

NAIST-IS-DD0561006

博士論文

時間展開モデル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

岩田 浩幸

2007年 3月 16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。

岩田 浩幸

審査委員：

藤原 秀雄 教授

松本 健一 教授

井上 美智子 助教授

# 時間展開モデル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\*

岩田 浩幸

## 内容梗概

VLSIの大規模化に伴い，VLSIのテスト費用の増加が重要な問題となっている．テスト費用を削減するために，回路にハードウェアを付加することによりテストを容易にするテスト容易化設計が提案されている．完全故障検出効率を達成する実用的なテスト容易化設計法として完全スキャン設計がある．しかし，この手法では，面積オーバーヘッドが大きい，テスト実行時間が非常に長いという問題点を抱えている．近年，レジスタ転送レベル(RTL)のコントローラ/データパス回路を対象とした非スキャン方式によるテスト容易化設計法が提案されており，テスト実行時間を大幅に削減することに成功している．しかし，依然として面積オーバーヘッドが大きいという問題が残されている．

本論文では，従来のRTL回路を対象とした非スキャン設計法における面積オーバーヘッドを削減することを目的とする．まず，RTL回路に対するテスト容易性として部分強可検査性を定義する．さらに，RTL回路を部分強可検査にするテスト容易化設計法及び時間展開モデルに基づく部分強可検査RTL回路に対するテスト生成法を提案する．評価実験では，提案手法は実用的なテスト生成時間で完全故障検出効率を達成し，従来のRTL回路を対象とした非スキャン設計法と比較して，面積オーバーヘッド及びテスト系列長を大幅に改善できたことを示す．

## キーワード

テスト容易化設計，RTL回路，時間展開モデル，部分強可検査，完全故障検出効率

\*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-DD0561006, 2007年3月16日.

# **Studies on a Non-Scan Design for Testability Method Based on Time Expansion Model for Register Transfer Level Circuits\***

Hiroyuki Iwata

## **Abstract**

With the progress of semiconductor technology, testing of VLSI circuits becomes more difficult and the cost is increasing. To reduce the cost of testing, design-for-testability (DFT) techniques have been proposed. The most widely used DFT technique is the full scan approach. However, it requires large hardware overhead and long test application time. Recently, a non-scan DFT approach for register transfer level (RTL) controller-datapath circuits has been proposed to reduce the test application time. However, the required hardware overhead becomes too large.

In this thesis, the main objective is to reduce the hardware overhead caused by the previous non-scan DFT approach for RTL circuits. First, partially strongly testability is defined as the testability of RTL circuits. Then, we propose a DFT method to make an RTL circuit partially strongly testable and a test generation method for partially strongly testable circuits based on time expansion model. In the experimental results, the proposed method can achieve complete fault efficiency within practical test generation time and drastically reduce hardware overhead and test application time compared to the previous approach.

## **Keywords:**

design-for-testability, register transfer level circuit, time expansion model, partially strong testability, complete fault efficiency

---

\*Doctoral Dissertation, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0561006, March 16, 2007.

# 研究業績

- 学術論文誌

1. 岩田浩幸, 米田友和, 大竹哲史, 藤原秀雄, “完全故障検出効率を保証する RTL データパスの部分強可検査性に基づくテスト容易化設計法,” 電子情報通信学会論文誌 (DI), Vol. J89-D No.8, pp.1463-1653, Aug. 2006.

- 国際会議 (査読付き)

1. Hiroyuki Iwata, Tomokazu Yoneda, Satoshi Ohtake and Hideo Fujiwara, “A DFT Method for RTL Data Paths Based on partially strong testability to Guarantee Complete Fault Efficiency,” 14th Asian Test Symposium, pp. 306-311, Dec. 2005.
2. Hiroyuki Iwata, Tomokazu Yoneda and Hideo Fujiwara, “A New Non-Scan DFT Method Based on the Time Expansion Model for RTL Controller-Datapath Circuits,” 7th Workshop on RTL and High Level Testing, pp. 7-12, Nov. 2006.
3. Hiroyuki Iwata, Tomokazu Yoneda and Hideo Fujiwara, “A Non-Scan DFT Method for the Time Expansion Model at the Register Transfer Level,” 44th Design Automation Conference (To appear)

- 研究会発表

1. 岩田 浩幸, 米田 友和, 大竹 哲史, 藤原 秀雄, “完全故障検出効率を保證するデータパスの部分強可検査設計について,” 第 52 回 FTC 研究会, Jan. 2005.
2. 岩田 浩幸, 米田 友和, 大竹 哲史, 藤原 秀雄, “完全故障検出効率を保證するデータパスの部分強可検査設計,” 信学技報 (DC2004-92), Vol. 104, No. 664, pp.1-6, Feb. 2005.

# 目次

|   |    |
|---|----|
| 第1章 序論                                  | 1  |
| 第2章 テスト容易化設計法                           | 4  |
| 2.1. 論理回路のテスト                           | 4  |
| 2.1.1 故障モデル                             | 4  |
| 2.1.2 テストの評価尺度                          | 4  |
| 2.1.3 回路のテスト生成                          | 5  |
| 2.2. テスト容易化設計                           | 6  |
| 2.2.1 完全スキャン設計                          | 6  |
| 2.2.2 レジスタ転送レベル回路                       | 8  |
| 2.2.3 RTL 回路の強可検査テスト容易化設計法              | 9  |
| 第3章 諸定義                                 | 12 |
| 3.1. 対象 RTL 回路                          | 12 |
| 3.2. 部分強可検査性                            | 14 |
| 第4章 部分強可検査性に基づくレジスタ転送レベルデータパスのテスト容易化設計法 | 19 |
| 4.1. 問題の定式化                             | 19 |
| 4.2. テスト容易化設計法                          | 20 |
| 4.2.1 制御線の決定                            | 21 |
| 4.2.2 依存関係の解消                           | 25 |
| 4.2.3 時間展開モデルの生成                        | 26 |
| 4.3. 実験結果                               | 29 |
| 4.4. むすび                                | 33 |

|       |                                  |    |
|-------|----------------------------------|----|
| 第 5 章 | 部分強可検査性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 | 34 |
| 5.1.  | 問題の定式化                           | 34 |
| 5.2.  | 提案手法の概要                          | 35 |
| 5.3.  | テスト容易化設計法                        | 36 |
| 5.3.1 | 制御林の決定                           | 37 |
| 5.3.2 | 観測林の決定                           | 39 |
| 5.3.3 | 依存関係の解消                          | 40 |
| 5.3.4 | テストコントローラの生成                     | 40 |
| 5.4.  | 部分強可検査 RTL 回路に対するテスト生成法          | 42 |
| 5.5.  | 実験結果                             | 45 |
| 5.6.  | むすび                              | 50 |
| 第 6 章 | 結論                               | 51 |
|       | 謝辞                               | 53 |
|       | 参考文献                             | 54 |

# 目次

|      |                              |    |
|------|------------------------------|----|
| 2.1  | 順序回路の時間展開モデル . . . . .       | 6  |
| 2.2  | 完全スキャン設計 . . . . .           | 7  |
| 2.3  | レジスタ転送レベル回路 . . . . .        | 8  |
| 2.4  | 文献 [14] の手法 . . . . .        | 9  |
| 2.5  | データパスの例 . . . . .            | 10 |
| 2.6  | ADD に対するテストプラン . . . . .     | 10 |
| 4.1  | スルー機能の実現例 . . . . .          | 20 |
| 4.2  | データパスの例 . . . . .            | 22 |
| 4.3  | 制御林の例 . . . . .              | 23 |
| 4.4  | m2 の制御経路 . . . . .           | 25 |
| 4.5  | 時間展開モデルの例 . . . . .          | 26 |
| 5.1  | [14] の DFT 例 . . . . .       | 36 |
| 5.2  | 提案手法の DFT 例 . . . . .        | 36 |
| 5.3  | RTL 回路の例 . . . . .           | 37 |
| 5.4  | 制御林の例 . . . . .              | 38 |
| 5.5  | 観測林の例 . . . . .              | 38 |
| 5.6  | ホールド機能の付加 . . . . .          | 39 |
| 5.7  | m2 の制御林 . . . . .            | 39 |
| 5.8  | テストコントローラの実出力パターン例 . . . . . | 42 |
| 5.9  | 部分強可検査 RTL 回路 . . . . .      | 43 |
| 5.10 | 時間展開モデルの例 . . . . .          | 44 |

# 表目次

|     |                       |    |
|-----|-----------------------|----|
| 4.1 | データパスの特性 . . . . .    | 29 |
| 4.2 | 面積オーバーヘッド . . . . .   | 30 |
| 4.3 | 故障検出効率 . . . . .      | 30 |
| 4.4 | テスト生成時間 . . . . .     | 31 |
| 4.5 | テスト実行時間 . . . . .     | 31 |
| 5.1 | RTL 回路全体の面積 . . . . . | 45 |
| 5.2 | コントローラの特徴 . . . . .   | 45 |
| 5.3 | データパスの特性 . . . . .    | 46 |
| 5.4 | 面積オーバーヘッド . . . . .   | 47 |
| 5.5 | 外部ピンオーバーヘッド . . . . . | 48 |
| 5.6 | 故障検出効率 . . . . .      | 48 |
| 5.7 | テスト生成時間 . . . . .     | 49 |
| 5.8 | テスト実行時間 . . . . .     | 49 |

# 第 1 章

## 序論

VLSIの大規模化，複雑化に伴い，VLSIのテストはますます困難な問題となっており，テストの費用の削減及びテストの質の向上が求められている．一般的なテストは，故障の有無で出力が異なるような入力系列（テスト系列）を外部テストタで印加しその応答を観測することで行われる．このようなテスト系列を求める操作をテスト生成という．テスト費用を示す評価尺度として，テスト生成時間やテスト実行時間がある．テストの質を示す評価尺度として，故障検出効率がある．故障検出効率とは，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，回路中のテスト生成の対象となる故障を検出または冗長と判定できる割合のことをいう．特に，故障検出効率が 100 % のとき，完全故障検出効率という．ここで，冗長故障数とは，回路の動作に影響を与えない故障をいう．本研究で対象とする故障モデルは，単一縮退故障とする．

組合せ回路のテスト生成は，効率の良いテスト生成アルゴリズムが考案されており，実用的なテスト生成時間で完全故障検出効率を達成するテスト系列が生成可能である [1]．これに対して，順序回路に対するテスト生成では，一般に回路を時間展開する方法が用いられる [1]．これは，順序回路をフリップフロップ (FF) のない組合せ回路（時間展開モデル）として考えてテスト生成を行う方法である．この時間展開モデルにおいて，検出可能な故障を全て検出するのに必要な時間展開数は，最悪の場合，回路中の FF 数の指数のオーダーの複雑度になる可能性があり，順序回路のテスト生成に膨大な時間がかかる．しかし，無閉路順序回路の

時間展開モデルに対しては，実用的なテスト生成時間によるテスト生成が可能である [2, 3] .

