

# ロバスト性を考慮したスタック・オープン故障 検出向きのテスト容易化設計について

三浦 幸也 樹下 行三

大阪大学工学部

スタック・オープン故障時に、ゲートは見かけ上トライステート素子になることに着目したテスト方法を提案する。提案した方法では、スタック・オープン故障の存在時にゲート出力をハイ・インピーダンス状態にするテスト・パターンを印加したあとに出力を直接アクセスし、スタック・オープン故障の有無をテストする。本方法を用いることにより1パターンでスタック・オープン故障のテストが可能となる。また、本方法の実装について述べる。更に、ICチップ内部のテスト・ポイントを再収斂ゲート出力点とし、テストのロバスト性を失うことなくテスト・ポイント数を減少させることができる。最後に、組合せ回路のベンチマーク回路に対するロバスト・テスト・パターン生成の結果を示す。

## Testable Design for Stuck-open Faults with the Robustness

Yukiya MIURA and Kozo KINOSHITA

Faculty of Engineering, Osaka University

2-1 Yamadaoka, Suita, Osaka, 565

We propose a new testable design method with the robustness. Since a faulty gate is regarded as a tri-state element, the gate output node can be set to arbitrary logic value from the outside of a circuit. In the proposed method, the testing is done by driving the gate output from the outside of the circuit under test and the method can be implemented relatively easily. To reduce the number of test points, we also consider a method for selecting internal test points without losing the property of the robustness. As a result, output nodes of reconvergent gates are used as internal test points. Experimental results of the pattern generation for some benchmark circuits are given.

## 1. はじめに

近年、論理回路の機能テストにおいて、MOSトランジスタにおいて縮退故障モデルでは対応しきれない故障がかなりあることがわかり、MOSトランジスタ特有の故障モデルとしてスタック・オープン故障が考えられるようになってきた<sup>(1)・(2)</sup>。

スタック・オープン故障を検出するためには、連続した2パターンのテスト・パターンが必要であることが知られているが<sup>(3)~(6)</sup>、2パターン間でハザードが発生し、スタック・オープン故障の検出が不可能となる場合もある<sup>(7)・(8)</sup>。これに対処するため、ローバスト・テストが提案されているが<sup>(7)・(8)</sup>、そのパターン生成は困難であると考えられ、また、回路構造によっては必ずしもローバスト・パターンが得られるとは限らない。

一方、スタック・オープン故障に対するテスト容易化設計もいくつか提案されている<sup>(9)~(10)</sup>。しかし、いずれも1ゲート当り2~4個の付加トランジスタが必要であり、また、付加トランジスタの駆動能力を考慮した場合、かなりの面積オーバー・ヘッドになるものと考えられる。

集積回路が大規模化、高集積化するに従い、1ピン当りの素子数が増加し、入出力ピンからの可制御性・可観測性が低下してきているため、テストが困難になってきている。可観測性を向上させる機能テスト環境として、非接触系である電子ビーム・プロービングを用いたEBテスト<sup>(11)</sup>があるが、ICチップの最上位金属配線が露出していなければならないという観測条件が必要である。一方、任意のゲート出力を電気的に観測できるテスト技術としてクロスチェックが提案されている<sup>(12)</sup>。

これまでに提案されているスタック・オープン故障に対するテストのほとんどは、スタック・オープン故障活性時のチャージ効果を利用しているため、その周囲の影響を受け易いという欠点がある。本研究では、スタック・オープン故障の存在時に出力をハイ・インピーダンス状態にするテスト・パターンを印加したあとにゲート出力をアクセスし、スタック・オープン故障の有無をテストするローバスト性を考慮したテスト方法を提案する。また、本方法の実装についても検討する。更に、内部テスト・ポイントを減らすための一選択方法を示す。2章で用語の定義について、3章で従来のテスト方法とその問題点について述べる。本研究で提案するテスト方法とその実装については4章

