

# レジスタ転送レベルデータパスの単一制御可検査性に基づく 組み込み自己テスト容易化設計法

井筒 稔<sup>†\*</sup>      和田 弘樹<sup>†</sup>      増澤 利光<sup>†</sup>      藤原 秀雄<sup>†</sup>

A DFT Method for BIST of RTL Data Paths Based on Single-Control Testability  
Minoru IZUTSU<sup>†\*</sup>, Hiroki WADA<sup>†</sup>, Toshimitsu MASUZAWA<sup>†</sup>, and Hideo FUJIWARA<sup>†</sup>

あらまし 本論文では、レジスタ転送レベルデータパスの組み込み自己テスト方式（BIST）として、階層テストに基づく test per clock 方式の BIST を提案する。この手法では、テストパターン生成器、応答解析器をテスト対象回路の外部入力、外部出力のみに付加し、各組合せ回路要素に対して、データパスの経路を用いてテストパターン、応答を伝搬する。本論文では、この階層テストに基づく BIST が可能なデータパスとして単一制御可検査データパスを定義し、与えられたデータパスを単一制御可検査データパスに設計変更するためのテスト容易化設計法を提案する。また、ベンチマーク回路を用いて、提案手法の評価を行う。

キーワード テスト容易化設計、レジスタ転送レベルデータパス、階層テスト、組み込み自己テスト

## 1. ま え が き

VLSI の大規模化、複雑化に伴い、テストパターン生成、応答解析を VLSI 上で行う組み込み自己テスト (Built-In Self-Test. 以下、BIST)[1] の重要性がますます高まっている。BIST を実現するために、テスト対象回路の外部入力、外部出力に、それぞれ、テストパターン生成器、応答解析器を付加する。しかし、テスト対象回路に閉路が含まれている場合には、外部入出力にテストパターン生成器、応答解析器を付加するだけでは高い故障検出率を得ることができない。そのため、高い故障検出率を得るために、回路内部にテストのためのハードウェアを付加する方法が数多く提案されている [2]。

BIST は、test per scan 方式と test per clock 方式に分類できる。test per scan 方式では、回路中の（一部の）レジスタをスキャンレジスタに変更し、スキャン操作により、テストパターン生成器で生成したテストパターンをスキャンレジスタにシフトし、スキャンレジスタに格納された応答を応答解析器にシフトする。

test per scan 方式では、スキャン操作によりテストパターンをシフトインするので、連続したシステムクロックでテストパターンを印加できず、テスト実行時間も長い。

一方、test per clock 方式では、回路中の（一部の）レジスタをテストパターン生成器、応答解析器に変更する。このようなテストレジスタとしては、BILBO (Built-In Logic Block Observer)[3]、CBILBO (Concurrent BILBO) が用いられる。test per clock 方式では、連続クロックでテストパターンの生成/印加、応答の解析が可能であり、実動作速度でのテスト (at-speed test) が可能である。このため、テスト実行時間が短く、更に、テストパターンの連続印加を必要とする遅延故障などのテストにも適用可能である。しかし、一般に、test per scan 方式に比べ、ハードウェアオーバーヘッドが大きくなる。

test per clock 方式の BIST として、文献 [4] は、回路中のすべての閉路が少なくとも二つの BILBO かつ一つの CBILBO を含むように設計変更するテスト容易化設計法を示している。

本論文では、test per clock 方式の BIST として、階層テスト [5], [6] に基づく方法を提案する。この手法では、内部レジスタを BILBO や CBILBO に変更せず、テストパターン生成器、応答解析器は、それぞれテ

<sup>†</sup> 奈良先端科学技術大学院大学情報科学研究科，生駒市  
Graduate School of Information of Science, Nara Institute of  
Science and Technology, 8916-5 Takayama-cho, Ikoma-shi,  
630-0101 Japan

\* 現在，株式会社日立製作所ソフトウェア事業部企画本部生産技術部

ト対象回路の外部入力，外部出力のみに付加する．そして，データパス中の各組合せ回路要素（演算器，マルチプレクサなど）ごとにテストを行う．つまり，テストパターンをテストパターン生成器から各組合せ回路要素まで伝搬し，応答をその組合せ回路要素から応答解析器まで伝搬する．このテストパターン，応答の伝搬は，データパス中の経路を利用する．連続クロックでテストパターンの生成/印加，応答の解析を可能にするには，これらの経路が共通部分をもたないことが必要である．もしそのような経路がデータパス中に存在しなければ，データパスに経路を付加する必要がある．また，この経路に演算モジュールが現れる場合には，この演算モジュールの入力端子から出力端子へテストパターン，応答を伝搬できるように，この演算モジュールにスルー機能を付加する必要がある．