テスト困難な順序回路に対して，テスト容易な回路に設計変更するテスト容易化設計が提案されている．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],[4] がある．この手法では，スキャン FF を擬似外部入力及び擬似外部出力とみなすことで，組合せ回路用のテスト生成アルゴリズムが適用可能である．従って，実用的なテスト生成時間で完全故障検出効率を達成するテスト系列を生成することができる．しかし，生成したテスト系列の印加のためにはスキャンシフト動作が必要となるため，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る．また，FF の数に比例してテスト実行時間が爆発的に大きくなるという問題点もある．このため，将来的には，外部テストのメモリにテスト系列が格納できなくなり，テスト実行が困難になる問題が発生する可能性がある．これらの問題点を解消するために，レジスタ転送レベル (RTL) のコントローラ / データパスを対象とする非スキャンテスト容易化設計法が提案されている [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18] .

文献 [14] では，RTL コントローラ / データパス回路全体に対して完全故障検出効率を達成するテスト容易化設計法が提案されている．文献 [14] の手法では，コントローラに対しては文献 [15] の方法を適用し，データパスに対しては文献 [16] の手法を適用する．そして，テストコントローラを回路内部に付加し，そのテストコントローラからデータパスへ制御信号を供給することで，少ないピンオーバーヘッドで実動作速度テストを可能としている．文献 [14] の手法は，組合せテスト生成が適用可能であり，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，完全スキャン設計法と比較して，大幅にテスト実行時間を削減することに成功している．しかしながら，テストコントローラは，各回路要素に対するテストプランおよび，その回路要素の制御入力へのテストパターンを発生する必要がある．その面積は大きい．また，テストコントローラからの制御入力と通常動作時のコントローラからの制御入力を切り替えるための MUX も必要となる．このため，面積オーバーヘッドは完全スキャン設計法よりも大きくなるという問題点がある．

そこで、本研究では、RTL コントローラ/データパスを対象とし、面積オーバーヘッドを削減するテスト容易化設計法を提案する。提案手法では、RTL 回路の時間展開モデルに対して、実用的なテスト生成時間で完全故障検出効率を達成するための性質として、部分強可検査性を定義する。部分強可検査 RTL 回路は、文献 [14] と比較してテスト実行時間をさらに削減可能であり、完全故障検出効率を達成可能、実動作速度テストが可能であるという多くの利点を持つ。まず、4 章では、RTL データパスのみを対象とし、[16] と比較して面積オーバーヘッドを削減しテスト容易化設計法を提案する。次に、5 章では、4 章の手法を RTL 回路全体に拡張したテスト容易化設計法を提案する。部分強可検査 RTL 回路では、データパスへの制御信号は、テストコントローラではなく可能な限り通常動作のコントローラを用いてデータを供給する。また、コントローラへの状態信号は、データパスを用いて供給する。これにより、文献 [14] に比べ、データパスとコントローラを分離するための MUX およびテストコントローラに要求される出力パターン数は少なくなり、その面積および遅延オーバーヘッドを削減することが可能となる。また、対象回路全体を同時にテスト実行することにより、テスト実行時間の短縮が可能となる。

## 第 2 章

# テスト容易化設計法

本章では，故障モデルと回路のテスト生成について述べる．次に，従来のテスト容易化設計法について説明する．

### 2.1. 論理回路のテスト

#### 2.1.1 故障モデル

論理回路（以降回路と呼ぶ）を構成する要素に存在する物理的欠陥を回路の故障という．回路が実現している論理機能を変化させる故障を論理故障とよぶ．代表的な論理故障には，回路内の信号線の値が 1 または 0 に固定される縮退故障がある．信号線  $l$  に発生し得る縮退故障は， $l$  が論理値 1 に固定される 1 縮退故障と，0 に固定される 0 縮退故障に分けられる．1 個の論理回路において同時に 1 個の故障しか存在しないと仮定したものを単一縮退故障という．本論文では，単一縮退故障のみを対象とする．

#### 2.1.2 テストの評価尺度

回路のテストとは，論理回路上の故障の有無を調べることを指す．回路にテスト系列を印加したときの出力応答を観測し，期待値と比較することで故障を検出

する。テストは、テスト系列を生成するテスト生成と、外部テストを用いてテスト系列を回路に印加しその応答を観測するテスト実行から成る。一般にテスト生成は自動テストパターン生成系 (ATPG) を用いて行われる。ATPG は、故障に対してテストパターンの生成を試み、故障に対するパターンを求めめるか、冗長であるかを判断する。その際、計算時間の問題から冗長判定を打ち切ることがある。テスト生成は設計した回路に対して一度行われるのに対し、テスト実行は製造した回路毎に必要となる。

テスト費用を示す評価尺度として、テスト生成時間やテスト実行時間がある。また、テストの質を示す評価尺度として、故障検出効率がある。故障検出効率とは、ATPG によって生成されたテスト系列が、回路中のテスト生成の対象となる故障を検出または冗長と判定できる割合のことをいい、次式で与えられる。

$$\text{故障検出効率} = \frac{\text{検出故障数} + \text{冗長故障数}}{\text{全故障数}}$$

特に、故障検出効率が 100 % のとき、完全故障検出効率という。ここで冗長故障とは、任意の入力系列に対して故障が回路の出力応答に影響を与えない故障をいう。また、回路中のテスト生成の対象となる故障を検出できる割合を故障検出率という。

### 2.1.3 回路のテスト生成

回路には、組合せ回路と順序回路がある。組合せ回路は、回路の出力値が印加された入力値にのみ依存する回路である。これに対して順序回路は、回路の出力値が印加された入力値と回路の内部状態に依存する回路である。現在、組合せ ATPG は効率のよいテスト生成アルゴリズムが提案されており、実用的なテスト生成時間で完全故障検出率を達成することができる。

順序回路に対するテスト生成は、図 2.1 に示すように回路を時間展開することにより全体として FF のない組合せ回路 (時間展開モデル) として考えることができる。時間展開モデルでは、元の順序回路の単一故障が、同一の組合せ論理部が繰り返し現れることにより、多重故障として現れる。従って、時間展開モデルに対するテスト生成では多重故障を扱う必要がある。図 2.1 において、 $Y(t)$  は時

刻  $t$  における FF の状態に対応し  $X(t)$  および  $Z(t)$  は  $t$  の外部入力および外部出力を表す．また，回路  $C$  は順序回路の組合せ回路部を表す．時間展開モデルに対して組合せ ATPG をして求めたテスト系列を元の順序回路に適用できるように変換することで，順序回路中の故障を発見できる．

順序回路中の故障を検出するのに必要な時間展開数は，最悪の場合，回路中の FF が全ての状態に遷移する時間展開数が必要となり，多くの計算時間を必要とする．順序 ATPG では，一般的に実用的な時間で高い故障検出効率を持つテスト系列を求めることが困難である．そこで，順序回路のテスト生成が容易となるようにハードウェアを付加することで，故障検出効率又はテスト実行時間を改善するテスト容易化設計法が提案されている．

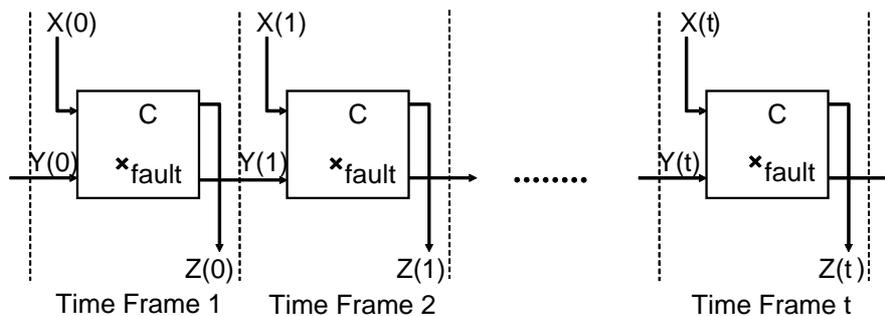


図 2.1 順序回路の時間展開モデル

## 2.2. テスト容易化設計

### 2.2.1 完全スキャン設計

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がある．完全スキャン設計法では，順序回路に含まれる全ての FF をスキャン FF に置き換える．スキャン FF は，通常動作時のデータ入力とスキャン入力を切替えて FF に入力可能にしたもので，FF に MUX を付加することで実現できる．全てのスキャン FF が，外部から制御及び観測可能な 1 個かまたは複数個のシフトレジスタとして動作する

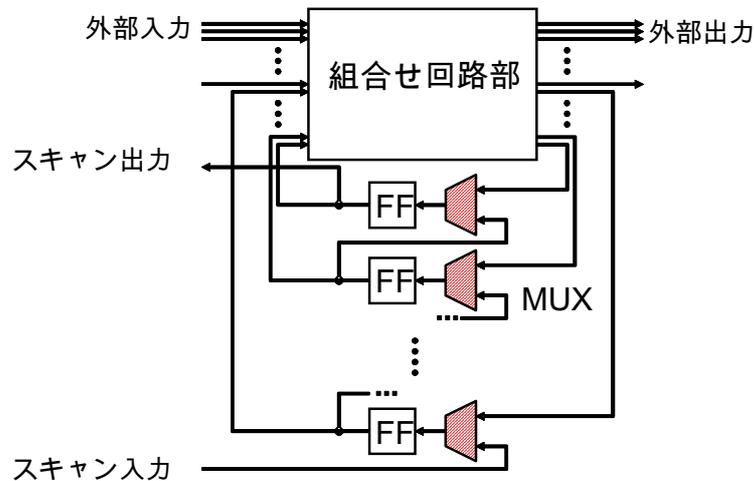


図 2.2 完全スキャン設計

ようにスキャンモードを回路に付加する．回路をスキャンモードに設定しシフトレジスタのシフト動作させることにより，全ての FF に対して任意の値の印加及び観測ができるようになる．従って，全てのスキャン FF を擬似外部入力及び擬似外部出力とみなすことができるので，テスト生成の対象となる回路（核回路）は組合せ回路となる．この組合せに対して，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で，実用的な時間で完全故障検出効率を達成するテスト系列を生成することができる．しかし，この手法には以下のような問題点がある．

