固定制御可検査性に基づくRTL回路の非スキャンテスト容易化設計法

永井 慎太郎,　和田 弘樹,　大竹 哲史,　藤原 秀雄

奈良先端科学技術大学院大学　情報科学研究科
〒630-0101　奈良県生駒市高山町8916-5
TEL: 0743-72-5226　FAX: 0743-72-5229
E-mail: {*shinta-n, hiroki-w, ohtake, fujiwara*}*@is.aist-nara.ac.jp*

あらまし　本論文では，階層テスト生成が容易なデータパスの性質として固定制御可検査性を新しく定義し，それに基づくレジスタ転送レベル回路のテスト容易化設計法を提案する．提案手法では，組合せテスト生成法を用いた階層テスト生成および非スキャン設計に基づいているため，テスト生成時間およびテスト実行時間を完全スキャン設計法に比べて大幅に短縮でき，実動作速度でのテスト（at-speed test）が可能で，完全故障検出効率を保証することができる．さらに，ベンチマーク回路を用いた実験により提案手法の有効性を示す．

キーワード　固定制御可検査性，完全故障検出効率，レジスタ転送レベル，階層テスト

A Non-Scan DFT Method for RTL Circuits Based on Fixed-Control Testability

Shintaro Nagai，　Hiroki Wada，　Satoshi Ohtake　and　Hideo Fujiwara

Graduate School of Information Science
Nara Institute of Science and Technology
8916-5, Takayama, Ikoma, Nara 630-0101
TEL: +81-743-72-5226　FAX: +81-743-72-5229
E-mail: {*shinta-n, hiroki-w, ohtake, fujiwara*}*@is.aist-nara.ac.jp*

Abstract　In this paper, we introduce a new testability called ”fixed-control testability” for data paths at register transfer level (RTL) for which hierarchical test generation using combinational ATPG can be applied easily, and present a non-scan design for testability (DFT) method based on the fixed-control testability for RTL circuits with controllers and data paths. Thanks to non-scan DFT and hierarchical test generation using combinational ATPG, the proposed method can reduce significantly both test generation time and test application time compared with the full-scan design, and can achieve complete fault efficiency and at-speed testing. Experimental results are presented to show the effectiveness of the proposed method.

Key words　fixed-control testability，complete fault efficiency, register transfer level, hierarchical test

1 はじめに

近年のVLSI回路の大規模化・高集積化は，VLSI回路のテストを困難な問題にしており，テスト費用の削減およびテストの質の向上が求められている．テスト費用はテスト生成やテスト実行にかかる時間で評価できる．テストの質は故障検出効率で評価できる．故障検出効率とは，回路中のテスト生成の対象となる全故障数に対する，テスト生成アルゴリズムによって生成されたテスト系列が検出可能な故障数とテスト生成アルゴリズムが冗長と判定した故障数の和の割合をいう．特に故障検出効率が100%の場合を完全故障検出効率という．

組合せ回路に対しては，実用的なテスト生成時間で完全故障検出効率を達成できるテスト生成アルゴリズムが提案されている[1]．これに対して，順序回路では，一般に実用的なテスト生成時間で高い故障検出効率を達成することは困難である．そのため，順序回路をテスト生成の容易な回路に設計変更するテスト容易化設計法が提案されている．

代表的なテスト容易化設計法として，完全スキャン設計法[1]，[2]がある．この手法では，順序回路中の全てのフリップフロップ(FF)をスキャンFFに置き換えることにより，FFの値を外部から直接制御および観測を可能にする．これにより，テスト生成の際に全てのFFを外部入出力とみなすことができる．そのため，組合せ回路部分に対して組合せ回路用のテスト生成アルゴリズムを用いてテスト生成を行うことができるので，完全故障検出効率を達成できる．しかし，この手法には，大きい面積オーバヘッドおよび長いテスト実行時間を要し，実動作速度でのテスト(**at-speed test**)[3]が困難であるなどの問題点がある．

これらの問題点を解消するために，近年，レジスタ転送(**RT**)レベルでのコントローラ/データパス回路を対象とするテスト生成やテスト容易化設計法が提案されている[4],[5],[7]-[9]．RTレベルでのコントローラは状態遷移図で記述され，データパスはレジスタ，組合せ回路で構成される演算モジュールなどの回路要素とそれらの回路要素を接続する信号線で記述される．

コントローラのテスト容易化設計法として，文献[7]の手法がある．この手法では，はじめに組合せ回路部に対して組合せ回路用のテスト生成アルゴリズムを用いてテストベクトル集合を求める．次に，コントローラ中の内部レジスタにテストベクトルを構成する値を外部から設定できるかどうかを考える．値を設定できる場合は，テスト容易化設計を必要としない．値を設定できない場合は，無効テスト状態生成器と呼ばれる回路を付加する．この手法では，無効テスト状態生成器による面積オーバヘッドは小さく，短いテスト生成時間で完全故障検出効率を達成できる．

データパスのテスト容易化設計法として，**Genesis** [4],[5]や強可検査テスト容易化設計法[8]などがある．これらの手法は階層テスト生成法[6]に基づいている．階層テスト生成法では，組合せ回路で構成される回路要素単体に対してテスト生成およびテストプラン生成を行う．テストプランとは，外部入力からその回路要素の入力へテストベクトルを伝搬(正当化)し，その回路要素の出力応答を外部出力へ伝搬するための，制御ベクトルの時系列をいう．一般にテストプランはテストベクトルの正当化，テスト，出力応答の伝搬の3フェーズで構成され，各フェーズの制御ベクトルは，時刻毎に変化する．

コントローラ/データパス回路全体に対する非スキャンテスト容易化設計法として，コントローラおよびデータパスそれぞれに対して文献[7]の手法および強可検査テスト容易化設計法[8]を適用した手法[9]がある．この手法では，コントローラとデータパスそれぞれのテスト容易性を保証するために，回路内部にテスト用回路を付加する．強可検査テスト容易化設計後のデータパスに対して，テストプランを供給するテストプラン生成回路を回路内部に付加することにより，小さい外部ピンオーバヘッドで，実動作速度でのテスト実行を可能にしている．さらにこの手法では，テスト生成時間およびテスト実行時間を完全スキャン設計法に比べて大幅に短縮している．しかし，テストプラン生成回路は順序回路で構成されるので，その面積オーバヘッドが大きいという問題がある．