本論文では，階層テストに基づく test per clock 方式の BIST が可能なデータパスとして，単一制御可検査データパスを定義し，与えられたデータパスが単一制御可検査になるように設計変更するテスト容易化設計法を提案する．本論文で提案する BIST の特徴は以下のとおりである．

- 高い故障検出率：テストはデータパス中の組合せ回路要素ごとに行われる．実際のデータパスで使用されるほとんどの組合せ回路要素（加算器，減算器，乗算器，シフタ，マルチプレクサなど）は，テストパターンとしてランダムパターンを用いることにより，縮退故障に対しては高い故障検出率が得られることが知られており [7]，本手法で高い故障検出率を得ることが期待できる．

- 低いハードウェアオーバーヘッド：テストパターン生成器，応答解析器を外部入出力のみに付加するので，文献 [4] の手法に比べ，ハードウェアオーバーヘッドが小さいことが期待できる．

- test per clock 方式：連続クロックで，テストパターンの生成，応答の解析が可能であり，実動作速度でのテストが可能である．

以下，2. では本論文で対象とするデータパスを定義し，データパスグラフについて述べる．3. ではデータパスの単一制御可検査性を定義する．4. では単一制御可検査性に基づくテスト容易化設計法を述べる．5. ではベンチマーク回路を用いた実験により，提案手法を評価する．

## 2. データパス

本論文で対象とするデータパス (data path) は以下の構成要素からなる．

- 回路要素
- データ信号線：回路要素を相互に接続．
- 制御信号線：制御回路から回路要素へ制御信号を伝達．
- 状態信号線：回路要素から制御回路へ状態信号を伝達．

以下に各構成要素について説明する．

- 回路要素：回路要素は外部入力，外部出力，ラッチ，レジスタ，マルチプレクサ，演算モジュール，観測モジュールに分類される．このうちマルチプレクサ，演算モジュール，観測モジュールを組合せ回路要素と呼ぶ．

回路要素に対してデータ信号線が接続される端子をデータ端子，制御信号線が接続される端子を制御端子，状態信号線が接続される端子を状態端子と呼ぶ．データ端子は回路要素に信号を入力するためのデータ入力端子と回路要素からデータを出力するためのデータ出力端子に分類される（以下，データ入力端子を入力端子，データ出力端子を出力端子と呼ぶ）．データパス上のすべての回路要素のデータ端子は等しいビット幅をもつものとする．

外部入力，外部出力：外部入力 (PI) はデータパス外部からデータパスにデータを入力するための端子，外部出力 (PO) はデータパスから外部にデータを出力するための端子である．便宜上，外部入力は出力端子のみをもち，外部出力は一つの入力端子のみをもちものとする．

マルチプレクサ：マルチプレクサは二つの入力端子と一つの出力端子，1 ビットの制御端子をもつ．制御端子の値に従って，対応する入力端子の値をそのまま出力端子に出力する．

演算モジュール：演算モジュールは一つまたは二つの入力端子，一つの出力端子，たかだか一つの制御端子，たかだか一つの状態端子をもつ．入力端子に与えられた値に対して演算を行って出力端子に出力する．

観測モジュール：観測モジュールは一つまたは二つの入力端子，たかだか一つの制御端子，一つの状態端子をもち，出力端子をもたない．観測モジュールは制御回路に伝達するための状態信号を生成することのみを目的とした比較器 (コンパレータ) 等をモデル化し

たものである。

ラッチ、レジスタ：ラッチ、レジスタはいずれも記憶素子である。ラッチは一つの入力端子と一つの出力端子をもつ。入力端子に与えられた値を記憶し、その値を次のクロックサイクルで出力端子に出力する。レジスタは一つの入力端子と一つの出力端子、1ビットの制御端子をもつ。制御端子の値によって、入力端子の値を新たに記憶するか（ロード）、既に記憶している値を保持する（ホールド）。記憶している値は次のクロックサイクルで出力端子に出力する。

- データ信号線：データ信号線（以降、単に信号線と呼ぶ）は相異なる回路要素の出力端子と入力端子を接続する。複数の信号線を同一の出力端子に接続できる（ファンアウト可能）が、入力端子に接続する信号線は1本のみとする。

- 制御信号線，状態信号線：制御信号線は制御回路からデータパス上の回路要素に対して制御信号を伝達し，状態信号線は回路要素から制御回路に対して状態信号を伝達する。本論文でも文献 [6] と同様に制御信号線には回路外部から任意の時刻に任意の値が設定可能であり，状態信号線は回路外部から任意の時刻での値が観測可能であると仮定する。

