

NAIST-IS-DT0061020

博士論文

階層テスト生成に基づくレジスタ転送レベル回路の
非スキャンテスト容易化設計法に関する研究

永井 慎太郎

2003年3月1日

奈良先端科学技術大学院大学
情報科学研究科 情報処理学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に
博士(工学)授与の要件として提出した博士論文である。

永井 慎太郎

審査委員： 藤原 秀雄 教授
渡邊 勝正 教授
井上 美智子 助教授

階層テスト生成に基づくレジスタ転送レベル回路の 非スキャンテスト容易化設計法に関する研究*

永井 慎太郎

内容梗概

近年の半導体技術の進歩により，VLSIの大規模化，高集積化，高性能化が進み，VLSIのテストは困難な問題となっており，テストの費用の削減およびテストの質の向上が求められている．テストは製造されたVLSIに故障があるかどうかを調べることをいい，テスト生成とテスト実行からなる．テスト生成では故障を検出するための入力系列(テスト系列)を求め，テスト実行ではテスト系列に対する出力応答を期待値と比較し，回路に故障があるかどうかを調べる．テストの費用は，テスト生成時間やテスト実行時間で評価できる．テストの質は故障検出効率などで評価できる．故障検出効率とは，回路中のテスト生成の対象となるすべての故障数に対する，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で検出可能な故障数と冗長と判定された故障数の和の割合をいう．

テストの費用の削減およびテストの質の向上を達成するために，回路をテスト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．ゲートレベルでのテスト容易化設計では，回路の大規模化に伴い，扱う回路要素数が多くなるという問題や，論理合成ツールに入力した回路設計者の設計制約をテスト容易化により満たさなくなる可能性がある．そのため，近年，レジスタ転送レベル回路を対象としたテスト容易化設計が行われている．レジスタ転送レベル回路は一般に，データの演算処理を行うデータパスとそれを制御するコントローラで構成される．データパスは演算モジュールやレジスタなどの回路要素と信号線で記述され，コントローラは状態遷移図で記述される．

*奈良先端科学技術大学院大学 情報科学研究科 情報処理学専攻 博士論文，NAIST-IS-DT0061020, 2003年3月1日.

代表的なテスト容易化設計法としてスキャン方式がある。スキャン方式は回路中のレジスタをスキャンレジスタに置き換え，回路をスキャンモードに切り替えてスキャンシフト動作を実行することでレジスタの値を外部から直接制御および観測を可能とするテスト容易化設計法である。しかしスキャン方式では，面積オーバーヘッドが大きくなる，テスト生成時間やテスト実行時間が長くなる，回路の実動作速度でのテスト実行が困難であるといった問題がある。

そこでこれらの問題を解消するために，非スキャン方式が提案されている。非スキャン方式は回路の通常動作時に用いるデータ転送経路上にテスト系列の印加，出力応答の伝搬ができるように回路を設計変更するテスト容易化設計法である。したがって，非スキャン方式では，スキャン方式に比べてテスト実行時間が短くなり，回路の実動作速度でのテスト実行が可能となる利点を持つ。

また，テスト生成時間を削減するための手法として，階層テスト生成法が提案されている。階層テスト生成法は次の 2 段階からなる。はじめにゲートレベルにおいて演算モジュールやマルチプレクサなどの組合せ回路要素単体に対してテスト生成を行う。次にレジスタ転送レベルにおいてテストプランを生成する。テストプランは，テスト系列および出力応答を伝搬できるための制御ベクトル系列をいい，一般にテストプランでの制御ベクトルは時刻ごとに変化する。階層テスト生成法では，組合せ回路要素単体に対して短いテスト生成時間で 100%の故障検出効率を達成できる利点を持つ。しかし一般に，各組合せ回路要素に対してテストプランを生成するのは困難であるため，高い故障検出効率を保証できない。

そこで本論文では，はじめに階層テスト生成が容易な回路構造の性質であるデータパスの固定制御可検査性を導入し，固定制御可検査性に基づくレジスタ転送レベルデータパスの非スキャンテスト容易化設計法およびテストアーキテクチャを提案する。提案手法では，データパスの組合せ回路で構成される各モジュールに対するテストプランをデータパスへ供給するためのテストプラン生成回路を付加する。固定制御可検査性を満たすデータパスでは，各テストプランでの制御ベクトル系列をたかだか 3 個の制御ベクトルで構成できるため，テストプラン生成回路を組合せ回路で実現できる。また提案手法では，100%の故障検出効率を達成でき，テスト生成時間やテスト実行時間がスキャン設計法に比べて短い利点を持つ。また実動作速度でのテスト実行が可能である。さらに提案手法の有効性をベ

ンチマーク回路および実設計回路を用いた実験によって評価する．

次に，データパス中の各組合せ回路要素に対するテストプランを，オリジナルコントローラが出力する制御ベクトル系列を用いて構成できるためのデータフロー依存型回路の非スキャンテスト容易化設計法を提案する．この手法では，固定制御可検査性に基づく非スキャンテスト容易化設計法と同じ利点を持つ．さらにこの手法では，テストプランをオリジナルコントローラからデータパスへ伝搬するので，固定制御可検査性に基づく非スキャンテスト容易化設計法に比べて面積オーバーヘッドを削減できる．

キーワード

レジスタ転送レベル，テスト容易化設計，固定制御可検査性，階層テスト生成，データフローグラフ

Studies on Non-Scan DFT Methods for Register Transfer Level Circuits Based on Hierarchical Test Generation*

Shintaro Nagai

Abstract

With the advance in semiconductor technology, since the complexity of VLSI designs is growing, the cost of *testing* is increasing. Testing of the circuit consists of two stages: *test generation* and *test application*. The former denotes to generate a *test sequence* which is an input sequence to detect faults in the circuit. The later denotes to apply the test sequence to the circuit. The cost of testing is estimated by *test generation time* and *test application time*. The quality of testing is estimated by *fault efficiency* of test sequences. The fault efficiency denotes the ratio of the sum of the number of faults detected and the number of faults proved redundant to the total number of target faults under test generation. In order to reduce test generation time and test application time and to enhance the fault efficiency, *design for testability* (DFT) techniques which makes testing of the circuit easy have been proposed.

DFT methods at gate level face common problems. Since the number of the circuit elements at gate level is increasing, the computational complexity of DFT becomes huge. Also, the design constraints inputted into a logic synthesis tool may not be satisfied by adding DFT elements to the original circuit. Recently, in order to solve these problems, DFT methods at *register transfer level* are proposed.

*Doctor's Thesis, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DT0061020, March 1, 2003.

A register transfer level circuit generally consists of two sub-circuits: a data path and a controller. The data path which processes data is represented by hardware elements (e.g. registers, operational modules and multiplexers) and signal lines connecting them. The controller which regulates the datapath is represented by a state transition graph.

The most commonly used DFT techniques are scan based approaches. In scan techniques, some registers in a circuit are replaced into scan registers which can be controlled/observed from outside of the circuit. However, scan techniques have disadvantages such as high hardware overhead, increased test application time. Moreover, in scan techniques, *at-speed testing* is difficult.

To avoid these disadvantages of scan based approaches, several non-scan based approaches have been investigated. In non-scan techniques, values of all registers are justified/observed through data transmission paths used for normal operation. Therefore, test application time of non-scan techniques is shorter than that of scan techniques. Also, in non-scan techniques, *at-speed testing* can be performed.

To reduce test generation time, *hierarchical test generation* method have been proposed. Hierarchical test generation method consists of two steps. Firstly, test patterns are generated for each combinational module in the data path using a combinational automatic test pattern generation tool at gate level. Secondly, *test plans* are generated for each combinational module at register transfer level. A test plan is a control vector sequence that can justify any value to the inputs of the combinational module under test from some primary inputs, and that can propagate any value from the output of the combinational module to a primary output. However, there does not always exist a test plan for each combinational module.

In this dissertation, we first propose a new property of circuit structure of data paths called *fixed-control testability*. We also propose a non-scan DFT method for data paths which makes a data path fixed-control testable and a test architecture for a whole circuit consisting of both a controller and a data path. In this method, a test plan generator which can generate test plans for the data path is embedded.

If a data path is fixed-control testable, hierarchical test generation can be applied and the test plan for each combinational module can be composed of at most three control vectors. Therefore, the test plan generator can be designed as a combinational logic. This method can achieve 100% fault efficiency, and allows at-speed testing. Furthermore, from our experimental results, test generation time and test application time of this method are shorter than those of the full-scan design.

Next, we propose a non-scan DFT method for hierarchical testability of a data path using control vector sequences generated by an original controller without adding any function to it. This method also guarantees the applicability of hierarchical test generation for each combinational module. In order to generate a control vector sequence for a test plan from the original controller, we extract an extended test control data flow graph from the data path and the controller. Also, this method has same advantages as our DFT method based on fixed-control testability. Furthermore, from our experimental results, the area overhead of this method is smaller than that of our DFT method based on fixed-control testability since the area overhead for the test plan generator to supply such test plans to the data path is small.

Keywords:

register transfer level, design-for-testability, fixed-control testability, hierarchical test generation, data flow graph

目次

1 序章	1
1.1. VLSI のテスト	1
1.2. テスト容易化設計	2
1.2.1 レジスタ転送レベル回路	2
1.2.2 スキャン設計法	3
1.2.3 非スキャン設計法	4
1.3. 提案手法	6
2 固定制御可検査性に基づくレジスタ転送レベル回路の非スキャンテスト容易化設計法	8
2.1. はじめに	8
2.2. 諸定義	8
2.2.1 対象回路	8
2.2.2 階層テスト生成法	10
2.2.3 データパスの強可検査性	10
2.3. テスト容易化設計	11
2.3.1 概要	11
2.3.2 データパスの固定制御可検査性	12
2.3.3 データパスのテスト容易化設計法	13
2.3.4 テストプラン生成法	25
2.3.5 テストコントローラ	28
2.4. データパスの制約緩和	29
2.5. レジスタファイルを含むデータパスのテスト容易化設計	30
2.5.1 レジスタファイル	30
2.5.2 レジスタファイルの設計変更	30

2.5.3	レジスタファイル以外のデータパス部分に対する DFT 法	32
2.5.4	レジスタファイル以外のデータパス部分に対するテストプラン生成法	32
2.5.5	レジスタファイルのテストおよびテストプラン	34
2.6.	実験結果	36
2.7.	むすび	45
3	レジスタ転送レベルでのデータフロー依存型回路の階層テスト容易化設計法	47
3.1.	はじめに	47
3.2.	諸定義	47
3.2.1	データフロー依存型回路	47
3.2.2	拡張データフローグラフ	48
3.3.	テスト容易化設計	51
3.3.1	概要	51
3.3.2	レジスタ初期化 MUX 付加	54
3.3.3	ETCDF 生成手続き	55
3.3.4	リセット MUX 可制御化	56
3.3.5	可検査性解析	57
3.3.6	データパスのテスト容易化設計	60
3.4.	従来法との比較	64
3.5.	実験結果	65
3.6.	むすび	69
4	結論	71
	謝辞	75
	参考文献	77

目 次

2.1	RT レベル回路	9
2.2	テストコントローラを含むレジスタ転送レベル回路	12
2.3	データパス例	14
2.4	(a) 制御林, (b) 観測林	15
2.5	スルー機能の実現例	16
2.6	図 2.4(a) に対するステップ 3 実行例	17
2.7	テスト MUX(TM) の付加例	20
2.8	バイパスレジスタ (BR) の付加例	21
2.9	図 2.3 に対するステップ 4 実行例	22
2.10	固定制御可検査データパス	24
2.11	M の制御経路, 観測経路, 副テスト用の経路	25
2.12	テストプラン例 (図 4:adder1)	26
2.13	1 入力 2 出力レジスタファイル	31
2.14	拡張後のレジスタファイル	32
2.15	拡張後のレジスタファイルの構造モデル疑似変換	33
2.16	レジスタファイル以外のデータパス部に対するテスト	34
2.17	GCD データパス	38
2.18	固定制御可検査 GCD データパス	38
2.19	Paulin データパス	39
2.20	固定制御可検査 Paulin データパス	39
2.21	LWF データパス	40
2.22	固定制御可検査 LWF データパス	40
2.23	JWF データパス (制御信号線は省略)	41
2.24	固定制御可検査 JWF データパス (制御信号線は省略)	41
3.1	レジスタ転送レベルにおけるデータフロー依存型回路	48

3.2	<i>Tseng</i> データパス	49
3.3	<i>Tseng</i> コントローラ状態遷移表	50
3.4	拡張データフローグラフ (<i>Tseng</i>)	51
3.5	テスト容易化設計手続き	52
3.6	テストプランの供給方法	53
3.7	レジスタ初期化 MUX 付加	54
3.8	リセット MUX 可制御化	57

表 目 次

2.1	テストピンの制御	13
2.2	テストコントローラの制御	28
2.3	回路全体の面積	37
2.4	回路特性 (コントローラ)	37
2.5	回路特性 (データパス)	37
2.6	テスト生成時間 (秒)	43
2.7	テスト実行時間 (クロックサイクル)	43
2.8	故障検出効率 (%)	43
2.9	面積オーバーヘッド (%)	44
2.10	外部ピンオーバーヘッド (#)	44
3.1	テストプラン生成対象頂点の入出力辺に対する尺度	58
3.2	尺度変換表	59
3.3	回路全体の面積	66
3.4	回路特性 (コントローラ)	66
3.5	回路特性 (データパス)	66
3.6	面積オーバーヘッド (%)	67
3.7	外部ピンオーバーヘッド (#)	67
3.8	テスト生成時間 (秒)	68
3.9	テスト実行時間 (クロックサイクル)	68

関連発表一覧

- 学術論文誌

1. 永井慎太郎, 和田弘樹, 大竹哲史, 藤原秀雄, ”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”, 電子情報通信学会論文誌 (DI), Vol.J84-D-I, No.5, pp.454-465, May 2001 .
2. 永井慎太郎, 大竹哲史, 藤原秀雄, ”レジスタ転送レベルでのデータフロー依存型回路の階層テスト容易化設計法”, 情報処理学会論文誌, Vol.43, No.5, pp.1278-1289, May 2002 .

- 国際会議 (査読付き)

1. S. Ohtake, S. Nagai, H. Wada and H. Fujiwara, ”A DFT method at RTL based on fixed-control testability to achieve 100% fault efficiency”, Workshop on RTL ATPG & DFT 2000, Sep. 2000 .
2. S. Ohtake, S. Nagai, H. Wada and H. Fujiwara, ”A DFT method for RTL circuits to achieve complete fault efficiency based on fixed-control testability”, Asia and South Pacific Design Automation Conference 2001, pp.331-334, Jan. 2001 .
3. S. Nagai, S. Ohtake and H. Fujiwara, ”A method of design for hierarchical testability for RTL data paths using extended data flow graphs”, Workshop on RTL ATPG & DFT 2001, Nov. 2001 .

- 研究会報告

1. 永井慎太郎, 和田弘樹, 大竹哲史, 藤原秀雄, ”固定制御可検査性に基づく RTL データパスの非スキャンテスト容易化設計法”, 第 41 回 FTC 研究会, July 1999 .

2. 永井慎太郎, 和田弘樹, 大竹哲史, 藤原秀雄, ”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”, 信学技報 (VLD99-101), VLD 研究会, pp.29-36, Jan. 2000 .
3. 永井慎太郎, 大竹哲史, 藤原秀雄, ”拡張データフローグラフを用いた RT レベルデータパスの階層テスト容易化設計法”, 第 45 回 FTC 研究会, July 2001 .
4. 永井慎太郎, 大竹哲史, 藤原秀雄, ”拡張データフローグラフを用いた RT レベルデータパスの階層テスト容易化設計法”, 信学技報 (VLD2001-106, ICD2001-151, FTS2001-53), FTS 研究会, pp.103-108, Nov. 2001 .

第 1 章

序章

1.1. VLSI のテスト

VLSI は論理回路でモデル化でき，一般に回路中の故障は，回路の論理機能が故障によって別の論理機能に変化する論理故障でモデル化できる．代表的な論理故障モデルとして，回路内部の信号線の値が 0 または 1 に縮退する縮退故障モデルがある．回路には一般に，同時に複数の縮退故障が存在する場合（多重縮退故障モデル）があるが，同時に 1 つの縮退故障しか存在しないと仮定した単一縮退故障モデルがある [1]．本論文では，単一縮退故障モデルのみを対象とする．

回路に故障があるかどうかを調べることをテストという．テストは，テスト生成とテスト実行から成る．テスト生成とは，故障を検出するための入力系列を求めることをいい，その入力系列をテスト系列（テストベクトル集合）という．ある入力系列に対する出力応答が故障により期待値と異なるとき，その故障を検出したという．テスト実行とは，テスト系列に対する出力応答を期待値と比較し，回路に故障があるかどうかを調べることをいう．

近年の VLSI の大規模化，高集積化に伴い，VLSI のテストは困難な問題となっており，テストの費用の削減およびテストの質の向上が目標となっている．テストの費用はテスト生成時間やテスト実行時間で評価できる．また，テストの質は故障検出率や故障検出効率で評価できる．故障検出率とは，回路中のテスト生成の対象となるすべての故障数に対する，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で検出可能な故障数の割合をいう．故障検出効率とは，回路中のテスト生成の対象となるすべての故障数に対する，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で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．故障検出効率が 100% の場合を完全故障検出効率という．ここで冗長とは，テスト生成アルゴリズムによってテスト系列が存

在しないと判定された場合をいい，そのような故障を冗長故障という．

回路は，記憶素子（フリップフロップ，FF）を含まない組合せ回路と，FF を含む順序回路に分類される．組合せ回路に対しては，高い故障検出効率を達成できるテスト系列を実用的なテスト生成時間で生成できる，効率の良いテスト生成アルゴリズムが提案されている [2]-[3]．これに対して，順序回路には FF による内部状態が存在し，テスト生成アルゴリズムの探索空間はその内部状態数に依存する．したがって，順序回路のテスト生成には膨大な時間がかかり，高い故障検出効率を達成できるテスト系列を生成するのは困難である．そのため，回路設計の段階から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が行われている．

1.2. テスト容易化設計

現在も，ゲートレベル回路に対するテスト容易化設計が提案されているが，回路の大規模化に伴い，扱う回路要素数が膨大となり，テスト容易化設計時の計算複雑度が大きくなるという問題や，テスト容易化のために付加した回路によって論理合成時での回路設計者の設計制約を満たさなくなる可能性があるという問題が指摘されてきた．そこで近年，レジスタ転送レベル回路を対象としたテスト容易化設計が提案されている．その利点は，レジスタ転送レベルにおけるテスト容易化設計で扱う回路要素数がゲートレベルでの回路に比べて少ないので，テスト容易化設計時の計算複雑度が小さいことである．以下では，レジスタ転送レベル回路について概説し，レジスタ転送レベル回路を対象とした代表的なテスト容易化設計法について紹介する．