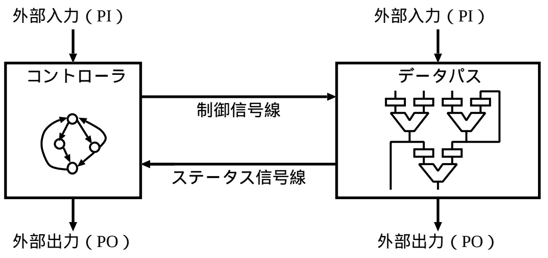


図 1: RT レベル回路

テストプラン生成回路の面積を縮小するために，本論文ではデータパスの新しい回路構造の性質として，固定制御可検査性を導入し，固定制御可検査性に基づくデータパスのテスト容易化設計法を提案する．固定制御可検査性を満たすデータパス中の各回路要素に対するテストでは，テストプランの各フェーズにおいて，1 個の制御ベクトルを制御入力に固定しておくことができる．従って，固定制御可検査性を満たすデータパスでは，各組合せ回路要素に対するテストプランは，高々3 個の制御ベクトルで構成され，そのデータパスに対するテストプラン生成回路は組合せ回路で実現できる．実験では，提案手法は文献 [9] でのテスト生成時間およびテスト実行時間と同等であることを示す．データパスの面積オーバヘッドは，文献 [9] と同等であり，テストプラン生成回路および回路全体の面積オーバヘッドは文献 [9] に比べて縮小できたことを示す．

以下，2 章では本論文で対象となるコントローラ/データパス回路，階層テスト生成法，データパスの強可検査性を定義する．3 章では固定制御可検査性に基づくデータパスのテスト容易化設計法およびテストプラン生成法を提案する．4 章ではベンチマーク回路および実設計回路を用いた実験により，提案手法の有効性を示す．

2　諸定義

2.1　コントローラとデータパス

RT レベルでの VLSI 回路は一般に，コントローラとデータパスから成る (図 1)．コントローラは状態遷移図，データパスは回路要素と回路要素を接続する信号線で記述される．回路要素は，外部入力，外部出力，ホールド機能を持つレジスタと持たないレジスタ，マルチプレクサ (MUX)，加算器や乗算器などの演算モジュールおよび比較器などの観測モジュールに分類される．各回路要素はポートを持ち，ポートはデータポート，制御ポートおよび観測ポートに分類される．データポートには，回路要素にデータを入力する入力ポートと回路要素からデータを出力する出力ポートがある．制御ポートは，コントローラから制御信号を入力するポートである．観測ポートは，コントローラへステータス信号を出力するポートである．信号線は，データ信号線，制御信号線およびステータス信号線に分類される．データ信号線は，2 つの回路要素のデータポートを接続する．制御信号線は，コントローラと制御ポートを接続する．ステータス信号線は，観測ポートとコントローラを接続する．本論文では議論を簡単にするために，以下の制約を満たすデータパスを考える．以下の制約のいずれかを満たさないデータパスが与えられた場合，3 章で提案するテスト容易化設計法をそのデータパスに適用できるようにする必要がある．これについては 3.4 節で述べる．

A1: データポートのビット幅は全て等しい．

A2: 観測モジュール以外の全ての回路要素は，1 または 2 個の入力ポート，1 個の出力ポート，高々 1 個の制御ポートと観測ポートを持つ．観測モジュールは，1 または 2 個の入力ポート，高々 1 個の制御ポートおよび 1 個の観測ポートを持つが，出力ポートを持たない．

A3: 外部入力から各回路要素の入力ポート，各回路要素の出力ポートから外部出力へ経路が存在する．

A4: 制御信号線は回路要素の制御ポートに直接コントローラから接続される．ステータス信号線は回路要素の観測ポートから直接コントローラへ接続される．

以下，組合せ回路で構成される MUX，演算モジュールおよび観測モジュールを組合せ回路要素と呼ぶ．

2.2 階層テスト生成法

階層テスト生成法は，次の 2 段階から成る．第 1 段階ではゲートレベルにおいて，各組合せ回路要素単体に対して組合せ回路用のテスト生成アルゴリズムを用いてテストベクトル集合 V を生成する．第 2 段階では RT レベルにおいて，V を外部入力から組合せ回路要素へ正当化し，その出力応答を外部出力へ伝搬するためのテストプラン (制御ベクトル系列) を求める．

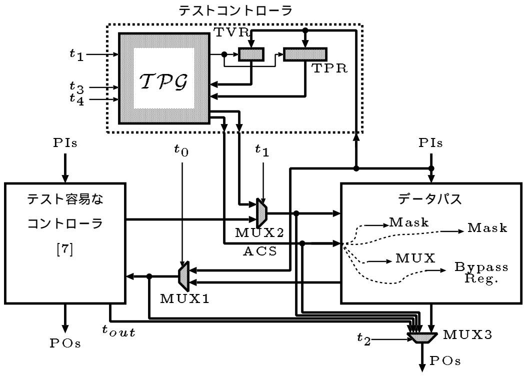
2.3 データパスの強可検査性 [8]

定義 1 (強可検査性) データパス DP 中の各組合せ回路要素 M に対してテストプラン TP が存在し，その TP で外部入力から M の入力ポートへ任意の値を正当化，かつ M の出力ポートから任意の値を外部出力へ伝搬できるとき，DP は強可検査であるという． □

強可検査データパスには，次の特長がある．

- 各組合せ回路要素単体に対して，組合せ回路用のテスト生成アルゴリズムを適用するので，短いテスト生成時間で完全故障検出効率を達成できる．
- テストプラン生成は RT レベルで生成されるため，テストプラン生成時間が短い．
- 各組合せ回路要素に対してテストプランが生成できることを保証しているので，データパス全体に対して完全故障検出効率を達成できる．

強可検査データパスに対するテストプラン TP は，外部入力から M の入力ポートへの経路集合 CP 上でテストベクトル集合 V を正当化でき，かつ M の出力ポート z から外部出力への経路集合 OP 上で出力応答 O を伝搬できる．一般に TP は以下の 3 フェーズで構成される．ここで $R(CP)$ を CP 上に存在し，かつ M の入力ポートの直前のレジスタ集合とする．$R(OP)$ を OP 上に存在し，かつ z の直後のレジスタ集合とする．



t_i: テストピン
t_{out}: 状態レジスタ出力
ACS: 付加制御信号線
TVR: テストベクトルレジスタ
TPR: 対象テストプランレジスタ
$\mathcal{TPG}$: テストプラン生成回路

図 2: テストコントローラを含む RT レベル回路

正当化フェーズ: $R(CP) \neq \phi$ のとき，正当化フェーズの制御ベクトル系列は，$R(CP)$ へ V を正当化する．$R(CP) = \phi$ のとき，正当化フェーズは必要ない．

テストフェーズ: テストフェーズの制御ベクトル系列は，V を $R(CP)$ の全てのレジスタから M の全ての入力ポートへ正当化し，z から $R(OP)$ の全てのレジスタへ O を伝搬する．CP の経路のうち，外部入力から M の入力ポート x までにレジスタがないときは，外部入力から x へ直接 V を正当化する．OP の経路のうち，z から外部出力までレジスタがないときは，z から外部出力へ O を直接伝搬する．