データパスに対してデータパスグラフ (data path graph)  $G = (V, A)$  を次の有向グラフとして定義する。

- $V = V_1 \cup V_2$

ここで  $V_1$  はデータパス中のすべての回路要素の集合， $V_2$  はデータパス中のすべてのデータ端子の集合とする。

- $A = A_1 \cup A_2 \cup A_3$

ここで  $A_1$  はデータ信号線に対応し， $A_1 = \{(x, y) \in V_2 \times V_2 \mid \text{出力端子 } x \text{ と入力端子 } y \text{ がデータ信号線で接続}\}$  とする。また， $A_2, A_3$  はそれぞれ，入力端子と回路要素の対応，回路要素と出力端子の対応を表す。つまり， $A_2 = \{(x, u) \in V_2 \times V_1 \mid x \text{ は } u \text{ の入力端子}\}$ ， $A_3 = \{(u, x) \in V_1 \times V_2 \mid x \text{ は } u \text{ の出力端子}\}$  とする。

図 1(a) のデータパスに対するデータパスグラフを図 1(b) に示す。経路，単純経路，サイクルなどのグラフ用語をデータパスグラフに対しても用いる。

本論文で対象とするデータパスは，そのデータパスグラフにおいてすべての入力端子は，少なくとも一つの外部入力から到達可能であり，すべての出力端子は少なくとも一つの外部出力に到達可能であるものとする。

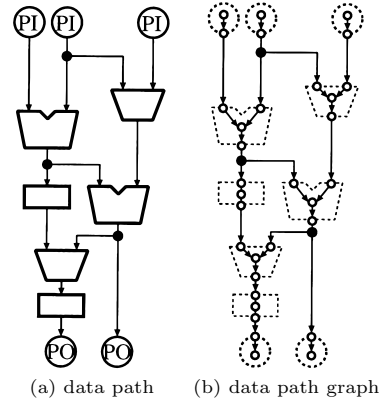


図 1 データパスとデータパスグラフ  
Fig. 1 Data path and data path graph.

### 3. 単一制御可検査性

本論文では，test per clock 方式の BIST で高い故障検出率を達成できるデータパスのテスト容易化設計法を提案する。実際のデータパスで用いられるほとんどの組合せ回路要素（加算器，減算器，乗算器，シフト，マルチプレクサなど）に対しては，ランダムパターンをテストパターンとして用いることにより，縮退故障に対しては高い故障検出率が得られる [7]。比較器については，ランダムパターンでは高い故障検出率を得るのが困難だが，制御点，観測点を付加することにより，ランダムパターンで高い故障検出率を得ることができる [7]。このことから，データパス中の各組合せ回路要素  $M$  に対して，以下の 2 条件が成り立てば， $M$  にランダムパターンを用いてテストを実行することにより，データパス全体で高い故障検出率を得られる。

(1)  $M$  の各入力端子まで入力端子ごとに異なるテストパターン生成器で生成したランダムパターンを伝搬可能。

(2)  $M$  の出力端子の値を応答解析器まで伝搬可能。

本論文のテスト容易化設計法では，ハードウェアオーバーヘッドを低く抑えるために，テストパターン生成器は外部入力のみ，応答解析器は外部出力のみに配置する。この制約上ですべての組合せ回路要素が上記の 2 条件を満たすための十分条件として，データパスの単一制御可検査性を次のように定義する。

[定義 1] (単一制御可検査データパス  $DP$ )

対応するデータパスグラフにおいて，各組合せ回路要素  $M(\in V_1)$  に対して，以下の条件を満たす互いに

共通部分をもたない単純経路  $P_1, P_2, P_3$  が存在し、各経路に対応するデータバス上の経路に沿って任意の値が伝搬可能であるとき、データバス  $DP$  は単一制御可検査であるという。

- 外部入力を始点とし、 $M$  の入力端子を終点とする経路  $P_1$  及び  $P_2$ <sup>(注1)</sup>。

ただし、 $M$  が一つの入力端子しかもたない場合、 $P_2$  を空経路とする。

- $M$  の出力端子を始点とし、外部出力を終点とする経路  $P_3$ 。

ただし、 $M$  が出力端子をもたない(観測モジュール)場合、 $P_3$  を空経路とする。 □

$P_1, P_2$  を  $M$  の制御経路、 $P_3$  を  $M$  の観測経路と呼ぶ。

単一制御可検査データバスにおいて、組合せ回路要素  $M$  に対して、制御経路を用いて外部入力から  $M$  のデータ入力端子に連続したテストパターンを印加し、観測経路を用いてデータ出力端子に現れる  $M$  の応答を連続して外部出力で観測できる。ここでデータバスに関する仮定から、 $M$  の制御端子に回路外部から直接連続したテストパターンの印加が可能であり、かつ  $M$  の状態端子の値は回路外部で直接連続して観測可能である。よって  $M$  が単一制御可検査であれば、 $M$  に対して連続したテストパターンの系列を用いたテストが実行可能となる。これらの経路上には、演算モジュールやマルチプレクサが現れるが、この経路上の入力端子の値が出力端子に伝搬するように制御する<sup>(注2)</sup>。この制御は、 $M$  のテストの間固定しておけばよく、各組合せ回路要素に対して、一つの制御パターンで十分なので、単一制御可検査性と呼んでいる。