1.2.1 レジスタ転送レベル回路

レジスタ転送レベル回路は一般に，データの演算処理を行うデータパスとそれを制御するコントローラで構成される．データパスは回路要素と回路要素同士間を接続する信号線で記述される．コントローラは状態遷移図で記述される．回路要素には，外部入力，外部出力，ホールド機能を持つレジスタ（FF の集合）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，

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つ。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に分類される。制御ポートは、回路要素を制御するための信号をコントローラから入力するためのポートである。観測ポートは、回路要素のステータスをコントローラへ出力するためのポートである。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、各回路要素の入力ポートおよび出力ポートに接続され、2つの回路要素を接続する。制御信号線は、コントローラと回路要素の制御ポートを接続し、回路要素の制御を行うための信号をコントローラから供給するための信号線である。ステータス信号線は、コントローラと回路要素のステータスポートを接続し、回路要素のステータスをコントローラへ供給するための信号線である。以下では、MUX、演算モジュールおよび観測モジュールなどの組合せ回路で構成される回路要素を組合せ回路要素と呼ぶ。

1.2.2 スキャン設計法

代表的なテスト容易化設計法として、スキャン設計法がある。スキャン設計法は、回路中のレジスタをスキャンレジスタに置き換えることで、そのレジスタをスキャンチェーンとよばれる1本または複数本のシフトレジスタとしても動作(スキャンシフト動作)できるように設計変更する手法である。スキャンレジスタは、レジスタにMUXを付加することで実現でき、そのMUXの制御を変えることによりスキャンシフト動作を行うことができる。回路にスキャン入力およびスキャン出力を付加し、回路をスキャンモードでスキャンシフト動作させることによって、スキャンレジスタの値を外部から直接制御および観測ができる。

すべてのレジスタをスキャンレジスタに置き換える手法を、完全スキャン設計法という。完全スキャン設計法では、テスト生成の際に回路中のすべてのレジスタを擬似的に外部入出力として見なすことができるので、テスト生成の対象回路が組合せ回路となる。したがって、完全スキャン設計法では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、一般に短いテスト生成時間で高い故障検出効率を達成できる。しかし、完全スキャン設計法には以下の問題点がある。

1. テスト容易化設計後の回路面積がレジスタ数に比例して大きくなる．
2. スキャンシフト動作を行うため，テスト実行時間が長くなる．
3. 回路の実動作速度でのテスト実行 (at-speed testing) が困難である．
4. テスト生成対象の回路規模の増大により，対象とする故障数が多くなり，テスト生成時間が長くなる．

完全スキャン設計法での面積オーバーヘッドの問題やテスト実行時間が長いという問題を解消するためのスキャン設計法として，一部のレジスタをスキャンレジスタに置き換える部分スキャン設計法が提案されている [4]-[8]．部分スキャン設計法では，スキャンレジスタに置き換えるレジスタの選択によって回路のテスト容易性が変化するので，その選択が部分スキャン設計法での主な課題となる．部分スキャン設計法として，初期化しにくい閉路上のレジスタのみをスキャンレジスタに置き換える手法が提案されている [4] が，テスト生成の対象が依然として順序回路であり，順序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しなければならないので，高い故障検出効率を達成するのは困難である．また，組合せ回路用のテスト生成アルゴリズムを用いて，短いテスト生成時間で高い故障検出効率を達成できる無閉路構造 [5]，平衡構造 [6]，内部平衡構造 [7, 8] に着目した手法が提案されている．しかし，これらの部分スキャン設計法では依然として，回路の実動作速度でのテスト実行 [9] が困難であり，スキャンシフト動作を行うためテスト実行時間は長くなる．

1.2.3 非スキャン設計法

非スキャン設計法は，データ信号線を用いて，回路中のレジスタの制御および観測を行うことができるように回路を設計変更する手法である．したがって，非スキャン設計された回路に対しては，回路の通常動作時のクロック速度 (実動作速度) でのテスト実行が可能となる．またテスト実行時間は，スキャン設計法に比べて短い．代表的な非スキャン設計法として，直交スキャン設計法 [10, 11]，Genesis [12]-[17]，強可検査法 [18]，鈴木らの手法 [24]，岡本らの手法 [25] がある．

直交スキャン設計法 [10, 11] は、レジスタ転送レベルデータパスを対象とした非スキャン設計法である。直交スキャン設計法では、テスト生成の対象が完全スキャン設計法と同じであり、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。また直交スキャン設計法では、データパス中のすべてのレジスタへ外部入力から値を伝搬し、すべてのレジスタから外部出力へ値を伝搬するためのデータ転送経路 (スキャンパス) を探索し、そのスキャンパスを用いてテスト系列および出力応答の伝搬を行う。しかし、直交スキャン設計法では、完全スキャン設計法と同様に、テスト生成対象の回路規模が大きくなれば、テスト生成時間が増大する問題がある。

テスト生成時間を短縮する手法として、階層テスト生成法 [19] が提案されている。階層テスト生成法は、次の 2 段階からなる。第 1 段階ではゲートレベルにおいて、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。第 2 段階ではレジスタ転送レベルにおいて、各組合せ回路要素に対してテストプランを生成する。テストプランは、テスト系列を外部入力からテスト対象となる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ごとに変化する。階層テスト生成に基づく非スキャン設計法として、Genesis、強可検査法が提案されている。

Genesis[12]-[17] は、データパス中の演算モジュールに対するテストプランをコントローラの通常動作時の出力系列 (以下では、コントローラの制御系列と呼ぶ) を用いて構成するための、レジスタ転送レベルデータパスを対象とした非スキャンテスト容易化設計法である。Genesis ではコントローラの制御系列を抽出するために、データフローグラフを生成し、そのデータフローグラフ上で、データパス中の各演算モジュールに対するテストプランをコントローラの制御系列で構成できるかどうかを解析している。テストプランをコントローラの制御系列を用いて構成できない場合は、MUX を用いてデータパスを設計変更する。各演算モジュールのテストプランは付加した MUX の制御ベクトルとコントローラの制御系列で構成されるため、テストプランを供給するための回路の面積オーバーヘッドが小さい。しかし、データパス中の MUX や付加した MUX をテストの対象としてお

らず、それらの MUX に対してテストプランを生成していないため、データパスに対して完全故障検出効率を保証できない。

強可検査法 [18] は、レジスタ転送レベル回路に対して、コントローラおよびデータパスそれぞれが独立してテストが可能となるように設計変更する非スキャンテスト容易化設計法である。この手法では、コントローラに対しては文献 [20] の手法を適用し、データパスに対しては文献 [21] の手法を適用している。またこの手法では、文献 [21] の手法を適用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ーヘッドで、実動作速度でのテスト実行を可能にしている。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。さらにこの手法では、組合せ回路用のテスト生成アルゴリズムを用いることができるように回路全体を設計変更しているため、コントローラの組合せ回路およびデータパスのすべ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る。しかし、文献 [21] の手法を適用後のデータパス中の各回路要素に対するテストプランでは、制御ベクトルが時刻ごとに変化するため、テストプラン生成回路は順序回路で構成され、その面積が大きくなり、回路全体の面積オーバーヘッドが完全スキャン設計法に比べて大きいという問題がある。

コントローラの機能を利用したデータパスのテスト容易化設計法 [24, 25] では、データパスの各回路要素に対してテストプランの存在を保証するために、レジスタのホールド機能および演算モジュールのスルー機能を用いてデータパスを設計変更する。また、データパスの各回路要素に対してテストプランを供給する機能をコントローラに付加しており、データパスのすべての組合せ回路要素に対して完全故障検出効率を保証している。

1.3. 提案手法

本論文では、第 2 章で、強可検査法 [18] の利点を持ち、またテストプラン生成回路の面積を縮小するために、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する。また固定制御可検査性に基づくレジスタ転送レベルデータパスの非スキャンテスト容易化設計法を提案し、強可検査法と同様に、コント

ローラに対しては文献 [20] の手法を適用する．さらにコントローラとデータパスそれぞれが独立にテストが可能となるように回路を設計変更する．固定制御可検査性を満たすデータパス中の各組合せ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．したがって，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，提案手法は強可検査法でのテスト生成時間，テスト実行時間およびデータパスの面積オーバーヘッドと同等であることを示す．また提案手法では，テストプラン生成回路および回路全体の面積オーバーヘッドが強可検査法に比べて縮小できることを示す．

さらに本論文では，第3 章で，Genesis での利点を持ち，またデータパス中のすべての組合せ回路要素に対するテストプランをコントローラの制御系列を用いて構成できるための，データパスの非スキャンテスト容易化設計法を提案する．提案手法では，テストプランをデータパスへ供給するための機能をコントローラに付加せずに，コントローラの制御系列を用いてテストプランを構成する．また提案手法では，拡張データフローグラフを抽出し，各組合せ回路要素に対してテストプランをコントローラの制御系列を用いて構成できるかどうかを調べる．テストプランをコントローラの制御系列を用いて構成できない場合は，演算モジュールのスルー機能，定数発生器および MUX を用いてデータパスを設計変更する．Genesis ではデータパスに対して完全故障検出効率を保証できないのに対して，提案手法では Genesis と同等の面積オーバーヘッドで，強可検査法と同様に完全故障検出効率を保証できる．実験では，提案手法は強可検査法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．また提案手法では，強可検査法に比べて面積オーバーヘッドおよび外部ピンオーバーヘッドを削減できることを示す．

第 2 章

固定制御可検査性に基づくレジスタ転送レベル回路の非スキャンテスト容易化設計法

2.1. はじめに

本章では，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，組合せ回路用のテスト生成アルゴリズムを用いた階層テスト生成および非スキャンテスト容易化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実行が可能で，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，提案手法の有効性を示す．

2.2. 諸定義

2.2.1 対象回路

レジスタ転送レベルでの VLSI 回路は一般に，コントローラとデータパスから成る (図 2.1)．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

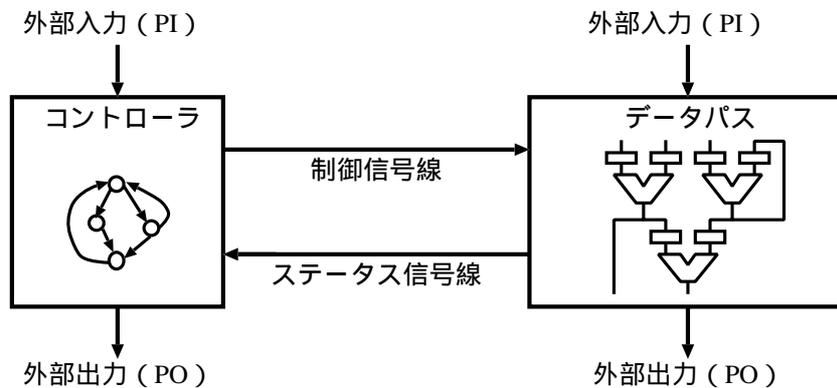


図 2.1 RT レベル回路

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．本章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，2.3節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2.4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．

2.2.2 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 段階ではレジスタ転送レベルにおいて， 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．

2.2.3 データパスの強可検査性

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レジスタ転送レベルで行われるため，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．

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， $R(CP) \wedge V$ 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.

2.3. テスト容易化設計

2.3.1 概要

提案手法では, 与えられたレジスタ転送レベルのコントローラ/データパス回路 (図 2.1) に対して, 文献 [18] 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 [20] の手法を適用し, データパスに対しては, 本章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 2.2). 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} [20] 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

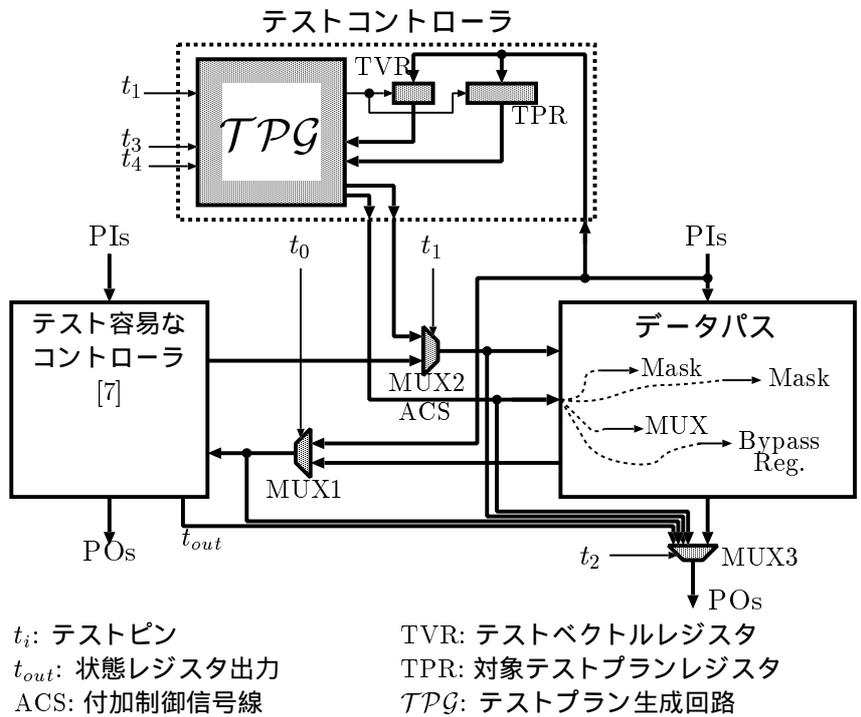


図 2.2 テストコントローラを含むレジスタ転送レベル回路

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 2.1 に示す．表 2.1 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．

2.3.2 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． □

表 2.1 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

定義より，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 [18] に比べて小さい．

2.3.3 データパスの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が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，

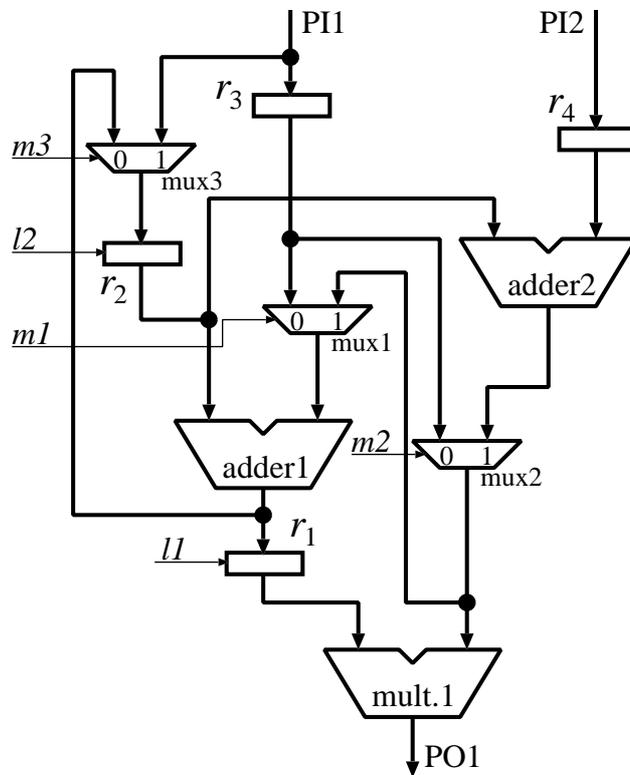


図 2.3 データパス例

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 [21] 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．

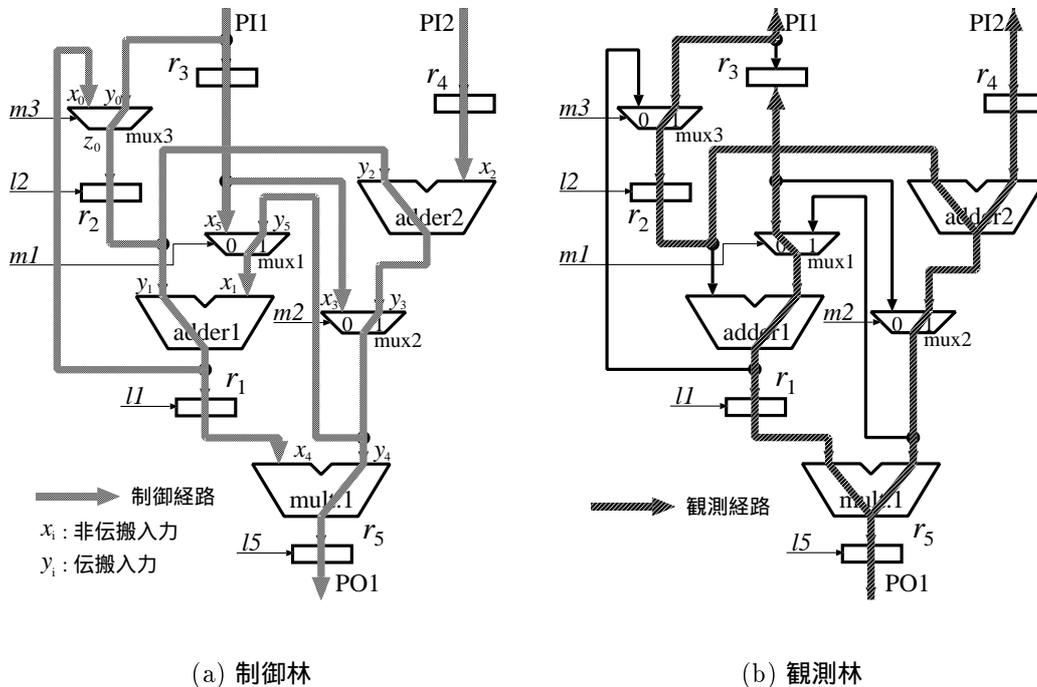


図 2.4 (a) 制御林, (b) 観測林

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 2.3 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 2.4(a) に示す．図 2.4(a) 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．

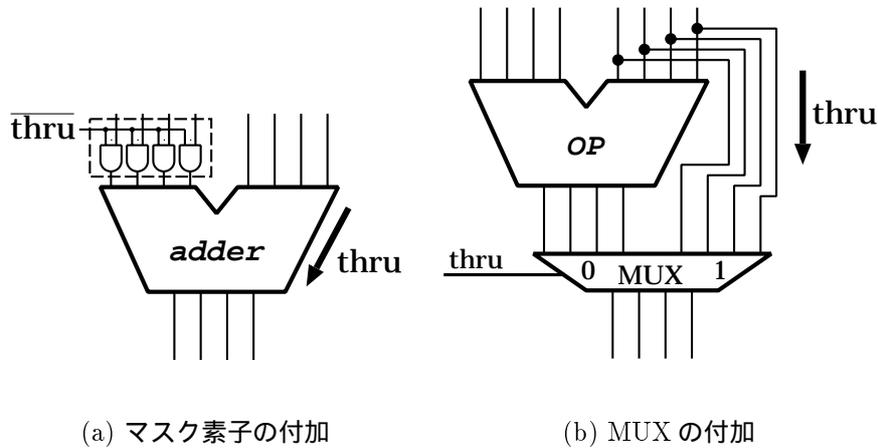


図 2.5 スルー機能の実現例

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 2.3 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 2.4(b) 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 2.5) ．

