

## 内部状態を利用した順序回路のテスト生成手法

尾野年信 吉田正昭  
日本電気(株) C&C システム研究所

あらまし 順序回路を対象としたテストパターン生成手法を提案した。提案手法は、回路の内部状態を最大限に利用することを特徴とし、信号値の変化が最小となるようにテストパターンの生成を行なう。提案手法を用いることにより従来手法と比べ、テストパターン生成に要する時間とテストパターン長を短縮でき、生成したテストパターンによるタイミングの問題の発生も減少させることが可能となった。また、内部状態をより有効に利用するための、故障信号の伝搬した距離による対象故障の選択アルゴリズムを提案し、実験によりその有効性を確認した。さらに、大規模な回路を扱うための局所目標の設定による処理の分割、信号値変化に着目した場合の再収斂ファンアウトの処理などに関しても検討を加えた。

### Test Generation Method for Sequential Circuits Based on Maximum Utilization of Internal States

Toshinobu ONO Masaaki YOSHIDA  
C&C Systems Research Laboratories  
NEC Corporation  
4-1-1, Miyazaki, Miyamae-ku,  
Kawasaki 213, JAPAN

Abstract In this paper, we present a novel deterministic test pattern generation method for sequential circuits. Our approach is characterized as maximum utilization of internal states and has several advantages over conventional methods. Making the best use of internal states of the circuit allows us to reduce the computational time, the test pattern length and the timing problems, which are serious especially in sequential circuit test pattern generation. Our method selects a fault as a target fault so that the internal state is suitable for detecting it. Only the value transitions necessary to detect a fault are traced during the pattern generation procedure. A target fault selecting algorithm is also proposed in order to utilize the internal state maximally in detecting the target fault. The fault selecting algorithm is shown to be efficient with experimental results.

## 1 はじめに

近年の半導体技術の発達に伴う LSI の大規模化、複雑化により、LSI のテストは非常に深刻な問題となっている。この問題に対処するための技術として、スキャンパスなどのテスト容易化設計があるが、スキャンパスが適用できるのは同期式回路に限られ、十分な面積効率や性能が得られない場合も多い。このようなことから、大規模な順序回路に対するテストパターンの自動生成技術が要望されている。

順序回路のテストパターン自動生成は、現在においてまだ実用化のレベルには達しておらず、人手による生成に頼らざるを得ない状況にある。組合せ回路のテストパターン生成 [2, 3, 4] と比較すると、順序回路のテストパターン生成は以下の点で困難である。

- 回路の内部状態を観測すること、及び、要求される値に設定することが困難である。
- 1 個の故障の検出のためにも長いテストシーケンスが必要となる場合がある。このような場合は、テストパターン生成の処理の中で多くの時間枠における回路の内部状態として、保持しなければならないデータ量が膨大となる。
- スパイクやハザードなどのタイミングの問題を含むようなシーケンスは、テストパターンとして用いることができない。

これまでに順序回路のテストパターン生成に対していくつかのアプローチが提案されている。これらの多くは、D アルゴリズムに代表される組合せ回路のための手法を拡張したものであり、順序回路を組合せ回路の繰り返しに変換した後、組合せ回路のために開発された手法を適用するというものである [5, 6, 7]。しかし、これらの手法は、同期式回路にしか適用できず、FF の出力が他の FF のクロック入力となっているような回路も扱うことができない。また、変換後の回路の繰り返し回数、つまり故障を検出するために必要となる時間枠の数を予測するのも困難である。さらに、対象とする回路の規模が大きく故障検出に長いシーケンスが必要となる場合には、テストパターン生成の処理は非常に複雑で、莫大な時間とメモリを消費するものとなる。

従来手法の欠点を克服するため、順序回路を対象とした新しい決定性テストパターン生成手法を提案する。本手法は、回路の内部状態を最大限に利用するという考え方に基づいている。

## 2 提案手法

### 2.1 基本方針

組合せ回路と順序回路の最も大きな違いは、順序回路は内部状態を持つということである。順序回路では、ある時刻において各信号線は何らかの値を持っており、これらの値が内部状態を形成している。これまでに発表された方法の多くは、内部状態が未知の状態から各故障のテストパターンの生成を行っていた [5, 6, 7, 8, 9]。つまり、各故障毎に独立にテストパターン生成を行っていた。その結果、各故障について状態初期化のシーケンスを含むテストパターンが生成されることになる。順序回路を組合せ回路に変換し拡張 D アルゴリズムを適用する手法もこれに含まれる。