## 4. テスト容易化設計法

### 4.1 テスト容易化設計

本節では、与えられたデータバスを単一制御可検査データバスに設計変更するためのテスト容易化設計法(Design For Testability, 以下 DFT)を示す。単一制御可検査データバスは、データバスの各組合せ回路要素  $M$  に対して、互いに共通部分をもたない制御経路と観測経路をもち、それぞれの経路に沿って任意の値を伝搬できる。

与えられたデータバスにおいて、ある組合せ回路要素  $M$  に対して、互いに共通部分をもたない制御経路、観測経路が存在しない場合、単一制御可検査にするためには、データバスに新たな経路を付加しなければ

ならない。提案する DFT では、この経路付加は、マルチプレクサを用いて実現する。また、制御経路、観測経路に  $M$  以外の演算モジュールが現れる場合、任意の値を伝搬可能とするために、この演算モジュールにスルー機能を付加する。そこで、単一制御可検査のための DFT を、次の最適化問題として定式化する。

[定義 2] (単一制御可検査 DFT)

単一制御可検査のための DFT を次の最適化問題として定義する。

- ・入力：データバス
- ・出力：単一制御可検査データバス
- ・最適化目標：付加する DFT 要素(マルチプレクサ、スルー機能)のハードウェア量最小化 □

### 4.2 テスト容易化設計法

単一制御可検査 DFT のための発見的アルゴリズムを示す。本アルゴリズムは、以下の 2 段階からなる。

1. 制御経路の決定と DFT 要素付加
2. 観測経路の決定と DFT 要素付加

組合せ回路要素  $M$  に対して、互いに共通部分をもたない制御経路、観測経路が存在するかどうかを判定する問題は、本質的に 2 品種フロー問題である。2 品種フロー問題は NP 完全であるため [8]、本アルゴリズムでは、制御経路と観測経路に分離して決定する。

#### 4.2.1 制御経路の決定と DFT 要素付加

各組合せ回路要素に対し、最小の付加ハードウェアで実現できる制御経路(2 入力組合せ回路要素の場合は、互いに共通部分をもたない二つの制御経路)を決定し、DFT 要素(マルチプレクサ、スルー機能)を付加する。制御経路を求める組合せ回路要素は、一つずつ処理していくが、先の処理で付加した DFT 要素は後の処理でも利用できるため、組合せ回路要素を処理する順序によって、全体のハードウェアオーバーヘッドは異なる。全体のハードウェアオーバーヘッドを低く抑えるには、なるべく必要性の高い DFT 要素から付加していくことが望ましい。そこで、前処理として、必ず付加が必要な DFT 要素を付加する。次に、外部入力に近い組合せ回路要素から順に制御経路を決定する。これは、外部入力に近い組合せ回路要素ほど、制御経路の選択肢が少なく、そこで付加する DFT 要素は必要性が高いと考えられるからである。また、外部入力

(注 1):  $P_1, P_2$  は共通部分をもたないので、 $P_1$  と  $P_2$  の始点、終点は相異なる。

(注 2): 演算モジュールでは、テスト容易化設計で付加するスルー機能を利用する。

に近い部分の DFT 要素ほど、以降の処理で再利用できる可能性が高いと考えられるからである。

・前処理

(a) データパスに外部入力が一つしか存在しない場合：組合せ回路要素に対して、互いに共通部分をもたない二つの制御経路が存在しないことは明らかである。これを解決するため、外部入力の付加を行う(図 2(a))。実際には、外部入力の代わりにテストパターン生成器をデータパスに付加する。しかしながら、以下のアルゴリズム記述の便宜上、外部入力の付加として扱う。前処理 (a) で新たに付加された外部入力とデータパスの他の回路要素を接続するデータ信号線は後続の前処理 (b) 若しくは制御経路の決定時に付加される。

(b) 2 入力組合せ回路要素  $M$  のそれぞれの入力端子に対して、データパスグラフでその先祖をたどったときに最初に現れる 2 入力組合せ回路要素が同じものであるとき： $M$  に対して、互いに共通部分をもたない二つの制御経路が存在しない。これを解決するため、以下のようにマルチプレクサを付加することにより、新たな経路を付加する(図 2(b))。