で述べる。内部テスト・ポイントの選択については5章に、実験結果については6章に示す。

## 2. 諸定義

[定義1] あるテスト・パターンを印加したとき、スタック・オープン故障の影響のため、ゲート出力がハイ・インピーダンス状態になることをスタック・オープン故障が活性化されたという。また、このとき印加したテスト・パターンをテスト入力という。

[定義2] 1スタック・オープン故障を検出するために、連続した2テスト・パターン $\langle T_1, T_2 \rangle$ の印加を必要とするテストを2パターン・テストという<sup>(6)</sup>。また、この2パターンのうち、第1のパターン( $T_1$ )を初期化入力、第2のパターン( $T_2$ )をテスト入力という。

[定義3] 初期化入力によるゲート出力値を初期化情報、または、故障情報という。

[定義4] 外部入力信号線へのテスト・パターン印加時のタイミング・スキューや回路遅延などによるハザードのため、初期化情報が破壊され、スタック・オープン故障が検出できなくなることをテストが無効になるという<sup>(6)・(8)</sup>。

[定義5] タイミング・スキューや回路遅延などの存在にかかわらず、スタック・オープン故障の検出を正常に行えるテストをローバスト・テストという<sup>(7)・(8)</sup>。また、このときのテスト・パターンをローバスト・パターン、または、ローバスト・テスト・パターンという。

## 3. 従来のテスト方法について

通常、CMOSゲートは図1に示すようにnMOSトランジスタとpMOSトランジスタとで構成されている。もしゲート内のトランジスタのスタック・オープン故障により、ある入力に対し、VDD、および、GNDと出力との間に導通経路が形成されない場合には、出力はハイ・インピーダンス状態になる。よって、このゲートの出力は寄生容量 $C_L$ 内に充電された電位を保持する。従って、図2のトランジスタPBのスタック・オープン故障に対し、テスト・パターンとして $\langle T_1(A, B), T_2(A, B) \rangle = \langle (1, 1), (1, 0) \rangle$ を印加しなければならない。ここで、出力値についてみれば、 $T_2$ が印加されたとき、PBがスタック・オープン故障ならば図中の容量 $C_L$ が $T_2$ 印加直前の電位を保持することから $Out = "0"$ となる。