本稿では、各故障に対し連続的にテストパターン生成を行なう手法を提案する。つまり、ある故障に対するテストパターン生成が終了すると、その時の回路の内部状態から次の故障のテストパターン生成を行なうのである。これを可能にする提案手法は、内部状態を最大限に利用した手法であり、以下の 3 つの方針に基づいている。

1. 信号値の変化だけに着目する。
2. 信号値の変化を最小化する。
3. 内部状態により対象故障を選択する。

回路内部の信号線に設定されている値を最大限に利用することにより、テストパターン生成に要する時間とテストパターン長をともに短くすることが可能である。テストパターン生成において、ある信号線に設定されている値が故障の検出に都合が良い値であれば、その信号線の値を回路の外部入力から設定する必要はない。その状態で信号線に設定されている値が故障検出に使えない場合に限り、その値を変えるための入力パターンを生成しなければならない。いくつかの信号線に関しては、値を設定するために何の処理も行なわれないため、パターン長とパターン生成時間をともに短くすることができる。

テストパターン生成の各処理において、信号線の値の変化が最小となるようにテストパターンを生成する。テストパターンの探索において複数の選択肢がある場合には、信号値の変化が少なくなる順に探索を行う。複数の候補について信号値変化の数が同数の場合は、可観測性や可制御性などを選択の基準として用いる。

さらに、提案手法の効果は内部状態に大きく依存するため、回路の内部状態により対象故障を選ぶ。回路の状態がある故障を検出するのに都合が良い値であったならば、その故障に対するテストパターン生成は成功する確率が高い。そこで、与えられた内部状態の下での、対象故障の選択アルゴリズム

を提案する。この選択アルゴリズムに関しては、次節で詳細に述べる。

## 2.2 効果

回路の内部状態を最大限に利用することで、テストパターン生成において以下の点で有利となる。

### パターン生成時間

回路全体のうち信号線の値を変えなければならない一部分だけを処理すればよく、他の部分は処理が不要となるため、全体の処理量が削減される。従って、テストパターン生成に要する計算時間は減少する。

図1に例を示す。図1に示す部分回路の出力線の値を1にしなければならない場合を考える。従来手法では、その時刻における信号値の設定しか考慮していないため、バックトレースの処理は図1(a)に示す5本の経路にそって行なわれる。しかし、既に回路に図1(b)に示す初期値が既に設定されていたとすると、既に設定されている値が回路の出力値を1にするために都合が悪い信号線だけに着目すればよい。つまり、既設定値からの変化に着目すると、この場合、バックトレースの処理は3本の経路で行うだけでよい。他の初期値が設定されている場合でも、バックトレースの経路はほとんどの場合で従来手法より減少する。最初に与えられた値が全く利用できないという最悪の場合において、本手法におけるバックトレースの処理量が従来手法と同じとなる。

### テストパターン長

回路の内部状態が未知の状態からテストパターンを生成するという事は、生成されたシーケンスに内部状態を既知にするための初期化シーケンスが含まれるということである。ある故障に対するテストパターンが生成され、対象故障が切り替えられるたびに、初期化シーケンスがテストパターンに加えられることになる。しかし、提案手法では、ある故障に対するテストパターンが生成された後、その時の新しいの内部状態から次の故障に対するテストパターンを生成する。この新しい内部状態は、前の対象故障に対するテストシーケンスを回路に印加した後で得られる状態である。つまり、回路の内部状態はテストパターン生成処理の最初にただ1度初期化されるだけであり、2番目以降の対象故障に対するパターンには回路の初期化シーケンスが含まれない。従来手法における回路の初期化シーケンスの長さは、本手法において回路中のすべての信号線の値を変更するために必要なシーケンスの長さと同じであるから、ほとんどの場合、初期化シーケンスを含まないテストパターンの方が初期化シーケンスを含む