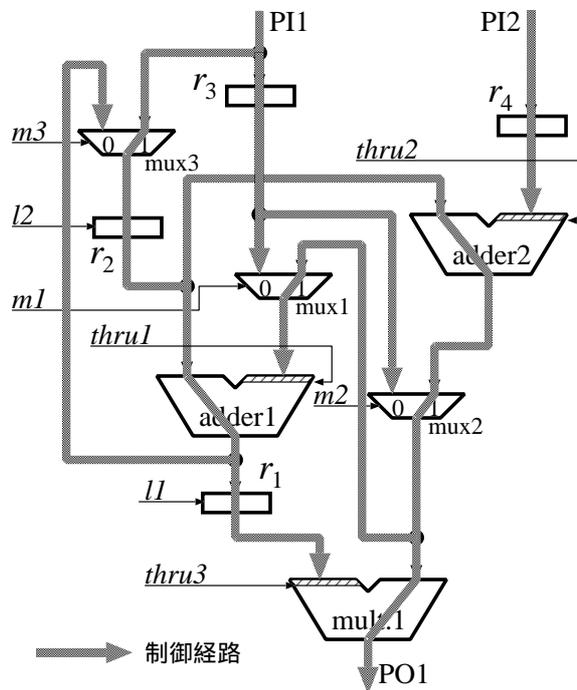


図 2.6 図 2.4(a) に対するステップ 3 実行例

図 2.4(a) に対するステップ 3 の実行例を 2.6 に示す。図 2.4(a) 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ー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，面積オーバー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．

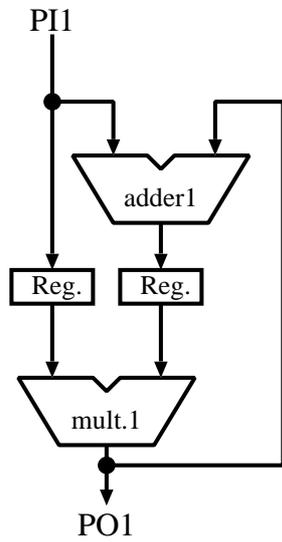
TM のテストを考える．TM の観測経路は p_c と共有しているので，TM の出力応答の伝搬のための観測経路を構成する必要はない．また TM の各入力ポートを終点とする制御経路の対は再収斂経路を構成するが，順序深度が異なるので，任意の値の伝搬を保証できる．

テスト MUX の付加例 テスト MUX(TM) の付加例を図 2.7 に示す．与えられたデータパス (図 2.7(a)) に対してステップ 1 を実行した結果を図 2.7(b) に示す．生成された制御林上の各経路において任意の値が固定制御で伝搬できるように adder1 および mult.1 にスルー機能を付加する (演算モジュール内の斜線部)．mult.1 の各入力ポートに対する制御経路の対は外部入力 PI1 を始点とし，mult.1 を終点とする順序深度の等しい再収斂経路となっており，任意の異なる値を固定制御で伝搬できるように TM を付加する (図 2.7(c))．TM を付加した後の mult.1 に対する制御経路を図 2.7(d) に示す．TM の直前のレジスタをバイパスすることにより，TM を付加した経路上の終点となる mult.1 の入力ポートに対して，外部入力からその入力ポートまでのレジスタ数は 1 つ減る．従って，同じ時刻に相異なる任意の値を mult.1 の各入力ポートに対して正当化できる．

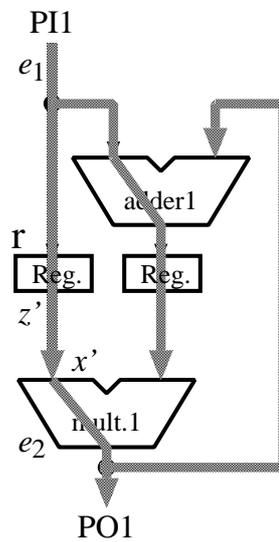
(2) p_c 上にレジスタが存在しない場合

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ため，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，順序深度が異なるため，任意の値の伝搬を保証できる．

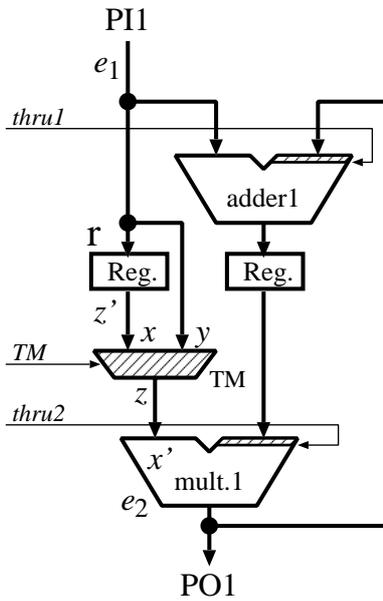
バイパスレジスタの付加例 バイパスレジスタ (BR) の付加例を図 2.8 に示す．与えられたデータパス (図 2.8(a)) に対してステップ 1 を実行した結果を図 2.8(b) に示す．生成した制御林上の各経路において任意の値が固定制御で伝搬できるように adder1 にスルー機能を付加する (演算モジュール内の斜線部)．adder1 の各入力ポートに対する制御経路の対は外部入力 PI1 を始点とし，adder1 を終点とする順序深度が 0 の再収斂経路となっており，任意の異なる値を固定制御で伝搬できるように BR を付加する (図 2.8(c))．BR を付加した後の adder1 に対する制御経路を図 2.8(d) に示す．BR を付加することにより，BR を付加した経路上の終点となる adder1 の入力ポートに対して，外部入力からその入力ポートまでのレジスタ数は 1 つ増える．従って，同じ時刻に相異なる任意の値を adder1 の各入力ポートに対して正当化できる．



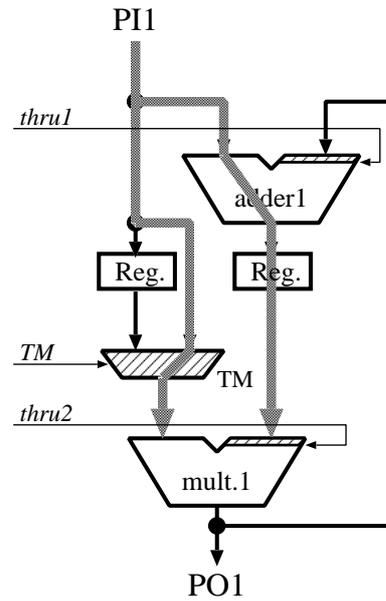
(a) データパスの例



(b)(a) の制御経路

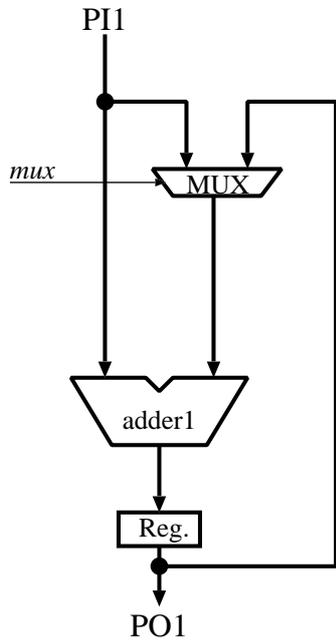


(c) ステップ 3 および 4(場合 1) 実行例

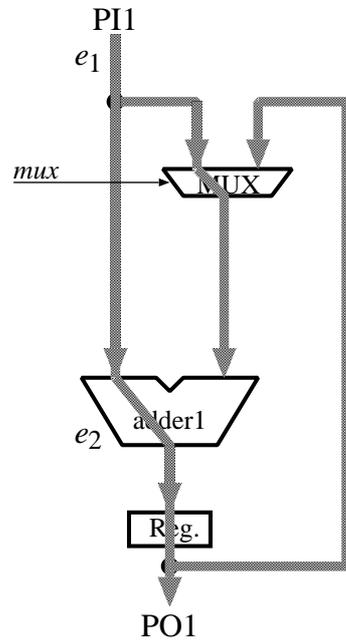


(d) TM 付加後の mult.1 に対する制御経路

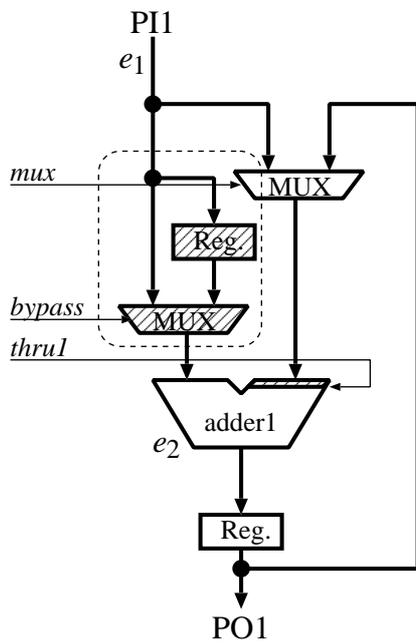
図 2.7 テスト MUX(TM) の付加例



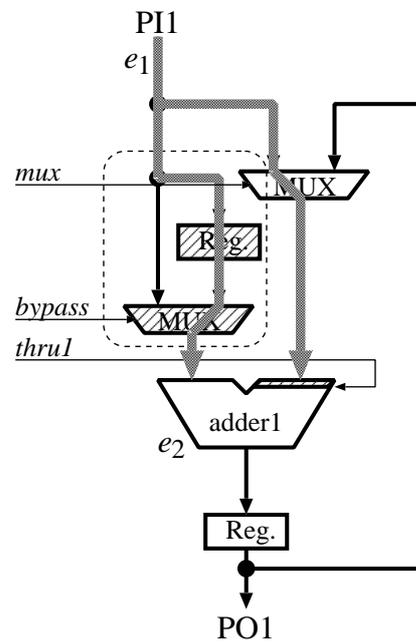
(a) データパスの例



(b)(a) の制御林



(c) ステップ 3 および 4(場合 2) 実行例



(d)(場合 2) 実行後のmult.1 に対する制御経路

図 2.8 バイパスレジスタ (BR) の付加例

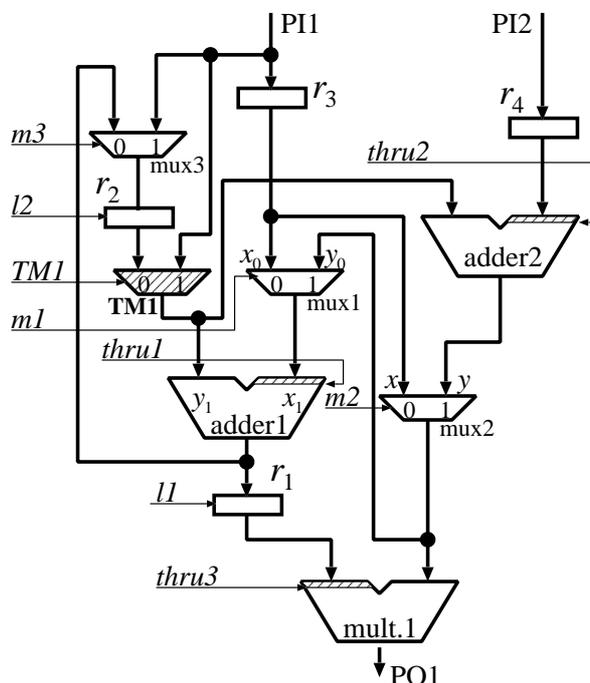


図 2.9 図 2.3 に対するステップ 4 実行例

ステップ 4 実行例 図 2.4 に対するステップ 4 の実行例を図 2.9 に示す．図 2.4(a) 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

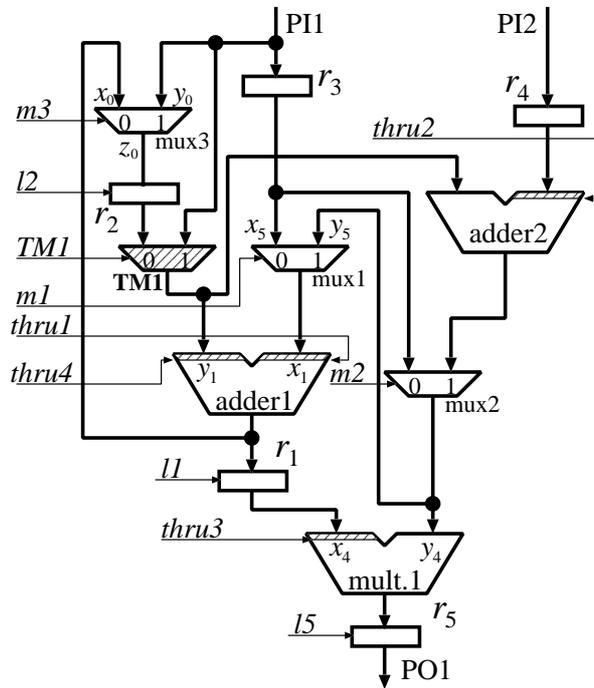


図 2.10 固定制御可検査データパス

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 2.3 に対する固定制御可検査データパスを図 2.10 に示す．図 2.4(b) 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

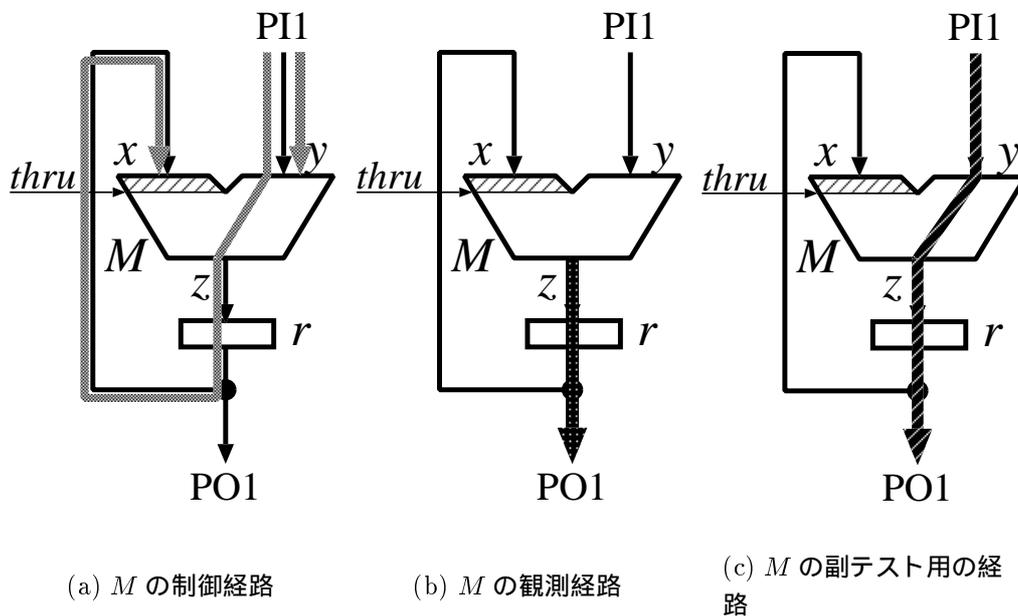


図 2.11 M の制御経路，観測経路，副テスト用の経路

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 $\text{PI1} \rightarrow \text{TM1} \rightarrow \text{adder2} \rightarrow \text{mux2} \rightarrow \text{mult.1}$ を利用できる．

2.3.4 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

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

図 2.12 テストプラン例 (図 4:adder1)

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 2.11 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 2.11(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 2.11(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．

テストプラン

2.3.3 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接続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 2.10 の adder1 に対するテストプランを図 2.12 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 2.2 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 \leftarrow PI	リセット
	0	1	$Z \leftarrow TP_J(TPR)$	正当化
	1	0	$Z \leftarrow TP_T(TPR, TVR)$	テスト
	1	1	$Z \leftarrow TP_P(TPR)$	伝搬

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
 TP_T : テストフェーズの制御ベクトル $mode$: モード切替信号
 TP_P : 伝搬フェーズの制御ベクトル Z: 制御信号線

2.3.5 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。レジスタ転送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 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。 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。 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 2.2 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 2.2 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 2.3.3 節で付加した回路の制御ポートに印加する。 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。

2.4. データパスの制約緩和

2.2.1 節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

演算モジュール 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．

2.5. レジスタファイルを含むデータパスのテスト容易化設計

2.5.1 レジスタファイル

レジスタファイルは，1 個の入力ポートおよび 2 個の出力ポートを持つ (図 2.13)． k 個のホールド機能を持つレジスタと 2 個の MUX ネットワークで構成される．MUX ネットワークは， $k - 1$ 個の MUX で構成され，レジスタファイル中の 1 個のレジスタからレジスタファイルの入力ポートへロードした値をどちらかの出力ポートへ伝搬できる．

2.5.2 レジスタファイルの設計変更

レジスタファイルの 2 個の出力ポートに対して，レジスタファイルの入力ポート RFinput1 を始点として，順序深度の等しい再収斂経路を構成する．この場合，RFoutput1 および RFoutput2 にデータパスの外部入力から異なる任意の値を同

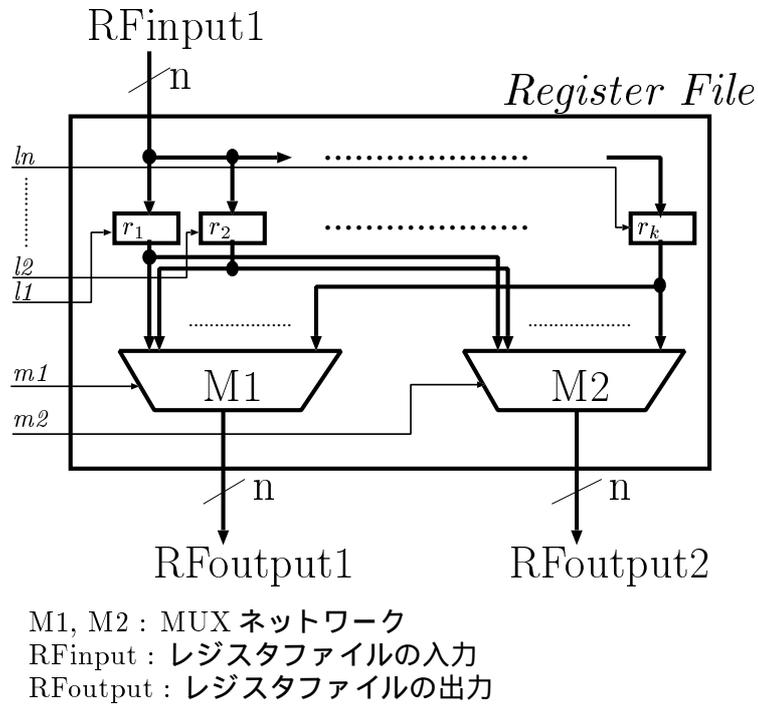


図 2.13 1 入力 2 出力レジスタファイル

時に伝搬することができない．そこで，以下のステップを実現できるように r_1 の直前に 1 個のテスト MUX(以下，TM) を付加する．TM の一方の入力ポートは $RFinput1$ から直接接続し，もう一方の入力ポートは r_2 の出力ポートから接続する．また，TM の出力ポートは r_1 の入力ポートへ接続する (図 2.14) ．