伝搬フェーズ: $R(OP) \neq \phi$ のとき，伝搬フェーズでの制御ベクトル系列は，$R(OP)$ の全てのレジスタから外部出力へ O を伝搬する．$R(OP) = \phi$ のとき，伝搬フェーズは必要ない．

3 RT レベルコントローラ/データパス回路のテスト容易化設計法

3.1 概要

提案手法では，与えられた RT レベルのコントローラ / データパス回路 (図 1) に対して，文献 [9] の手法と同様に，コントローラとデータパスそれぞれに対してテスト容易化設計を行う．コントローラに対しては，文献 [7] の手法を適用し，データパスに対し

ては，本論文で提案する固定制御可検査性に基づくテスト容易化設計法を適用する (3.2 節)．これらの手法では，制御信号線およびステータス信号線は外部から直接制御 / 観測が可能であると仮定している．この仮定を取り除くために，コントローラとデータパス間にテスト用の回路を付加する (図 2)．ステータス信号線上に MUX1 を付加し，データパスの外部出力側に MUX3 を付加することにより，コントローラのテストに必要な外部からのステータス信号線の制御，制御信号線の外部での観測およびコントローラ内部の状態レジスタ出力 t_{out}[7] の観測が可能となる．さらに，データパスのテスト用の制御ベクトル系列をデータパスへ供給するためのテストコントローラを回路内部に付加し (3.3 節)，制御信号線上に MUX2 を付加することにより，データパスのテストに必要な制御信号線の制御が可能となる．また，MUX1 および MUX3 を利用することにより，データパスからのステータス信号線およびデータ信号線の観測が可能となる．データパスのテスト容易化設計において付加した回路要素は，テストコントローラから付加制御信号線 ACS を介して制御する．テストコントローラおよびコントローラ/データパス間に付加した MUX は直接外部からテストピンを介して制御する．これらの付加回路のテストピンの制御を表 1 に示す．表 1 において，* はテストベクトルまたはテストプランに依存することを示す．

3.2　データパスのテスト容易化設計法

3.2.1　データパスの固定制御可検査性

定義 2 (固定制御可検査性) データパス DP が以下の条件を満たすとき，DP は固定制御可検査であるという．

C1: DP は強可検査である．

C2: DP 中の各組合せ回路要素に対するテストプランの正当化/テスト/伝搬の各フェーズは，各々 *1* 個の制御ベクトルで構成できる．□

定義 2 より，固定制御可検査データパスは強可検査データパスの特長を持つ．さらに，各組合せ回路要素のテストプランにおける制御ベクトル数が高々 3 個である．従って，固定制御可検査データパスに対するテストプラン生成回路の回路面積が，文献 [9] に比べて小さい．

表 1: テストピンの制御

テストピン					動作
t_0	t_1	t_2	t_3	t_4	
0	0	0	0	0	通常動作
1	0	1	*	*	コントローラのテスト
0	1	*	*	*	データパスのテスト

3.2.2　テスト容易化設計法

提案するテスト容易化設計法では，与えられたデータパス中の各組合せ回路要素に対して，任意のテストベクトルを外部入力から正当化するための経路 (制御経路)，および任意の出力応答を外部出力へ伝搬するための経路 (観測経路) を決定する．次にそれらの経路に対して以下 2 点を実現するようにテスト用の回路を付加することで，任意の値の伝搬を保証する．

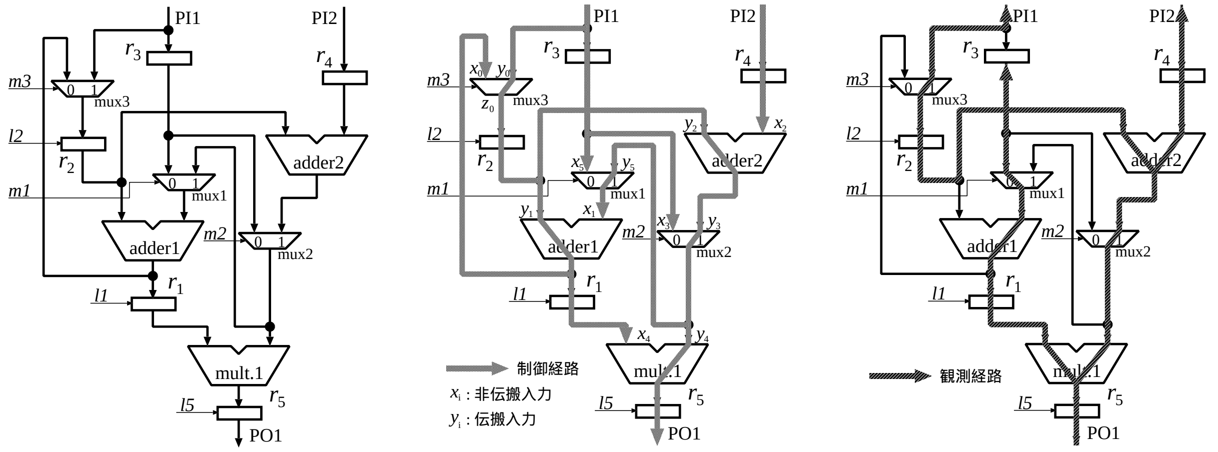
1. 演算モジュールの入出力ポート間での値の伝搬

2. 順序深度の等しい再収斂経路を構成する制御経路上での値の伝搬のタイミング衝突の解消

順序深度とは，経路上にあるレジスタ数をいう．再収斂経路とは，異なる回路要素 e_1 および 2 入力ポートを持つ e_2 に対して，e_1 を始点，e_2 を終点とする任意の異なる経路の対のうち，e_1，e_2 以外に共通部分を含まないものをいう．タイミング衝突とは，2 つの異なる値が同時刻に外部入力で必要とされる場合をいう．

提案手法では，演算モジュール M の入出力ポート間に存在する制御経路または観測経路上で任意の値が伝搬できることを保証するために，上記 1 に対して，スルー機能を M に付加する．スルー機能は，M の入力ポートに印加した値を変えることなく出力ポートへ伝搬させる機能である．

2 入力ポートを持つ組合せ回路要素 M の各入力ポートに対する制御経路の対が再収斂経路を構成し，かつその各経路の順序深度が等しい場合がある．この場合，任意の異なる値を各入力ポートに対して同



(a) データパス例　(b) 制御林　(c) 観測林

図 3: (a) データパス例, (b) 制御林, (c) 観測林

時に与えることができない．上記 2 に対して，強可検査テスト容易化設計法 [8] では，レジスタのホールド機能を用いて，値の伝搬のタイミング衝突を解消するが，固定制御可検査性を満たすためには，レジスタのホールド機能を利用できない．提案手法では，テスト MUX またはバイパスレジスタを再収斂経路の一方の経路上に付加することにより，タイミング衝突を解消する．