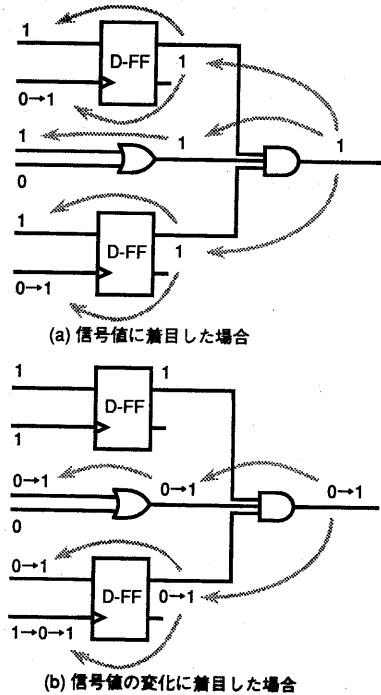


図1: バックトレースの経路

ものより短くなる。

このように、与えられた信号値をできるだけそのまま利用することにより、各故障に対するテストパターン長を短くすることができる。すべての信号線の値が未知の状態から始めるということは、それまでに生成したパターンで決まる回路の内部状態に関する情報を全く利用していないということである。

### タイミングの問題

順序回路のテストパターンは、タイミングに関する問題を持たないということが重要である。生成されたテストパターンで、スパイクやハザードなどのタイミングの問題が生じてはならない。しかし、テストパターン生成の段階でタイミングの問題を考慮することは非常に難しい。この問題を回避するため、前後する入力パターンで値の変化を1ビットに制限するという手法がいくつか提案されている [10, 11, 12]。しかし、この制限があるためにテストパターンの生成ができない場合もあり得る。

一方、本手法では信号値の変化に着目しその数を最小化しているだけであるため、必要ならば同時に複数の入力線の変化も許す。しかし、信号値変化の数が最小限に押えられるた

め、タイミングを考慮していない他の手法と比較すると、本手法で生成したテストパターンでは、タイミングの問題の発生する確率は非常に低くなる。つまり、入力の変化に制限を加えることなくタイミングの問題を緩和することができる。

### 3 対象故障の選択

#### 3.1 伝搬距離

先に述べたように、各故障に対し連続的にテストパターン生成を行なう本手法においては、対象故障の選択が非常に重要である。次の対象故障は、それまでに生成されたパターンによって決定される回路の内部状態から決定される。

故障シミュレーションの結果として、各未検出故障の故障信号 ( $D, \bar{D}$ ) の位置に関する情報が利用できる。ここで、ある故障の故障信号のうち最も可観測性の良いものを、先頭故障信号と定義する。すなわち、先頭故障信号は、故障信号が設定されている中で最も回路の外部出力に近い信号線である。

可観測性の最もよい先頭故障信号を持つ故障を対象故障として選んだ場合、その故障に対するテストパターン生成は成功する確率が高い。しかし、他の故障は対象故障より検出が難しいため、その時に生成されたパターンで検出される他の故障は少ないと考えられる。一方、最も可観測性の悪い故障を対象故障に選んだ場合には、故障の検出に成功する確率が低くなる。しかし、一度検出に成功すると、生成されたパターンで同時に他の多くの故障を検出できる可能性がある。

従って、ここでは、故障の位置と先頭故障信号の位置の両方を考慮に入れた選択基準を提案する。対象故障を、故障と先頭故障信号の距離によって選択する。2本の信号線の距離は、両線の可観測性の差として定義する。さらに、故障の伝搬距離は、故障線とその先頭故障信号の距離として定義される。伝搬距離が大きいということは、故障線で発生した故障信号が回路の外部出力線により近くまで伝搬してしているということである。

故障  $f$  の伝搬距離  $P(f)$  は、

$$P(f) = B(l_f) - B(l_{s_f})$$

と定義される。ここで、 $l_f$  と  $l_{s_f}$  はそれぞれ故障線と先頭故障信号線、 $B(l)$  は信号線  $l$  の可観測性を表すものとする。故障について故障信号が発生していない場合の  $B(l_{s_f})$  は十分大きな値と定義される。

伝搬距離が最大の故障信号を持つ故障が次の対象故障として選択される。ある故障の故障信号が遠くまで伝わっているなら、内部状態はその故障を検出するのに都合が良いと考えられる。本手法は、与えられたどの内部状態からでもテスト