ステップ 1: $RFinput1$ へ連続して任意の値 a および b を伝搬する．また，それぞれの値をロードするレジスタを r_1 および r_2 とする．1 時刻目に値 a を r_2 にロードする．

ステップ 2: 2 時刻目に値 a を r_1 にロードし，値 b を $RFinput1$ から r_2 へロードする．

ステップ 3: 3 時刻目に r_1 から $M1$ を通して $RFoutput1$ へ値 a を伝搬， r_2 から $M2$ を通して $RFoutput2$ へ値 b を伝搬する．

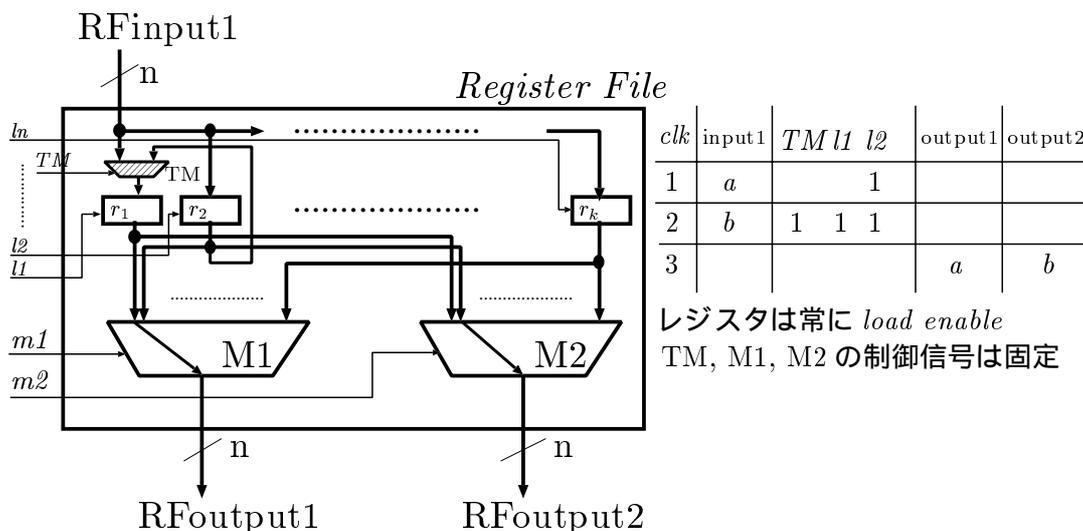


図 2.14 拡張後のレジスタファイル

2.5.3 レジスタファイル以外のデータパス部分に対する DFT 法

図 2.14のレジスタファイルの構造は機能的に図 2.15と等価であり，レジスタファイル以外のデータパス部分に対するテスト容易化設計では，レジスタファイルを図 2.15に示す構造に置き換える．図 2.15の疑似変換された回路を含めたデータパスに対して，第 2.3 章で提案したテスト容易化設計アルゴリズムを適用する．

2.5.4 レジスタファイル以外のデータパス部分に対するテストプラン生成法

2.3.4 節のテストプラン生成法と同様に，テスト容易化設計後の図 2.15の疑似変換された回路を含むデータパスに対して，任意の値が制御経路，観測経路および補助経路上を伝搬するタイミングの調整ができた経路集合が与えられるものとする．ただし，レジスタファイルの入出力ポートに対して生成した制御経路および観測経路上に，2.3.3 節のテスト容易化設計アルゴリズムのステップ 4 でテスト MUX またはバイパスレジスタを付加している場合がある．この場合，それらの付加回路要素を利用して RFinput1 までの順序深度，あるいは RFoutput から外部出力までの順序深度を変更することなく，2.3.3 節のテスト容易化設計アルゴ

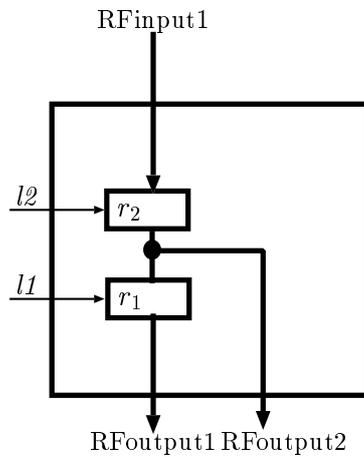


図 2.15 拡張後のレジスタファイルの構造モデル疑似変換

リズムのステップ 1 およびステップ 2 で生成した制御経路および観測経路を利用して任意の値を伝搬する．テスト対象となる組合せ回路要素に対する制御経路または観測経路上にレジスタファイルが存在する場合，そのような組合せ回路要素に対するテストプラン生成を考える．

RFoutput 上に制御経路が存在する組合せ回路要素をテストする場合:

レジスタファイルの一方の出力ポートからテストベクトルを伝搬する場合は，その出力ポートに接続するレジスタ (r_1 または r_2) にテストベクトルをロードする．2 個の出力ポートから異なるテストベクトルを同時に伝搬する場合は， r_1 および r_2 を利用して，テストベクトルをロードする (図 2.16(a)) ．

RFinput1 を介して出力応答を外部出力へ伝搬する場合:

この場合は，RFoutput1 または RFoutput2 を利用し，データパスの外部出力までの観測経路上で出力応答を伝搬する (図 2.16(b)) ．

上記のいずれの場合でも，各組合せ回路要素に対する制御経路上に存在する回路要素，レジスタファイル，および各組合せ回路要素に対する観測経路上に存在する回路要素に対して，固定制御で制御できる．

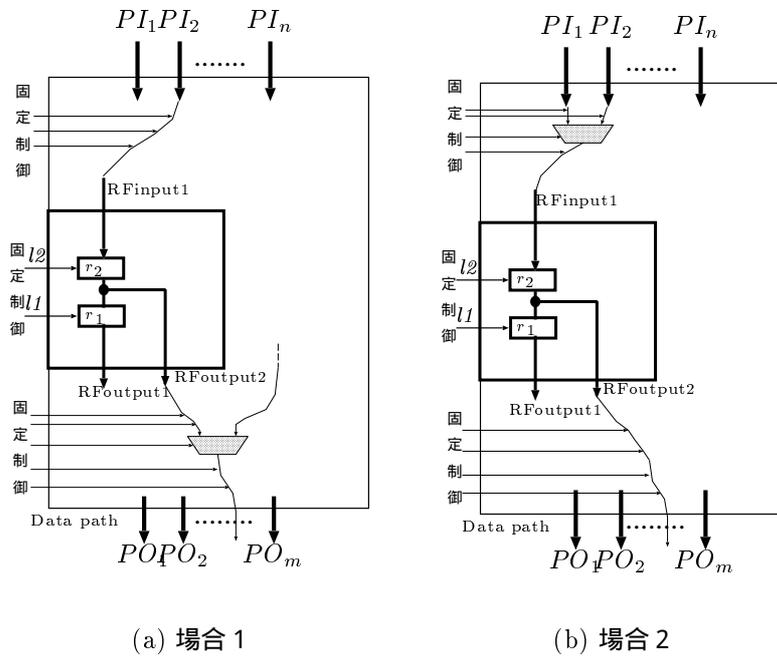


図 2.16 レジスタファイル以外のデータパス部に対するテスト

2.5.5 レジスタファイルのテストおよびテストプラン

レジスタファイルに対する制御経路は，外部入力から RFininput1 までの経路であり，観測経路は，2 個の RFoutput から外部出力までの経路である．また，レジスタファイルを構成する各回路要素に対する制御経路は，RFininput1 から各回路要素の入力ポートまでの経路であり，観測経路は，各回路要素の出力ポートから各 RFoutput までの経路である．

MUX ネットワークを構成する各 MUX の 2 個の入力ポートに接続されるレジスタファイル中のレジスタを r_a および r_b とする．各 MUX に対する正当化，テストおよび伝搬の各フェーズに分類して，レジスタファイル中の MUX ネットワークを構成する MUX のテストおよびテストプランについて述べる．

正当化フェーズ: データパスの外部入力から r_a へテストベクトル t_1 を正当化する．次に，データパスの外部入力から r_b へテストベクトル t_2 を正当化する．

t_2 が r_b へロードされる間は, r_a において t_1 をホールドする. 正当化フェーズにおけるテストプラン長は, [データパスの外部入力から RFininput1 までの制御経路上に存在するレジスタ数+2] である.

テストフェーズ: r_a および r_b からそれぞれ, t_1 および t_2 を MUX の各入力ポートへ正当化する. それらのテストベクトルに対する出力応答を MUX の出力ポートに最も近い観測経路上のレジスタ r_c へ伝搬する. r_c が観測経路上になければ, 出力応答は直接外部出力へ伝搬し, 伝搬フェーズはない. テストフェーズにおけるテストプラン長は, 1 である.

伝搬フェーズ: MUX の出力ポートに最も近い観測経路上のレジスタ r_c が存在すれば, 出力応答をその r_c から外部出力へ伝搬する. 伝搬フェーズにおけるテストプラン長は, [Rfoutput からデータパスの外部出力までの観測経路上に存在するレジスタ数] である.

レジスタファイルを設計変更した際に付加したテスト MUX の各入力ポートに接続される 2 つの制御経路は, RFininput1 を始点として再収斂経路を構成する制御経路であるが, 各制御経路の順序深度は, レジスタファイル中のレジスタ 1 個 (r_1) 分だけ異なるので, 新たにテスト容易化設計を必要としない. この付加したテスト MUX に対するテストは, 次のように行う. RFininput1 の直前のレジスタがあれば, そのレジスタを r_a とする. また, 付加したテスト MUX の一方の入力ポートの直前に存在するレジスタファイル中のレジスタを r_b とする. 外部出力から r_a および r_b それぞれに対して, テストベクトル t_1 および t_2 ($t_1 \neq t_2$) を正当化する (正当化フェーズ). 各レジスタからテストベクトルを, 付加したテスト MUX の各入力ポートへ正当化し, その出力応答を直後のレジスタ r_c へ伝搬する (テストフェーズ). r_c から MUX ネットワークを介して外部出力へ出力応答を伝搬する (伝搬フェーズ).

2.6. 実験結果

本節では、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、強可検査法 [18] および提案手法を、テスト容易化設計に伴う面積オーバーヘッド、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。実験に使用したレジスタ転送レベルベンチマーク回路は、GCD と PAULIN[13]、LWF と JWF[22] 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)、MPEG 回路 (以下、MPEG と呼ぶ) を使用した。MPEG 回路は 2 つのコントローラと 1 つのデータパスから成る。これらの回路の面積を表 2.3 に示す。またコントローラおよびデータパスの特性をそれぞれ、表 2.4 および表 2.5 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。

表 2.3 回路全体の面積

回路	Area(#gates)
GCD	1524.5
JWF	6875.4
LWF	1986.2
Paulin	24965.6
RISC	62287.6
MPEG	79788.6

表 2.4 回路特性 (コントローラ)

回路	#PI	#PO	#State	#Stat.	#Cont.	Area(#gates)	
GCD	1	1	4	3	7	169.4	
JWF	1	0	8	0	38	199.5	
LWF	1	0	4	0	8	57.7	
Paulin	1	0	6	0	16	123.5	
RISC	1	2	11	54	62	3986.9	
MPEG	ctrl1	3	0	129	0	256	9522.6
	ctrl2	3	0	34	0	15	781.1

表 2.5 回路特性 (データパス)

回路	#PI	#PO	bit	#Reg.	#Mod.	Area(#gates)
GCD	32	16	16	3	1	1350.9
JWF	80	80	16	14	3	6671.7
LWF	32	32	16	5	3	1924.3
Paulin	64	64	32	7	4	24833.7
RISC	32	96	32	40	4	58157.9
MPEG	56	148	8	241	161	69245.5

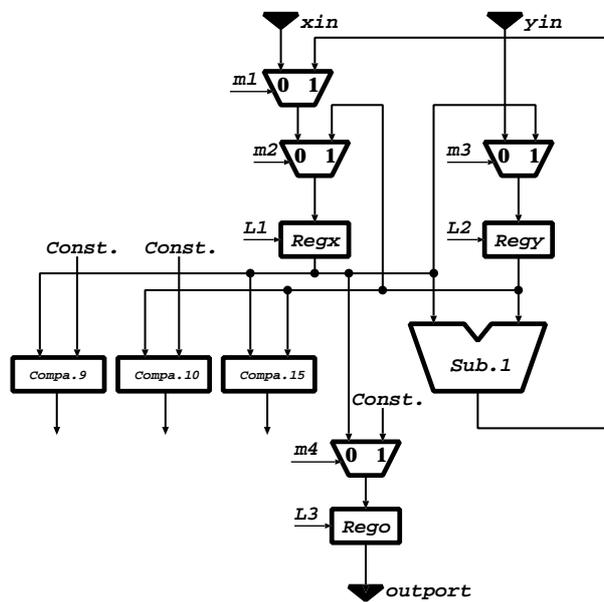


図 2.17 GCD データパス

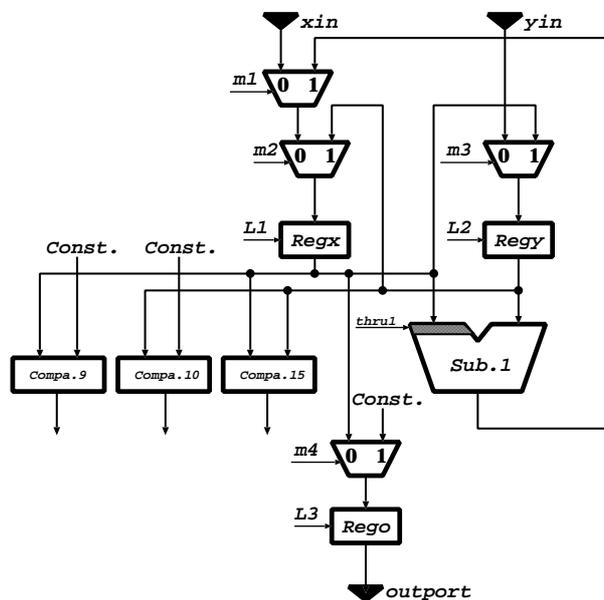


図 2.18 固定制御可検査 GCD データパス

図 2.20 固定制御可検査 Paulin テータパス

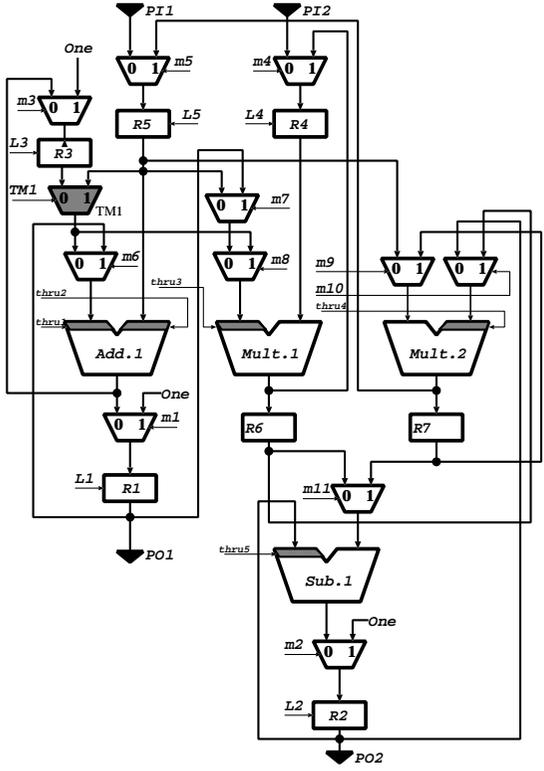
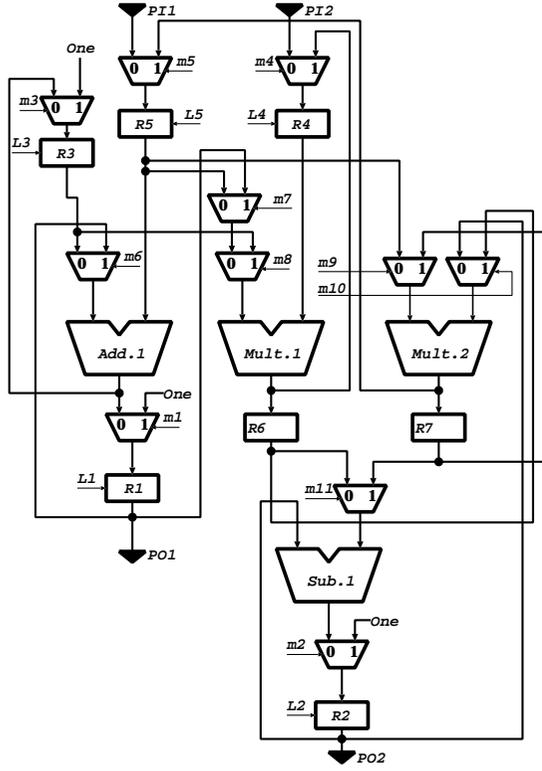