- 全ての FF をスキャン FF に置き換えるので，付加ハードウェアによる面積オーバーヘッドが大きい．
- テスト実行の際に，スキャンシフト動作を行うので，テスト実行時間が長くなる．
- スキャンシフト動作は，通常動作時におけるクロックよりも遅いクロックを用いるため，実動作速度テスト実行（at-speed test）が困難である．
- FF 数に比例してテスト実行時間が増大するため，将来的には外部テストの

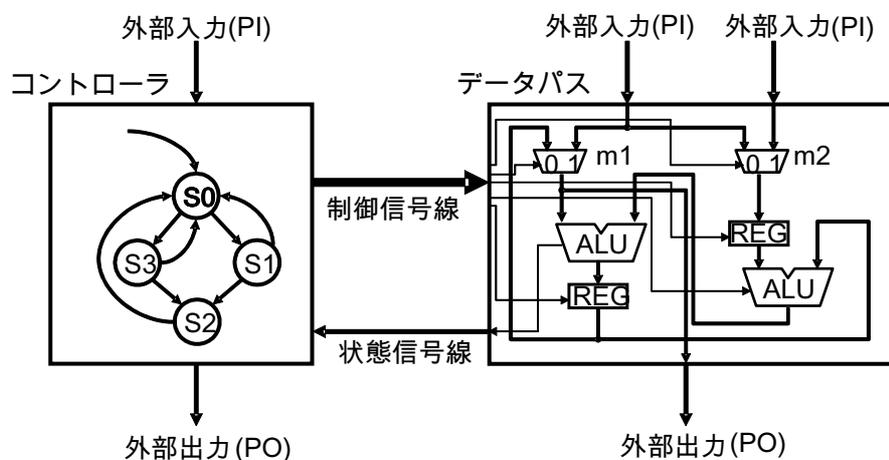


図 2.3 レジスタ転送レベル回路

メモリにテスト系列が収まりきらなくなり，テスト実行が困難になる可能性がある．

スキャン設計におけるテスト実行時間及び実動作速度テストが困難である点を改善するためのテスト容易化設計として，非スキャンテスト容易化設計法 [13, 14, 16, 17] がある．非スキャンテスト容易化設計では，回路の通常動作時に用いるデータ転送経路を用いて FF の制御及び観測ができるように回路を設計変更する．従って，実動作速度テスト実行が可能となり，テスト実行時間を短くできる．

## 2.2.2 レジスタ転送レベル回路

近年，論理合成前のレジスタ転送レベル (RTL) 回路の情報を用いることで，完全スキャン設計法と比較してテスト実行時間が削減可能なテスト容易化設計法が提案されている．RTL 回路は，一般にコントローラとデータパスからなる．データパスは入力されるデータを処理し，コントローラはデータパスの動作を制御する．コントローラは状態遷移図で記述され，データパスは回路要素と回路要素同士を接続する信号線で記述される．回路要素は，レジスタ，マルチプレクサ (MUX)，演算モジュールなどに分類される．信号線は，データ信号線，制御信

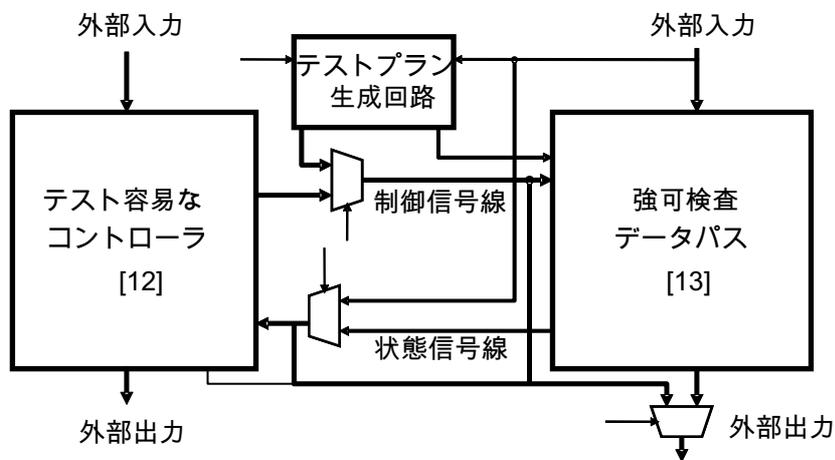


図 2.4 文献 [14] の手法

号線，状態信号線に分類される．データ信号線は各回路要素の入出力間を接続する．制御信号線は，コントローラからデータパスの各回路要素を制御するための信号を与える信号線である．状態信号線は，データパスの各回路要素の状態をコントローラへ出力するための信号線である．

### 2.2.3 RTL 回路の強可検査テスト容易化設計法

実動作速度テストが可能であり，完全故障検出効率を達成できる RTL 回路全体を対象としたテスト容易化設計の一つとして，文献 [14] の手法がある．この手法では，与えられた RTL コントローラ / データパス回路に対して，コントローラに対しては文献 [15] の手法と適用し，データパスに対しては文献 [16] の手法を適用する．これらの手法は，制御信号線および状態信号線が外部から制御及び観測が可能であると仮定していたため，適用するために，コントローラとデータパス間に MUX を付加する（図 2.4）．これにより，コントローラに対しては，データパスの外部入出力を用いることで，任意の値の印加及び観測が可能となる．従って，文献 [15] の手法が適用可能となり，完全故障検出効率が達成できる．

データパスに対しては，テスト用のコントローラ（テストプラン生成回路）を用いて，データパスに必要な制御信号を供給する．テストプランとは，外部入力

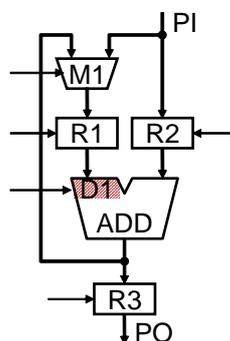


図 2.5 データパスの例

| 時刻 | PI | PO | M1 | R1 | R2 | R3 | D1 |
|----|----|----|----|----|----|----|----|
| 1  | T  | x  | x  | x  | 1  | x  | x  |
| 2  | x  | x  | 1  | 1  | 0  | x  | x  |
| 3  | x  | x  | x  | x  | x  | 1  | T  |
| 4  | x  | R  | x  | x  | x  | x  | x  |

T: テストパターン  
R: 応答観測  
x: don't care

図 2.6 ADD に対するテストプラン

からテスト対象となる回路要素の入力へテストパターンを印加し，その出力応答を外部出力へ伝播するための制御入力の時系列をいう．文献 [16] の手法は，階層テスト生成 [19, 20] に基づいており，以下の 2 段階でテスト生成を行う．初めに，組合せ回路である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してテスト生成を行う．次に，各回路要素に対して外部入力から回路要素の入力へ任意の値を伝搬し，その回路要素の任意の出力を外部出力へ伝搬するためのテストプランを求める．文献 [16] では，データパスの全ての回路要素に対してテストプランの存在を保証する性質として強可検査性を定義している．

定義 1 (データパスの強可検査性) データパス  $DP$  の任意の回路要素  $M$  が，以下の 2 つの条件を満たすとき， $DP$  は強可検査であるという．

1. 外部入力から任意の値を  $M$  のデータ入力端子に伝達可能 (強可制御性)
2.  $M$  のデータ出力端子に表れる任意の値を外部出力まで伝達可能 (強可観測性)

□

文献 [16] のテスト容易化設計法では，データパス中の全ての回路要素に対してスルー機能を付加することで，データパスを強可検査にする．図 2.5 に強可検査データパスの例を示す．図中の ADD に対するテストプランを図 2.6 に示す．ADD 単体に対してテスト生成を行い，図 2.6 のテストプランを用いて生成したテストパターンの印可及び応答の観測を行うことで，ADD に対するテスト実行が可能

である．各回路要素に対するテスト生成では，組合せ回路用のテスト生成アルゴリズムが適用可能であり，容易に完全故障検出効率を達成できる．データパスが強可検査性を満たすとき，各回路要素毎にテストプランを用いて，各回路要素に対するテスト系列の印加及び応答の観測を行うことで，データパス全体に対して完全故障検出効率を達成可能である．従って，RTL 回路全体での完全故障検出率が達成可能となる．

文献 [14] の手法では，完全スキャン設計法と比較してテスト実行時間を大幅に削減することに成功している．しかしながら，データパスの回路要素数の増加に伴い面積オーバーヘッドは大きくなる．また，テストコントローラのアレイオーバーヘッドが大きいため，結果としてテスト容易化設計に必要な面積オーバーヘッドは完全スキャン設計よりも大きくなるという問題点が残されている．

文献 [17] の手法では，テストコントローラに代わり既存のコントローラに対して，テストプランを出力させる状態遷移を追加する．これにより，面積オーバーヘッドを大幅に削減することに成功している．しかし，テストコントローラを使う場合よりテストプラン長が長くなる場合が存在するため，テスト実行時間が長くなる場合がある．また，既存のコントローラに対して状態遷移を追加していくため，テストコントローラの付加に比べて遅延オーバーヘッドが大きくなる可能性があるという問題点がある．

## 第 3 章

### 諸定義

本論文では，文献 [14] の手法の利点である短いテスト実行時間及び実動作速度テスト可能を達成し，文献 [14] の手法より面積オーバーヘッドを削減したテスト容易化設計法の提案を目標とする．また，本論文で提案するテスト容易化設計法は，レジスタ転送レベル（RTL）回路を対象とする．提案手法では，実用的なテスト生成時間で完全故障検出効率を達成する時間展開モデルを生成可能な RTL 回路の性質として部分強可検査性を新しく導入し，それに基づくテスト容易化設計法及びテスト生成法を提案する．本章では，提案手法が対象とする RTL 回路及び，部分強可検査性の定義を行う．

#### 3.1. 対象 RTL 回路

RTL 回路はコントローラとデータパスで構成される．コントローラは，状態レジスタ（SR）と組合せ論理回路（CL）で構成される．データパスは，外部入力（PI），外部出力（PO），ホールドレジスタ（HR），ロードレジスタ（LR），マルチプレクサ（MUX），演算モジュール及び観測モジュールで構成されとする．各回路要素は，入出力端子を持ち，入出力端子は，データ入力端子，データ出力端子，制御入力端子，制御出力端子，状態入力端子，状態出力端子に分類される．