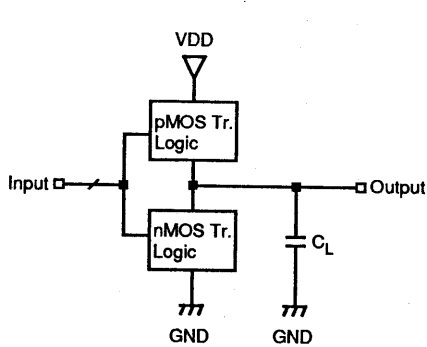


図1 CMOSゲート・ブロック図

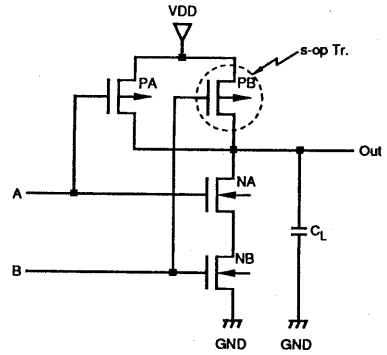


図2 2入力NANDゲート

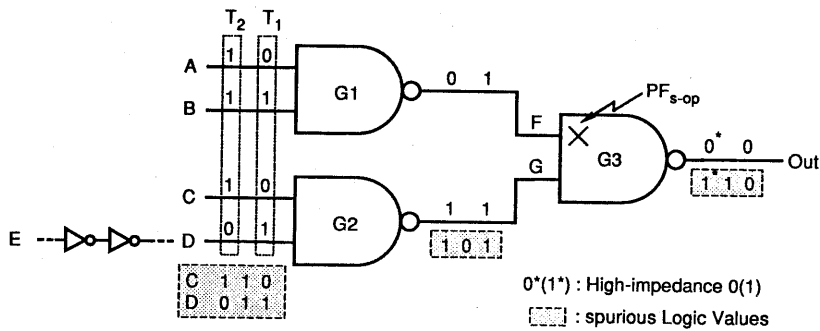


図3 回路例1

上述したように、スタック・オープン故障に対する2パターン・テストでは、スタック・オープン故障時における寄生容量内の充(放)電された電位を利用しているため、 $T_1$  から  $T_2$  への入力遷移のとき、タイミング・スキューや回路遅延のために容量に充電されていた初期化情報が破壊されテストが無効、すなわち、故障検出が不可能となる場合もあり得る。

図3の回路を例として2パターン・テストが無効となる場合について考えてみる。いま、ゲートG3内のpMOSトランジスタPFのスタック・オープン故障について考える。テスト・パターンとして  $\langle T_1(A, B, C, D), T_2(A, B, C, D) \rangle = \langle (0, 1, 0, 1), (1, 1, 1, 0) \rangle$  を選んだとする。ここで、 $T_1$  から  $T_2$  への入力遷移が行なわれたとき、タイミング・スキューや回路遅延のため入力Dが入力Cより遅れてデータ遷移したならば、ハザード  $T_h(A, B, C, D) = (1, 1, 1, 1)$  が入力パターン遷移の間に発生する。この  $T_h$  はトランジスタPGを導通させるのでG3の出力論理値

は不正に"1"になり、 $T_1$  によって、設定された初期化情報が破壊される。続いて安定な入力  $T_2$  が印加され、G3の出力をハイ・インピーダンス状態にする。このとき出力論理値はトランジスタPFが故障であるにもかかわらず"1"であり、 $T_2$  印加時の正常なゲート出力の期待値と同一である。従って、テスト  $\langle T_1, T_2 \rangle = \langle (0, 1, 0, 1), (1, 1, 1, 0) \rangle$  はトランジスタPFのスタック・オープン故障の検出に失敗する可能性がある。

このようなタイミング・スキューや回路遅延による2パターン・テストの潜在的な無効を避けるテストとして、ローバスト・テストが提案されている<sup>(7)・(8)</sup>。通常、ローバスト・テストでは2パターン・テストの  $T_1$  と  $T_2$  との間で、ハミング距離が1であるテスト・パターンが使用される。先の図3の回路例においては、 $T_1(A, B, C, D) = (0, 1, 0, 1)$  のとき  $T_2(A, B, C, D) = (1, 1, 0, 1)$  を選ぶことによりローバスト・テスト・パターンを得ることができる。

#### 4. テスト方法とその実装

##### 4.1 テスト方法

3章では2パターン・テストであってもローバスト・テストを用いることにより、タイミング・スキューや回路遅延の影響を受けずにスタック・オープン故障のテストを正常に行えると述べたが、ローバスト性を有するテスト・パターン生成は困難であり、また、回路構成によっては必ずしもローバスト・パターンを生成できるわけではない。

2パターン・テストでは、寄生容量のチャージ効果を利用してスタック・オープン故障の検出を行なっているが、ここにローバスト性に関し問題があると考えられる。すなわち、2パターン・テストではゲート出力がハイ・インピーダンス状態になったとき、出力が初期化情報を保持しているか否かという動作を利用している。よって回路動作の観点からとらえた場合、この回路は不安定な状態にあり、回路遅延などの周囲の影響を受け易いという問題がある。

スタック・オープン故障が活性化され出力がハイ・インピーダンスになったとき、このゲートは見かけ上、トライステート素子となっている。このとき、このゲート出力は外部から任意の論理値に設定可能となる。従って、あるテスト・パターンを印加したとき、ゲート出力がハイ・インピーダンス状態であるかを確認できればローバスト性を有するスタック・オープン故障のテストが可能になると考えられる。

次に、本研究で提案するテスト方法について説明する。提案した方法では、被テスト・ゲートにテスト入力を印加しておいた状態で回路外からこのゲート出力を"0"、および、"1"に直接駆動し、その結果、出力