図 2.19 Paulin テータパス



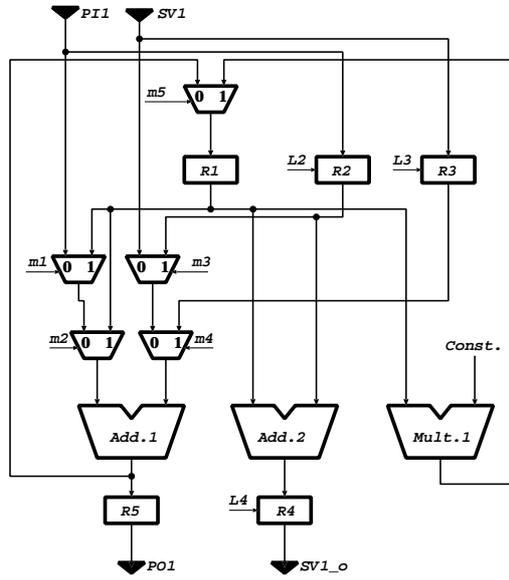


図 2.21 LWF データパス

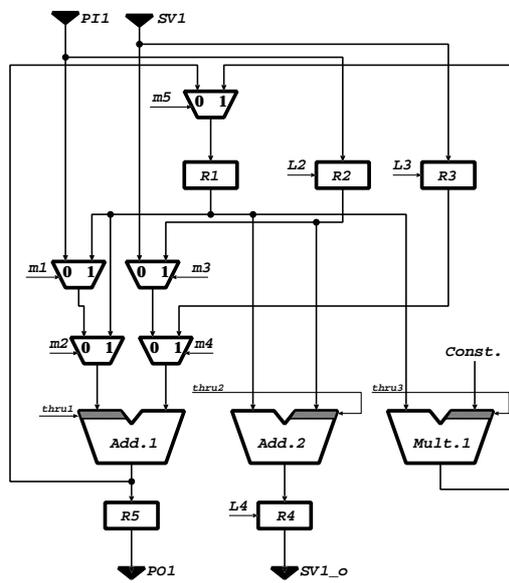


図 2.22 固定制御可検査 LWF データパス

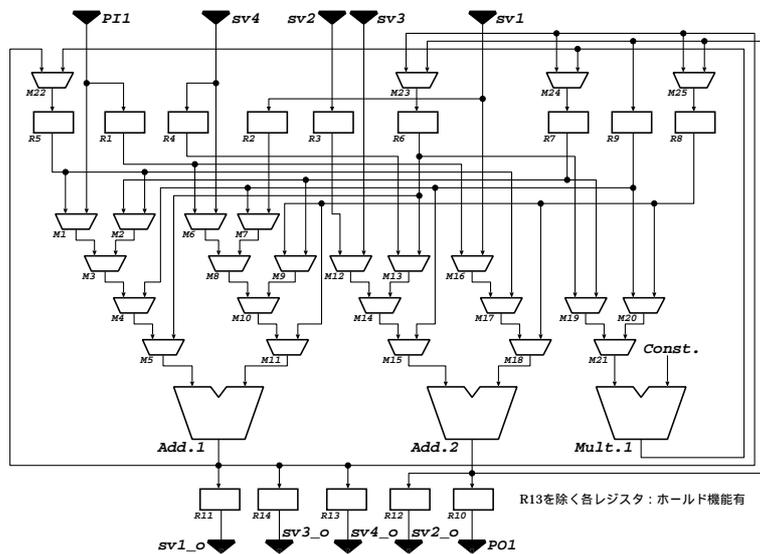


図 2.23 JWF データパス (制御信号線は省略)

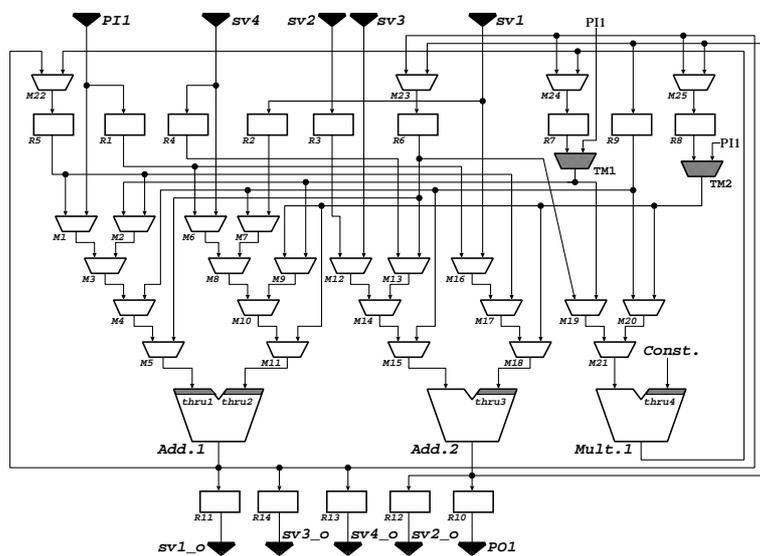


図 2.24 固定制御可検査 JWF データパス (制御信号線は省略)

各手法のテスト生成時間，テスト実行時間および故障検出効率をそれぞれ，表 2.6，表 2.7 および表 2.8 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．強可検査法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，ベンチマーク回路および MPEG では完全スキャン設計に比べて，平均で約 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ーヘッドおよび外部ピンオーバーヘッドをそれぞれ，表 2.9 および表 2.10 に示す．本章では，テストプラン生成回路 *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．*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，*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．MPEG に対して，強可検査法と提案手法を適用後，同じデータパスが得られ，強可検査法の各組合せ回路要素に対するテストプランは，提案手法と同じであった．そのため，提案手法の MPEG に対する面積オーバーヘッドは，強可検査法と同じであった．提案手法は *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

表 2.6 テスト生成時間 (秒)

回路	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
LWF	171.68	0.47	0.27	0.27
JWF	2348.24	2.88	0.37	0.27
Paulin	20362.55	4.68	2.11	2.20
RISC	288102.05	51740.92	71.50	72.29
MPEG	N/A	224.74	17.64	17.64

表 2.7 テスト実行時間 (クロックサイクル)

回路	DFT 前	完全スキャン	強可検査法	提案手法
GCD	9	6629	504	504
LWF	488	20519	1497	1621
JWF	322	4066	517	443
Paulin	283	16187	2193	2172
RISC	4298	1006154	9674	7768
MPEG	N/A	423573	150019	150019

表 2.8 故障検出効率 (%)

回路	DFT 前	完全スキャン	強可検査法	提案手法
GCD	4.92	100.00	100.00	100.00
LWF	98.14	100.00	100.00	100.00
JWF	99.64	100.00	100.00	100.00
Paulin	97.01	100.00	100.00	100.00
RISC	62.31	99.97	99.99	99.99
MPEG	N/A	100.00	100.00	100.00

表 2.9 面積オーバーヘッド (%)

回路	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
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
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
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
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
MPEG	19.7	24.9	0.2	4.0	12.9	0.1	7.2	24.9	0.2	4.0	12.9	0.1	7.2

表 2.10 外部ピンオーバーヘッド (#)

回路	完全スキャン	強可検査法	提案手法
GCD	3	4	5
LWF	3	3	4
JWF	3	3	4
Paulin	3	4	4
RISC	3	6	6
MPEG	3	7	7

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ー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ー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ー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ーヘッドについて、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 の面積オーバーヘッドは強可検査法より大きいが，その差は小さい．

外部ピンオーバーヘッドについて，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 および MPEG における外部ピンオーバーヘッドは，他の回路の外部ピンオーバーヘッドより多い．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 RISC では 2 段必要となり，また MPEG では 3 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ですが，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．

2.7. むすび

本章では，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 TPG を組合せ回路で構成できる．したがって，提案手法では TPG の面積は強可検査法 [18] に比べて小さくできる．提案手法では，データパスに対する制約を強可検査法に比べて緩和している．したがって，提案手法は強可検査法に比べて適用範囲が広い．さらに提案手法では強可検査法と同様に非スキャン設計法であるので，

実動作速度でのテスト実行が可能である．

ベンチマーク回路および実設計回路を用いた実験では，提案手法でのテスト生成時間，テスト実行時間は強可検査法と同等で，完全スキャン設計法に比べて短い．特に RISC 回路においては，提案手法のテスト生成時間は約 $1/700$ ，テスト実行時間は約 $1/130$ と大幅に短縮できたことを示した．データパスの固定制御可検査性は，データパスの強可検査性に比べてデータパスに対する制約が強いため，データパスのテスト容易化設計による付加回路の面積が増大する可能性があると考えられるが，実験に用いた回路では，提案手法と文献 [21] の手法での面積オーバーヘッドはほぼ同等であると言える．

提案手法の面積オーバーヘッドは完全スキャン設計法に比べて依然として大きいいため，今後は，テストプラン生成回路の面積を削減できるようなデータパスの回路構造の性質，データパスの非スキャンテスト容易化設計法を提案することが必要である．

第 3 章

レジスタ転送レベルでのデータフロー依存型回路の階層テスト容易化設計法

3.1. はじめに

本章では，コントローラの機能を考慮したデータパスの階層テスト容易化設計法を提案する．データパスの階層テスト生成法では，各回路要素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，外部入力から回路要素の入力へテストベクトルを正当化し，その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．提案手法では，拡張データフローグラフを用いてコントローラから制御ベクトル系列を抽出し，これを用いてテストプランを構成する．これにより，データパスへテストプランを供給するための付加回路の面積を小さくできる．提案手法はさらに，実動作速度でのテスト実行 (at-speed test) が可能で，データパスに対して完全故障検出効率を保証できる．

3.2. 諸定義

3.2.1 データフロー依存型回路

レジスタ転送レベルにおいて，回路はコントローラとデータパスから構成される．データフロー依存型回路では，コントローラとデータパスは制御信号線のみで接続され (図 3.1)，コントローラはリセット入力のみを持つ．レジスタ転送レベルにおいてコントローラは状態遷移図，データパスは回路要素および回路要素を相互に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，定数入

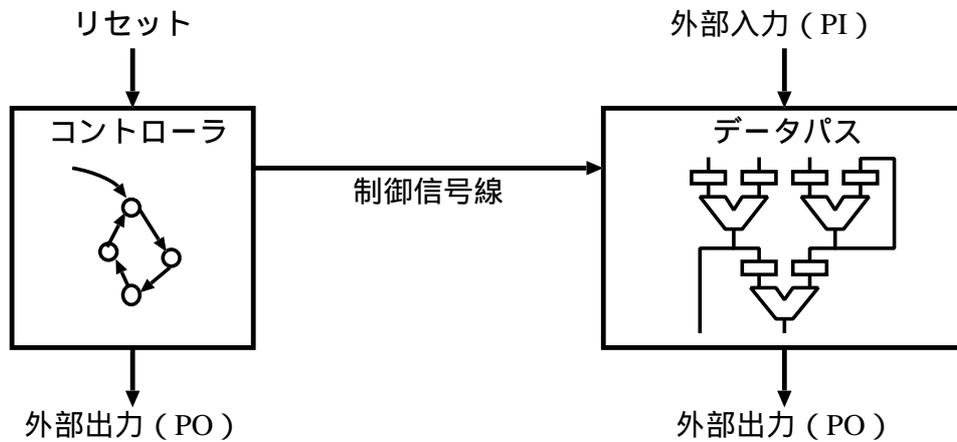


図 3.1 レジスタ転送レベルにおけるデータフロー依存型回路

力，ホールド機能を持つレジスタと持たないレジスタ，MUX，加算器や乗算器などの演算モジュールに分類される．以下では，組合せ回路で構成される MUX および演算モジュールを組合せ回路要素と呼ぶ．各回路要素は入出力を持ち，入力にはデータを入力するデータ入力とコントローラから制御値を入力する制御入力に分類され，出力はデータを出力するデータ出力がある．信号線は回路要素のデータ入出力を接続するためのデータ信号線とコントローラと制御入力を接続するための制御信号線に分類される．本論文では，以下の条件を満たすデータパスを対象とする．

- A1: 回路要素の各データ入出力のビット幅は全て等しい．
- A2: 各回路要素は，1 個または 2 個のデータ入力，1 個のデータ出力，高々 1 個の制御入力を持つ．

3.2.2 拡張データフローグラフ

ここでは，新しいデータフローグラフとして拡張データフローグラフを導入する．拡張データフローグラフ (以下では ETCDF, Extended Test Control Data Flow と呼ぶ) は，有向グラフ $G = (V, E, c)$ で与えられる． V は演算を表す頂点の集合， E はデータ転送を表す辺の集合， $c: V \mapsto \mathcal{N}$ (\mathcal{N} : 自然数) は制御ステップであ

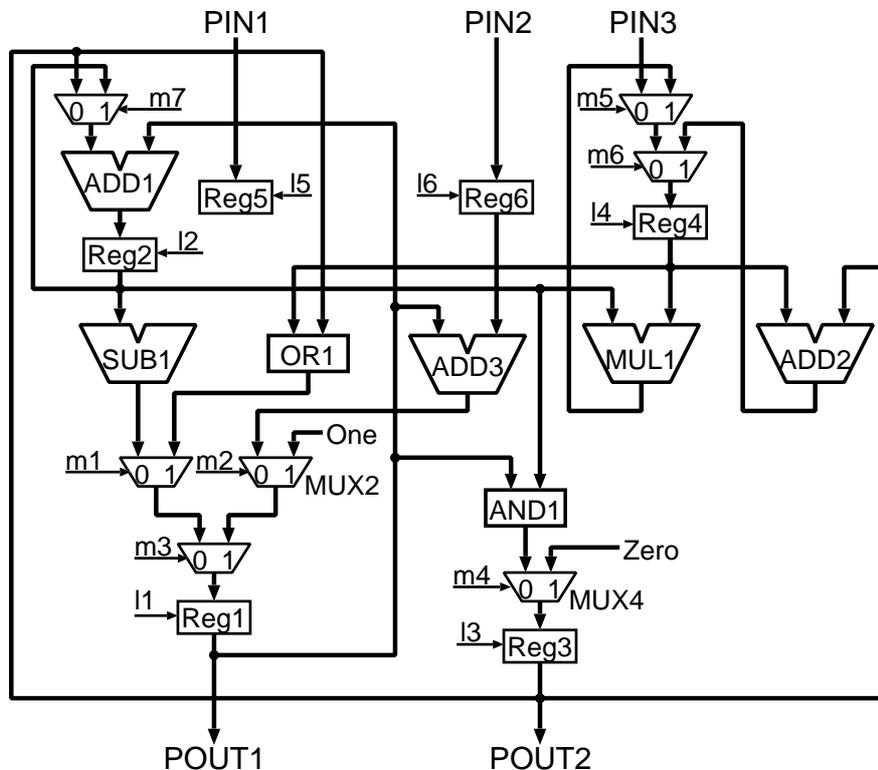


図 3.2 Tseng データパス

る．頂点 $v \in V$ は，データパスの外部入力，外部出力，定数入力，演算モジュールまたは MUX に対応する．辺 $e(v_i, v_j) \in E$ が $c(v_i) \neq c(v_j)$ を満たすならば，辺 e は 2 つの演算頂点 v_i と v_j に対応する回路要素間に存在するレジスタとそれらの回路要素を接続するデータ信号線に対応する．辺 $e(v_i, v_j) \in E$ が $c(v_i) = c(v_j)$ を満たすならば，辺 e はデータ信号線に対応する．

ETCDF は 2 種類の頂点で構成される．1 つは実行される演算に対応する頂点で，その頂点に対応する回路要素の入力に値が伝搬され，レジスタまたは外部出力にその回路要素の出力応答を伝搬する．もう 1 つは実行されない演算に対応する頂点 (区別するために以下では，ダミー頂点と呼ぶ) で，そのダミー頂点に対応する回路要素の出力応答がレジスタおよび外部出力に伝搬しない．ETCDF は 2 種類の辺で構成される．1 つは実行される 2 つの演算頂点を接続する辺である．

Input reset	State		Outputs													
	PS	NS	l1	l2	l3	l4	l5	l6	m1	m2	m3	m4	m5	m6	m7	
1	Any	S1	1	0	1	1	1	1	0	1	1	1	0	0	0	
0	S1	S2	0	1	0	0	0	0	0	0	0	0	0	0	0	
0	S2	S3	1	0	0	1	0	0	0	0	0	0	1	0	0	
0	S3	S4	1	1	0	1	0	0	0	0	1	0	0	1	1	
0	S4	S5	1	0	1	0	0	0	1	0	0	0	0	0	0	
0	S5	S1	0	0	0	1	1	1	0	0	0	0	0	0	0	

図 3.3 *Tseng* コントローラ状態遷移表

もう1つはレジスタとダミー頂点またはダミー頂点とMUXに対応する頂点間を接続する辺(区別するために以下では,ダミー辺と呼ぶ)である.

ここで,演算モジュールのスルー機能について考える.ある制御ステップで,スルー機能付き演算モジュール m の機能がスルー演算に選択されている場合, m に対応する頂点はETCDF上には表さず,スルー演算によるデータフローについては単に辺で表す.

ベンチマーク回路 *Tseng* のデータパスとコントローラの状態遷移表をそれぞれ,図3.2と図3.3に示す.制御ステップ数を6としたときの *Tseng* に対するETCDFを図3.4に示す.図3.4において,大きい頂点は演算モジュールに対応し,小さい頂点はMUXに対応する.図3.4において,破線で示した頂点がダミー頂点を表し,破線で示した辺がダミー辺を表す.Genesis[13]のデータフローグラフ(TCDF, Test Control Data Flow)では演算モジュールに対応する頂点のみ表しているのに対して,提案手法のETCDFでは演算モジュールおよびMUXに対応する頂点を表している.

ETCDF上のダミー頂点およびダミー辺に関する回路の機能は回路外部に影響しないが,テストプランを構成するためのテスト容易化の際にこの機能を考慮することで,面積オーバーヘッドを軽減することができる.またETCDFでは,コントローラのリセットは最初の制御ステップでのみ実行されるものとする.