以下に，具体的な各回路要素の入出力端子の構成を示す．

外部入力（PI）: 1 個のデータ出力端子を持つ．

外部出力 (PO): 1 個のデータ入力端子を持つ。

組合せ論理回路 (CL): 2 個のデータ入力端子, 高々2 個のデータ出力端子, 複数の制御出力端子及び状態入力端子を持つ。CL とデータパスの各回路要素を接続する入出力端子が, CL の制御出力端子及び状態入力端子となる。

状態レジスタ (SR): 1 個のデータ入力端子, 1 個のデータ出力端子を持つ。

ホールドレジスタ (HR): 1 個のデータ入力端子, 1 個のデータ出力端子および, 1 個の制御端子を持つ。

ロードレジスタ (LR): 1 個のデータ入力端子と 1 個のデータ出力端子を持つ。

マルチプレクサ (MUX): 2 個のデータ入力端子, 1 個のデータ出力端子。1 個の制御端子を持つ。

演算モジュール (CM): 1 個または 2 個のデータ入力端子, 1 個のデータ出力端子, 高々1 個の制御端子, 高々1 個の観測端子の観測端子を持つ。CM はさらに CMA と CMB の 2 つの種類に分類する。CMA は出力端子の値を入力端子の値の関数で表したとき, 出力関数が全射となる CM とし, CMB は全射とならない CM をとする。具体的には, CMA はデータ入力端子のビット幅が等しい加算器や減算器や乗算器である。また, CMB は定数倍の乗算器のように出力の一部のビットが常に固定値となるような演算モジュールである。

観測モジュール (OM): 1 個または 2 個のデータ入力端子, 高々1 個の制御端子, 1 個の観測端子を持つ。

各回路要素のデータ入力端子は, 少なくとも 1 つの外部入力から到達可能であり, 各回路要素のデータ出力端子は, 少なくとも 1 つの外部出力へ到達可能であるとする。CL, MUX, CM, OM を組合せ回路要素と呼ぶ。

各回路要素は信号線で接続される。信号線は, データ信号線, 制御信号線, 状態信号線に分類する。データ信号線は, 回路要素のデータ出力端子とデータ入力端子を接続する。制御信号線は, CL の各制御出力端子から, データパスの各回

路要素の制御入力端子に接続される．状態信号線は，データパスの各回路要素の状態出力端子から，CLの各状態入力端子に接続される．データ入力端子に接続可能なデータ信号線は1本とし，データ出力端子には複数のデータ信号線を接続（ファンアウト）可能とする．本研究では，データパスに対する設計制約として，データパスのデータ信号線のビット幅は全て等しいものとする．あらゆるデータパスにおいて，データパスの信号線のビット幅は等しくなるように設計変更可能である．

$e_i$  入出力端子とし， $l_i$  を信号線とする経路  $p = (e_1, l_1, e_2, \dots, e_{k-1}, l_{k-1}, e_k)$  について考える． $p$  にあられるレジスタ数を  $p$  の順序深度という．また，経路  $p_1$  の終点及び経路  $p_2$  の始点が同一であるとき，それらを接続した経路を  $(p_1, p_2)$  と表す． $p$  の全ての  $e_i$  が異なるとき， $p$  を単純経路という． $p$  の始点  $e_1$  と終点  $e_k$  が同一であり， $e_2$  から  $e_{k-1}$  までの経路が単純経路であるとき， $p$  を閉路という．異なる2つの頂点  $e_i, e_j$  に対して， $e_i$  を始点とし， $e_j$  を終点とする任意の異なる単純経路の組を再収斂経路の組という． $p$  上に存在する回路要素の入出力端子に対して， $p$  上に存在する入出力端子を経路上入力端子，経路上出力端子， $p$  上に存在しない入出力端子を経路外入力端子，経路外出力端子という．

## 3.2. 部分強可検査性

まず，部分強可検査性の定義のために必要な要素として，信号線の値域及び，信号線の依存関係を定義する．それから，部分強可検査性の定義とその定理を述べる．

定義 2 (信号線の値域)  $PI$  から各信号線  $l$  に正当化可能な全ての値の集合を  $l$  の値域という．信号線に接続する入力 / 出力端子の値域は，信号線の値域で定義する．また，外部入力から各回路要素  $m$  の全ての入力端子に同時に正当化可能な全ての値の集合を  $m$  の入力端子の値域という． □

定義 3 (信号線の依存関係) 時刻  $t$  における信号線  $l_i$  と時刻  $t'$  における信号線  $l_j$  において，各々の値域の任意の値が独立して設定不可能なとき， $t$  における  $l_i$  と  $t'$  に

おける  $l_j$  は依存関係を持つという．入出力端子における依存関係は，入出力端子に接続する信号線  $l_i, l_j$  の依存関係で定義する．  $\square$

定義 4 (RTL 回路の部分強可検査性) RTL 回路  $CD$  が，以下の条件 (a) (b) を満たすとき， $CD$  は部分強可検査であるという．

(a)  $CD$  に存在する閉路の集合を  $L$  とする．任意の閉路  $c \in L$  に対して，以下の条件を満たす経路  $p_c$  が存在する．

1. ある  $PI$  から  $c$  上の回路要素  $m_{cin}$  の  $c$  の経路上出力端子  $out_{cin}$  までの単純経路を  $p_{cin}$  とする．ただし， $m_{cin}$  以外は， $c$  上の回路要素を含まない． $c$  上の回路要素  $m_{cout}$  の  $c$  の経路上出力端子からある  $PO$  までの単純経路を  $p_{cout}$  とする． $out_{cin}$  を始点且つ終点とする  $c$  を  $p_{c1}$  とする． $out_{cin}$  から  $m_{cout}$  の経路上出力端子までの  $c$  上の単純経路を  $p_{c2}$  とする．このとき， $p_c = (p_{cin}, p_{c1}, p_{c2}, p_{cout})$  である．
2.  $p_{cin}$  上の任意の回路要素  $m_i$  の  $p_{cin}$  の経路上入力端子の任意の値を経路上出力端子に伝搬させる値が  $m_i$  の経路外入力端子の値域に対して存在する．ただし， $m_{cin}$  については  $in_a$  の値域に対してその値が存在する．ここで  $in_a$  は， $m_{cin}$  の  $p_{cin}$  の経路上入力端子及び  $c$  の経路上入力端子以外の入力端子とする．
3.  $(p_{c1}, p_{c2}, p_{cout})$  上の任意の回路要素  $m_i$  の経路上入力端子の任意の値の変化を経路上出力端子に伝搬させる値が  $m_i$  の経路外入力端子の値域に対して存在する．
4.  $p_c$  上の任意の回路要素  $m_i$  に対して， $p_c$  上の  $PI$  から  $m_i$  までの順序深度を  $d_i$  とする．このとき，時刻  $t$  における  $p_c$  上の  $PI$  と，時刻  $t + d_i$  における  $m_i$  の経路外入力端子は依存関係を持たない．ただし， $p_{cin}$  上における  $m_{cin}$  については，時刻  $t$  における  $p_c$  上の  $PI$  と，時刻  $t + d_i$  における  $in_a$  のみが依存関係を持たない．
5.  $p_c$  上に存在する任意の回路要素  $m_i, m_j$  に対して， $p_c$  上の  $PI$  から  $m_i, m_j$  までの順序深度を  $d_i, d_j$  とする． $d_i = d_j$  且つ  $m_i$  が  $m_j$  で再収斂する順序深度 0 の再収斂経路上に存在するとき，時刻  $t + d_i$  における  $m_i$  及び  $m_j$  の  $p_c$  の経路

外入力端子に，経路上入力端子の任意の値又は任意の変化を経路上出力端子に伝搬させる値が印加可能．それ以外するとき，時刻  $t + d_i$  における  $m_i$  の経路外入力端子は，時刻  $t + d_j$  における  $m_j$  の経路外入力端子とそれぞれ依存関係を持たない．ただし， $p_{cin}$  上における  $m_{cin}$  については， $in_a$  についてのみ同様の条件を満たす．

(*b*)  $CD$  の回路要素について，任意の  $PI$  から入力端子までの単純経路のうち，閉路の一部を通る単純経路が存在する回路要素の集合を  $M$  とする．(*a*) の条件を満たす経路の集合を  $P$  とする．任意の回路要素  $m \in M$  に対して， $m$  を通る経路  $p \in P$  が存在する． □

**定理 1** 部分強可検査  $RTL$  回路を  $CD$  とする． $CD$  中の任意の回路要素を  $m$  とし， $m$  に存在する故障  $f_m$  に対して， $f_m$  を検出可能な  $CD$  への入力系列が存在するとする．ある  $PI$  から  $m$  の入力端子までの単純経路と  $m$  の出力端子から  $PO$  までの単純経路を接続した経路のうち，順序深度が最大となる経路の順序深度を  $d_p$  とする． $CD$  中に存在するレジスタの総数を  $n_{REG}$  とする．このとき，時間展開数が高々  $d_p + n_{REG}$  で， $f_m$  を検出可能な時間展開モデルが生成可能である． □

**証明 1** 回路要素  $m$  がある経路  $p_c \in P$  上に存在する場合について考える．部分強可検査性の条件 (*a*) の (4)(5) より， $m_{cin}$  を除く  $p_{cin}$  上の任意の回路要素の入力端子に対して，その値域の任意の値が正当化可能である．さらに，部分強可検査性の条件 (*a*) の (2) より， $p_{cin}$  上の全ての経路上入出力端子には任意の値が正当化可能であるため， $out_{cin}$  には任意の値の正当化が可能である．これより， $p_{cin}$  上の回路要素と同様に， $(p_{c1}, p_{c2}, p_{cout})$  上の任意の回路要素の入力端子についても，その値域の任意の値が正当化可能である． $m_{cin}$  については， $(p_{c1}, p_{c2}, p_{cout})$  上において， $m_{cin}$  の入力端子にその値域の任意の値が正当化可能である．以上より， $p_c$  上の全ての回路要素の入力端子には，その値域の任意の値が正当化可能である．従って， $m$  が  $p_c$  上に存在し， $f_m$  を検出可能な  $CD$  への入力系列が存在するとき， $f_m$  を活性化可能である．同様に，部分強可検査性の条件 (*a*) の (2)(3)(4)(5) より， $p_c$  上の任意の回路要素は経路上入出力端子間の任意の誤りを伝搬可能であるため， $f_m$  の誤りを  $PO$  まで伝搬可能である．