提案するテスト容易化設計法は以下の 5 つのステップから成る．

ステップ **1:** 制御経路の決定

データパス DP 中の各回路要素に対して制御経路を決定する．一般には，ある外部入力からある入力ポートへの経路は複数存在するが，テスト実行時間を短縮するには，順序深度が最小の経路を選択することが望ましい．またステップ 3 においてスルー機能が付加される場所を極小化するためには，制御経路同士ができるだけ多くの共有部分を持つことが望ましい．そこで，各回路要素に対する制御経路の集合を DP の外部入力を根として，経路の順序深度に関する最短経路林として生成する．これにより，テスト実行時間の短縮およびスルー機能数の削減が可能となる．ここで，組合せ回路要素の入力ポートについて，制御林の葉となるものを非伝搬入力，そうでないものを伝搬入力と呼ぶ．

図 3(a) のデータパス中の各回路要素に対して制御経路を生成した例を図 3(b) に示す．図 3(b) において，各組合せ回路要素の入力ポート x_i および y_i は，それぞれ，非伝搬入力および伝搬入力を表す．

ステップ **2:** 観測経路の決定

DP 中の各組合せ回路要素に対して観測経路を決定する．一般には，ある出力ポートからある外部出力への経路は複数存在するが，制御経路との共有部分が極大となる経路を選択することにより，ステップ 5 で付加されるスルー機能数を削減できる．従って，各回路要素に対する観測経路の集合を DP の外部出力を根として，経路上の非伝搬入力数に関する最小経路林を生成する．

図 3(a) のデータパス中の各回路要素に対して観測経路を生成した例を図 3(c) に示す．

ステップ **3:** スルー機能の付加 (制御経路上の値の伝搬の保証)

ある制御経路が演算モジュール M の伝搬入力 y と出力ポート z 間を通る場合を考える．$y-z$ 間にスルー機能がなければ，スルー機能を M に付加する．スルー機能は，加算器や乗算器などの演算モジュールであれば，マスク素子を利用することでスルー機能を低面積で実現できる．マスク素子とは，演算モジュールの伝搬入力と出力ポート間に任意の値を伝

搬するのに必要な定数を非伝搬入力に対して発生する回路である．マスク素子を用いてスルー機能を実現できない場合は，MUX を付加してスルー機能を実現する．

図 3(b) において，adder1，adder2 および mult.1 の伝搬入力と出力ポート間のスルー機能 (マスク素子) をそれらの演算モジュールに付加する．

ステップ 4: テスト MUX またはバイパスレジスタの付加

順序深度の等しい経路対で構成される再収斂経路ごとに，テスト MUX またはバイパスレジスタを付加すると，付加回路による面積オーバヘッドが大きくなる可能性がある．従って，付加回路数を少なくするために，以下のようにテスト容易化設計を行う．

外部入力からステップ 1 で生成した制御経路上の伝搬入力に関する幅優先探索を行い，2 入力ポートを持つ組合せ回路要素の伝搬入力に到達した順番に，その組合せ回路要素に順番を付ける．この順に，外部入力から組合せ回路要素 M の各入力ポートまでの制御経路の対が，各入力ポートを終点として，順序深度の等しい再収斂経路を構成するかどうかを調べる．順序深度の等しい再収斂経路が構成される場合には，その再収斂経路を構成する経路上で任意の値が伝搬できるようにタイミング調整を行う．再収斂経路を始点として，再収斂経路を構成する一方の経路上にテスト MUX(TM) またはバイパスレジスタ (BR) が既に付加されている場合がある．これらの付加回路を用いて，一方の経路上の順序深度を変更することができれば，新たにテスト容易化設計を行う必要はない．この場合，M の制御経路をこれらの付加回路を用いることができるように変更する．再収斂経路を構成する経路対のどちらにも TM または BR が存在しない場合は，以下のように TM または BR を付加する．ここで，再収斂経路の始点および終点となる回路要素をそれぞれ e_1 および e_2 とする．再収斂経路を構成する経路で，e_1 の出力ポートから e_2 の伝搬入力までの経路を p_c とする．

(1)p_c 上にレジスタが存在する場合

p_c 上にレジスタが存在する場合は TM を付加する．ここで，p_c 上にある e_1 の直後のレジスタを r とする．r の出力ポートを z' とし，r の直後の回路要素の伝搬入力を x' とする．また，TM の 2 つの入力ポートを x および y とし，出力ポートを z とする．このとき，TM を付加する場所について考える．TM を外部入力に近い場所に付加すれば，TM を再利用できる可能性が高くなり，面積オーバヘッドを縮小できると期待できる．そこで，TM を r の直後に付加する．

$z'-x'$ 間の信号線を除去し，TM の x と z はそれぞれ，z' および x' を接続する．r が自己ループを持つ場合，r の直前の信号線と TM の入力ポート y を接続すると，非同期ループを作ることになる．そこで，非同期ループを作らない接続を考える．外部入力から r の入力ポートまでの制御経路を p_r とし，p_r 上にあるレジスタ集合を R_i とする．$R_i \neq \phi$ のとき，r に最も近いレジスタの出力ポートと y を接続する．$R_i = \phi$ のとき，p_r の始点である外部入力と y を接続する．

TM のテストを考える．TM の観測経路は p_c と共有しているので，TM の出力応答の伝搬のための観測経路を構成する必要はない．また TM の各入力ポートを終点とする制御経路の対は再収斂経路を構成するが，順序深度が異なるので，任意の値の伝搬を保証できる．

(2)p_c 上にレジスタが存在しない場合

p_c 上にレジスタが存在しない場合は，TM の付加の場合と同様に，p_c 上の e_1 の直後に BR を付加する．BR を構成する MUX のテストについて考える．付加した BR の出力ポートは e_2 の伝搬入力に接続しているので，その MUX の出力応答の伝搬のための観測経路を新たに構成する必要はない．また，その MUX の入力ポートへの経路の対は再収斂経路となるが，順序深度が異なるので，任意の値の伝搬を保証できる．

図 3(b) において，mux3，adder1，adder2，mux2，mult.1，mux1 の順で，これらの組合せ回路要素の各入力ポートまでの制御経路が順序深度の等しい再収斂経路を構成しているかどうかを調べる．adder1 の入力ポートまでの各制御経路は，順序深度が 1 であ