マルチプレクサの付加： $M$  の入力端子を  $x, y$  とすると、 $x$  の直前にマルチプレクサを挿入し、マルチプレクサのもう一方の入力端子は外部入力に接続する。このとき、 $y$  へ到達可能な外部入力が一つかない場合には、その外部入力以外の外部入力に接続する。

・制御経路の決定

外部入力に近い組合せ回路要素から順に、制御経路を決定し、DFT 要素を付加する。具体的には、以下の方法で対象とする組合せ回路要素を決定し、その組合せ回路要素に対して、データパスグラフ上で最小費用流問題を解くことで、最小の付加ハードウェアで実現できる制御経路を決定する。

対象となる組合せ回路要素の決定方法：組合せ回路要素  $M$  の入力端子の先祖をたどったときに最初に現れる組合せ回路要素 ( $M$  が 2 入力組合せ回路要素の場合は、それぞれの入力端子の先祖をたどったときに最初に現れる組合せ回路要素  $M_a, M_b$  の双方) が既に制御経路の決定している組合せ回路要素のとき、 $M$  を対象とする(便宜上、外部入力は、制御経路を決定済み組合せ回路要素とみなす)。この場合、制御経路を実現するためにマルチプレクサの付加は必要ない。このような組合せ回路要素が存在しない場合は、 $M_a, M_b$  の一方だけが既に制御経路が決定済みの 2 入力組

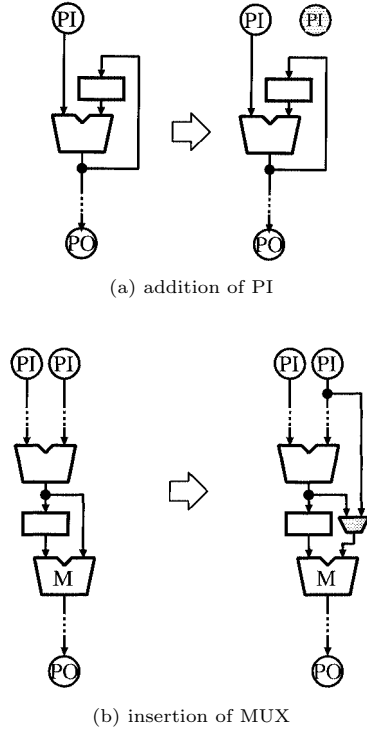


図2 前処理  
Fig.2 Preprocess.

合せ回路要素  $M$  が存在し、 $M$  を対象とする。この条件を満たす組合せ回路要素が複数ある場合には、データパスグラフにおいて、 $M$  から  $M'$  へ到達可能な場合、 $M$  を優先する。更に、互いに到達可能、あるいは互いに到達不可能の組合せ回路要素に関しては、マルチプレクサを優先する。

図 3 にこの手続きの適用例を示す。組合せ回路要素に付加した番号は選択された順番を示す。

このように優先度を定めることにより、外部入力に近い組合せ回路要素から順に制御経路が決定されることとなり、スルー機能を共有しやすくなるので、ハードウェアオーバーヘッドが低くなることが期待できる。

制御経路の決定方法：組合せ回路要素  $M$  に対して、最小の付加ハードウェアで実現できる制御経路を以下のように決定する。

データパスグラフにすべての外部入力への辺をもつダミー頂点を付加する。各演算モジュール  $u$  の各入力端子  $x$  に対し、辺  $(x, u) \in A_2$  のコストを、 $x$  の値を  $u$  の出力に伝搬するスルー機能を実現するのに必要な付加ハードウェア量(既にスルー機能が付加済み

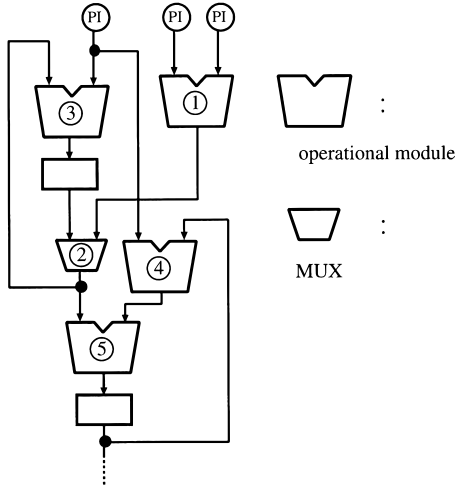


図3 制御経路の決定順  
Fig. 3 Order of control path determination.