次に、 $m$  が任意の経路  $p \in P$  上に存在しない場合について考える。このとき、任意の  $PI$  から  $m$  の入力端子までの経路は閉路を含まないため、 $PI$  から  $m$  の入力端子までは無閉路構造である。従って、 $m$  の入力端子に対してはその値域の任意の値が印加可能であり、 $f_m$  を検出可能な  $CD$  への入力系列が存在するとき、 $f_m$  を活性化可能である。 $m$  から任意の  $PO$  までの単純経路の集合を  $P_{mo}$  とする。ある経路  $p \in P_{mo}$  が  $P$  のどの経路とも共有部分を持たないとき、 $m$  から  $p$  上の  $PO$  までは無閉路構造である。ある経路  $p \in P_{mo}$  が  $P$  のある経路と共有部分を持つとき、 $m$  から共有部分までは無閉路構造であり、共有部分は部分強可検査の条件 (a) を満たす経路であるため、その経路を通り  $PO$  まで誤りが伝搬可能である。従って、 $f_m$  を検出可能な  $CD$  への入力系列が存在するとき、 $f_m$  の誤りを  $PO$  まで伝搬可能である。

ある  $PI$  から  $m$  の入力端子まで及び  $m$  の出力端子からある  $PO$  までを接続した単純経路を  $p_m$  とするとき、その順序深度は高々  $d_p$  である。 $p_m$  上の回路要素  $m_i$  及び  $m_j$  に対して、 $p_m$  上における  $PI$  からの順序深度をそれぞれ  $d_i$  及び  $d_j$  とする。このとき、時刻  $t + d_i$  における  $m_i$  の経路外入力と時刻  $t + d_j$  における  $m_j$  の経路外入力には、依存関係のない値域の任意の値を高々  $n_{REG}$  の時間展開数で設定可能である。従って、最も時間展開数が大きくなるのは  $p_m$  の順序深度が  $d_p$  のとき、 $p_m$  上の順序深度が 0 である  $m_p$  の経路外入力に対して  $n_{REG}$  の時間展開数が必要になるときである。以上より、時間展開数が高々  $d_p + n_{REG}$  で  $f_m$  の活性化及び伝搬が可能である。□

$f_m$  を検出可能な  $CD$  への入力系列が存在しないとき、各回路要素の入力端子に、その値域の任意の値を印加しても  $f_m$  の活性化又は伝搬ができないため、 $f_m$  は冗長故障である。順序回路におけるレジスタの初期値によっては検出可能となる故障 (potentially detectable faults) は時間展開モデル上では冗長故障として判定される。生成する時間展開モデルでは各信号線における値域の値が正当化及び伝搬可能であるため、リセット状態からの回路動作で検出可能な全ての故障は検出可能である。時間展開モデル上で冗長故障と判別される故障は、リセット状態からの回路動作で検出不可能であるため通常の回路動作に影響しない。従って、冗長故障として判定しても問題ない。本論文では、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が達成可能であると考える。また、 $d_p + n_{REG}$  程度の時間展開数の時間展開モデルに対しては、実用的なテスト生成時間で完全故障検出効率が達成可能であると考える。これらの前提のもとでは、定理 1 から、部分強可検査 RTL 回路は実用的なテスト生成時間で完全故障検出効率が達成可能であるといえる。

## 第 4 章

# 部分強可検査性に基づくレジスタ転送 レベルデータパスのテスト容易化設 計法

初めに，RTL データパスのみを対象とし，部分強可検査性に基づくテスト容易化設計法を提案する．提案手法では，部分強可検査性に基づいたテスト容易化設計を行うため，強可検査性に基づくデータパスのテスト容易化設計法 [16] と比較して，面積オーバーヘッドの削減が期待できる．また，提案手法では，完全スキャン設計法と比べてテスト実行時間を大幅に短縮でき，実動作速度テストが可能である．本章では，始めに問題の定式化を行い，次に，提案するテスト容易化設計法の説明をする．最後に，ベンチマーク回路及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．

### 4.1. 問題の定式化

提案するテスト容易化設計では，与えられたデータパスを部分強可検査にする．与えられたデータパスにおいて，CM のデータ出力端子に任意の値を設定可能とするためにスルー機能を付加する．また，CM のデータ入力端子間に依存関係を持つ場合は，CM に隣接する LR にホールド機能を付加し，依存関係を解消する．

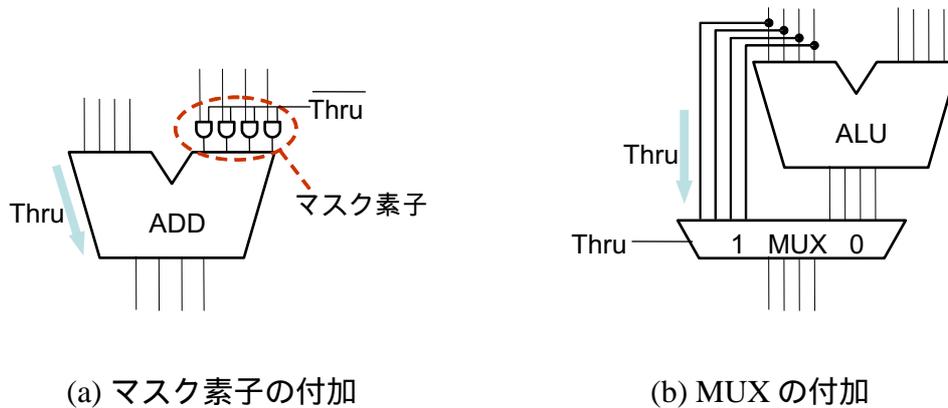


図 4.1 スルー機能の実現例

従って、付加ハードウェアはスルー機能及びホールド機能となる。スルー機能は、加算器や乗算器などの2入力演算モジュールであれば、マスク素子を利用することで実現できる。マスク素子とは、2入力演算モジュールの一方のデータ入力端子とデータ出力端子間の値を伝搬するために、もう一方のデータ入力端子に必要な定数を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演算モジュールは、MUXを付加することによりスルー機能を実現する。また、ホールド機能は、LRにMUXを付加することにより実現する。部分強可検査性のためのテスト容易化設計を、次の最適化問題として定式化する。

定義 5 (部分強可検査性に基づくテスト容易化設計問題)

- 入力：レジスタ転送レベルデータパス
- 出力：部分強可検査データパス，時間展開モデル
- 最適化目標：付加ハードウェア量最小

□

## 4.2. テスト容易化設計法

部分強可検査性に基づくテスト容易化設計のための発見的アルゴリズムの概要を示す。アルゴリズムは以下の4段階からなる。

1. 制御林の決定：与えられたデータパス中の各データ入力端子に対して，PI から値域の任意の値の伝搬を保証する単純経路（制御経路）を決定する．この制御経路の集合を制御林と呼ぶ．この制御経路で値域の任意の値が伝搬できない場合は，値域の任意の値の伝搬を保証するために，経路中の CM にスルー機能を付加する．
2. 観測林の決定：  
任意の回路要素からの PO までの経路上に存在する回路要素のデータ入力端子に依存関係を持ち，その依存関係が通常動作時に解消できる場合には，レジスタのホールド機能を用いて依存関係を解消する．レジスタにホールド機能がない場合は，ホールド機能を付加する．
3. 依存関係の解消：制御林上の経路を用いて値を伝搬したときに，各回路要素のデータ入力端子間に依存関係を持つかどうかを調べる．依存関係を持つ場合，各回路要素に対して依存関係を解消できるレジスタおよび，そのレジスタのホールドを行うサイクル数を決定する．レジスタにホールド機能がない場合は，ホールド機能を付加する．
4. 時間展開モデルの生成：部分強可検査データパスのテスト生成のために全回路要素に対して単一の時間展開モデルを生成する．任意の回路要素に対して，制御林上の経路を用いて値が伝搬できるように時間展開モデルを生成する．依存関係を解消する場合に，LR にホールド機能が必要となる場合は，ホールド機能を付加する．

#### 4.2.1 制御林の決定

提案する DFT アルゴリズムでは，PI から順に，スルー機能を付加せずに値域の値が伝搬できる回路要素及び経路を優先的に探索し，制御林として決定する．以下に制御林決定の手続きに用いる変数及び集合を定義する．任意の回路要素を  $m$  とする． $m$  のデータ出力端子に接続している回路要素のうち， $m$  と回路要素間

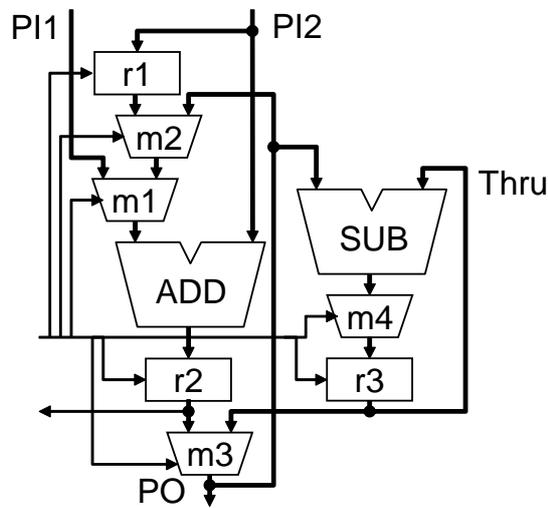


図 4.2 データパスの例

の経路が未決定である回路要素を  $U(m)$  と表し、その回路要素の数を  $n_m$  とする。データ入力端子に接続している経路が制御林に決定されており、データ出力端子に接続している経路が探索の対象となる回路要素の集合を  $C_C$ 、対象でない組合せ回路要素の集合を  $C_{CM}$ 、対象でないレジスタの集合を  $C_{REG}$  とする。 $C_C$  中の回路要素のうち、 $n_m$  が最小となる回路要素の集合を  $C_M$  とする。制御林を決定する手順を次に示す。

1. 全ての回路要素と経路を未決定に設定する。 $C_C$  を PI の集合で初期化し、 $C_{REG}$ 、 $C_{CM}$  を空集合に設定する。
2. 以下の操作を  $C_M$  が空になるまで繰り返し実行する。回路要素  $m \in C_M$  を選択し、 $m$  を  $C_M$  から削除する。 $U(m) = \phi$  であるとき、 $C_C$  から  $m$  を削除する。 $U(m) \neq \phi$  であるとき、1 個の  $m_t \in U(m)$  に対し、 $m_t$  及び  $(m, m_t)$  間の経路を制御林に決定し以下の操作を行う。
  - $m_t$  が MUX であり、 $m_t$  のデータ出力端子に接続する経路が未決定であるとき、 $m_t$  を  $C_C$  に追加する。
  - $m_t$  がレジスタであるとき、 $m_t$  を  $C_{REG}$  に追加する。

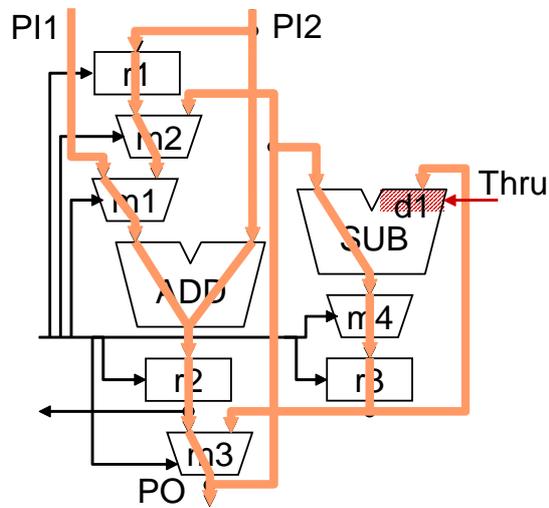


図 4.3 制御林の例

- $m_t$  が CMA であり,  $m_t$  の全てのデータ入力端子に接続する経路が制御林に決定されており,  $m_t$  が  $C_C$  に属していないとき,  $m_t$  を  $C_C$  に追加する. このとき,  $m_t$  が  $C_{CM}$  に存在するときは,  $m_t$  を  $C_{CM}$  から削除する.
- $m_t$  が CMA であり上記の条件を満たしていない, または  $m_t$  が CMB であるとき,  $m_t$  を  $C_{CM}$  に追加する.