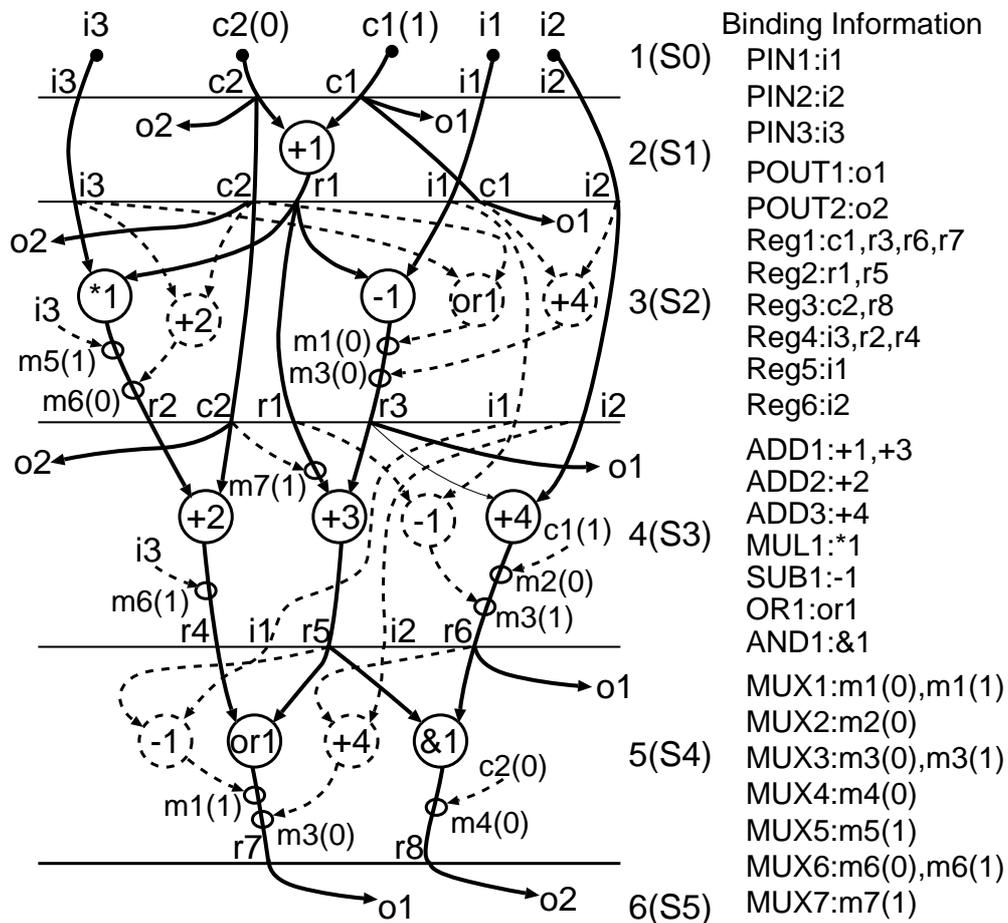
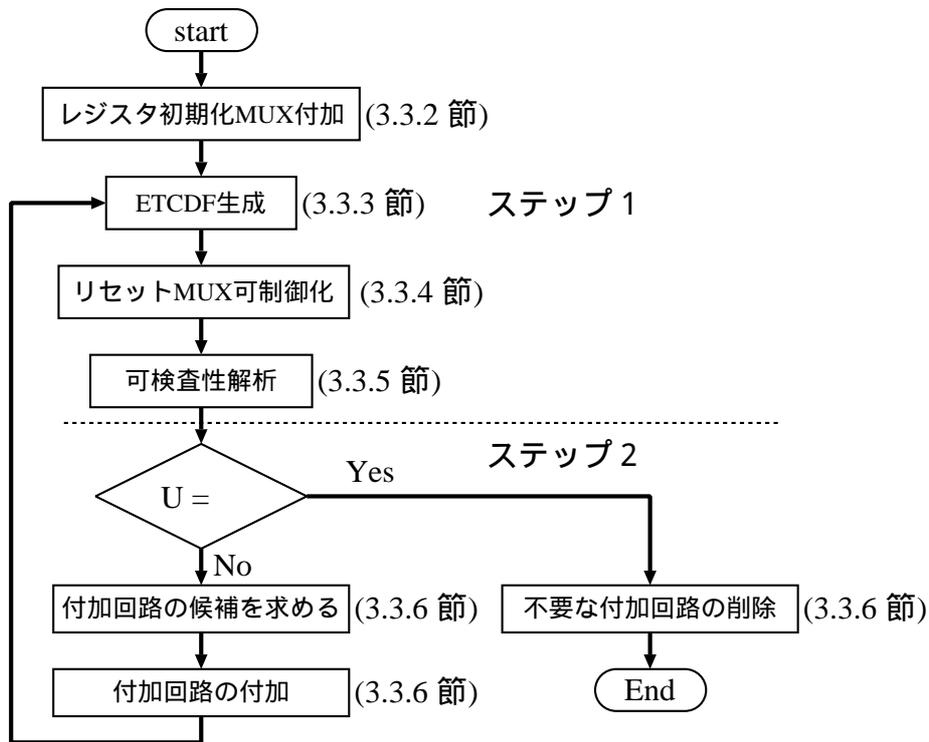


図 3.4 拡張データフローグラフ (Tseng)

3.3. テスト容易化設計

3.3.1 概要

提案するテスト容易化設計法では、演算モジュールのスルー機能、定数発生器および MUX を用いて、与えられたデータパスを強可検査データパスに設計変更する。テストプランをデータパスへ印加するのに全ての制御信号線（付加した回路要素の制御信号線も含む）を付加回路によって制御すれば、付加回路の面積が大きくなる。そのため提案手法では、データパスに付加した回路要素以外の回路



U: テストプランをコントローラの制御系列を用いて
構成できない組み合わせ回路要素の集合

図 3.5 テスト容易化設計手続き

要素はコントローラの制御系列を用いて制御する。提案手法は以下の 2 ステップからなる (図 3.5 参照)。

ステップ 1: データパス中の各組合せ回路要素に対してコントローラの制御系列を用いてテストプランを構成できるかどうかを調べる。

ステップ 2: ステップ 1 でテストプランを構成できなかった組合せ回路要素に対してテスト容易化を行う。

ステップ 1 では、コントローラの制御系列を抽出するために、コントローラとデータパスから ETCDF を生成する (3.3.3 節)。ETCDF 上で、テスト対象となる組合せ回路要素に対応する頂点の入力と出力に直接接続する辺の可制御性および

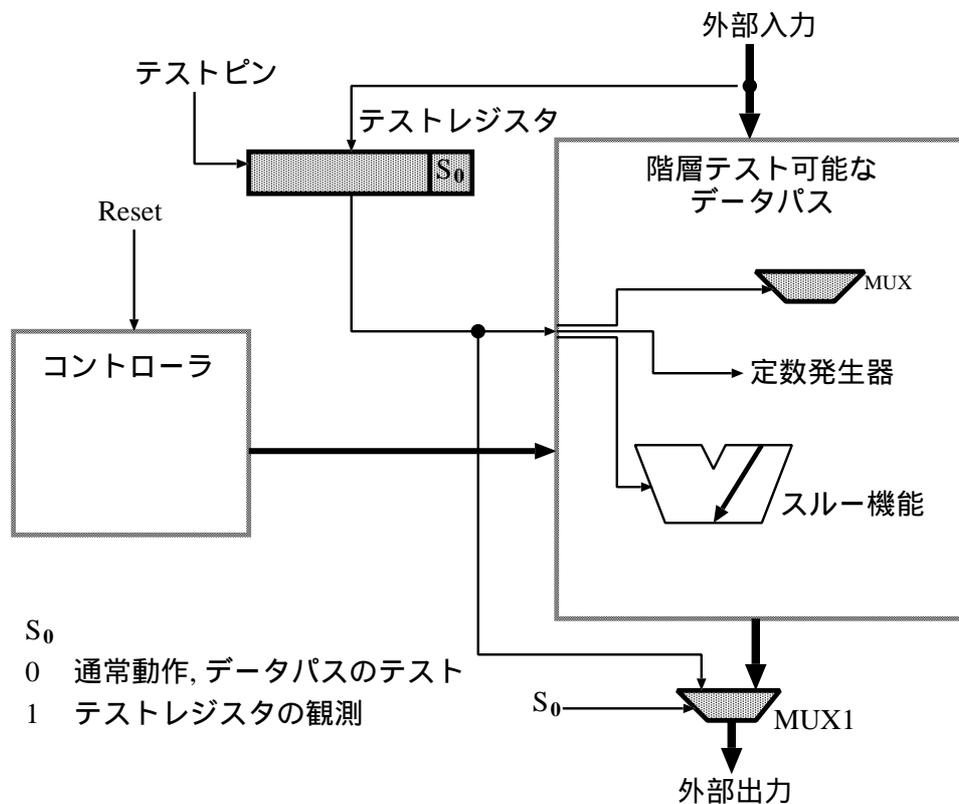


図 3.6 テストプランの供給方法

可観測性を調べることで、テストプランをコントローラの制御系列を用いて構成できるかどうかを調べる (3.3.5節)。

ステップ 2 では、テストプランをコントローラの制御系列を用いて構成できるように、データパスを設計変更する (3.3.6 節)。テスト容易化で用いる付加回路としては、面積オーバーヘッドをできるだけ小さく抑えるために、演算モジュールのスルー機能、定数発生器および MUX を用いる。このとき、各組合せ回路要素に対するテストプランの付加回路の制御については、1 つの制御ベクトルで構成できるようにする。

テストプランの供給方法を図 3.6 に示す。提案手法では各テストプランについて、付加回路の制御を 1 つの制御ベクトルで構成するので、付加回路の制御用のハードウェアをレジスタ (テストレジスタと呼ぶ) のみで構成できる。従って提案

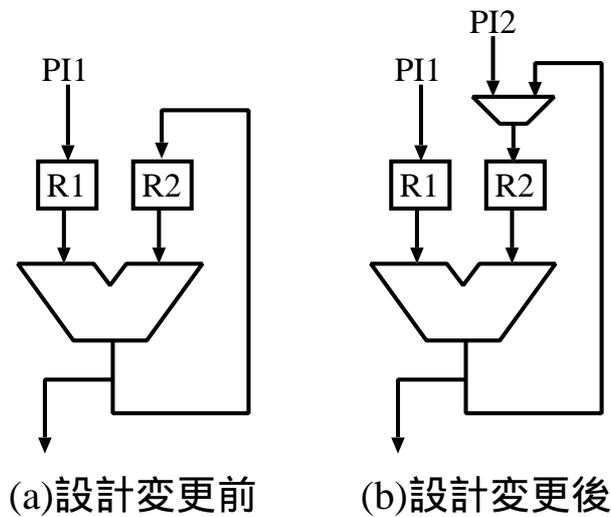


図 3.7 レジスタ初期化 MUX 付加

手法では、強可検査法および固定制御可検査法に比べて少ない面積でテストプランの供給を実現している。テストレジスタの故障によって、データパスの付加回路に誤った制御ベクトルが供給される場合があるので、データパスのテストができない可能性がある。そのため、データパスをテストする前にテストレジスタからデータパスの付加回路へ供給する制御ベクトルを外部で観測する必要がある。そこで、データパスの外部出力側に MUX を付加し (図 3.6 中の MUX₁)、テストレジスタの値を観測する。テストレジスタの故障を検出すればテストは終了する。この MUX はテストレジスタから制御する (図 3.6 中の S_0)。テストピンはテストレジスタのホールド制御用の外部ピンである。

3.3.2 レジスタ初期化 MUX 付加

データパス中の全ての組合せ回路要素に対応する頂点を ETCDF 上に表すことができるように前処理としてデータパスを設計変更する。与えられたデータパス中のあるレジスタを外部から何らかの値に初期化できないとき、外部からそのレジスタまでのデータフローは存在しない。前処理では、初期化できないレジスタの直前に MUX を付加して、外部入力とその MUX を接続し、外部から初期化で

きるように設計変更する．例えば，図 3.7(a) のレジスタ R2 は外部から初期化できないので，R2 の直前に MUX を付加し，外部入力とその MUX を接続する (図 3.7(b)) ．テストの際 (テストプランの生成の時も含む) に，外部入力から初期化できないレジスタへのデータフローが存在するように，付加した MUX の制御を固定する．

3.3.3 ETCDF 生成手続き

コントローラに接続された制御信号線以外の制御信号が決められたとき，ETCDF の生成は，以下の ETCDF 生成手続きおよび ETCDF 更新手続きにより行う．

ETCDF 生成手続き: 与えられたデータパスおよびコントローラの状態遷移表をそれぞれ， DP および FSM とする．また， l を状態遷移回数， s_0 をリセット状態とする．レジスタ初期化 MUX 付加により，外部入力または定数入力から全てのレジスタへのデータフローが存在することが保証されている．従って，全てのレジスタを初期化するのに必要な状態遷移を起こせば，全ての回路要素に対応する頂点を ETCDF 上に表現することができる．ここで， l は与えられるものとする．ただし， l はレジスタ初期化に必要な状態遷移回数以上の整数値とする． s_0 に対応する 1 番目の制御ステップから l 番目の制御ステップまで ETCDF 更新手続きを繰り返す．

ETCDF 更新手続き (i 番目の制御ステップを ETCDF に追加する手続き): $i-1$ 番目の制御ステップまでの ETCDF を G_{i-1} とする． s_{i-1} ， s_i をそれぞれ， $i-1, i$ 番目の制御ステップに対応する FSM の状態とする． E_{i-1} を $i-1$ 番目の制御ステップの辺の集合とする． s_i において， G_{i-1} に追加する頂点の集合および辺の集合をそれぞれ， V_i ， E_i とし，各集合は最初は空とする．このとき，以下の手続きによって得られる V_i ， E_i を G_{i-1} に加える．

1. s_i でホールドモードになっているレジスタに対応する辺が E_{i-1} に存在すれば，その辺を E_i に追加する．
2. 次の条件を同時に満たす演算モジュールに対応する頂点を V_i に追加する．

c1: 演算モジュールの入力へ外部入力または E_{i-1} の辺に対応するレジスタからの経路が存在する．

- c2: 演算モジュールの出力から外部出力または s_i でロードモードになっているレジスタへの経路が存在する .
3. 2 で V_i の頂点として追加した演算モジュールへレジスタを介さない経路をもつ外部入力および外部出力に対応する頂点を V_i に追加する .
 4. 2 で V_i に追加した演算モジュール間 , 演算モジュールとレジスタ間または演算モジュールと外部入出力間に存在する MUX を考える . MUX の s_i で選択されていない入力とレジスタまたは外部入力との接続関係を調べ , MUX の選択されていない入力へ E_{i-1} の辺に対応するレジスタからの経路があるならば , この MUX に対応する頂点 v を V_i に追加する . さらに v に対応する MUX の選択されていない入力とその入力への経路を持つレジスタ間に演算モジュールがあれば , その演算モジュールをダミー頂点として V_i に追加する .
 5. 各頂点間を接続する辺 , ダミー辺およびロードモードになっているレジスタに対応する辺を E_i に追加する .

3.3.4 リセット MUX 可制御化

リセット状態において制御信号が v , 他の状態において制御信号が \bar{v} であるような MUX をリセット MUX と呼ぶことにする (図 3.2 中の MUX2, MUX4) . リセット MUX は , レジスタを定数または外部入力から初期化するために用いられる . ETCDF の 1 番目の制御ステップにリセット MUX に対応する頂点が現れない場合は , リセット状態にのみ活性化されるリセット MUX の故障を検出するためのテストプランを生成することができない . この場合にはテストレジスタから任意の制御値をリセット MUX の制御入力に与えるための回路をリセット MUX の制御信号線上に付加する (図 3.8) .

リセット状態において制御信号が 1 , 他の状態において制御信号が 0 のリセット MUX を考える . 制御信号 0 が印加される制御ステップで , リセット MUX を通してデータ転送が行われるとき , リセット MUX に対応する頂点はその制御ステップに現れる . その頂点に対してテストプランの存在を保証すれば , リセット

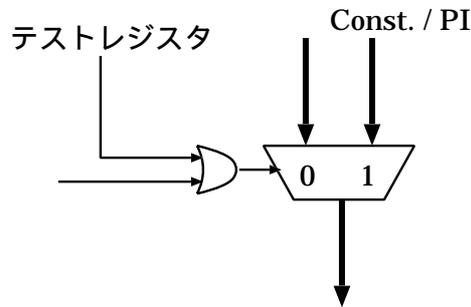


図 3.8 リセット MUX 可制御化

MUXの制御入力に1を印加することで、リセット状態にのみ活性化されるリセット MUX の故障を検出できる。

リセット MUX の制御のための付加回路については、テストレジスタから制御する。テストプランの生成の際には、リセット MUX に対してリセット状態で制御信号が v のときのみ活性化されるリセット MUX の故障を検出するためのテストプランを生成することはできないが、 \bar{v} が印加される制御ステップに存在するリセット MUX に対応する頂点に対してテストプラン p を生成できる。リセット状態で制御信号が v のときのみ活性化されるリセット MUX の故障を検出するためのテストプランは、付加回路を用いて制御入力に v を与えることによって p で代用できる。

3.3.5 可検査性解析

データパス中の各組合せ回路要素 m に対してテストプランをコントローラの制御系列を用いて構成できるかどうかを調べるために、ETCDF 上で m に対応する頂点 n の入力に接続された辺 (入力辺) の可制御性および出力に接続された辺 (出力辺) の可観測性を調べる。可検査性解析は次の 2 段階からなる。第 1 段階では、 n の各入出力辺に対して表 3.1 に基づいて尺度を付ける。尺度とは、レジスタまたはデータ信号線上の値の制御および観測が可能かどうかを示す性質である。尺度として以下の 6 つの性質を用いる。

表 3.1 テストプラン生成対象頂点の入出力辺に対する尺度

可検査性解析 対象回路要素	入力辺	出力辺
2 入力演算モジュール	C_g, C_g	O
1 入力演算モジュール	C_g	O
MUX(制御信号 0) (左入力を選択)	左入力: C_{all1} , 右入力 C_0	O
	左入力: C_0 , 右入力:—	O
MUX(制御信号 1) (右入力を選択)	左入力: C_0 , 右入力 C_{all1}	O
	左入力:—, 右入力: C_0	O

—:尺度を付けないことを表す.

一般可制御性: 辺 x に任意の値を制御可能かどうかを示す尺度をいい, $C_g(x)$ で表す.

0 可制御性: 辺 x に値 0 を制御可能かどうかを示す尺度をいい, $C_0(x)$ で表す.

1 可制御性: 辺 x に値 1 を制御可能かどうかを示す尺度をいい, $C_1(x)$ で表す.

全 1 可制御性: 辺 x に全て 1 からなる値を制御可能かどうかを示す尺度をいい, $C_{all1}(x)$ で表す.

可観測性: 辺 x の値を観測可能かどうかを示す尺度をいい, $O(x)$ で表す.

可検証性: 辺 x の値を観測可能かどうか, または, 辺 x に任意の値を制御可能かどうかを示す尺度をいい, $V(x)$ で表す.