のときはコスト0とする)とし、他のすべての辺のコストを0とする。また、すべての辺の容量を1とする(図4)(注3)。

このグラフにおいて、ダミー頂点を始点、 $M$  を終点とする最小費用流問題を解くことにより、最小の付加ハードウェアで実現できる制御経路を決定する。つまり、 $M$  が1入力組合せ回路要素のときは、流量1の最小費用流が  $M$  の制御経路を表す。また、 $M$  が2入力組合せ回路要素のときは流量2の最小費用流が  $M$  の二つの制御経路を表す。このとき、データパスグラフの作り方から、この二つの制御経路は共通部分をもたない。

上記の最小費用流問題において、2入力組合せ回路要素  $M$  に対し、流量2のフローが存在しないことがある。この場合、 $M$  に対して互いに共通部分をもたない二つの制御経路が存在しないということであり、マルチプレクサを付加することにより、新たな経路を付加する。マルチプレクサの付加は前処理と同様に行う。このマルチプレクサの制御経路も、前出の方法で決定するが、マルチプレクサの付加方法から、共通部分をもたない制御経路が存在するのは明らかである。

#### 4.2.2 観測経路の決定とDFT要素付加

観測経路は、出力端子をもつ各組合せ回路要素  $M$  に対して、制御経路と共通部分をもたず、最小の付加ハードウェアで実現できる観測経路を求める。制御経路の場合と同様に、観測経路を求める組合せ回路要素

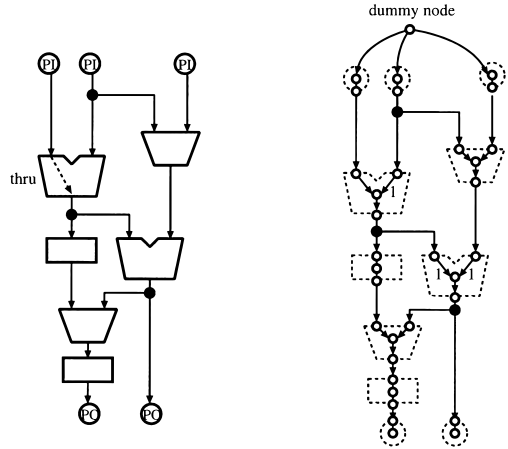


図4 データパスグラフのコスト  
Fig. 4 Cost of data path graph.

は一つずつ処理していく。ただし、外部出力に近い組合せ回路要素から順に観測経路を決定し、DFT要素を付加する。具体的には、以下の方法で対象とする組合せ回路要素を決定し、その組合せ回路要素に対してデータパスグラフ上で最小費用流問題を解くことで、最小の付加ハードウェアを実現できる観測経路を決定する。

対象とする組合せ回路要素の決定方法：組合せ回路要素  $M$  の出力端子  $ot$  の子孫をたどったときに最初に現れる組合せ回路要素の集合を  $M$  とする。 $M$  に既に観測経路の決定している組合せ回路要素が含まれるとき、対象とする組合せ回路要素を  $M$  とする。ただし、この条件を満たす組合せ回路要素が複数ある場合には、 $M$  が観測経路を決定済みのMUX、若しくは観測経路決定済みの演算モジュールで、 $ot$  からたどった入力端子にスルー機能が付加されているものを優先する。更に、そのような組合せ回路要素が複数存在する場合には、組合せ回路要素  $M'$  から組合せ回路要素  $M$  に到達可能な場合には  $M$  を優先する。

図5にこの手続きの適用例を示す。組合せ回路要素に付加された番号は選択された順番を示す。

観測経路の決定方法：組合せ回路要素  $M$  に対して、最小の付加ハードウェアで実現できる観測経路を以下のように決定する。データパスグラフにすべての外部出力からの辺をもつダミー頂点を付加する。更に  $M$

(注3): 図4では、演算モジュールに対するスルー機能のコストはすべて1とした。コストを記入していない辺のコストはすべて0。

の制御経路に現れるすべての辺を取り去る．こうして得られたグラフに，制御経路を求めるときと同様のコスト，容量を与える（既に付加済みのスルー機能についてはコスト 0 とすることに注意）．このグラフで， $M$  を始点，ダミー頂点を終点とする流量 1 の最小費用流が， $M$  の観測経路を表す．

上記の最小費用流問題において，流量 1 のフローが存在しないことがある．この場合， $M$  の制御経路と共通部分をもたない観測経路が存在しないということであり，マルチプレクサを付加することにより，新たな経路を付加する．マルチプレクサの付加は以下のように行う（図 6）．

マルチプレクサの付加：対象となる組合せ回路要素  $M$  の出力端子を  $z$ ，任意の外部出力を  $d$  とする． $d$

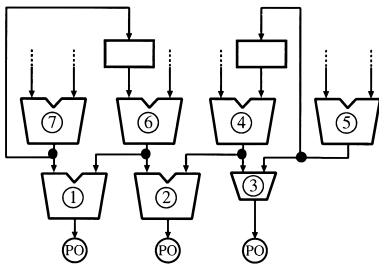


図 5 観測経路の決定順  
Fig. 5 Order of observation path determination.

