに最も近い SFF の値は"0"か"1"どちらでも良いため、この部分もドントケアとして表記することができる。

表1の与えられたテストパターン集合に対しドントケア抽出を行い、その後2パターン変換を行い表2のようなテストパターン集合 T2 を生成する。

4. 遷移故障検出を指向したドントケア値の再割当

縮退故障を1回検出するテストパターンに対しドントケア抽出を行った表2のテストパターン集合 T2 を用いて、遷移故障検出率が向上するようにドントケアの値の再割当てを行っていく。以下にその再割当てアルゴリズムを示す。

ATPGより縮退故障を一回のみ検出するテストパターン集合を生成し、それに対しドントケア抽出を行ったものをテストパターン集合 T とする。テストパターン集合 T を2パターンテスト用に変換させて、対象回路 C に対して遷移故障シミュレーションを実行する(1~6行目)。

全ての信号線に対し、その信号線の遷移故障が検出できたか否かを判定し、信号線 l の遷移故障が未検出ならば un_detect transition_fault とし、さらに何が原因で遷移故障を検出できなかったのかを判別する(9~20行目)。

2時刻目に l の v 縮退故障を検出している(detect_s-a-v)が、1時刻目に l に値 v が割当てられていない場合を casel(10~11行目)、1時刻目に l に値 v が割当てられているが、2時刻目に l の v 縮退故障を検出できていない場合を

```

1: Procedure X-delay(T,C)
2: Test set T;
3: Circuit under the test C;
4: {
5:   T' = 2_pattern_conversion(T);
6:   fault_simulation(T',C);
7:   for each line "l" in C;
8:   {
9:     if (l == un_detect transition_fault){
10:      if ((t1')="X") && (t1) == detect_s-a-v){
11:        case = 1;
12:        assignment_value_1(t1');
13:      }else if ((t1) = "v") && (t1) == un_detect_s-a-v){
14:        case = 2;
15:        assignment_value_2(t1);
16:      }else if ((t1) = "X") && (t1) == un_detect_s-a-v){
17:        case = 3;
18:        assignment_value_3(t1',t1);
19:      }
20:    }
21:    if ("X" in t1,t1'){
22:      "X" = assign 0 or 1;
23:    }
24:  }
25: }

```

図3：ドントケア再割当てアルゴリズム

case2 (13~14 行目), 1 時刻目に l に値 v が割当てられておらず, さらに 2 時刻目で l の v 縮退故障を検出できていない(un_detect_s-a-v)場合を case3 (16~17 行目)とする. なお, 「検出できていない場合」とは, 対象信号線 l の v 縮退故障がテストパターン ti で未検出のことを言う.

各 case に分類された未検出故障は, それぞれ対応した再割当関数(assignment_value)が実行され, 検出可能ならそれに応じた X の値の再割当を行う(12,15,18 行目). 最後に再割当が行われなかったドントケアに対して①ランダム, ②全て 0, ③全て 1, ④ローパワー[7]のいずれかの方法で割り当てる. 本来, 遷移故障には立上り遷移故障と立下り遅延故障が存在するが, 図 3 のアルゴリズムでは特に区別をしていない.

case1 で ti' において, 信号線 l に対して値を割当てるようにドントケアの再割当てを行いながら正当化操作を行う. case2 では ti において信号線 l の v 縮退故障を検出するようにドントケアの再割り当てを行う. case3 では ti' によって信号線 l に値 v の割当てと正当化処理を行い ti の縮退故障検出をするようにドントケアの再割り当てを行う. 以下にその例を示す.

図 4 はフルスキャン設計された回路の組合せ回路部分を示す. この時のスキャンチェーンは, スキャンイン→ $s1$ → $s2$ →スキャンアウトの順に接続していると仮定する. 信号線 e の立ち上がり遷移故障を検出する場合, 各 case における再割り当ては以下ようになる.

case1 では 2 時刻目に信号線 e の 0 縮退故障がすでに検出されており, 1 時刻目で信号線 e に "0" を割り当て正当化するために図 5 に示すように $p4$ のドントケアを "0" と再割り当てすればよい. case2 では, 1 時刻目で信号線 e に "0" がすでに割り当てられているが, 2 時刻目で e の 0 縮退故障が検出できていない状態である. このときは 2 時刻目に存在するドントケアを用いて故障を励起, 伝搬する. この例では $p2$ と $p4$ のドントケアを "0" と "1" に再割り当てすることで e の 0 縮退故障の検出が可能となる. case3 では 1 時刻目に e に "0" が割り当てられておらず, かつ 2 時刻目でも e の 0 縮退故障が検出されていない. このときは 1 時刻目, 2 時刻目共に存在するドントケアを用いて値を設定していかなければならない. 図 7 の例であると, 1 時刻目の $p4$ のド