論理値がどうなるかを観測する。図4にここで提案するテスト回路を示す。図4では説明を簡単にするため被テスト・ゲート(G. U. T)をインバータとした。ゲート出力にはテスト信号の印加ドライバであるトライステート・インバータをテスト用に付加してある。通常動作では $E = "0"$ 、テスト時には $E = "1"$ である。ゲート入力 $In$ にテスト入力を印加しておいた状態でテスト信号として"0"、および、"1"を印加する。ここで、被テスト・ゲート出力への印加値として、ゲート出力の期待値に対し、補数となる論理値"0"、または、"1"のどちらか一方のみを印加すれば良いが、各テスト・パターンに対しテスト信号を制御する複雑さを回避するため"0"と"1"とを印加することにする。 $E = "1"$ としたとき、もしこのゲートにスタック・オープン故障があり、かつ、 $Out$ がハイ・インピーダンス状態ならば $Out$ は"0"、および、"1"となる。しかし、ゲートが正常な場合、 $E = "1"$ としたとき節点Aで信号衝突が起き $Out$ は不定となる可能性がある。この影響を避けるためには、テスト信号印加ドライバの駆動能力は弱くなくてはならない。すなわち、ラッチ回路などのフィード・バック部のインバータに用いられるような $W/L$ の小さい駆動能力の弱いインバータをドライバとして使用する。こうすることで $E = "1"$ としたとき被テスト・ゲートが正常な場合には、 $Out$ を被テスト・ゲート論理で決定される論理値に設定することができる。表1に図4に示したテスト回路による出力値を示す。正常時にはテスト信号にかかわらず、ゲート入力値によって決まる論理値が出力される。スタック・オープン故障が活性化されたとき、出力は印加したテスト信号値の反転値となる。また、

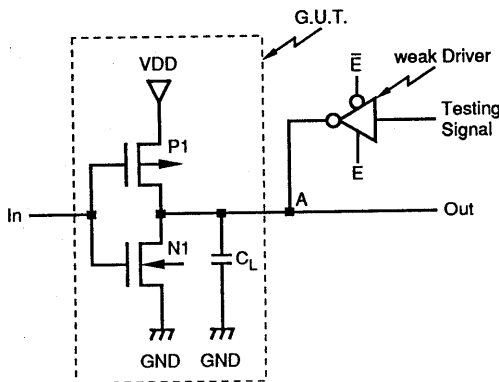


図4 テスト回路

表1 テスト回路(図4)による出力値

In	1		0	
Testing Signal	0	1	0	1
Normal	0	0	1	1
N1 : s-op	1	0	1	1
P1 : s-op	0	0	1	0
Out : s-a-0 In : s-a-1	0	0	0	0
Out : s-a-1 In : s-a-0	1	1	1	1

} Output Values

縮退故障が存在するとき、ゲート入力値、テスト信号にかかわらず、“0”か“1”かのどちらか一方の論理値が出力される。以上述べたように、本テスト方法では、スタック・オープン故障時の寄生容量によるチャージ効果を利用せずにスタック・オープン故障の検出を行なうため、1パターンでテスト可能であり、また、回路遅延などによるテストの無効がない、ローバスト性を有したテストが可能となる。

#### 4.2 実装方法

被テスト回路に図4に示したテスト回路を適用した場合、可観測性・可制御性は高くなるが、1ゲート当たりトランジスタ4個、信号線3本が必要となり面積オーバー・ヘッドが大きくなると予想される。

チップ内の任意のゲート出力を電氣的に観測可能とする技術としてクロスチェック方式が提案されている<sup>(12)</sup>。これは図5に示すように、ゲート出力にスイッチ用MOSトランジスタを1個付加し、このトランジスタをコントロールしてゲート出力を観測可能にする技術である。スイッチ用トランジスタはクロスポイント・スイッチ(CPS)と呼ばれ、そのゲート端子はコントロール信号線であるプローブ線に接続されており、また、ソース、ドレインの一方はゲート出力に、他方はゲート出力読出し信号線であるセンス線に接続されている。