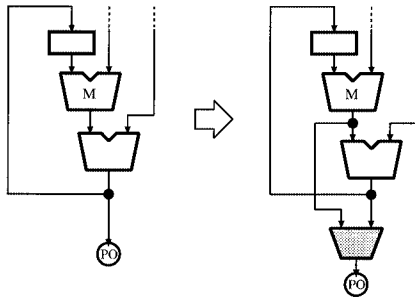


図 6 MUX の挿入  
Fig. 6 Insert of multiplexer.

の直前にマルチプレクサを挿入し，他方の入力に  $z$  を接続する．

このマルチプレクサの制御経路は，前述の制御経路の決定方法に従う．

### 5. 実験結果

従来手法として文献 [4] による方法，及び提案手法（単一制御可検査テスト容易化設計法）を，テスト容易化設計後のハードウェアオーバヘッド及び故障検出率について比較した．実験に使用した RTL ベンチマーク回路は，GCD と Paulin [9]，3rd Lattice Wave Filter (LWF)，4th IIR である．また実設計回路として，小規模の RISC プロセッサ（以下，RISC と呼ぶ）を使用した．これらのベンチマーク回路の回路特性を表 1 に示す．「circuit」は回路名を表し，「bit width」，「#PI」，「#PO」，「#Reg.」，「#MUX」，「#OP」はそれぞれビット幅，外部入力数，外部出力数，レジスタ数，マルチプレクサ数，演算モジュール数を表す．ハードウェアオーバヘッドの算出に必要な回路面積は，論理合成ツール「AutoLogicII (Mentor Graphics)」及び ALTERA 社の論理合成ライブラリを用いて求めた．表 1 の Area に回路面積 (gate equivalent) を示す．

各手法でのテスト容易化設計に伴う付加回路によるハードウェアオーバヘッドを表 2 に示す．

ほとんどの事例で，提案手法のハードウェアオーバヘッドは従来手法より低いことがわかった．

ただし，従来手法，提案手法ともすべての外部入出力に対してテストパターン生成器及び応答解析器を付加するので，表 2 のハードウェアオーバヘッドの項は，外部入出力に付加されるテストパターン生成器及び応答解析器を含まないものとした．

次に，各手法での最終的な故障検出率とその故障検出率が得られた最小のクロックサイクル数を表 3 に示す．実験では両手法において発生可能なすべてのテストパターンを C 言語で記述したプログラムであらかじめ生

表 1 データバスの回路特性

Table 1 Character of circuits.

circuit	bit width	#PI	#PO	#Reg.	#MUX	#OP	Area
GCD	16	2	1	3	4	1	530.6
Paulin	16	2	2	7	11	4	3818.9
LWF	16	2	2	5	5	3	735.0
4th IIR	16	1	1	12	3	5	1728.3
RISC	32	1	3	40	84	19	25731.5

表2 ハードウェアオーバーヘッド (HWOH)  
Table 2 Hardware overheads (HWOH).

circuit	bit width	従来手法 [4]			提案手法			
		HWOH (%)	#BILBO	#CBILBO	HWOH (%)	#MUX	#THRU	#LFSR
GCD	8	89.30	1	1	18.67	1	1	0
	16	74.54			18.36			
	32	66.67			18.06			
Paulin	8	80.74	0	6	22.50	8	5	0
	16	40.25			13.23			
	32	20.07			7.27			
LWF	8	43.23	0	1	35.10	4	3	0
	16	34.86			33.32			
	32	30.87			32.47			
4th IIR	8	41.47	0	2	33.73	5	6	1
	16	29.65			27.02			
	32	25.70			25.10			
RISC	32	59.12	0	33	18.42	36	18	1

表3 故障検出率  
Table 3 Fault coverage.

回路名	従来手法 [4]		提案手法	
	故障検出率 (%)	クロック数	故障検出率 (%)	クロック数
GCD	96.07	556	99.83	10178
Paulin	99.84	1108	99.89	20531
LWF	81.44	143	99.82	554
4th IIR	97.89	136	99.57	604

成した後に故障シミュレータ (TestGen(Synopsys)) を用いて印加パターン数と故障検出率の関係を調べた。

すべての場合において、提案手法は従来手法よりも高い故障検出率を得ることができたが、それを得るためのテスト実行時間は大きくなった。これは、提案手法が組合せ回路要素ごとにテストを行っているのに対し、従来手法では閉路を含まない組合せ回路要素の集合ごとに並列にテストを行っているからである。このとき、検出できなかった故障に対してテスト生成したところ、これらの故障は冗長故障であるか、「すべて0のパターン」がテストパターンとなる故障であった。