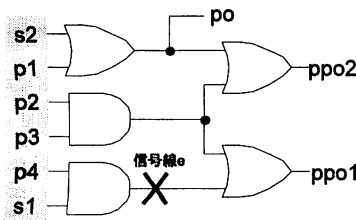


図 4: X 再割り当ての回路例

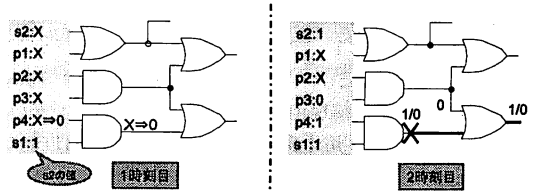


図 5: case1 に対する再割当て

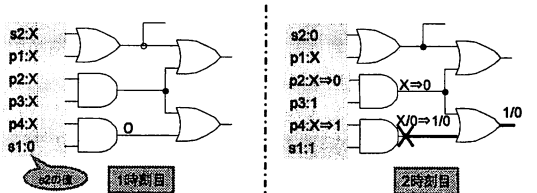


図 6: case2 に対する再割当て

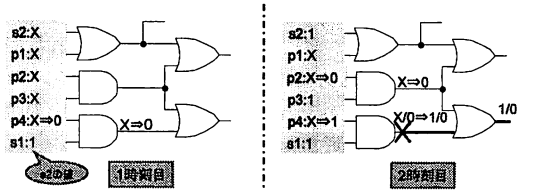


図 7: case3 に対する再割当て

ントケアを "0" にし, 2 時刻目の $p2, p4$ のドントケアをそれぞれ "0" と "1" に再割り当てしている.

全ての未検出故障について上記 case1 から case3 を実行し, 各 case 終了後に, 初期パターンと遷移パターンとの関係を考慮しながら "0" または "1" を割当てていく. また, 全ての case 終了後にまだ割当てが行われていないドントケアに対しても, 同様の動作を行う. 以下にその例を示す.

case1 から case3 までを実行した後, 表 3 のように再割り当てが行われなかったドントケアが存在したとする. スキャンチェーンはスキャンイン→ $s1$ → $s2$ → $s3$ → $s4$ → $s5$ →スキャンアウトのように接続されているものとする.

(1)外部入力($p1 \sim p3$)にドントケアが存在した場合, そのドントケアには "0" または "1" を①ランダム②全て 0③すべて 1④ローパワーのいずれかの方法に割り当てる.

(2)擬似外部入力($s1 \sim s5$)にドントケアが存在した場合, 3 つの場合が考えられる.

(2-1)初期パターンにドントケアが存在し, 1 ビット逆シフトする前の値がドントケアでない場合.

(2-2)遷移パターンにドントケアが存在し, 1 ビット逆シフトした後の値がドントケアである場合.

(2-3)遷移パターンにドントケアが存在し, 1 ビット逆シフトした後の初期パターンの値もドントケアである場合.

(1)は $t1$ の $p2$ の場合であり, ここには "0" または "1" をランダムで割当てて. ここでは 0 を割当てたとする. (2-1)は

表 3: 残ったドントケアの再割当

	p1	p2	p3	s1	s2	s3	s4	s5
t1'	0	1	1	1	0	0	1	1
t1	0	X	0	1	1	0	X	1
t2'	0	0	1	1	0	X	X	0
t2	0	0	0	1	1	0	0	X

表 4: 割当て後のテストパターン

	p1	p2	p3	s1	s2	s3	s4	s5
t1'	0	1	1	1	0	0	1	1
t1	0	0	0	1	1	0	0	1
t2'	0	0	1	1	0	0	1	0
t2	0	0	0	1	1	0	0	1

t2'の入力 s3 の場合であり、逆シフトする前の値 t2 の入力 s4 の値すなわち 0 が割当られる。(2-2)は t1 の s4 が該当し、逆シフト後の値 t1' の s3 の値である 0 が割当てられる。(2-3)は t2' の s4 と t2 の s5 が該当し、この 2 箇所には"0"または"1"どちらでも良いが、共に同じ値を割当てて必要がある。ここでは 1 を割当てることとする。表 4 は割当て後のテストパターンである。

5. 実験結果

本手法を C 言語を用いて実装し、ITC' 99 ベンチマーク回路に対して実験を行った。計算機は Celeron(R) CPU 2.40GHz、メモリ 512MB のものを用いた。テストパターンは縮退故障用 ATPG によって生成され、動的・静的圧縮されたものを用いている。