$m_t$  のデータ出力端子に接続する経路が未決定であるとき, 別の  $m_t$  があれば選択し上記の操作を繰り返して行う.

3.  $C_C$  が空でないとき, 手順 (2) に戻る.
4.  $C_{REG}$  が空でないとき,  $C_{REG}$  中の全てのレジスタを  $C_C$  に追加し,  $C_{REG}$  を空集合に設定する. そして, 手順 (2) に戻る.
5.  $C_{CM}$  が空でないとき, CMA が存在すれば CMA を 1 個, CMB のみ存在する場合は CMB を 1 個  $C_{CM}$  から削除し  $C_C$  に追加した後手順 (2) に戻る.
6. 回路要素  $m$  のデータ入力端子のうち, 最初に制御林に決定される経路に接続する  $m$  のデータ入力端子を  $in1_m$  とする.  $m$  のデータ出力端子を  $out_m$  と

する． $m$  が以下の条件の何れかを満たすとき， $in_{1_m}$  から  $out_m$  へのスルー機能を  $m$  に付加する．

- $m$  が閉路上に存在するか  $m$  の出力先に閉路が存在し，制御林が  $m$  で再収斂する再収斂経路であり，その再収斂経路上にレジスタが存在しない．
- $m$  が手順 (5) のとき  $C_{CM}$  に属した CMA であり，閉路上に存在する．
- $m$  が CMB であり， $m$  が閉路上に存在するか  $m$  の出力先に閉路が存在する．

$m$  が MUX 又はスルー機能を付加した回路要素である場合， $in_{1_m}$  から  $out_m$  までの経路を制御林に決定し，それ以外の場合は，各データ入力端子からデータ出力端子までの経路を制御林に決定する．

図 4.2 のデータパス中の回路要素に対して，制御林の生成とスルー機能を付加した例を図 4.3 に示す．図 4.2 の ADD1，SUB，m4 は CMA である．また，図中の  $m_1$ ， $m_2$ ， $m_3$  は MUX であり，図中の  $r_1$ ， $r_2$ ， $r_3$  は HR である．提案する制御林の決定アルゴリズムは，手順 (1) で， $C_C$  を  $PI_1$ ， $PI_2$  で初期化する．手順 (2) で  $C_C$  の中で  $n_{PI_1}$  が最小となるので  $PI_1$  から探索を開始する． $PI_1$  と接続する  $m_1$  を  $C_C$  に追加し， $PI_2$  から ADD 及び  $r_1$  までの経路を制御林に決定し  $C_C$  から  $PI_1$  を削除する．次に， $n_{m_1}$  が最小であるため  $m_1$  から探索し，ADD までの経路を制御林に決定し， $C_C$  から  $m_1$  を削除する．次に， $PI_2$  と接続する ADD を  $C_{CM}$  に追加し， $r_1$  を  $C_{REG}$  に追加する．また， $PI_2$  から ADD 及び  $r_1$  までの経路を制御林に決定し， $C_C$  から  $PI_2$  を削除する．このとき，ADD の全てのデータ入力端子に接続する経路が制御林に決定されたため，ADD を  $C_C$  に追加する．次に，ADD から探索し  $r_2$  を  $C_{REG}$  に追加して， $C_C$  から ADD を削除する．次に， $C_C$  が空となるため， $r_1$  を  $C_{REG}$  から削除し  $C_C$  に追加する．以降，同様に手順 (2)-(4) を繰り返して行う．手順 (5) において，SUB は  $C_{CM}$  に属するので， $C_{CM}$  から削除し  $C_C$  に追加する．以下，同様に手順 (2)-(4) を行う．手順 (6) において，SUB に左側のデータ入力端子とデータ出力端子間のスルー機能を付加する．

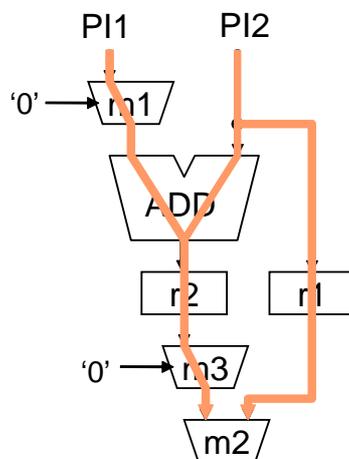


図 4.4  $m_2$  の制御経路

#### 4.2.2 依存関係の解消

ある 2 入力回路要素  $m$  の制御経路が、 $m$  で再収斂する再収斂経路であり、その再収斂経路の順序深度が等しいとする。PI から  $m$  までの制御経路の順序深度を  $D$  とする。時刻 0 において PI から値を印加したとき、時刻  $D$  において、 $m$  のデータ入力端子間は依存関係を持つ。

データ入力端子間に依存関係を持つ時刻が存在する回路要素を  $m_c$  とする。 $m_c$  が閉路上に存在するとき、部分強可検査性の定義の条件 (a)4 を満たすために、依存関係を解消する必要がある。また、 $m_c$  の制御経路上に HR が存在するとき、値域の任意の値の設定のために、依存関係を解消する必要がある。依存関係を解消するために、 $m_c$  のデータ入力端子に隣接するレジスタのうち、順序深度が大きい制御経路上に存在するレジスタに対して、依存関係が解消できるホールドのサイクル数を計算する。順序深度が大きな経路上に存在するレジスタに対して、ホールド機能を用いることで、少ないホールド回数で依存関係を解消できる。用いるホールド回数の情報は、時間展開モデル生成時に利用される。隣接するレジスタが LR である場合は、ホールド機能を追加する。

図 4.3 の制御経路から、 $m_2$  に対する制御経路のみを抜き出したものを図 4.4 に示す。 $m_2$  の両方のデータ入力端子には、PI2 からの制御経路を通り順序深度が 1

で到達する．従って，m2 のデータ入力端子間には依存関係を持つ時刻が存在する．m2 は閉路上に存在するので，部分強可検査性の定義の条件 (a)4 を満たすために，m2 のデータ入力端子間の依存関係を解消する必要がある．m2 に隣接するレジスタは，どちらも順序深度が等しいので，どちらかのレジスタに対してホールド機能を用いる．

### 4.2.3 時間展開モデルの生成

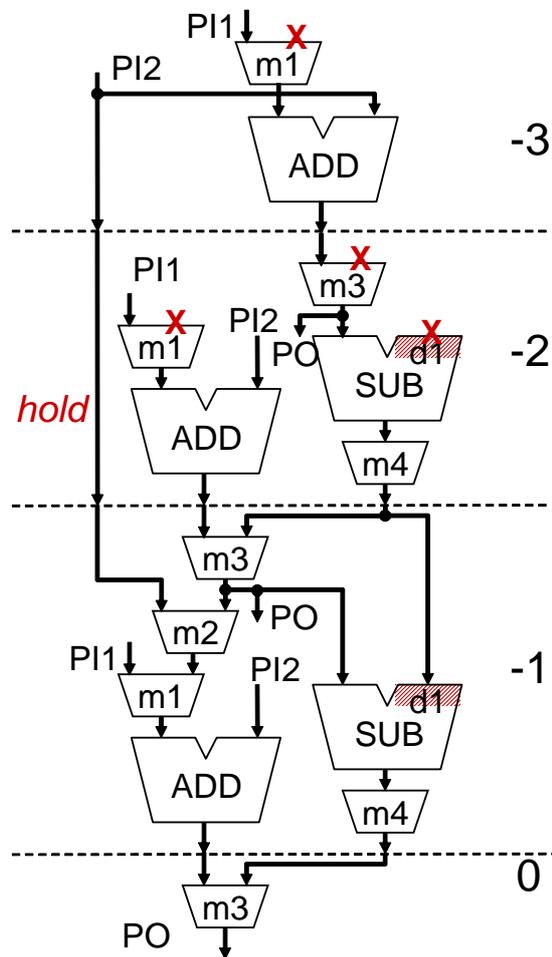


図 4.5 時間展開モデルの例

部分強可検査データパスに対するテスト生成では、単一の時間展開モデルを生成し、その時間展開モデル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。以下に時間展開モデルの生成に用いる変数及び集合を定義する。時刻  $t$  における時間展開フレームを  $f_t$  とする。 $f_t$  への追加対象となる組合せ回路要素の集合を  $C_O$  とする。また、データ入力端子に接続する回路要素が、 $f_t$  への追加候補となるレジスタの集合を  $C_{REG}$  とする。 $C_{REG}$  のレジスタのうち、 $t$  でロードするレジスタの集合を  $C_{LOAD}$  とする。時間展開モデルの生成手順を以下に示す。