パターンを生成できるため、故障を検出するのに都合の良い内部状態を有効に利用することができる。

#### 3.2 可観測性

本手法では、対象故障を選択する際の伝搬距離の計算に可観測性が用いられる。現在は、可観測性の計算に単純な方法を用いている。

先に述べたように、信号線  $l$  の可観測性を  $B(l)$  と表す。信号線  $l$  の可観測性は、外部入力線から  $l$  までのゲートの段数として定義される。つまり、 $B(l)$  は、

$$B(l) = \begin{cases} 0 & l \text{ が外部出力の場合} \\ W(g_l) + \min\{B(l_1), \dots, B(l_n)\} & \text{その他の場合} \end{cases}$$

と定義される。ここで、 $g_l$  は  $l$  を入力線としているゲート、 $l_1, \dots, l_n$  は  $g_l$  の出力線である。FF などの記憶素子及びファンアウトもゲートの1種として扱われる。また、 $W(g)$  はゲート  $g$  の重みであり、次のように定義される。

$$W(g) = \begin{cases} 0 & \text{ファンアウト} \\ 1 & \text{組合せゲート} \\ w & \text{記憶素子} \end{cases}$$

ここで、 $w$  は組合せゲートと比較して記憶素子を故障信号が通過する困難さを表す定数である。

可制御性も可観測性と同様に、現在は外部入力線からのゲートの段数として定義している。可観測性と可制御性の定義はともに今後さらに複雑で正確なものに改良する予定である。

#### 3.3 例

図2に伝搬距離による対象故障選択の例を示す。図2(a)の数字は  $w = 5$  とした場合の各信号線の可観測性を示している。図2(b)と(c)は、それぞれ故障  $f_1$  と  $f_2$  が存在する場合の各信号線の値を表している。故障信号は太線で表されている。 $f_1$  の場合には、 $B(l_{f_1}) = 12$  で  $B(l_{s_{f_1}}) = 6$  であるから、 $P(f_1) = 6$  である。同様に、 $P(f_2) = 1$  と計算される。未検出故障が  $f_1$  と  $f_2$  だけであるとすると、伝搬距離の大きい  $f_1$  が次の対象故障に選ばれる。

$f_1$  の故障信号は D-FF を越えて伝搬しているが、 $f_2$  の故障信号は越えていない。従って、この内部状態は  $f_2$  よりも  $f_1$  を検出するのに都合が良いと考えられる。2つの故障の伝搬距離の差がこの状況を忠実に表現している。

### 4 順序回路のテストパターン生成のための工夫

ここでは、順序回路のテストパターン生成特有の問題点と、それらに対処するための工夫について述べる。

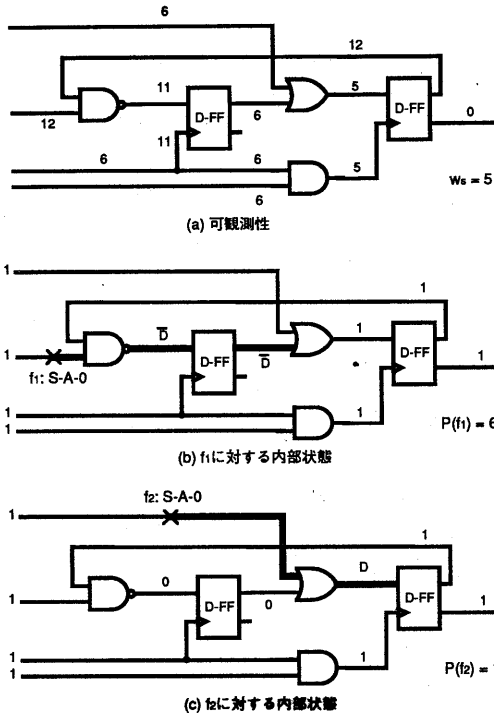


図 2: 可観測性と伝搬距離

#### 4.1 局所目標の設定

ある故障に対するテストパターン生成の処理は、故障を活性化し故障信号を外部出力まで伝搬させる入力シーケンスを求めることである。大規模な回路に対してはテストシーケンスの長さは非常に長大なものとなることがある。そのような場合、多くの時間枠にわたる回路の状態を保持するためには膨大な記憶容量が必要となる。そこで、保持するデータの量を減らすためにテストパターン生成の処理をいくつかの処理に分割した方がよい。