り，外部入力 PI1 を始点として再収斂経路を構成する．PI1 と adder1 の伝搬入力 y_1 間にレジスタ r_2 のみ存在するので，r_2 の直後に TM1 を付加する．これにより，y_1 までの制御経路の順序深度が 0 となる経路を用いてタイミング調整を行う．mux2 の伝搬入力 y_3 の取り得る順序深度は，y_3 までの制御経路上に TM1 が存在するので，0 または 1 となる．mux2 の非伝搬入力 x_3 までの制御経路の順序深度は 1 であるので，y_3 までの制御経路の順序深度が 0 となるように y_3 までの制御経路を変更することで，タイミング調整を行う．mux1 の伝搬入力 y_5 の取り得る順序深度は，y_0 までの制御経路上に TM1 が存在するので，0 または 1 となる．mux1 の非伝搬入力 x_5 までの制御経路の順序深度は 1 であるので，y_5 までの制御経路の順序深度が 0 となるように y_5 の制御経路を変更することで，タイミング調整を行う．他の組合せ回路要素は，順序深度または制御経路の始点となる外部入力が異なるので，タイミング衝突は起きない．

ステップ 5: スルー機能の付加 (観測経路上の値の伝搬の保証)

ステップ 2 で生成した組合せ回路要素 M に対する観測経路が 2 入力演算モジュール M_j を通る場合を考える．観測経路が M_j の非伝搬入力 x 上を通る場合，M_j の伝搬入力 x と出力ポート z 間にスルー機能がない場合には，任意の値を伝搬できない．ここで，M_j の y に定数を与えて $x-z$ 間のスルー機能を実現できる場合について考える．外部入力から M_j の y へ定数を印加できれば，M_j の $x-z$ 間のスルー機能を新たに付加する必要はないので，スルー機能実現のための付加回路面積を削減できる．そこで，外部入力から M_j の $x-z$ 間のスルー機能を実現するのに必要な定数を M_j の y へ印加するための経路 (補助経路) の利用について考える．ただし，y に対する補助経路と，M の入力ポートへ任意の値を印加するのに用いる経路または M_j とは別の演算モジュール M' のスルー機能を実現するための定数の印加に用いる補助経路間でタイミング衝突が起きる場合がある．そこで，以下の 3 つのいずれかの条件を満たすかどうかを調べることにより，タイミング衝突が起きるかどうかを判定する．ここで，M の出力ポートから M_j の x までの観測経路上に存在し，かつ M_j に最も近い組合せ回路要素を M' とする．また，I を M の入力ポートへテストベクトルを印加し，かつその出力応答を x へ伝搬するのに必要な外部入力の集合とする．I 中の全ての外部入力の制御時刻はステップ 4 までに決定しているものとする．

C1: M_j の y への補助経路の始点となる外部入力が I のどの外部入力とも異なる．

C2: 条件 C1 を満たさないが，M_j の y への補助経路の始点となる外部入力の制御時刻が他の入力ポートに値を伝搬するのに必要な外部入力の制御時刻と異なる．

C3: 条件 C1，C2 を満たさないが，M_j の y への補助経路の始点となる外部入力の制御時刻を，補助経路上にステップ 4 で付加したテスト MUX またはバイパスレジスタを用いて他の入力ポートに値を伝搬するのに必要な外部入力の制御時刻と異なる制御時刻に変更できる．または，他の入力ポートに値を伝搬するのに必要な外部入力の制御時刻が，M' の出力ポートから M_j の x までの観測経路上にステップ 4 で付加したテスト MUX またはバイパスレジスタを用いて変更できる．

以上のいずれかを満たす場合は，補助経路を用いて，適切な時刻に補助経路に対する外部入力を制御できるように，補助経路および観測経路を決定する．以上のどの 3 つの条件も満たさない場合は，M_j に対して x-z 間のスルー機能を付加する．

図 3(a) に対する固定制御可検査データパスを図 4 に示す．図 3(c) において，mux1 の観測経路上に adder1 および mult.1 の非伝搬入力が存在する．mux1 の各入力ポートに対する各制御経路の始点は共に外部入力 PI1 である．また，PI1 から mux1 の y_5 および x_5 までの順序深度はそれぞれ，0 および 1 である．mux1 の出力ポートから adder1 の非伝搬入力 x_1 までの観測経路の順序深度は 0 である．adder1 の伝搬入力 y_1 に対する制御経路の始点は外部入力 PI1 であり，その制御経路上に TM1 が存在するの

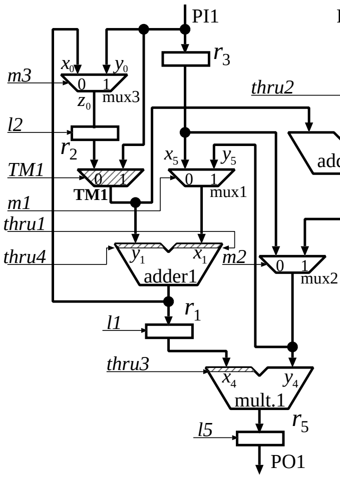
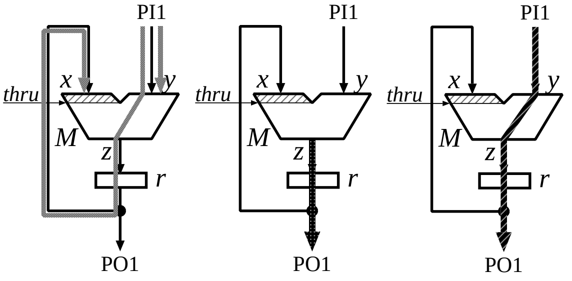


図 4: 固定制御可検査データパス



(a) M の制御経路

(b) M の観測経路

(c) M の副テスト用の経路

図 5: M の制御経路，観測経路，および副テスト用の経路

で，y_1 までの制御経路の順序深度は 0 である．従って，y_1 に対する制御経路を補助経路として利用できる条件を満たさないので，adder1 の非伝搬入力と出力ポート間のスルー機能 (thru4) を adder1 に付加する．PI1 から mult.1 の非伝搬入力 x_4 までの制御経路の順序深度は 1 および 2 である．mult.1 の伝搬入力 y_4 に対する制御経路上に TM1 が存在するので，y_4 までの制御経路の順序深度は 0 である．従って，条件 C2 を満たすので，y_4 に対しては補助経路 PI1 → TM1 → adder2 → mux2 → mult.1 を利用できる．

3.2.3　テストプラン生成法

主テストと副テスト　データパスの回路要素のテストには，主テストと副テストがある．主テストでは，外部入力から回路要素の入力ポートまでの制御経路を用いてテストベクトルを印加し，出力ポートから外部出力までの観測経路を用いてその出力応答を伝搬する．ここで，図 5 の 2 入力ポート x,y を持つ回路要素 M の制御経路について考える．M の入力ポート x の制御経路が図 5(a) のように M 自身を通る場合，主テストの際に，M の故障によって x に誤ったテストベクトルが印加される場合があるので，故障による誤りが M の出力ポート z に現れない可能性がある．そこで M の主テストを行う前に，y までの制御経路，y-z 間の経路および M の観測経路からなる経路上 (図 5(c) 参照) に，x へ印加するテストベクトルを伝搬させて，y-z 間の経路上でその値が変化しないことを外部出力で観測する．これを副テストといい，M の副テストで故障が検出されれば，テストは終了する．M の副テストで故障が検出されなければ，M の主テストを行う．副テストのテストプランは，主テストのテストプランと同様に生成できるので，以下では主テストのテストプラン生成について述べる．