表 5 は ITC'99 ベンチマーク回路における実験結果である。また、case1~case3 実行後のテストパターン中に残存するドントケアは全て"0"または"1"の値をランダムに再割当てを行った。

『適用前』は ATPG(Synopsys 社 TetraMAX)により縮退故障のために生成したテストパターンによる結果であり、『適用後』は本論文の手法を適用し、変更されたテストパターンによる結果である。『検出率』は各手法が実行された時点での遷移故障検出率を表している。『X 抽出』は ATPG より与えられた縮退故障用テストパターンに対してドントケア抽出を行い、ドントケアがテストパターン中に含まれている状態での遷移故障検出率である。『case1~3』は各 case 実行後の遷移故障検出率を表しており、『Random』は case1~case3 を適用後に、残ったドントケアに対し"0"または"1"をランダムに割り当てた後、つまり本手法適用後の遷移故障検出率を表している。『X 使用量』は各段階で使用したテストパターン中に含まれているドントケアの割合を示している。『消費電力』は SFF がキャプチャ時に発生する値の遷移率を表しており、ATPG により生成された縮退故

障用テストパターンで故障シミュレーションを実行した時の値を 100 としたときの、本論文の提案手法適用後の割合を示している。また、表 6 は b12 回路において、case1~case3 適用後に残ったドントケアに対し『Random』以外の方法で値を割当てた場合の結果であり、ドントケアの割当て方は『Random』と『全て 0(All_0)』、『全て 1(All_1)』に割り当てた場合のものである。『縮退故障』は縮退故障検出率を表しており、『遷移故障』は故障検出率、故障検出数を表している。

表 5 より、ドントケア抽出を行った後、各 case を実行することにより、最終的には全ての回路において平均約 8.3% の遷移故障検出率の向上が見られた。中でも b02, b07 回路ではそれぞれ 27%, 26% 向上し、b15 回路においては 2500 個以上の遷移故障が新たに検出可能となった。また、各 case 別の遷移故障検出率を比較すると、case1 での検出率の向上が全 case の 80~90% を占めていた。これは遷移パターンに ATPG より出力された縮退故障テストパターンを用いているため、case1 に該当するテストパターンが非常に多くなっているためである。しかしながら、消費電力が平均 7%, 最大 18% 増加した。これは、より多くの遷移故障を検出しようとするためにキャプチャ時のスキャン FF の値の遷移確率が高くなってしまったためであると考えられる。

表 6 に、b12 回路に対して残ったドントケアを"0"または"1"の値をランダムに割り当てた場合、全て 0 を割当てた場合、全て 1 に割り当てた場合の遷移故障検出率、消費電力を示す。比較した結果、ランダムに割り当てたものがより多くの遷移故障を検出可能となったが、逆に全て 0 または全て 1 に割り当てたほうがランダムに割当てより消費電力を抑えることができた。ここで、全て 0 に割り当てているものが、手法適用前のものより消費電力が低くなっている。

6. おわりに

本論文ではドントケア抽出の応用の 1 つとして、スキュードロード方式における遷移故障の検出率を効率的に向上させる方法を提案した。

今後は、今回増大してしまった消費電力の削減を指向しながら遷移故障検出率を向上する方法や、さらに遷移故障検出率を向上させるために特定ビットにおけるドントケアを抽出する方法を検討していく予定である。

参 考 文 献

- [1] Toshinori Hosokawa, Masayoshi Yoshikawa, and Mitsuyasu Ohta. Novel DFT Strategies Using Full/Partial Scan Designs and Test Point Insertion to Reduce Test Application Time. IEICE A publication of the engineering sciences society, Nov. 2001.
- [2] A. Krstic, and K.-T. Cheng, "Delay Fault Testing for VLSI Circuits," Kluwer Academic Publishers, 1998.