ここでは、全体の処理を分割するために局所的な目標を導入する。テストパターン生成の各段階で局所目標が設定され、それを達成するためのパターンが生成される。局所目標の導入により大規模な順序回路が扱えるようになる。

局所目標としては以下のものがある。

- (a) 故障信号を故障線に発生させる。
- (b) 故障信号を FF の入力まで伝える。
- (c) 故障信号を FF の入力から出力に伝える。

記憶素子として D-FF しか含まない同期式回路では、各時刻にすべての FF にクロック信号が供給され FF の入力の故障信号は自動的に出力まで伝わるため、局所目標 (c) は必要

ない。外部出力線は FF の入力と同じと見なされる。故障信号が外部出力に到達するとその故障に対するテストパターン生成は終了する。

#### 4.2 FF の表現

順序回路のテストパターン生成においては、FF などの記憶素子をどのように扱うかが問題となる。記憶素子は、AND や NOR などの組合せゲートと同じように扱うことはできない。しかし、連続する 2 時刻を考え、前の時刻の状態出力を入力と見ると、記憶素子の動作は組合せゲートと同様にキューブ表現で表すことができる。

記憶素子の例として D-FF のキューブ表現を表 1 に示す。クロック入力  $C$  も特別な扱いはされず、データ入力  $D$  と同じく扱われる。表 1 のキューブ表現は、時刻  $t$  における  $D, C, Q, \bar{Q}$  と時刻  $t+1$  における  $D, C$  を入力とし、時刻  $t+1$  における  $Q, \bar{Q}$  を出力とする関数を表している。

記憶素子をキューブ表現で表すことにより、クロック線の故障の扱いも可能となる。順序回路を組合せ回路に変換してから扱う従来手法では、クロック線の故障を検出するテストパターンを生成することはできない。

$t$				$t+1$			
$D$	$C$	$Q$	$\bar{Q}$	$D$	$C$	$Q$	$\bar{Q}$
-	-	0	1	-	0	0	1
-	-	1	0	-	0	1	0
-	1	0	1	-	-	0	1
-	1	1	0	-	-	1	0
0	0	-	-	-	1	0	1
1	0	-	-	-	1	1	0

(-: don't care)

表 1: D-FF のキューブ表現

### 5 テストパターン生成処理

#### 5.1 全体の処理フロー

テストパターン生成処理の全体フローを図 3 に示す。まず、回路の全信号線の可制御性と可観測性を計算する。次に、回路の内部状態を既知とするための初期化シーケンスが生成される。

その後、ループに入り、それまでに生成されたテストパターンにより未検出故障の故障シミュレーションが実行される。故障シミュレーションによりそれまでに検出される故障と検

出されない故障が判定され、未検出故障のリストが作成される。すべての故障が検出されたらテストパターン生成は終了する。未検出故障が残っている場合には、そのうちの1つが次の対象故障として選択される。対象故障の選択は、前述の伝搬距離に従って行なわれる。

故障が選択されると、それまでに生成されたパターンによって形成される内部状態から、その故障を検出するテストパターンが生成される。対象故障の故障信号が回路中に現れていない場合には、まず故障を活性化するパターンが生成される。故障の活性化に成功し故障信号が発生すると、故障信号を伝搬させるためのシーケンスが生成される。この伝搬シーケンスは、故障信号のうちの1つを外部出力まで伝える。故障の活性化が故障信号の伝搬に失敗した場合は、対象故障が他の未検出故障に切り替えられ、その故障に関して同様の処理が繰り返される。