時刻	*PI1*	*PI2*	*m1*	*m2*	*m3*	*TM1*	*l1*	*l2*	*l5*	*thru1*	*thru2*	*thru3*	*thru4*	*PO1*	フェーズ
1	a	X	X	X	X	X	X	X	X	X	X	X	X	-	正当化
2	b	X	0	X	X	1	1	X	X	T	X	X	T	-	テスト
3	1	X	X	1	X	1	X	X	1	X	1	X	X	-	伝搬
4	X	X												O	

a, b: 外部入力へのテストパタン印加タイミング
O: 外部出力での出力応答の観測タイミング
T: 制御ポートへのテストパタンの印加タイミング
X: ドントケア
-:　観測不要

図 6: テストプラン例 (図 4:adder1)

テストプラン　3.2.2 節で提案したテスト容易化設計では，与えられたデータパス DP にテスト用の回路要素を付加したことにより，DP 中の組合せ回路要素 M に対して，M をテストするための制御経路，観測経路および補助経路を用いて，任意のテストベクトルの外部入力からの印加およびその出力応答の外部出力からの観測ができることを保証した．これらの経路上に存在する回路要素に適切な制御信号を印加することで，それらの経路は活性化できる．従って，M のテストに対して，M に対する全ての制御経路および補助経路は，制御ベクトル C をそれらの経路上に存在する回路要素の制御ポートへ印加することで活性化できる．同様に，M に対する観測経路

と全ての補助経路は，制御ベクトル O をそれらの経路上に存在する回路要素の制御ポートへ与えることで活性化できる．

はじめに，M に対するテストプランの正当化フェーズおよび観測フェーズにおける制御ベクトル系列を考える．正当化フェーズにおける制御ベクトル系列は制御ベクトル C のみで構成できる．観測フェーズにおける制御ベクトル系列は制御ベクトル O のみで構成できる．正当化フェーズおよび観測フェーズでの制御ベクトル系列長を考える．ここで，d_c および d_o をそれぞれ，制御経路および観測経路の最大順序深度とする．また，d_s を観測経路と補助経路を連接してできる経路の最大順序深度とする．正当化フェーズおよび観測フェーズにおける制御ベクトル系列長はそれぞれ，$max(d_c, d_s - d_o)$ および d_o である．

次に，M に対するテストプランのテストフェーズにおける制御ベクトル T を考える．以下の経路は制御ベクトル T をそれらの経路上に存在する回路要素の制御ポートへ与えることで活性化できる．

- M の制御経路上に存在し，かつ M の直前のレジスタから M の入力ポートまでの部分制御経路
- M の出力ポートから，M の観測経路上に存在し，かつ M の直後のレジスタまでの部分観測経路
- 外部入力から M の入力ポートまでの順序深度 0 の制御経路
- M の出力ポートから外部出力までの順序深度 0 の観測経路
- 観測経路上に存在する 2 入力演算モジュールの伝搬入力までの補助経路

図 4 の adder1 に対するテストプランを図 6 に示す．テストプランは，3 つの制御ベクトル C，T および O で構成できる．テストプラン長は，$max(d_c, d_s - d_o) + d_o + 1$ である．制御ポートに対応するテストベクトルについては次節で述べる．

表 2: テストコントローラの制御

t_1	*mode*		動作	
	t_3	t_4		
0	x	x	通常動作	
1	0	0	TVR, TPR ⟵ PI	リセット
	0	1	$Z \longleftarrow \mathrm{TP_J}(\mathit{TPR})$	正当化
	1	0	$Z \longleftarrow \mathrm{TP_T}(\mathit{TPR}, \mathit{TVR})$	テスト
	1	1	$Z \longleftarrow \mathrm{TP_P}(\mathit{TPR})$	伝搬

$\mathrm{TP_J}$: 正当化フェーズの制御ベクトル　t_i: テストピン　X: ドントケア
$\mathrm{TP_T}$: テストフェーズの制御ベクトル　*mode* : モード切替信号
$\mathrm{TP_P}$: 伝搬フェーズの制御ベクトル　Z : 制御信号線

3.3 テストコントローラ

テストプランはデータパス中の全ての組合せ回路要素に対して生成される．テストプランのデータパスへの供給は，制御信号線を用いて行う．一般にコントローラの外部入力のビット幅は，制御信号線のビット幅より小さいので，コントローラの外部入力から直接テストプランを供給することはできない．そのため回路内部でテストプランを生成することを考える．RT レベル回路において，全てのテストプランはテストコントローラによって生成する (図 2)．テストコントローラはテストプラン生成回路 $\mathcal{TPG}$，テストベクトルレジスタ (**TVR**) および対象テストプランレジスタ (**TPR**) から成る．TPR は，テストプランに対応する番号を格納する．TPR のビット幅は，データパス中に存在する組合せ回路要素のテストプランの総和を m(副テストプランを含む) とすると，$\lceil log_2 m \rceil$ である．$\mathcal{TPG}$ は，TPR に設定されたテストプラン番号に対応するテストプランを生成する．$\mathcal{TPG}$ はテスト対象となる組合せ回路要素 M が制御ポートを持つ場合，テストフェーズにおいて，TVR にあらかじめ格納されたテストベクトルを制御ポートに対して印加する．TVR のビット幅は，テスト対象となる組合せ回路要素に接続される制御信号線の最大ビット幅である．

テストコントローラの制御を表 2 に示す．TPR および TVR は t_3 および t_4 を制御することで，値をロードする．表 2 において，$mode(t_3,t_4)$ が (0,0) の場合，TPR および TVR に対してそれぞれ，テストプラン番号および制御入力に対するテストベクトルをロードする．$mode(t_3,t_4)$ が (0,1) の場合は TPR

に格納されたテストプラン番号に対応するテストプランの正当化フェーズの制御ベクトル，(1,0) の場合はテストフェーズの制御ベクトル，(1,1) の場合は伝搬フェーズの制御ベクトルを出力する．また，t_1 が 0 の場合，制御信号線には回路の通常動作を実行するための制御ベクトルを 3.2.2 節で付加した回路の制御ポートに印加する．

$\mathcal{TPG}$ の $mode(t_3,t_4)$ は実動作速度で制御可能なので，データパスの各組合せ回路要素に対して，実動作速度でのテスト実行が可能である．

3.4　データパスの制約緩和

2 章で述べたデータパスの 4 つの制約のいずれかを満たさないデータパスが与えられた場合，以下に述べる処理を行うことにより，固定制御可検査テスト容易化設計法をそのデータパスに適用できる．