ここで、本テスト方法の実装を少ないハードウェア量で実現するために、CPSをゲート出力の観測・制御用に導入することを試みる。本テスト方法をCPSを用いて実現した場合、図4のドライバはトライーステート・インバータの代わりに通常の駆動能力の弱いインバータで構成でき、また、各ゲート出力にドライバを付加する必要はなく、駆動能力を十分考慮すれば同一センス線上には1個のドライバを付加すれば十分である。クロスチェック方式を用いて本テスト方法を回路に適用した場合のテスト手順(HOCT-P)、

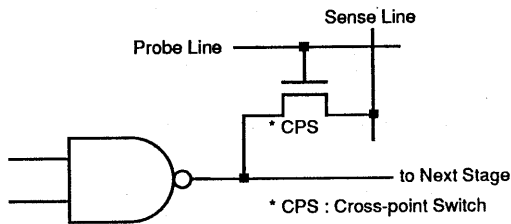


図5 クロスチェック

および、チップ全体のテスト機構をそれぞれ図6、および、図7に示す。テスト手順は、外部入力信号線にテスト入力を印加しておいた状態で(line 2)すべてのセンス線に“0”(“1”)を印加する(line 4)。次に、クロスポイント・スイッチをオンにする(line 5)。このとき故障があればセンス線上の論理値は表1に示した値になり、故障がなければ被テスト・ゲート論理で決定される論理値になる。クロスポイント・スイッチをオフにし(line 6)、すべてのセンス線の論理値を観測する(line 7)。以上の操作をすべてのテスト入力について繰返す(line 1)。なお、図7において、内部テスト・ポイント数は $m \cdot n$ である。外部入力信号線への印加テスト・パターン数を $k$ とした場合、出力パターン数(出力系列長)は $2 \cdot k \cdot m$ であり、また、1出力パターンビット幅は $n$ ビットである。本方法をそのままICチップに適用するとピン数が増加するので、チップ内で出力データを圧縮する必要がある。図7では、出力データを圧縮器で圧縮するコンパクト・テストとして表わしているが、これはLFSRなどで実現できる。

```

procedure HOCT-P
/* testing procedure of the proposed method */
begin
   $v=\{0,1\}$ ;
  (1) while given faults exist do begin
  (2)   apply a testing input to primary inputs;
  (3)   for each  $v$  do begin
  (4)     all sense lines= $v$ ;
  (5)     cross-point switches=on;
  (6)     cross-point switches=off;
  (7)     observe all sense lines;
  end
  end
end;

```

図6 テスト手順

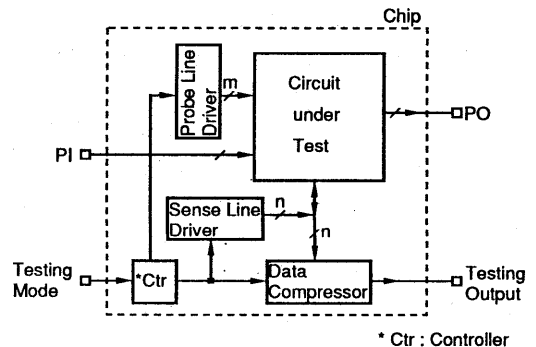


図7 本テスト方法のブロック図

## 5. テスト・ポイントの選択

回路に連続した異なる2テスト・パターンを印加したとき、タイミング・スキューや回路遅延が存在していても必ずしもハザードが発生するものとは限らない。しかし、これを検証するためには実際の被テスト回路を忠実にモデル化した正確な回路シミュレーションを実施しなければならずかなり困難である。よって、本研究では以下の仮定が成立するものとする。