表 5 : 実験結果

回路名	検出故障数		故障検出率(%)		遷移故障			検出率(%)				X含有量		X使用量(%)				消費電力
	対象故障数	適用前	適用後	縮退故障	適用前	適用後	X抽出	Case:1	Case:2	Case:3	Random	X抽出	Case:1	Case:2	Case:3	Random		
b01	222	130	178	100.00	58.55	79.27	36.03	71.17	75.52	77.70	79.27	36.97	18.06	9.70	3.36	5.88	107.69	
b02	122	67	101	100.00	54.91	82.78	50.81	79.50	82.78	82.78	82.78	20.00	10.00	10.00	0.00	0.00	114.63	
b03	732	634	677	100.00	86.61	92.48	58.33	81.14	85.73	91.09	92.48	57.69	9.90	1.93	8.83	37.03	103.31	
b04	3204	2717	2889	100.00	84.80	90.18	41.54	73.09	83.93	88.09	90.18	66.34	4.74	4.50	6.56	50.54	108.51	
b05	4612	2817	2916	100.00	61.07	63.22	54.29	59.28	60.93	61.99	63.22	35.25	3.31	3.37	8.44	20.13	100.79	
b06	254	209	226	100.00	82.28	88.97	51.98	86.22	87.81	88.97	88.97	69.70	14.07	1.52	3.46	50.65	100.70	
b07	2014	1139	1677	100.00	56.55	83.26	43.79	72.44	78.09	81.28	83.26	58.40	3.23	2.65	6.82	45.80	100.90	
b08	834	646	666	100.00	77.45	79.85	68.34	77.22	78.89	79.85	79.85	56.90	3.71	2.36	3.23	47.69	102.43	
b09	784	644	651	100.00	84.29	85.20	71.72	82.58	84.49	84.99	85.20	38.17	3.15	3.03	4.76	27.23	99.65	
b10	948	739	743	100.00	77.95	78.37	50.33	74.07	76.23	76.65	78.37	60.59	7.22	2.15	4.55	46.67	116.45	
b11	3340	2527	2643	100.00	75.65	79.13	51.25	72.45	75.74	77.98	79.13	61.08	4.14	3.78	8.79	44.39	118.54	
b12	5202	4294	4425	100.00	82.54	85.06	54.15	73.81	80.72	82.35	85.06	78.27	1.94	2.82	4.04	68.47	116.32	
b13	1588	1254	1325	100.00	78.98	83.43	58.18	76.65	80.41	81.73	83.43	67.65	4.39	2.80	6.63	53.83	101.55	
b15	41270	29897	32778	99.89	72.44	79.44	45.73	66.95	68.34	71.64	79.44	88.32	12.41	4.10	6.02	65.79	118.86	
平均					73.86	82.19	52.80	74.76	78.54	80.51	82.19	56.81	6.76	3.91	5.38	40.30	107.88	

表 6 : b12 回路実験結果

b12	割当て方式	縮退故障		遷移故障		消費電力
		検出率	検出数	検出率	検出数	
適用前		100.00	4293	82.54	4293	100.00
適用後	Random	100.00	4425	85.06	4425	116.32
	All_0	100.00	4310	82.87	4310	89.08
	All_1	100.00	4286	82.37	4286	109.21

- [3] Kohei Miyase, Kenta Terashima, Seiji Kajihara, Xiaqing Wen and Sudhakar M. Reddy "On Improving Defect Coverage of Stuck-at Fault Tests" Proceedings of the 14th Asian Test Symposium (ATS '05) PP. 216~223.
- [4] Y.Takamatsu, T.Shiosaka, T.Yamada, and, K.Yamazaki, "A Fault Model and Test Generation for Bridging Faults in CMOS Circuit," IEICE Trans. Vol.J81-D, No.6, pp. 872-879, Jun.1998.
- [5] Kohei Miyase and Seiji Kajihara, "XID: Don't Care Identification of Test Patterns for Combinational Circuits," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol.23, No2, pp.321-326, Feb.2004.
- [6] P.Goel, and B. C. Rosales, "Test Generation and Dynamic Compaction of Tests," Digest of Papers 1979 Test Conf., pp. 189-192, Oct. 1995.
- [7] Xiaqing Wen, Yoshiyuki Yamashita, Shohei Morishima, Seiji Kajihara, Laung-Terng Wang, Kewal K. Saluja, Kozo Kinoshita, "Low-Capture-Power Test Generation for At-Speed Scan Testing" International Test Conference, page 39.2 (10 pages), Nov. 2005.
- [8] J.Savir. Skewd-Load Transition Test: Part 1; Calculus. Proceedings of IEEE International Test Conference, pages 705-713 Oct. 1992.
- [9] J.Savir. Skewd-Load Transition Test: Part 2; Calculus. Proceedings of IEEE International Test Conference, pages 714-722 Oct. 1992.
- [10] "Cost-Effective Generation of Minimal Test Sets for Stuck-at Faultsin Combinational Logic Circuits," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, Vol. 14, No. 12, pp.1496-1504, Dec. 1995.
- [11] J. Savir and S. Patil, "Broad-side delay test," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, Vol. 13, No. 8, pp.1057-1064, 1994.
- [12] 富田 健, 細川 利典, 山崎 浩二, "故障活性化率向上のための可変 n 回テスト生成法とその品質評価に関する研究" DC2007-71, pp. 25-31, Feb. 2008.