演算モジュールのテストでは, ETCDF 上で対応する頂点の全ての入力辺が一般可制御性 C_g を満たし, かつ出力辺が可観測性 O を満たさなければならない. MUX のテストでは, 制御入力と各データ入力に対して, $(0, all1, 0)$, $(0, 0, -)$, $(1, 0, all1)$, $(1, -, 0)$ の 4 通りのテストパターンを正当化できれば, MUX の全ての故障をテストできる. ここで「-」はどんな値でもよいことを意味する. 表 3.1 の MUX において, 上段は MUX の制御信号線上の故障を検出するために各入出力辺に付ける尺度, 下段は MUX のデータ信号線上の故障を検出するために各入出力辺に付ける

表 3.2 尺度変換表

尺度	+	-	×	÷	AND	OR	XOR	NOT	MUX
$C_g(z)$	$C_g(x), V(y)$ $C_g(y), V(x)$	$C_g(x), V(y)$ $C_g(y), V(x)$	$C_g(x), C_1(y)$ $C_g(y), C_1(x)$	$C_g(x), C_1(y)$ $C_g(y), C_1(x)$	$C_g(x), C_{all}(y)$ $C_g(y), C_{all}(x)$	$C_g(x), C_0(y)$ $C_g(y), C_0(x)$	$C_g(x), V(y)$ $C_g(x), V(x)$	$C_g(x)$	$C_g(z)$
$C_1(z)$	$C_1(x), C_0(y)$ $C_1(y), C_0(x)$	$C_1(x), C_0(y)$	$C_1(x), C_1(y)$	$C_1(x), C_1(y)$ $C_{all}(x), C_{all}(y)$	$C_1(x), C_{all}(y)$ $C_1(y), C_{all}(x)$ $C_1(x), C_1(y)$	$C_1(x), C_0(y)$ $C_1(y), C_0(x)$ $C_1(x), C_1(y)$	$C_0(x), C_1(y)$ $C_0(y), C_1(x)$	$C_g(x)$	$C_1(z)$
$C_0(z)$	$C_0(x), C_0(y)$	$C_0(x), C_0(y)$ $C_1(x), C_1(y)$ $C_{all}(x), C_{all}(y)$	$C_0(x)$ $C_0(y)$	$C_0(x)$	$C_0(x)$ $C_0(y)$	$C_0(x), C_0(y)$	$C_0(x), C_0(y)$ $C_1(x), C_1(y)$ $C_{all}(x), C_{all}(y)$	$C_{all}(x)$	$C_0(z)$
$C_{all}(z)$	$C_{all}(x), C_0(y)$ $C_{all}(y), C_0(x)$	$C_{all}(x), C_0(y)$	$C_{all}(x), C_1(y)$ $C_{all}(y), C_1(x)$	$C_{all}(x), C_1(y)$	$C_{all}(x), C_{all}(y)$	$C_{all}(x)$ $C_{all}(y)$	$C_{all}(x), C_0(y)$ $C_{all}(y), C_0(x)$	$C_0(x)$	$C_{all}(z)$
$O(x)$	$V(y), O(z)$	$V(y), O(z)$	$C_1(y), O(z)$	$C_1(y), O(z)$	$C_{all}(y), O(z)$	$C_0(y), O(z)$	$V(y), O(z)$	$O(z)$	$O(x)$
$O(y)$	$V(x), O(z)$	$V(x), O(z)$	$C_1(x), O(z)$	$C_g(x), O(z)$	$C_{all}(x), O(z)$	$C_0(x), O(z)$	$V(x), O(z)$	-	$O(y)$
$V(x)$	$V(y), O(z)$	$V(y), O(z)$	$C_1(y), O(z)$	$C_1(y), O(z)$	$C_{all}(y), O(z)$	$C_0(y), O(z)$	$V(y), O(z)$	$O(z)$	$V(x)$
$V(y)$	$V(x), O(z)$	$V(x), O(z)$	$C_1(x), O(z)$	$C_g(x), O(z)$	$C_{all}(x), O(z)$	$C_0(x), O(z)$	$V(x), O(z)$	-	$V(y)$
$V(z)$	$V(x), V(y)$	$V(x), V(y)$	$V(x), V(y)$	$V(x), V(y)$	$V(x), V(y)$	$V(x), V(y)$	$V(x), V(y)$	$V(x)$	$V(z)$

$z = x \text{ (operation) } y$

尺度を表す。また、表 3.1 の MUX において、「—」は尺度を付ける必要がないことを表す。

第 2 段階では、尺度が付いた頂点 n の入出力辺ごとに、その尺度を尺度変換表 (表 3.2) に基づいて別の辺に対する尺度に変換していき、外部入力または外部出力に接続する辺に到達するまで尺度の変換を繰り返す (以下では、尺度変換と呼ぶ)。尺度変換表は、尺度変換において、各演算モジュールに対応する頂点に付いた尺度を、その演算モジュールの関数に基づいて別の辺の尺度にどう変換できるかを示した表である。尺度変換は以下の 2 ステップで行う。ここで、頂点 n に対して尺度を付けた辺の集合を $M(n)$ とする。また、 n の尺度変換によって尺度が付いた全ての辺の集合を $M'(n)$ とし、 $M'(n)$ は最初は空とする。

ステップ 1: $M(n) = \phi$ であれば尺度変換は終了し、そうでなければ以下を行う。 $M(n)$ から 1 個の尺度の付いた辺 e を削除し、その辺を $M'(n)$ に追加する。辺 e が外部入力または外部出力に接続する辺であればステップ 1 へ戻り、そうでなければステップ 2 を行う。

ステップ 2: 辺 e の尺度を尺度変換表に基づいて他の辺の尺度に変換し、尺度が付いた辺を $M(n)$ に追加してステップ 1 へ戻る。

テスト対象の組合せ回路要素 m に対応する頂点 n とは別の、 n より上流の m に対応する頂点 n' を介して値を伝搬するようなテストプランを構成したとき、 m を通して m のテストのための値を伝搬しなければならない。そのため、 m がその値を伝搬できるかを検証するために、頂点 n' の出力辺の可観測性を調べる必要がある。

また、テスト対象の組合せ回路要素 m' に対応する頂点 k の出力辺に付けた可観測性を外部出力に対応する頂点の入力辺まで尺度変換を行う際には、 k とは別の、 k より下流の m' に対応する頂点の出力辺への尺度変換を行わない。すなわち、 m' を含む閉路を通らずに、 m' の出力応答を外部出力へ伝搬する。これにより、不要な探索を削減することができる。

一般に組合せ回路要素に対応する頂点は ETCDF 上に複数存在するが、少なくとも 1 つの頂点に対する尺度変換で以下の問題が生じなければ、テストプランをコントローラの制御系列を用いて構成できる。

1. 定数入力に一般可制御性が付く、または異なる定数の可制御性が付く場合
2. ETCDF 上のある 1 つの辺において、尺度の衝突が生じる場合

ここで尺度の衝突とは、可検査性解析において尺度変換で複数の異なる尺度が 1 つの辺に対して割り当てられる場合、または、複数の一般可制御性または複数の可検証性が 1 つの辺に対して割り当てられる場合をいう。後者の場合、頂点 n に対応する回路要素の各データ入力に異なる任意の値が印加できない。可観測性は n の 1 つの出力辺に要求され、その可観測性は n とは別の頂点の出力辺の可観測性に変換されるので、1 つの辺に可観測性が同時に要求されることはない。

3.3.6 データパスのテスト容易化設計

可検査性解析でテストプランをコントローラの制御系列を用いて構成できないと判定された組合せ回路要素の集合を U とする。 U 中のある組合せ回路要素 m に対応する頂点は複数存在し、各頂点に対して考えられる尺度変換も複数存在するので、3.3.5 節で述べた問題点を解消するために必要な付加回路の候補を尺度変換ごとに求める。用いる付加回路は、演算モジュールのスルー機能、定数発生器および MUX である。スルー機能は、加算器や乗算器などの演算モジュールに対

しては，マスク素子を用いることで低面積で実現できる．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 を用いてスルー機能を実現する．定数発生器は，出力に定数を発生するマスク素子で実現できる． m に対してテストプランを生成できることを保証するために必要な付加回路の面積は，尺度変換ごとに異なる．そこで提案手法では， m に対応する全ての頂点に対して考えられる尺度変換ごとに必要な付加回路の面積を表す重みを付ける．各付加回路の重みを，マスク素子と定数発生器は 1，MUX は 3 とする． U 中の全ての組合せ回路要素の全ての尺度変換に対して付加回路の候補を求めた後に，重みの小さい順に付加回路をデータパスに付加する．全ての組合せ回路要素に対してテストプランをコントローラの制御系列を用いて構成できることを保証した後に，不要な付加回路が存在するかどうかを調べ，不要な付加回路があれば削除する．

付加回路候補の求め方

1 つの尺度変換に対して，3.3.5 節に示した問題点が発生していれば，それぞれの場合に応じて，以下のように付加回路の候補を求める．

問題点 (1):

i) 一般可制御性が定数入力に接続する辺に付いた場合: 定数入力の直後に MUX を付加すれば，外部入力から任意の値を直接制御できるようになる．この場合はこの MUX を候補とする．

ii) 定数 α の可制御性が定数 $c (\neq \alpha)$ に接続する辺に付いた場合: 定数 c の直後に α を発生する定数発生器を付加すれば，定数 α を制御できるようになる．この場合はこの定数発生器を候補とする．

問題点 (2):

i) 1 つの辺に一般可制御性が 2 回以上要求された場合:

a) テスト対象回路要素がデータ入力を 1 つ持つ場合，そのデータ入力の直前に MUX を付加すれば，外部入力から任意の値を直接制御できるようになる．この場合はこの MUX を候補とする．

b) テスト対象回路要素がデータ入力を 2 つ持つ場合，その回路要素に対応する頂点の入力辺までに再収斂経路の多い入力辺を e とする．その回路要素の e の終点に対応するデータ入力の直前に MUX を付加してこの問題が

解決できる場合は、この MUX を候補とする。ここで、ETCDF 上の再収斂経路とは、頂点 v_1 および 2 つのデータ入力を持つ v_1 とは異なる頂点 v_2 に対して、 v_1 を始点、 v_2 を終点とする任意の異なる経路の対をいう。

これで解決できなければ、両方のデータ入力の直前に MUX を付加することにより、この問題を解決できる。この場合はこれらの MUX を候補とする。

ii) $i)$ 以外で、1 つの辺に複数の尺度が要求された場合:

ここで、テストプラン生成対象の組合せ回路要素を m とする。

a) m とは別の 2 つのデータ入力を持つ回路要素 m' に対応する頂点を n' とする。 n' の入力辺を e_a および e_b 、出力辺を e_o とする。 e_o に付いた尺度を p_i とし、 e_a の p_i と e_b の p'_i に尺度変換されているとする。 e_b から外部入力に対応する頂点の出力辺までの間に尺度の衝突が発生している場合、 e_a に対応する m' のデータ入力にスルー機能を付加すれば、 e_b に無関係に、 e_o の p_i を e_a の p_i に尺度変換できるので、この問題を解決できる。この場合はこのスルー機能を候補とする。

b) m とは別の 2 つのデータ入力を持つ回路要素 m'' に対応する頂点を n'' とする。 n'' の入力辺を e'_a および e'_b 、出力辺を e'_o とする。 e'_a に付いた尺度を p_j とし、 e'_o の p_j と e'_b の p'_j に尺度変換されているとする。 e'_b から外部入力に対応する頂点の出力辺までの間に尺度の衝突が発生している場合、 e'_a に対応する m'' のデータ入力にスルー機能を付加すれば、 e'_b に無関係に、 e'_a の p_j を e'_o の p_j に尺度変換できるので、この問題を解決できる。この場合はこのスルー機能を候補とする。

外部入力から m の 1 つのデータ入力まで値を伝搬するのに 3 個*以上スルー機能を必要とする場合、または、 m のデータ出力から外部出力まで値を伝搬するのに 3 個以上スルー機能を必要とする場合は、付加回路の面積の増大を防ぐため、それらのスルー機能を候補とせず、1 個の MUX を候補として選ぶことを考える。前者の場合、 m のデータ入力の直前に MUX を付加すれば、外部入力から値を直接制御できるようになる。後者の場合、MUX を外部出力の直前に付加し、 m の

*1 個の MUX の面積は 3 個のスルー機能の面積に等しい。

データ出力とその MUX を接続すれば，外部出力で値を直接観測できるようになる．これらの場合はこの MUX を候補とする．

m とは別の 2 入力演算モジュールの両方のデータ入力にスルー機能が同時に候補として選ばれた場合には，同時に実現することができないので，どちらか一方の候補を選ばなければならない．候補として選ばれなかったスルー機能を MUX と置き換えて，外部入力からの m へのテストのための値の伝搬および外部出力への m の出力応答の伝搬を行うことを考える．ただし，付加回路の面積の増大を防ぐために，できるだけ多くのスルー機能を 1 個の MUX に置き換える．外部入力から m の 1 つのデータ入力へテストのための値を伝搬するのにスルー機能を最も多く必要とするならば，そのデータ入力の直前に MUX を付加すれば，外部入力から値を直接制御できるようになる． m のデータ出力から外部出力へ出力応答を伝搬するのにスルー機能を最も多く必要とするならば，MUX を外部出力の直前に付加し，そのデータ出力とその MUX を接続すれば，外部出力で値を直接観測できるようになる．これらの場合はこの MUX を候補とする．

付加回路の決め方

U 中の全ての組合せ回路要素に対応する各頂点について，全ての尺度変換ごとに求めた付加回路の候補の面積を表す重みを計算する．U が空になるまで以下の手続きを繰り返す．

1. 重みの最も小さい尺度変換を選び，その尺度変換で候補となっている付加回路を全てデータパスに付加する．
2. 付加回路の制御の全ての組合せについて ETCDF を生成し，U に付加回路を追加する．
3. 2 で生成した各 ETCDF 上で，U 中の全ての組合せ回路要素に対応する各頂点に対して可検査性解析を行う．

付加回路の削除

3.3.6 節で付加した回路要素のうち，不要な回路要素があるかどうかを調べる．付加回路を削除しても，コントローラの制御系列を用いて全ての組合せ回路要素に対してテストプランを構成できれば，その付加回路を削除する．

3.4. 従来法との比較

本節では，Genesis[16]，強可検査法 [18]，固定制御可検査法 [26] および提案手法を比較する．これらの手法はデータパスの階層テストを実現するための非スキャンテスト容易化設計法であり，実動作速度でのテスト実行が可能である．

Genesis:

コントローラの制御系列を用いて，各演算モジュールのテストプランを構成している．付加した回路要素 (MUX) の制御については，各演算モジュールのテスト間には付加した MUX の制御を固定にしているため，制御用の回路をレジスタのみで構成でき，少ない面積でテストプランを供給できる．しかし，MUX や付加した MUX をテストの対象としておらず，それらの MUX に対してテストプランを生成しないため，全ての組合せ回路要素に対して完全故障検出効率を保証できない．

強可検査法:

はじめにコントローラと独立に (制御入力を自由に制御できるものとして) データパスをテスト容易化する．次に制御入力へテストプランを供給できることを保証するために，制御入力に MUX およびテストプランを生成するテストコントローラを付加する．回路内部に付加したテストコントローラからデータパスへテストプランを供給できるので，実動作速度でのテスト実行が可能となる．また，全ての組合せ回路要素に対して完全故障検出効率を保証できる．テストプラン長を考慮して設計変更しているため，テスト実行時間が短くなる．強可検査法でのテストプランでは，制御ベクトルが時刻ごとに変化するので，強可検査法ではテストコントローラを順序回路で構成している．そのため，強可検査法ではテストコントローラや MUX の面積が大きくなり，面積オーバーヘッドが大きい．

固定制御可検査法:

各テストプランをテストベクトルの正当化，テスト，出力応答の伝搬の 3 つのフェーズに分けて，各フェーズにおける制御系列を 1 個の制御ベクトルで構成しているため，各テストプランは高々 3 個の制御ベクトルで構成できる．これにより，テストコントローラを組合せ回路で構成できるので，固定制御可検査法では強可検査法に比べてテストコントローラの実面積をより小さくすることができる．

しかし，制御信号線上に付加した MUX の面積が強可検査法と同程度であるため，依然として面積オーバーヘッドが大きい．

提案手法：

各テストプランを Genesis と同様にコントローラの制御系列を用いて構成している．従って，提案手法での面積オーバーヘッドは強可検査法および固定制御可検査法よりも大幅に削減できる．提案手法では，コントローラの制御系列を用いて各組合せ回路要素に対してテストプランを構成するため，テスト実行時間は強可検査法および固定制御可検査法でのテスト実行時間よりも長くなる．Genesis 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保証できないのに対して，提案手法では強可検査法および固定制御可検査法と同様に完全故障検出効率を保証できる．

3.5. 実験結果

Genesis，強可検査法，固定制御可検査法および提案手法を，面積オーバーヘッド，テスト生成時間およびテスト実行時間について比較した．実験に使用したレジスタ転送レベルベンチマーク回路は，LWF と JWF[22]，Paulin[13] および Tseng[23] である．これらの回路はデータフロー依存型回路である．4 つの回路の面積，コントローラおよびデータパスの特性をそれぞれ，表 3.3，3.4 および 3.5 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，#Stat. および #Control はそれぞれ，状態数，ステータス数および制御出力数を表す．データパスに関して，#Reg. および #Mod. はそれぞれ，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) を使用した．実験ではコントローラの状態数を k とすると， $2k + 1$ 番目の制御ステップまでの ETCDF を生成した．

テスト容易化設計に伴う付加回路のデータパスに対する面積オーバーヘッドおよび外部ピンオーバーヘッドをそれぞれ，表 3.6 および表 3.7 に示す．DP はデータパスのテスト容易化に伴う付加回路の面積オーバーヘッド，TC はテストプランをデータパスへ供給するための付加回路の面積オーバーヘッド，MUX はデー

表 3.3 回路全体の面積

回路	Area(#gates)
LWF	1986.2
JWF	6875.4
Paulin	24965.6
Tseng	15033.3

表 3.4 回路特性 (コントローラ)

回路	#PI	#PO	#State	#Stat.	#Cont.	Area(#gates)
LWF	1	0	4	0	8	57.7
JWF	1	0	8	0	38	199.5
Paulin	1	0	6	0	16	123.5
Tseng	1	0	5	0	13	95.0

表 3.5 回路特性 (データパス)

回路	#PI	#PO	bit	#Reg.	#Mod.	Area(#gates)
LWF	32	32	16	5	3	1924.3
JWF	80	80	16	14	3	6671.7
Paulin	64	64	32	7	4	24833.7
Tseng	96	64	32	6	7	14929.9

表 3.6 面積オーバーヘッド (%)

回路	Genesis*				強可検査法				固定制御可検査法				提案手法			
	DP	TC	MUX		DP	TC	MUX		DP	TC	MUX		DP	TC	MUX	
LWF	— (0)	— (0)	— (0)	— (0)	34.5	5.7	24.3	4.5	30.5	5.6	20.4	4.5	7.3	3.8	2.3	1.2
JWF	— (0)	— (0)	— (0)	— (0)	32.8	1.6	26.4	4.8	37.7	6.0	26.5	5.2	2.1	1.1	0.7	0.4
Paulin	— (1.3)	— (1.0)	— (0.2)	— (0.1)	8.0	2.0	5.4	0.6	6.1	2.3	3.1	0.7	2.5	1.6	0.6	0.3
Tseng	— (2.0)	— (1.7)	— (0.2)	— (0.1)	11.5	3.6	6.8	1.1	12.8	5.8	5.8	1.2	2.4	1.5	0.6	0.3

—: Genesis では演算モジュールのみテスト容易化しているため、全体のテスト容易化による面積オーバーヘッドを示せない。

*: Genesis では完全故障検出効率を達成できていないが、強可検査法、固定制御可検査法および提案手法では完全故障検出効率を達成。

表 3.7 外部ピンオーバーヘッド (#)