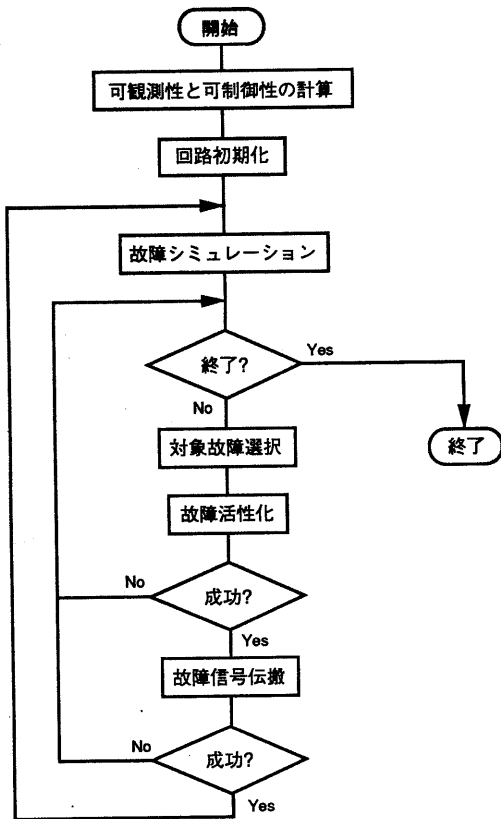


図 3: 全体の処理フロー

## 5.2 バックトレース

局所目標が設定されると、その目標を達成するための入力パターンが作成される。この処理は、各ゲートの出力線の値をある要求値にするための入力線の値を決めるという作業の繰り返しである。ある信号線のある値に設定しなければならない場合、これを目標と呼ぶことにする。目標は、目標線と目標値のペアで表される。バックトレースではこのような目標を多数処理しなければならない。バックトレースの処理は以下のように書くことができる。

- (1) 局所目標を達成するための目標を求める。
- (2) 目標を正当化リストに加える。
- (3) 正当化リストが空ならば、終了する。
- (4) 正当化リストから目標を1つ取り出す。
- (5) 目標線が既に目標値になっていれば、値を固定して(3)へ行く。
- (6) 目標線を目標値にし前方へ含意操作を行なう。
- (7) 目標線を目標値にするための新たな目標を求める。
- (8) 新たな目標を正当化リストに加える。
- (9) (3)へ行く。

各処理において矛盾が生じたならば、バックトラック処理が行なわれ別の選択肢が調べられる。順序回路の場合には、新たな目標は2時刻以上にわたるものとなる場合がある。

## 5.3 再収斂ファンアウト

回路に再収斂するファンアウトがある場合には、バックトレースの処理がもっと複雑になる。そのようなバックトレースの例を図4に示す。図4の回路の各信号線が矢印の左側の値にあらかじめ設定されている状態で、回路の出力を0に変化させることを考える。

まず、ゲートAの出力を1から0に変化させる。ゲートAの入力値は1と0であるから、0の入力だけを1に変えればよく、他の入力に変化させる必要はなく1に固定される。次に、ゲートCが処理され初期値として0が設定されていた入力が1に変化させられる。ゲートCの入力値の変化で前方に含意操作が行なわれ、その変化の影響が調べられる。図4(a)の場合には、その影響はどこへも伝搬しない。図4(b)の場合には、影響はゲートBを越えてその出力に伝わるが、その出力値は前に固定されている。従って、ゲートAの出力の変化を押えるために、Bの出力を1にするためのバックトレース処理が再び起動される。

このように、前方への含意操作で信号値の変化の影響が調べられ、前に値を固定した信号線に対して再び値を設定する

処理が起動される。信号値の変化が前に固定した値に影響を及ぼさない場合には、値の変化の必要の無い信号線については一切処理が行なわれない。

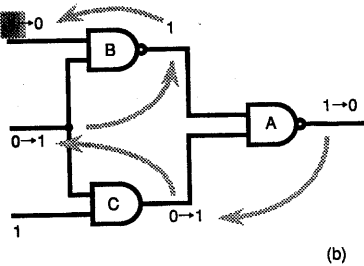
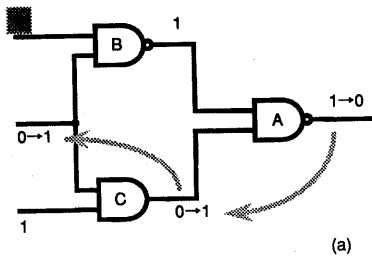


図 4: 再収斂ファンアウトの処理

## 6 実験結果

提案手法を Sun-3 ワークステーション上に C 言語でインプリメントした。伝搬距離による対象故障の選択アルゴリズムの効果について評価を行うため、いくつかの順序回路についてテストパターンの生成を行ない、ランダムな対象故障の選択と比較した。結果を表 2 に示す。時間は全対象故障のテストパターン生成時間の合計であり、故障シミュレーションに要した時間は含まれていない。