1.  $C_O$  を PI からの順序深度が最も大きい 1 個の PO で初期化する。 $C_{REG}$ 、 $C_{LOAD}$  を空集合に設定する。 $t$  を 0 に設定する。
2.  $C_O$  が空になるまで以下の操作を繰り返し実行する。回路要素  $m \in C_O$  を  $C_O$  から削除し、 $m$  を  $f_t$  に配置する。 $m$  がレジスタであるとき、 $m$  を  $C_{REG}$  に追加する。 $m$  が組合せ回路要素であるとき、初めて時間展開モデルに追加される場合は、 $m$  のデータ入力端子に接続する全ての回路要素を  $C_O$  に追加する。そうでない場合は、制御林上で  $m$  のデータ入力端子に接続する回路要素のみを  $C_O$  に追加する。このとき、4.4 で求めた  $m$  に対応するレジスタのホールドを行うサイクル数に従いレジスタをロードする時刻を決定する。また、 $m$  と  $C_O$  に追加される回路要素間の信号線を  $f_t$  に配置する。さらに、追加する信号線が PO と接続している場合、PO を  $f_t$  に配置する。
3.  $C_{REG}$  が空であるとき、手順(4)を行う。 $C_{REG}$  が空でないとき、以下の操作を行う。 $C_{REG}$  のうち  $t$  でロードされるレジスタで  $C_{LOAD}$  を初期化する。時間展開モデル上における、 $C_{LOAD}$  の各レジスタから到達可能な回路要素の集合を  $M_R$  とする。 $M_R$  の中で、閉路上に存在し 2 入力とも  $C_{LOAD}$  から到達可能な回路要素に対して、制御林を用いて依存関係を判別する。値域の任意の値の観測を保証するために、回路要素  $m \in M_R$  が依存関係を持つとき、 $m$  の一方のデータのデータ入力端子から到達する  $C_{LOAD}$  のレジスタに対して、依存関係が解消できるホールドのサイクル数を決定する。このとき、レジスタが LR であれば、ホールド機能を付加する。ホールドを行ったレジスタを  $C_{LOAD}$  から削除し、 $C_{LOAD}$  の各レジスタのデータ入力端子に接続する

回路要素を  $C_O$  に追加する． $t$  を 1 小さくし， $C_{LOAD}$  の各レジスタから  $C_O$  の要素までの信号線を  $f_t$  に配置する．また， $C_{LOAD}$  の全てのレジスタを  $C_{REG}$  から削除し， $C_{REG}$  の全てのレジスタに対して， $t+1$  から  $t$  までの信号線を  $f_t$  に配置する．手順 (2) に戻る．

4. 時間展開モデルに追加を行っていない PO 及び OM のうち 1 個を  $C_O$  に追加する． $t = 0$  に設定し手順 (2) に戻る．

図 4.3 に対する時間展開モデルの例を図 4.5 に示す．図 4.3 のデータパスでは， $m_2$  のデータ入力端子は依存関係を持つ．図 4.5 の例では，手順 (2) で時刻  $-1$  に  $m_2$  を追加したとき，依存関係を解消するために時刻  $-1$  で  $r_1$  のホールド機能を 1 サイクル使用し，時刻  $-3$  に  $PI_2$  を追加する．

以上のテスト容易化設計により，PI から決定した制御林上での値域の値の伝搬及び PO までの値の伝搬を保証している．このため，データパス中の任意の閉路に対して，依存関係のない任意の値の設定及び値域の任意の値の観測が可能である．従って，部分強可検査性の条件を満たす．

生成された時間展開モデル上において，信号線が接続されていないデータ入力端子を持つ回路要素はスルー機能を実現している．このスルー機能を実現する制御入力値およびスルーされないデータ入力端子のデータ入力値を不定値とする制約を設定し，この制約下で時間展開モデル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．時間展開モデルに対するテスト生成によって得られたテストパターンは，元の順序回路に対するテスト系列へと変換する．

時間展開モデルの生成の過程において，初めて時間展開モデル上に現れる回路要素に関しては，全てのデータ入力端子に接続する回路要素を時間展開モデルに追加し，二回以上現れる回路要素に関しては，制御林上の回路要素のみを時間展開モデルに追加する．このため，生成する時間展開モデル上において，部分強可検査データパスの全ての回路要素及び，外部入力から各回路要素に到達する制御林が現れる．また，4.4 のホールド機能を使うサイクル数の情報を用いることで，4.3 で決定した制御林を用いたとき，各回路要素のデータ入力端子間は依存関係を持たない．つまり，閉路を複数回展開することなく制御林のみを用いることで，値域の任意の値の印加を行う．従って，部分強可検査データパスの任意の回路要

素  $m$  は，生成した時間展開モデル上の  $m$  の全てのデータ入力端子に対して，値域の任意の値の印加が保証される．以上により，部分強可検査データパスで検出可能な全ての故障は，生成した時間展開モデル上で検出可能である．また，生成した時間展開モデルの時間展開数は，高々  $d_p + n_{REG}$  となる．従って，この時間展開モデルに対するテスト生成では，実用的なテスト生成時間で，完全故障検出効率が達成可能であると考えられる．

### 4.3. 実験結果

表 4.1 データパスの特性

| Circuits | bit | #PI | #PO | #Reg | #MUX | #CM | #OM | area    |
|----------|-----|-----|-----|------|------|-----|-----|---------|
| GCD      | 16  | 2   | 1   | 3    | 3    | 2   | 3   | 1010.7  |
| LWF      | 16  | 2   | 2   | 5    | 5    | 3   | 0   | 1364.7  |
| JWF      | 16  | 4   | 5   | 14   | 25   | 3   | 0   | 4208.3  |
| Paulin   | 16  | 2   | 2   | 7    | 8    | 7   | 0   | 4329.0  |
| Tseng    | 16  | 3   | 2   | 6    | 5    | 9   | 0   | 3163.9  |
| RISC     | 32  | 1   | 3   | 40   | 84   | 20  | 3   | 39834.8 |
| MPEG     | 8   | 56  | 148 | 241  | 207  | 161 | 0   | 48709.7 |

提案手法の有効性を実験により評価した．実験に使用したベンチマーク回路は，典型的なベンチマーク回路である GCD, LWF, JWF, PAULIN [16, 13] 及び，より実用的で大規模な回路である RISC と MPEG<sup>1</sup> [14] である．これらのベンチマーク回路の特性を表 4.1 に示す．「bit」, 「#PI」, 「#PO」, 「#Reg」, 「#MUX」, 「#CM」, 「#OM」は，それぞれデータパスのビット幅，外部入力数，外部出力数，レジスタ数，マルチプレクサ数，演算モジュール数，観測モジュール数を表す．論理合成ツールには AutoLogicII (MentorGraphics) を使用した．NOT ゲートの面積を

<sup>1</sup>半導体理工学研究センター (STARC) との共同研究 (1997-2001) において，提供を受けたものである．

表 4.2 面積オーバーヘッド

| 回路     | 完全スキャン | 強可検査法 [16] | 提案手法 |
|--------|--------|------------|------|
| GCD    | 16.5   | 3.8        | 0.0  |
| LWF    | 20.4   | 9.6        | 0.0  |
| JWF    | 18.5   | 3.1        | 0.0  |
| Paulin | 9.0    | 3.5        | 1.8  |
| Tseng  | 10.5   | 8.4        | 2.4  |
| RISC   | 16.9   | 4.3        | 0.0  |
| MPEG   | 14.0   | 8.0        | 0.8  |

表 4.3 故障検出効率

| 回路     | DFT 前  | 完全スキャン | 強可検査法 [16] | 提案手法   |
|--------|--------|--------|------------|--------|
| GCD    | 100.00 | 100.00 | 100.00     | 100.00 |
| LWF    | 100.00 | 100.00 | 100.00     | 100.00 |
| JWF    | 100.00 | 100.00 | 100.00     | 100.00 |
| Paulin | 99.23  | 100.00 | 100.00     | 100.00 |
| Tseng  | 99.01  | 100.00 | 100.00     | 100.00 |
| RISC   | 99.37  | 100.00 | 100.00     | 100.00 |
| MPEG   | 88.30  | 100.00 | 100.00     | 100.00 |

表 4.4 テスト生成時間

| 回路     | DFT 前    | 完全スキャン | 強可検査法 [16] | 提案手法  |
|--------|----------|--------|------------|-------|
| GCD    | 3.18     | 0.20   | 0.83       | 0.39  |
| LWF    | 0.33     | 0.21   | 0.62       | 0.24  |
| JWF    | 3.63     | 0.60   | 0.62       | 1.74  |
| Paulin | 297.50   | 0.83   | 0.86       | 4.69  |
| Tseng  | 703.90   | 0.34   | 1.08       | 0.87  |
| RISC   | 12210.81 | 55.17  | 76.56      | 60.02 |
| MPEG   | 68947.90 | 1.84   | 1.18       | 13.34 |

表 4.5 テスト実行時間

| 回路     | DFT 前 | 完全スキャン | 強可検査法 [16] | 提案手法 |
|--------|-------|--------|------------|------|
| GCD    | 133   | 2351   | 387        | 197  |
| LWF    | 56    | 2674   | 250        | 70   |
| JWF    | 244   | 14849  | 769        | 608  |
| Paulin | 147   | 2824   | 875        | 526  |
| Tseng  | 590   | 4752   | 633        | 523  |
| RISC   | 2271  | 621284 | 5520       | 3420 |
| MPEG   | 1216  | 185183 | 107359     | 8448 |

1としたときの、各ベンチマークの回路面積を表4.1の「area」に示す。提案手法の比較対象は、DFT前、完全スキャン、強可検査法[16]とした。

面積オーバーヘッドの結果を表4.2に示す。面積オーバーヘッドはデータパスに付加した回路のみを対象とした。完全スキャンにおける面積オーバーヘッドは、元のデータパスに対する、全てのFFに1ビットMUXを付加することによって増加した面積の割合となる。1ビットMUXは、AutoLogic IIを用いて合成を行った。強可検査法は完全スキャン設計法より、面積オーバーヘッドが削減されているが、提案手法は強可検査法と比較して平均1/8、完全スキャン設計法と比較して平均1/20に削減できた。特に、GCD、LWF、JWF、RISCはハードウェアの付加を必要とせず部分強可検査性の性質を満たすので、提案手法の面積オーバーヘッドは0となった。