〔仮定1〕わずかなタイミング・スキューや回路遅延によってでもハザードは発生する。

また、本研究では次に示すテスト・パターンをローバスト・パターンとする。

〔定義6〕外部入力信号線数 $m$ の回路内の $n$ 入力ゲート $G$ のテストについて考える。ここで、外部入力信号線を $p_j$  ( $1 \leq j \leq m$ )、ゲート $G$ の入力信号線を $i_h$  ( $1 \leq h \leq n$ )と表示する。 $\langle T_1, T_2 \rangle$ をゲート $G$ 内の入力が $i_k$  ( $1 \leq k \leq n$ )であるトランジスタ $Q$ に対する2パターン・テストであるとする。外部入力信号線にテスト・パターン $T$ を印加したときの信号線 $x$ の論理値を $T(x)$ と表示する。 $T_1(i_k)=v, T_2(i_k)=\bar{v}, v \in \{0, 1\}$ であり、かつ、 $T_1(i_g)=T_2(i_g), (i_g | 1 \leq g \leq n, g \neq k)$ で、更に、これら $T_1(i_g), T_2(i_g)$ を制御している $\{p_r | 1 \leq r \leq m\}$  ( $C(p_j)$ )から $i_g$ までの最低1本の経路 $r$ 上の各信号線では $T_1(s)=T_2(s)$ である $\langle T_1, T_2 \rangle$ をローバスト・パターンと呼ぶ。ここで、 $s$ は経路 $r$ 上の任意の信号線を表す。

定義6はハザードが発生しないための十分条件である。

〔定義7〕同一分岐点を起点とする複数の分岐枝経路を再び1本の経路にしているゲートを再収斂ゲートという。

図8に示した回路において、 $RG_x$  ( $1 \leq x \leq 3$ )は再収斂ゲートである。また、 $NRC_y$  ( $1 \leq y \leq 3$ )は非再収斂ゲートからなる非再収斂回路である。

本提案のテスト方法を用いて全ゲート出力を観測した場合、1パターンでスタック・オープン故障のテストができローバスト性を有したテストが可能である。しかし、テスト・ポイント数が多く多層配線技術を用いたとしても付加配線量が多いためレイアウト設計が困難になると予想される。よって、テスト・ポイントを適当なゲート出力に設定し、代表テスト・ポイントでテストすることが必要である。

回路内の適当なゲート出力点で信号値を観測する場合、故障による誤り信号を伝搬させるためにローバ

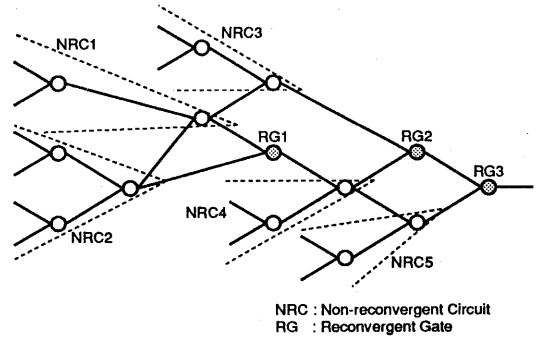


図8 回路例2

ト・2パターン・テストを行わなければならない。スタック・オープン故障のパターン生成において、それがローバスト性を有するようにするには一般に複雑かつ困難であるが、回路構造とテスト・ポイントをチップ内に任意に設定できるということを考慮した場合、必ずしもローバスト・パターン生成が困難であるというわけではない。図3に示したような樹状回路を例にとると、任意のゲートの各入力を任意の論理値に制御するとき、このゲートの他の入力に影響を与えることなく外部入力信号線への印加値を決定することができる。よって、ローバスト・パターンの生成は比較的容易である。