表 2 からわかるように、ほとんどの回路でランダムな選択に比べ、伝搬距離による選択により実行時間とパターン数がともに減少している。ランダムな選択の方が実行時間が短くなる回路もあるが、そのような回路でもパターン数は少なくなっている。表 2 により伝搬距離による故障選択の有効性がわかる。

現在用いている可観測性の計算方法は非常に単純なものであり、さらに正確な可観測性を用いることによりさらに良い結果が期待できる。詳細な評価には、もっと多くの大規模な回路で評価する必要がある。回路が大規模になるほど故障毎の伝搬距離の差が大きくなり、本手法の効果がより期待できる。

回路	ゲート数	FF数	伝搬距離による選択		ランダムな選択	
			実行時間(秒)	パターン数	実行時間(秒)	パターン数
A	10	2	4.5	17	3.4	17
B	19	3	4.4	28	8.0	34
C	20	4	32.5	17	33.7	17
D	26	6	2.4	42	1.9	47
E	28	6	15.4	45	28.9	48
平均			11.8	29.8	15.1	32.6

表 2: 実験結果

## 7 まとめ

回路の内部状態を最大限に利用する新しい順序回路のテストパターン生成手法について述べた。本手法は従来手法と比較していくつかの利点を持つ。回路の内部状態を最大限に利用することで、実行時間とテストパターン長の短縮、及び、タイミングの問題の緩和が可能となる。また、内部状態を有効に活かすための対象故障の選択アルゴリズム、及び、記憶素子の表現法とバックトレース処理についても述べた。さらに、提案した故障選択法の効果を実験により確認した。今後は、可観測性、可制御性の改良と提案手法の詳細な評価を行なう予定である。

## 謝辞

本研究を進めるに当たり御指導、御鞭撻戴いた日本電気(株)後藤所長代理、吉村部長、藤田課長に感謝致します。

## 参考文献

- [1] 尾野, 藤田, 吉村, “信号値の変化に着目したテスト生成手法”, 情報処理学会第 39 回全国大会, 1989 年 10 月.
- [2] J.P.Roth, “Diagnosis of Automata Failures: A Calculus and a Method”, *IBM J.Res. Develop.*, Vol.10, pp.278-291, July 1966.
- [3] P.Goel, “An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits”, *IEEE Trans.Computers*, Vol.C-30, No.3, pp.215-222, March 1981.
- [4] H.Fujiwara and T.Shimono, “On the Acceleration of Test Generation Algorithms”, *IEEE Trans.Computers*, Vol.C-32, No.12, pp.1137-1144, December 1983.

- [5] H.Kubo, "A Procedure for Generating Test Sequences to Detect Sequential Circuit Failures", *NEC Research & Development*, No.12, pp.69-78, October 1968.
- [6] G.R.Putzolu and J.P.Roth, "A Heuristic Algorithm for the Testing of Asynchronous Circuits", *IEEE Trans.Computers*, Vol.C-20, No.6, pp.639-647, June 1971.
- [7] P.Muth, "A Nine-Valued Circuit Model to Generate Tests for Sequential Circuits", *Proc.International Symp.on Fault-Tolerant Computing*, pp.43-49, June 1975.
- [8] R.Marlett, "An Effective Test Generation System for Sequential Circuits", *Proc.23rd Design Automation Conf.*, pp.250-256, June-July 1986.
- [9] H.T.Ma, S.Devadas, A.R.Newton and A.Sangiovanni-Vincentelli, "Test Generation for Sequential Circuits", *IEEE Trans.Computer-Aided Design*, Vol.7, No.10, pp.1081-1093, October 1988.
- [10] T.J.Snethen, "Simulator-Oriented Fault Test Generator", *Proc.14th Design Automation Conf.*, pp.88-93, June 1977.
- [11] K.Cheng, V.D.Agrawal and E.S.Kuh, "A Sequential Circuit Test Generator Using Threshold-Value Simulation", *Proc.International Symp.on Fault-Tolerant Computing*, pp.24-29, June 1988.
- [12] V.D.Agrawal, K.Cheng and P.Agrawal, "CONTEST:A Concurrent Test Generation for Sequential Circuits", *Proc.25th Design Automation Conf.*, pp.84-89, June 1988.