回路	Genesis	強可検査法	固定制御可検査法	提案手法
LWF	0	3	4	1
JWF	0	3	4	1
Paulin	1	3	4	1
Tseng	1	3	4	1

タパスの外部出力に付加した MUX の面積オーバーヘッドを示す。回路全体の面積オーバーヘッドは、回路全体に対する、データパスのテスト容易化に伴う付加回路の面積の割合を示す。回路全体の面積オーバーヘッドについて、提案手法は強可検査法および固定制御可検査法に比べて大幅に削減しており、Genesis とほぼ同等である。外部ピンオーバーヘッドについて、提案手法は Genesis と同様に、テストレジスタ (図 3.6) のロード/ホールド用のテストピン 1 本のみであり、強可検査法および固定制御可検査法に比べて小さい。

テスト生成時間およびテスト実行時間をそれぞれ、表 3.8 および表 3.9 に示す。Genesis および提案手法での括弧内のテスト生成時間およびテスト実行時間は演

表 3.8 テスト生成時間 (秒)

回路	Genesis ^{*1}	強可検査法 ^{*2}	固定制御可検査法 ^{*2}	提案手法
LWF	— (0.38)	0.83	0.85	0.78 (0.38)
JWF	— (0.38)	0.84	1.13	0.85 (0.67)
Paulin	— (2.44)	3.08	3.12	5.02 (4.51)
Tseng	— (2.83)	3.65	3.98	3.78 (3.26)

—: Genesis では MUX のテストをしていないため、全体のテスト生成時間を示せない。

*1, (): 演算モジュールのみのテスト生成時間。

*2: テストプラン生成時間を含む。

表 3.9 テスト実行時間 (クロックサイクル)

回路	Genesis ^{*1}	強可検査法	固定制御可検査法	提案手法
LWF	— (78)	229	229	196 (78)
JWF	— (78)	720	742	934 (78)
Paulin	— (1524)	1120	1334	1937 (1542)
Tseng	— (1131)	1135	1236	1547 (1281)

—: Genesis では MUX のテストをしていないため、全体のテスト実行時間を示せない。

*1, (): 演算モジュールのみのテスト実行時間。

算モジュールのみのテスト生成結果であり，Genesis では MUX に対してテストしていない．テスト生成時間はテストパタン生成時間とテストプラン生成時間からなる．Genesis および提案手法では，組合せ回路要素に対するテストプランは手動で求めたため，表 3.8 におけるテスト生成時間は組合せ回路要素に対するテストパタン生成時間のみを示している．一方，強可検査法および固定制御可検査法では，テスト生成時間にテストプラン生成時間が含まれているが，テストプラン生成時間は全ての回路に対して約 0.1～0.2 秒であった．従って，表 3.8 においてテスト生成時間はほとんどテストパタン生成時間で占められていることが分かる．これについては，提案手法でも同様である．提案手法でのテストプラン生成時間は強可検査法および固定制御可検査法に比べて長くなる可能性があるが，表 3.8 から分かるように，提案手法のテスト生成時間は強可検査法および固定制御可検査法とほぼ同じである．

強可検査法および固定制御可検査法ではテスト実行時間が短くなるように設計変更を行っているのに対して，提案手法ではコントローラの制御系列を用いて，各組合せ回路要素のテストプランを構成できるように設計変更を行う．従って，提案手法のテスト実行時間は，強可検査法および固定制御可検査法より長い．

提案手法でのテスト生成時間およびテスト実行時間は強可検査法および固定制御可検査法に比べて長くなった．これは面積オーバーヘッドとテスト生成時間およびテスト実行時間のトレードオフを示している．提案手法での面積オーバーヘッドは Genesis とほぼ同等であり，強可検査法および固定制御可検査法に比べて大幅に削減できることを実験で示した．

3.6. むすび

本章では，コントローラの制御系列を用いてデータパス中の各組合せ回路要素に対してテストプランを構成するための階層テスト容易化設計法を提案した．Genesis ではデータパス中の MUX や付加回路をテストの対象としておらず，それらの回路要素に対してテストプランを生成していないため，完全故障検出効率を保証できない．これに対して，提案手法では強可検査法および固定制御可検査法と同様に完全故障検出効率を保証できる．さらに提案手法での面積オーバーヘッ

ドは Genesis とほぼ同等で，強可検査法および固定制御可検査法に比べて大幅に削減できた．今後の課題としては，コントローラがステータス入力やリセット以外の外部入力を持ち，異なるビット幅，多入力多出力回路要素を持つレジスタ転送レベル回路を対象としたテスト容易化設計法に拡張することなどがある．

第 4 章

結論

近年の VLSI 回路の大規模化，高集積化技術の発展に伴い，回路のテストは困難な問題になっている．そのため，回路設計の段階から，回路をテストの容易な回路に設計変更するテスト容易化設計の研究が進められている．テスト容易化設計では，テスト容易化のための付加回路の面積オーバーヘッドおよび外部ピンオーバーヘッドをできるだけ小さく抑え，テスト生成時間やテスト実行時間の短縮や，故障検出効率の向上が目標である．

代表的なテスト容易化設計法として，スキャン設計法がある．スキャン設計法では一般に，長いテスト実行時間を要し，実動作速度でのテスト実行が困難であるという問題があり，また手法によっては依然として順序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高い故障検出効率を達成できない問題がある．

そこでスキャン設計法の問題を解消するために，回路の通常動作時に用いるデータ転送経路を介してテスト系列や出力応答の伝搬を行うことでレジスタの外部からの制御および外部での観測を可能にするための非スキャン設計法が提案されている．非スキャン設計法での利点としては，スキャン設計法と同等以上の故障検出効率を達成できること，またテスト生成時間およびテスト実行時間がスキャン設計法に比べて短縮できること，さらには実動作速度でのテスト実行が可能であること，などが挙げられる．代表的なレジスタ転送レベルでの非スキャンテスト容易化設計法として，直交スキャン設計法，Genesis や強可検査法などがある．

直交スキャン設計法ではテスト実行時間がスキャン設計法に比べて短くなるがテスト生成の対象が組合せ回路部分全体であるため，回路規模の増大に伴い，テスト生成時間が長くなるという問題がある．

テスト生成時間の削減を目的としたテスト生成法として，階層テスト生成法が提案されている．階層テスト生成法では組合せ回路要素単体に対してゲートレベ

ルでテスト生成を行い，レジスタ転送レベルでテストプラン生成を行う．したがって，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系列を生成することができる．階層テスト生成に基づく非スキャンテスト容易化設計法として，Genesis，強可検査法などがある．

Genesis では，データパス中の演算モジュールのみに対してテストプランをコントローラの制御系列を用いて構成できるように回路の設計変更を行う．Genesis では，テストプランをデータパスへ供給するための回路面積が小さいので，面積オーバーヘッドが小さい．しかし，Genesis では，データパス中の MUX をテストの対象としていないため，一般に完全故障検出効率を保証できない問題がある．

強可検査法では，コントローラおよびデータパスそれぞれに対して文献 [20] の手法，文献 [21] の手法を適用している．強可検査法では，テスト生成時間およびテスト実行時間はスキャン設計法に比べて短く，完全故障検出効率を達成でき，実動作速度でのテスト実行が可能である．しかし，回路内部に付加したテストプラン生成回路の面積が大きいため，回路全体の面積オーバーヘッドが完全スキャン設計法に比べて大きくなる問題がある．

本論文でははじめに，第 2 章で，強可検査法の利点を持ち，またテストプラン生成回路の面積を縮小するために，固定制御可検査性に基づくレジスタ転送レベルデータパスの非スキャンテスト容易化設計法を提案した．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．これに対して強可検査法でのテストプランは時刻ごとに制御ベクトルが変化するため，テストプラン生成回路は順序回路で構成される．したがって，提案手法のテストプラン生成回路の面積は，一般に強可検査法でのテストプラン生成回路の面積よりも小さくなる．実験結果から，提案手法は強可検査法でのテスト生成時間，テスト実行時間およびデータパスの面積オーバーヘッドと同等である．また提案手法では，テストプラン生成回路および回路全体の面積オーバーヘッドが強可検査法に比べて小さい．

次に，第 3 章で，Genesis での利点を持ち，またデータパス中のすべての組合せ回路要素に対するテストプランをコントローラの制御系列を用いて構成できるための，データパスの非スキャンテスト容易化設計法を提案した．提案手法では，

テストプランをデータパスへ供給するための機能をコントローラに付加せずに、コントローラの制御系列を用いてテストプランを構成するので、コントローラ的设计制約を壊さない利点がある。Genesis ではデータパスに対して完全故障検出効率を保証できないのに対して、提案手法では Genesis と同等の面積オーバーヘッドで、強可検査法と同様に完全故障検出効率を保証できる。一般に提案手法ではコントローラの制御系列を用いてテストプランを構成するため、テスト実行時間が強可検査法に比べて長くなるが、実験に用いた回路においてはその差は大きくない。また提案手法では、強可検査法に比べて面積オーバーヘッドおよび外部ピンオーバーヘッドが小さい。

VLSI 回路は今後もその高性能化が求められ、回路の大規模化が引き続き進展するものと考えられる。レジスタの段数の少ない回路を対象とした場合には、提案手法はテスト生成時間およびテスト実行時間がスキャン設計に比べて大幅に削減できる。しかし、レジスタの段数の多い回路を考えた場合に、本論文で提案した手法では、依然としてテストの費用は増大する傾向にある。提案した手法では、テストパターンごとにテストプランをデータパスへ供給するため、提案手法でのテスト実行時間はテストパターン数とテストプラン長の積の組合せ回路要素数の和となる。提案手法では、レジスタ数が多く、レジスタの段数が少ない回路に対しては、テスト実行時間が完全スキャン設計法に比べて大幅に削減できたことを実験で示したが、レジスタの段数の多く、回路要素数の多い回路に対しては、提案手法でのテスト実行時間は増大するものと考えられる。今後の課題としては、レジスタの段数に関係なく、どのような回路に対してでもテスト実行時間を削減することができるための非スキャンテスト容易化設計法が求められる。階層テスト生成法の利点を継承しつつテスト実行時間を削減するためには、組合せ回路要素ごとにテスト系列および出力応答の伝搬をパイプライン処理によって行う、または複数の組合せ回路要素を並行にテストすることなどができるためのテスト容易化設計法が必要になると考えられる。

また VLSI 回路においては、回路中の回路要素は異なるビット幅のデータ入出力ポートを持ち、また回路要素の出力が別の回路要素の制御入力に接続し、複数の回路要素がコントローラの同じ制御出力から制御され、さらには多入力多出力ポートを持つ回路要素が存在するなど、様々な回路構造を持つ。提案手法では、

ビット幅が同じで、たかだか 2 個の入力ポートとたかだか 1 個の出力ポートを持つ回路要素を扱い、信号線への制約があるなど、回路の構成条件が強い。そこで今後の課題としては、そのような回路の構成条件を緩和し、より実用的なテスト容易化設計法が求められる。

本論文での提案手法を含め、これまでに様々なテスト容易化設計法やテスト生成法が提案されている。しかしながら今後も引き続き VLSI 回路の大規模化、高集積化、高性能化が進展するにつれて、テストはますます困難な問題になる。したがって、VLSI の製造技術の発展とともに、そのテストの費用の削減ならびテストの質の向上を実現するために、より実用的なテスト容易化技術の開発を引き続き行わなければならない。

謝辞

本研究の機会を与えて下さるとともに、本研究の全過程を通じて、方針や問題点などのあらゆる面において、絶えず懇切丁寧な御指導と貴重な御助言、御討論を賜りました藤原秀雄教授に心から深く感謝の意を表します。

本研究を進めるにあたり、御指導と御助言を賜りました渡邊勝正教授に心から感謝の意を表します。

本研究を進めるにあたり、懇切丁寧な御指導と貴重な御助言、御討論を賜りました大阪大学基礎工学部の増澤利光教授に心から感謝の意を表します。

本研究の全過程を通じて、懇切丁寧な御指導と貴重な御助言、御討論を賜りました井上美智子助教授に心から感謝の意を表します。

本研究を進めるにあたり、懇切丁寧な御指導と貴重な御助言、御討論を賜りました広島市立大学情報機械システム工学科の井上智生助教授、市原英行助手に心から感謝の意を表します。

本研究の全過程を通じて、絶えず懇切丁寧な御指導と貴重な御助言、御討論を賜り、また情報論理学講座(現 コンピュータ設計学講座)CAD システムとその管理について御指導を賜りました大竹哲史助手に心から深く感謝の意を表します。

本研究を進めるにあたり、御指導と御助言、御討論を頂きました米田友和助手に感謝いたします。

本研究を進めるにあたり、貴重な御討論、御助言を賜りました(株)半導体理工学研究センター(STARC)の小澤時典取締役研究推進部長、(株)日立製作所中央研究所の畠山一実氏、ソニー(株)S&S アーキテクチャセンターコアテクノロジー部門の小野寺岳志氏、三洋電機(株)研究開発本部マイクロエレクトロニクス研究所の小椋功氏に感謝の意を表します。

本研究を進めるにあたり、貴重な御討論と御助言、御援助を賜りました(株)半導体理工学研究センター(STARC)設計技術開発本部の村岡道明 VCDS 開発室長、細川利典氏、宮崎政英氏、浜本征志氏ならびに松尾茂則氏、(株)システム・ジェイディーの伊達博氏に感謝の意を表します。

本研究における実験に用いた情報科学研究科 CAD システムの保守管理をして頂きました早稲田大学の木村晋二教授に感謝いたします。

本研究を進めるにあたり、御助言、御討論を賜り、また情報論理学講座(現 コンピュータ設計学講座)CAD システムとその管理について御指導を賜りました(株)日立製作所中央研究所の和田弘樹氏に心から感謝の意を表します。

また、日頃から御討論、御援助を頂きました情報論理学講座(現 コンピュータ設計学講座)の諸氏に感謝いたします。

参考文献

- [1] 藤原秀雄, コンピュータ設計概論, 工学図書, 1998.
- [2] H . Fujiwara, Logic testing and design for testability, The MIT Press, Cambridge, 1985.
- [3] M . Abramovici, M . A . Breuer and A . D . Friedman, Digital Systems Testing and Testable Design, Computer Science Press, 1990.
- [4] S . T . Chakradhar, A . Balakrishnan and V . D . Agrawal, "An exact algorithm for selecting partial scan flip-flops," in Proc. of Design Automation Conference, pp.81-86, 1994.
- [5] T . Inoue, T . Hosokawa and H . Fujiwara, "An optimum time expansion model based combinational atpg for rt level circuits," in Proc. of IEEE the 7th Asian Test Symposium, pp.190-197, 1998.
- [6] R . Gupta and M . A . Breuer, "Partial scan design of register-transfer level circuits," Journal of Electronic Testing: Theory and Applications, Vol.7, pp.25-46, 1995.
- [7] 藤原秀雄, 大竹哲史, 高崎智也, "組合せテスト生成複雑度でテスト生成可能な順序回路構造とその応用," 電子情報通信学会論文誌, Vol.J80-D-I, No.2, pp.155-163, February 1997.
- [8] 高崎智也, 井上智生, 藤原秀雄, "内部平衡構造に基づく部分スキャン設計法の考察," 電子情報通信学会論文誌, Vol.J81-D-I, No.8, pp.318-327, March 1998.

- [9] 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%?," in Proc. of International Test Conference, pp.358-364, 1991.
- [10] R . B . Norwood and E . J . McCluskey, "Orthogonal scan: Low overhead scan for data paths," in Proc. of International Test Conference, pp.659-668, 1996.
- [11] R . B . Norwood and E . J . McCluskey, "High-level synthesis for orthogonal scan," in Proc. of VLSI Test Symposium, pp.370-375, 1997.
- [12] S . Bhatia and N . K . Jha, "Genesis: A behavioral synthesis system for hierarchical testability," in Proc. of European Design and Test Conference, pp.272-276, 1993.
- [13] I . Ghosh, A . Raghunath and N . K . Jha, "Design for hierarchical testability of RTL circuits obtained by behavioral synthesis," in Proc. of IEEE International Conference on Computer Design, pp.173-179, 1995.
- [14] I . Ghosh, A . Raghunath and N . K . Jha, "A design for testability technique for RTL circuits using control / data flow extraction," in Proc of IEEE/ACM International Conference on CAD, pp.329-336, 1996.
- [15] I . Ghosh, A . Raghunath and N . K . Jha, "Design for hierarchical testability of RTL circuits obtained by behavioral synthesis," in Proc. of IEEE Transaction on CAD, Vol.16, No.9, pp.1001-1014, September 1997.
- [16] I . Ghosh, A . Raghunath and N . K . Jha, "A design for testability technique for RTL circuits using control / data flow extraction," in Proc of IEEE Transaction on CAD, Vol.17, No.8, pp.706-723, 1998.
- [17] I . Ghosh, A . Raghunath and N . K . Jha, "Hierarchical test generation and design for testability methods for ASPP's and ASIP's," in Proc. of IEEE Transaction on CAD, Vol.18, No.3, pp.357-370, March 1999.

- [18] S . Ohtake, H . Wada, T . Masuzawa and H . Fujiwara, "A non-scan DFT method at register-transfer level to achieve complete fault efficiency," in Proc. of Asia and South Pacific Design Automation Conference 2000, pp.599-604, 2000.
- [19] B . T . Murray and J . H . Hayes, "Hierarchical test generation using pre computed tests for modules," in Proc. of IEEE Transaction on CAD, Vol.9, No.6, pp.594-603, June 1990.
- [20] 大竹哲史, 増澤利光, 和田弘樹, 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, December 1998.
- [21] 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.
- [22] M . Inoue, T . Higashimura, K . Noda, T . Masuzawa and H . Fujiwara, "A high-level synthesis method for weakly testable data paths," in Proc. of IEEE the 7th Asia Test Symposium, pp.40-45, 1998.
- [23] I . Ghosh, N . K . Jha and S . Bhawmik, "A BIST scheme for RTL controller-data paths based on symbolic testability analysis," in Proc. of Design Automation Conference, pp.554-559, 1998.
- [24] 鈴木和博, 井上美智子, 藤原秀雄, "コントローラの機能を利用したデータパスのテスト容易化設計," 信学技報 (FTS2000-86), pp.1-8, February 2001.
- [25] 岡本紘征, 井上美智子, 藤原秀雄, "レジスタ転送レベル回路に対するテストプラン埋め込み型テスト容易化設計法," 信学技報 (FTS2001-83), pp.45-52, February 2002.
- [26] 永井慎太郎, 和田弘樹, 大竹哲史, 藤原秀雄, "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," 電子情報通信学会論文誌, Vol.J84-D-I, No.5, pp.454-465, May 2001.