- データポートのビット幅が異なる場合
  演算器 M(入力 n ビット，出力 m ビット) で $n \neq m$ のとき，M の出力ポートの $|n-m|$ ビットの信号線上にテスト MUX を付加する．$n > m$ のとき，M の伝搬入力に接続された $n-m$ ビットの信号線と付加したテスト MUX を接続する．$n < m$ のとき，新たに外部入力から の $m-n$ ビットの信号線を付加し，その信号線と付加したテスト MUX を接続する．

- $n(n > 2)$ 個の入力ポートを持つ回路要素が存在する場合
  はじめに回路要素 M を，2 入力の組合せ回路要素を $(n-1)$ 個直列に接続したもので置き換え，そのデータパスに対して，M 以外の回路要素のテストプランが存在するように 3.2.2 節のテスト容易化設計法を適用する．次に，M の n 本の制御経路を用いて，n 個の入力ポートを同時に制御できるようにテスト容易化設計 (例えば，n 本の制御経路全ての順序深度が異なるように設計変更) を行えば，M のテストプランの存在が保証できる．

- A3 を満たさない回路要素が存在する場合
  回路要素 M の入力ポートへ外部入力からの経路がない場合，M の入力ポートの直前にテスト MUX を付加し，その入力ポートを直接外部入力から制御可能にする．M の出力ポートから外部出力への経路がない場合，外部出力の直前にテスト MUX を付加し，その出力ポートを直接外部出力で観測可能にする．

- A4 を満たさない回路要素が存在する場合
  回路要素 M の制御入力がコントローラから直接制御できない場合，制御入力の直前にテスト MUX を付加し，その制御入力を直接制御可能にする．M のステータス出力がコントローラへ接続していない場合，ステータス出力を観測するための信号線およびテスト MUX を付加し，そのステータス出力を直接外部出力で観測可能にする．

4　実験結果

本章では，ベンチマーク回路および実設計回路を用いた実験により，完全スキャン設計法，文献 [9](以後，強可検査法と呼ぶ) および提案手法を，テスト容易化設計に伴う面積オーバヘッド，テスト生成時間，テスト実行時間および故障検出効率について比較した結果を述べる．実験に使用した RT レベルベンチマーク回路は，GCD と PAULIN[4]，LWF と JWF[10] である．また実設計回路として，RISC プロセッサ (以下，RISC と呼ぶ) を使用した．これらの回路の特性を表 3 に示す．#PI，#PO はコントローラおよびデータパスそれぞれの外部入力数および外部出力数を表す．コントローラに関して，#State，#Status および#Control はそれぞれ，状態数，ステータス入力数および制御出力数を表す．データパスに関して｜bit｜，#Reg および#Mod. はそれぞれ，ビット幅，レジスタ数および演算モジュール数を表す．

論理合成ツールには AutoLogicII(Mentor Graphics)，テスト生成ツールには TestGen(Synopsys)，計算機には SUN Ultra60 を使用した．

各手法のテスト生成結果を表 4 に示す．テスト生成時間は，強可検査法および提案手法では，完全ス

表 3: コントローラ/データパス回路の特性

回路	面積 $(gate)$	コントローラ						データパス					
		#PI	#PO	#State	#Status	#Control	面積 $(gate)$	#PI	#PO	|bit|	#Reg.	#Mod.	面積 $(gate)$
GCD	1524.50	1	1	4	3	7	169.40	32	16	16	3	1	1350.90
JWF	6875.40	1	0	8	0	38	199.50	80	80	16	14	3	6671.70
LWF	1986.20	1	0	4	0	8	57.70	32	32	16	5	3	1924.30
PAULIN	24965.60	1	0	6	0	16	123.50	64	64	32	7	4	24833.70
RISC	62287.60	1	2	11	54	62	3986.90	32	96	32	40	4	58157.90

表 4: テスト生成結果

回路	テスト生成時間 (秒)				テスト実行時間 (サイクル)				故障検出効率 (%)			
	DFT 前	完全スキャン	強可検査法	提案手法	DFT 前	完全スキャン	強可検査法	提案手法	DFT 前	完全スキャン	強可検査法	提案手法
GCD	18055.53	171.51	0.69	0.69	9	6629	504	504	4.92	100.00	100.00	100.00
JWF	2348.24	2.88	0.37	0.27	488	20519	1497	1621	98.14	100.00	100.00	100.00
LWF	171.68	0.47	0.27	0.27	322	4066	517	443	99.64	100.00	100.00	100.00
PAULIN	20362.55	4.68	2.11	2.20	283	16187	2193	2172	97.01	100.00	100.00	100.00
RISC	288102.05	51740.92	71.50	72.29	4298	1006154	9674	7768	62.31	99.97	99.99	99.99

表 5: 面積オーバヘッド

回路	面積オーバヘッド (%)													外部ピンオーバヘッド (#)		
	完全スキャン	強可検査法						提案手法						完全スキャン	強可検査法	提案手法
			C	DP	$\mathcal{TPG}$	TMR,TPR	MUX		C	DP	$\mathcal{TPG}$	TMR,TPR	MUX			
GCD	26.6	39.7	1.1	2.6	19.1	4.1	12.8	32.8	1.1	2.6	12.2	4.1	12.8	3	5	5
JWF	26.7	37.1	0.4	5.2	20.5	1.4	9.7	41.9	0.4	9.3	20.7	1.4	10.0	3	5	5
LWF	33.4	48.6	0.8	18.1	17.2	3.9	8.6	44.6	0.8	18.1	13.2	3.9	8.6	3	5	5
PAULIN	7.4	8.1	0.2	1.2	4.9	0.4	1.4	7.1	0.2	2.5	2.6	0.4	1.4	3	5	5
RISC	16.7	27.3	0.1	10.9	12.3	0.2	3.6	21.0	0.1	9.6	7.4	0.2	3.7	3	6	6

キャン設計法に比べて短い．特に RISC では，提案手法は完全スキャン設計に比べて約 1/700 と大幅に短縮している．強可検査法および提案手法では，組合せ回路要素単体に対してテスト生成を行うのに対して，完全スキャン設計法では組合せ回路全体に対してテスト生成を行う．従って，強可検査法および提案手法は，より大規模な回路に対するテスト生成時間を完全スキャン設計法よりも大幅に短縮できると期待できる．

完全スキャン設計法のテスト実行時間は「テストベクトル数」×(「FF 数」+1)+「FF 数」とした．強可検査および提案手法のテスト実行時間は，(全ての組合せ回路要素の「テストベクトル数」×「テストプラン長」の和)+(コントローラのテスト実行時間)とした．強可検査法および提案手法のテスト実行時間は，ベンチマーク回路では完全スキャン設計に比べて約 1/10 と短く，RISC では，約 1/130 と大幅に短縮している．