〔定理1〕再収斂ゲートをテスト・ポイントとすればローバスト・テストが可能である。

(証明) 非再収斂ゲートならば、その各入力の論理値はそれぞれ異なる外部入力信号線で制御できる。よって、ローバスト・パターンが存在するならば、定義6に示したローバスト・パターンは比較的容易に生成可能である。一方、再収斂ゲートでは、その入力信号線のうち少なくとも2本は最低1本の同一の外部入力信号線に接続している。従って、このゲートに対して2パターン・テストを実施したとき定義6に示したローバスト・パターンを生成できない場合がある。すなわち、再収斂ゲートのある1本の入力信号線の論理値を遷移させる $\langle T_1, T_2 \rangle$ を外部入力信号線に印加したとき、このゲートの他の入力信号線にハザードが発生する可能性がある。この場合、再収斂ゲートの出力を本テスト方法でテストすればローバスト性のあるテストが可能となる。よって、再収斂ゲートの出力をテスト・ポイントとすれば、被テスト回路はローバスト・テストが可能である。 (証明終)

以上のことから回路構造を考慮した場合、再収斂ゲートにはローバスト・パターンが存在しない場合もあり、内部テスト・ポイントとして再収斂ゲートの出力を選択することがローバスト・テストを可能にする一方法である。

## 6. 実験結果

テスト・パターン生成法と実験結果について述べる。パターン生成の前処理として、各種ゲートに対応するトランジスタ回路に変換し、次に、変換された回路の入力での2パターンを求める。これらのパターンに基づいて被テスト回路に対する2パターン $\langle T_1, T_2 \rangle$ を生成する。よって、基本的にはゲート・レベルでのパターン生成が可能である。以上の前処理を終了したのち、伝搬すべき故障（非再収斂ゲート内の故障）に対しては、再収斂ゲート、または、外部出力信号線まで伝搬させる2パターンの生成を、再収斂ゲート内の故障に対しては、テスト入力1パターンのみの生成をそれぞれ行なう。2パターン・テストでは、初期化入力は被テスト・ゲートの出力のみの初期化を行なえば良いのに対し、テスト入力は故障を活性化し、かつ、誤り信号をテスト・ポイントまで伝搬しなければならない。本研究の2パターン生成法では、初めにテスト入力を生成し、次にこれに見合う初期化入力を生成する。概念的に、この方法の方がローバスト・パターンを容易に生成できるものと考えて用いている。テスト入力生成時には、ハザード発生をチェックを行なう。図8の回路において、NRCy ( $1 \leq y \leq 5$ )内の各故障

に対しては、2パターン・テストを行なう。故障による誤り信号が最低1個のRGx ( $1 \leq x \leq 3$ )の出力に伝搬すれば、故障は検出可能である。RGxのテストに対しては、テスト入力の1パターンのみを生成する。なお、各テスト・パターンは、PODEMに基づいたアルゴリズムを用いて生成している。

表2に組合せ回路のベンチマーク回路<sup>(13)</sup>について、非再収斂ゲートのスタック・オープン故障を再収斂ゲート出力まで伝搬させた場合のローバスト・パターン生成結果を示す。ゲート内のシリーズに接続されたトランジスタ、例えば図2のNAとNB、の各故障は、同一のテスト・パターンで検出可能であるので、これらの故障を等価として扱う。AND (OR) ゲートは、NAND (NOR) ゲート+NOTゲートで構成されているものと仮定する。よって、このようなNOTゲート内の故障は、その前段のNAND (NOR) ゲートに対するテストによって検出できるので、これらのNOTゲート内の故障はNAND (NOR) ゲート内のそれと等価であると考え、これらの考え方に基づいた代表故障数を表2に示している。テスト不能故障数はテスト・パターンの存在しない故障数を表わしている。ここで、()内の数は、バックトラック10回でパターン生成を打切った故障数であり、内数である。故障検出率 $\rho$ は次式で示している。

$$\rho = \left(1 - \frac{\text{テスト不能故障数}}{\text{故障数}}\right) \times 100 \quad (1)$$