RISC については回路規模が非常に大きいため、故障検出率に関する実験を行っていないが、他のベンチマーク回路の結果から、高い故障検出率が得られることが予想される。

## 6. むすび

レジスタ転送レベルデータパスの組込み自己テストとして、階層テストに基づく test per clock 方式の BIST を提案した。提案方式は実動作速度でのテストが可能で、低いハードウェアオーバーヘッドでデータパスに対して高い故障検出率を得ることができる。

また、実験によって、提案手法によって生じるハードウェアオーバーヘッドは従来手法のものより低く、また従来手法よりも高い故障検出率が得られることを示した。

今後の課題としては、テスト実行の並列化が挙げられる。つまり、同時にテスト可能な組合せ回路要素を選択し、テストスケジューリングを行うことで、テスト実行時間が削減できると考えられる。

謝辞 本研究に際し、多くの貴重な意見を頂いた広島市立大学の井上智生助教授並びに本学の情報論理学講座の諸氏に深く感謝します。本研究は一部(株)半導体理工学研究センター (STARC) との共同研究、及び、文部省科学技術研究費補助金・基盤研究 B(2) (代表者: 藤原秀雄, 課題番号 09480054) の研究助成による。

## 文 献

- [1] P.H. Bardell, W.H. McAnney, and J. Savir, Built-In Test for VLSI: Pseudorandom Techniques, Wiley-Interscience, New York, 1987.
- [2] M. Abramovici, M.A. Breuer, and A.D. Freedman, Digital Systems Testing and Testable Design, Computer Science Press, New York, 1990.
- [3] B. Koenemann, J. Mucha, and G. Zwiehoff, "Built-in test for complex digital integrated circuits," IEEE



J. Solid-State Circuits, vol.SC-15, no.3, pp.315-318, 1980.

- [4] A.P. Stroele and H.-J. Wunderlich, "Hardware-optimal test register insertion," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol.17, no.6, pp.531-539, June 1998.
- [5] B.T. Murray and J.H. Hayes, "Hierarchical test generation using pre computed tests for modules," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol.16, no.9, pp.1001-1014, 1990.
- [6] 和田弘樹, 増澤利光, K.K. Saluja, 藤原秀雄, "完全故障検出効率を保証するデータパスの非スキャンテスト容易化設計法," 信学論 (D-I), vol.J82-D-I, no.7, pp.843-851, July 1999.
- [7] I. Ghosh, N.K. Jha, and S. Bhawmik, "A BIST scheme for RTL controller-data paths based on symbolic testability analysis," Proc. 35th Design Automation Conf., pp.554-559, San Francisco, USA, June 1998.
- [8] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, 1979.
- [9] I. Ghosh, A. Raghunathan, and N.K. Jha, "Design for hierarchical testability of RTL circuits obtained by behavioral synthesis," Proc. IEEE Int. Conf. on Computer Design: VLSI in Computers & Processors, pp.173-179, Austin, USA, Oct. 1995.

(平成 12 年 3 月 31 日受付, 8 月 22 日再受付)



井筒 稔

平 10 岡山大・工・情報卒。平 12 奈良先端科学技術大学院大学博士前期課程了。同年 (株) 日立製作所入社。テスト容易化設計に関する研究に従事。



和田 弘樹 (学生員)

平 8 阪大・工・通信卒。平 10 奈良先端科学技術大学院大学博士前期課程了。現在, 奈良先端科学技術大学院大学博士後期課程に在学中。現在, テスト容易化設計の研究に従事。



増澤 利光 (正員)

昭 57 阪大・基礎工・情報卒。昭 62 同大大学院博士後期課程了。同年同大情報処理教育センター助手。同大基礎工助教授を経て, 平 6 奈良先端科学技術大学院大学情報科学研究科助教授, 現在に至る。平 5 コーネル大客員準教授 (文部省在外研究員)。分散アルゴリズム, 並列アルゴリズム, テスト容易化設計, テスト容易化高位合成に関する研究に従事。工博。ACM, IEEE, EATCS, 情報処理学会各会員。



藤原 秀雄 (正員)

昭 44 阪大・工・電子卒。昭 49 同大大学院博士課程了。同大・工・電子助手, 明治大・工・電子通信助教授, 情報科学教授を経て, 現在奈良先端大・情報科学教授。昭 56 ウォータールー大客員助教授。昭 59 マッギル大客員準教授。論理設計論, フォールトトレランス, 設計自動化, テスト容易化設計, テスト生成, 並列処理, 計算複雑度に関する研究に従事。著書「Logic Testing and Design for Testability」(MIT Press) など。大川出版賞, IEEE Computer Society Meritorious Service Award 各受賞。情報処理学会会員, IEEE Computer Society Golden Core Member, IEEE Fellow。