故障検出効率は，強可検査法および提案手法では，ベンチマーク回路では全て 100%を達成している．RISC に対する故障検出効率は 99.99%である．これは，TestGen が RISC に含まれる乗算器に対してテストベクトルを一部生成できなかったためである．しかし，強可検査法や提案手法では，全ての組合せ回路要素に対して完全故障検出効率を達成できれば，データパス全体に対して完全故障検出効率を達成できることが保証されている．

テスト容易化設計に伴う付加回路による面積オーバヘッドおよび外部ピンオーバヘッドを表 5 に示す．本論文では，テストプラン生成回路 $\mathcal{TPG}$ の回路面積を強可検査法より縮小するために，データパスの固定制御可検査性を提案した．$\mathcal{TPG}$ の回路面積について，提案手法は JWF 以外は強可検査法に比べて縮小できた．特に RISC では，$\mathcal{TPG}$ の回路面積が強可検査法では約 1 万ゲートであったが，提案手法では約 5 千ゲートまで縮小できた．提案手法は $\mathcal{TPG}$ の回路面積を縮小するために，強可検査法と比べてデータパスに対する要求が強くなり，データパスの

面積オーバヘッドが大きくなると考えられるが，実験で使用したほとんどの回路で提案手法は強可検査テスト容易化設計法と同等の面積オーバヘッドであることを示している．特にRISCにおいては，提案手法のデータパスの面積オーバヘッドは強可検査法より小さくなった．回路全体の面積オーバヘッドについて，提案手法は強可検査法に比べてJWF以外は小さいが，完全スキャン設計法に比べて大きい．しかし，提案手法と完全スキャン設計法の差は小さい．JWFは，ほとんどのレジスタがホールド機能を持ち，強可検査テスト容易化設計ではスルー機能が3個だけ付加されるのに対し，提案手法ではテストMUXが2個とスルー機能が4個付加される．従って，新たに付加した制御信号線数が，提案手法は強可検査法より3本多い．また強可検査データパス中の各組合せ回路要素に対するほとんどのテストプランが3つの制御ベクトルで構成できる．従って，提案手法によるJWFの面積オーバヘッドは強可検査法より大きいが，その差は小さい．

外部ピンオーバヘッドについて，提案手法は強可検査法と同じであるが，完全スキャン設計法より多い．強可検査法と提案手法において，RISCにおける外部ピンオーバヘッドは，他の外部ピンオーバヘッドより多い．これは，データパスの外部出力のビット幅よりも制御信号線のビット幅が大きいので，データパスの外部出力でMUXが2段必要となったためである．

次に，テスト容易化設計による回路の性質劣化について考える．完全スキャン設計法では，MUXが全てのレジスタに付加される．提案手法では，レジスタ間の経路上にマスク素子とMUXが両方付加される場合がある．そのため，提案手法適用後の回路の性質劣化は完全スキャン設計法に比べて大きいが，実験で用いたJWF，PAULINおよびRISCそれぞれにおいて1箇所のみであった．本論文ではクリティカルパスを考慮していないが，クリティカルパス上にはマスク素子とMUXを両方付加しない，または何も付加しないなどの処理を考慮したテスト容易化設計を行うことで，回路の性質劣化を抑えることができると考えられる．

5　むすび

本論文では，データパスの新しい回路構造の性質として固定制御可検査性を定義し，固定制御可検査性に基づくテスト容易化設計法，およびテストプラン生成法を提案した．データパスのテスト生成法は，階層テスト生成法に基づいており，各組合せ回路要素単体に対して組合せ回路用のテスト生成アルゴリズムを用いてテスト生成を行うので，短いテスト生成時間で完全故障検出効率を達成できる．また固定制御可検査性は，データパス中の各組合せ回路要素のテストプランの制御ベクトル数は高々3個であるため，提案手法ではテストプラン生成回路$\mathcal{TPG}$を組合せ回路で構成できる．従って，提案手法では$\mathcal{TPG}$の面積は文献[9]に比べて小さくできる．さらに本論文では，データパスに対する制約を文献[9]に比べて緩和している．従って，提案手法は文献[9]に比べて適用範囲が広い．ベンチマーク回路および実設計回路を用いた実験では，提案手法は文献[9]と比較して，テスト生成時間，テスト実行時間の点で同等で，面積オーバヘッドは小さいことを示した．

謝辞　本研究に際し，多くの貴重な意見を頂いた奈良先端科学技術大学院大学の増澤利光助教授，井上美智子助手ならびに広島市立大学の井上智生助教授に深く感謝します．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)(課題番号09480054)の研究助成による．

参考文献

[1] H.Fujiwara, “Logic testing and design for testability”, The MIT press, Cambridge, 1985.

[2] M.Abramovici, M.A.Breuer and A.D. Friedman, “Digital Systems Testing and Testable Design,” Computer Science Press, New York, 1990.

[3] P.C.Maxwell, R.C.Aitken, V.Johansen and I.Chiang, “The effect of different test sets on quality level prediction: when is 80% better

than 90%?," Proc. of International Test Conference, pp.358-364, 1991.

[4] I.Ghosh, A.Raghunathan and N.K.Jha, "Design for hierarchical Testability of RTL circuits obtained by behavioral synthesis," Proc. of IEEE Transaction on Computer Aided Design, VOL.16, NO.9, pp.1001-1014, Sept. 1997.

[5] I.Ghosh, A.Raghunath and N.K.Jha, "Hierarchical test generation and design for testability methods for ASPP's and ASIP's," IEEE Trans. on Computer Aided Design, Vol.18, NO.3, pp.357-370, aMarch 1999.

[6] B.T.Murray and J.P.Hayes, "Hierarchical test generation using pre computed tests for modules," IEEE Trans.on Computer Aided Design, VOL.9, NO.6, pp.594-603, June 1990.

[7] 大竹哲史, 増澤利光, 藤原秀雄, "完全故障検出効率を保証するコントローラの非スキャンテスト容易化設計法," 電子情報通信学会論文誌, Vol.J81-D-I, No.12, pp.1259-1270, Dec. 1998.

[8] 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RTLデータパスの非スキャンテスト容易化設計法," 電子情報通信学会論文誌, Vol.J82-D-I, No.7, pp.843-851, July 1999.

[9] S.Ohtake, H.Wada, T.Masuzawa and H. Fujiwara, "A non-scan DFT method at register-transfer level to achieve complete fault efficiency," Proc. of Asia and South Pacific Design Automation Conference 2000, pp.599-604, 2000.

[10] M.Inoue, K.Noda, T.Higashimura, T. Masuzawa and H.Fujiwara, "High-level synthesis for weakly testable data paths," IEICE Trans. on Information and Systems, Vol.E81-D, No.7, pp.645-653, July 1998.