すべてのテストはハザードが発生していないのでローバスト・テストとなっている。

表2 実験結果

回路名	ゲート数	再収斂ゲート数	代表故障数	テスト不能故障数*	テストパターン数	故障検出率 $\rho$ [%]	ハザード
adder	45	10	121	0(0)	126	100.0	なし
C432	160	92	478	1(1)	328	99.8	なし
C499	202	82	506	0(0)	454	100.0	なし
C880	383	133	1112	0(0)	1304	100.0	なし
C1355	546	394	1610	8(0)	855	99.5	なし
C1908	876	250	2370	7(0)	1404	99.7	なし
C3540	1669	600	4608	170(4)	3078	96.3	なし

\*: ()内はバックトラック10回でパターン生成を打切った故障数であり、内数である。

## 7. むすび

本研究ではスタック・オープン故障が活性化されたとき、ゲートが見かけ上トライーステート素子になることを利用し、外部からゲート出力をアクセスし故障を検出するローバスト性を有するテスト方法の提案を行なった。本方法は、スタック・オープン故障のみならず、縮退故障やスタック・オン故障にも適用でき、また、順序回路のテストにも拡張できる。更に、チップ内部のテスト・ポイントを再収斂ゲート出力点とし、テストのローバスト性を失うことなくテスト・ポイント数を減少させることができた。今後はチップ内部のテスト用周辺回路の設計、耐故障性、および、一層テスト・ポイント数を減少させるための代表テスト・ポイントの決定方法、テスト・パターン数の削減などについて検討する必要がある。

## 文献

- (1) R.L. Wadsak : "Fault modeling and logic simulation of CMOS and MOS integrated circuits", Bell System Technical Journal, Vol. 57, pp.1499-1474, (May-June 1978).
- (2) J. Galiay, Y. Crouzet and M. Vergniault : "Physical versus logic fault models MOS LSI circuits : impact on their testability", IEEE Trans. Comput., Vol. C-29, No. 6, pp.527-531 (Jun. 1980).
- (3) H.K. Lee, D.S. Ha and K. Kim : "Test pattern generation for stuck-open faults using stuck-at test sets in CMOS combinational circuits", Proc. 26th Design Automation Conf., pp.345-350 (Jun. 1989).
- (4) Y.M. El-zig : "Automatic test generation for stuck-open faults in CMOS VLSI", Proc. 18th Design Automation Conf., pp.347-354 (Jun. 1981).
- (5) R. Chandranouli : "On testing stuck-open faults", Proc. 13th Int. Symp. Fault-Tolerant Comput., pp.258-265 (Jun. 1983).
- (6) S.K. Jain and V.D. Agrawal : "Test generation for MOS circuits using D-algorithm", Proc. 20th Design Automation Conf., pp.64-70 (June 1983).
- (7) S.M. Reddy, M.K. Reddy and V.D. Agrawal : "Robust tests for stuck-open faults in combinational circuits", Proc. 14th Int. Symp. Fault-Tolerant Comput., pp.44-49 (Jun. 1984).
- (8) S.M. Reddy, M.K. Reddy and J.G. Kuhl : "On testable design for CMOS logic circuits", Proc. 1983 Int. Test Conf., pp.435-445 (Oct. 1983).
- (9) D.L. Liu and E.J. McCluskey : "Designing CMOS circuits for switch-level testability", IEEE Design and Test, Vol. 4, No. 4, pp.42-49 (Aug. 1987).
- (10) R. Rajsuman, A.P. Jayasumana and Y.K. Malaiya : "CMOS stuck-open fault detection using single test patterns", Proc. 26th Design Automation Conf., pp.714-717 (Jun. 1989).
- (11) A. Tamama and N. Kuji : "Integrating an electron-beam system into VLSI fault diagnosis", IEEE Design and Test, Vol. 3, No. 4, pp.23-29 (Aug. 1986).
- (12) T. Gheewa : "CrossCheck : A cell based VLSI testability solution", Proc. 26th Design Automation Conf., pp.706-709 (Jun. 1989).
- (13) F. Brglez and H. Fujiwara : "A neutral netlist of 10 Combinational Benchmark Circuits and a target translator in FORTRAN", Int. Symp. Circuit and System, Special session on ATPG and fault simulator (Jun. 1985).