テスト生成結果を表4.3、4.4及び4.5に示す。テスト生成ツール(ATPG)はTestGen(Synopsys)、計算機はSunBlade2000を使用した。提案手法の時間展開モデルを用いたテスト生成では、多重故障を扱う必要がある。TestGenは多重故障を扱うことができないので、文献[21]の手法を用いて多重故障を単一縮退故障として扱える回路に変換し、TestGenを適用した。完全スキャンは、全てのFFをスキャンFF(MUX付きFF)に置き換え、シングルスキャンパスで構成した。完全スキャン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。また、強可検査法のテスト実行時間は、全ての「回路要素のテストベクトル数」×「テストプラン長」の和とした。完全スキャン設計法、強可検査法及び提案手法は完全故障検出効率を達成し、テスト容易化設計前の回路と比較して大幅にテスト生成時間を削減できた。特にRISCに関しては、テスト容易化設計前と比較して提案手法は、面積オーバーヘッド0で1/200に削減できた。また、完全スキャン設計法、強可検査法及び提案手法は、ほぼ同程度のテスト生成時間となった。RISCでは、各回路要素を通るPIからPOまでの経路の順序深度が小さいため、テスト生成の対象となる組合せ回路の規模は、完全スキャン設計と同程度になる。このため、比較的高速なテスト生成が可能になったと考えられる。MPEGに関しては、各回路要素を通るPIからPOまでの経路の順序深度が非常に大きい。このため、テスト生成の対象となる組合せ回路の規模が大きくなったため、完全

スキャン設計と比較してテスト生成時間が大きくなったと考えられる。一方，提案手法と強可検査法は，スキャンシフト操作を必要としないので完全スキャン設計法よりも大幅にテスト実行時間を短縮できた。RISC では提案手法は完全スキャン設計法の 1/180 に削減できた。また，提案手法と強可検査法は実動作速度での実行が可能であり，提案手法は強可検査法に比べてテスト実行時間を削減することができた。特に MPEG に関しては，強可検査法と比較して提案手法は 1/8 に削減できた。これは，強可検査法が回路要素毎にテスト生成をするのに対し，提案手法は回路全体を対象にテスト生成を行うので，故障シミュレーションにより効率よく故障が検出でき，系列長が短くなったためであると考えられる。以上のことから，提案手法は，テスト生成時間を増やすことなく面積オーバーヘッドを大幅に削減するのに成功している。又，結果としてテスト実行時間を強可検査法よりさらに短縮するのに成功している。

#### 4.4. むすび

本論文では，データパスの性質として部分強可検査性を定義し，部分強可検査性に基づくテスト容易化設計法及びテスト生成法を提案した。組合せ回路用のテスト生成アルゴリズムを用いることで，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。提案手法は，実動作速度でのテスト実行が可能である。また，強可検査と比較して面積オーバーヘッドが小さく，テスト実行時間が短くなることを示した。

本論文では，レジスタ転送レベル回路のデータパスのみを対象としており，コントローラの実在を考慮していない。次章では，コントローラを含むレジスタ転送レベル全体に対するテスト容易化設計法を提案する。

## 第 5 章

# 部分強可検査性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

文献 [14] の手法における面積オーバーヘッドを削減するために、4 章ではデータパスのみに注目し、面積オーバーヘッドを削減することに成功した。また、テスト実行時間を削減することにも成功した。しかし、文献 [14] の手法と同様に RTL 回路のデータパスとコントローラを分離し、データパスに対して 4 章の手法を適用する場合、必要なテストコントローラ的面積オーバーヘッドは大きくなる。そこで、本章では、レジスタ転送レベル (RTL) 回路全体を対象とした部分強可検査性に基づくテスト容易化設計法を提案する。通常のコントローラの出力を利用してデータパスへの制御信号を供給するので、テストコントローラに必要な出力パターン数は少なくなる。このため、文献 [14] の手法と比較してテストコントローラ的面積が削減可能である。

### 5.1. 問題の定式化

与えられた RTL 回路に対して、部分強可検査性を実現するためのテスト容易化設計を、次の最適化問題として定式化する。

定義 6 (部分強可検査性に基づくテスト容易化設計問題)

- 入力：RTL 回路
- 出力：部分強可検査 RTL 回路
- 最適化目標：付加ハードウェア量最小

□

## 5.2. 提案手法の概要

提案するテスト容易化設計法では、データパス中の各回路要素に対して、外部入力からそのデータ入力端子に値域の任意の値を伝搬するための制御線およびそのデータ出力端子の任意の誤りを外部出力まで伝搬する観測線を決定する。それらの経路を用いて値の伝搬を可能とするために、経路上の回路要素に(1)スルー機能の付加、(2)テストコントローラによる制御入力端子の制御を行う。さらにそれらの経路を用いて値を伝搬した場合に依存関係が存在する場合は、レジスタのホールド機能を用いてその依存関係を解消する。依存関係を解消するためのホールド機能の制御もテストコントローラによって行う。このように、各回路要素のデータ入出力端子への値の伝搬では、テストコントローラを用いて制御入出力端子の制御を行う。一方、データパスの各回路要素の制御入力端子へ値域の任意の値の伝搬を行う際は、テストコントローラではなく、通常のコントローラを用いる。通常のコントローラの制御はコントローラの制御経路及び、状態信号線を通じてデータパスの制御経路を用いて行う。データパスの各回路要素の状態出力端子からの誤りの伝搬も同様に通常のコントローラを通じて行う。上記はデータパスについて述べたがコントローラについても同様である。これは部分強可検査性がRTL回路全体を対象としており、データパス/コントローラ間の信号線に対して任意の値の印加・観測を必要とせず、値域の任意の値のみの印加・観測が可能であれば良いという性質による。これにより、文献[14]に比べ、データパスとコントローラを分離するためのMUXおよびテストコントローラに要求される出力パターン数は少なくなり、その面積および遅延オーバーヘッドを削減することが可能となる。

図5.1,5.2に文献[14]と提案手法の例を示す。図5.1,5.2のCCは組合せ論理回路、TCはテストコントローラである。図5.1,5.2のMUX m1は他の回路要素の制

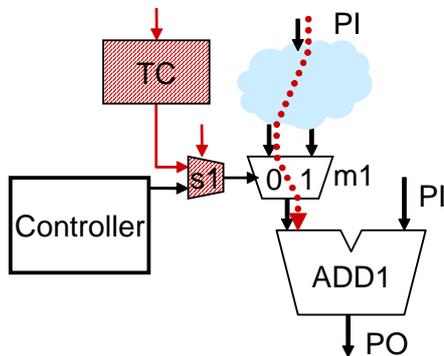


図 5.1 [14] の DFT 例

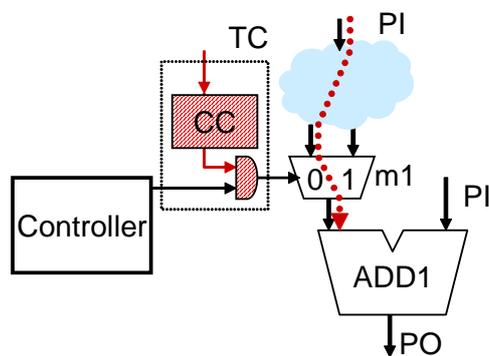


図 5.2 提案手法の DFT 例

御経路上にあると仮定する．この経路を用いて値の伝搬を行うためには， $m1$  の制御信号はテストコントローラを用いて'0'に制御される必要がある．さらに文献 [14] の場合，図 5.1 中の  $m1$  自身のテストのためにその制御入力に'0'と'1'の両方を設定可能である必要がある．これらの制御信号は全てテストコントローラから発生させる必要があり，そのためには通常のコントローラと  $m1$  の制御入力端子間には入力切替用の MUX  $s1$  も必要となる．これに対し提案手法の場合は，図 5.2 中の  $m2$  自身のテストのための制御信号は通常のコントローラを用いて発生させる．したがって，テストコントローラからは制御経路を用いて伝搬を行うための制御信号'0'のみを発生出来れば良く，通常のコントローラと  $m1$  の間には AND ゲートのみで良い．

### 5.3. テスト容易化設計法

部分強可検査性に基づくテスト容易化設計のための発見的アルゴリズムの概要を示す．アルゴリズムは以下の 4 ステップからなる．各ステップで DFT 回路が付加される．コントローラは，どの RTL 回路においても同じ構造となるようにモデル化するため，コントローラに対する DFT は常に同じである．

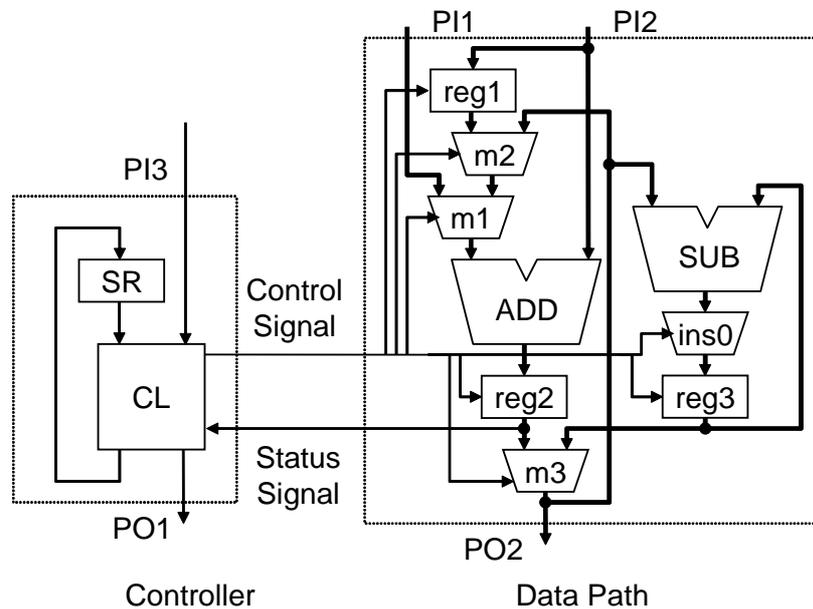


図 5.3 RTL 回路の例

### 5.3.1 制御林の決定

各回路要素に対して、PI からそのデータ入力端子に値域の任意の値を伝搬するための経路（制御経路）を決定する。この制御経路の集合を制御林とする。制御林は、PI から各データ入力端子までの任意の値の伝搬が可能な経路を優先し、順序深度に対する幅優先探索をして決定することにより、スルー機能の付加箇所を少なくしている。スルー機能は MUX がマスク素子を付加すること実現でき、データ出力端子に任意の値の伝搬が不可能な回路要素に対して付加する。

図 5.3 の RTL 回路に対する制御林及び、追加される DFT 回路を図 5.4 に示す。コントローラ中の CL を含むの閉路上のデータ入力端子に対しては、MUX (TM1) を付加し、データパスの PI からの付加経路を用いて、任意の値を設定可能とする。データパスに対する制御林の生成法は、4 章の制御林の生成と同様である。

制御林上での値を伝搬に必要なデータパス中の各回路要素に対する制御信号は、テストコントローラを用いて印加する。テストコントローラに必要な出力は、MUX では制御林で選択された経路の伝搬に必要な制御信号、スルー機能を持つ回路要素

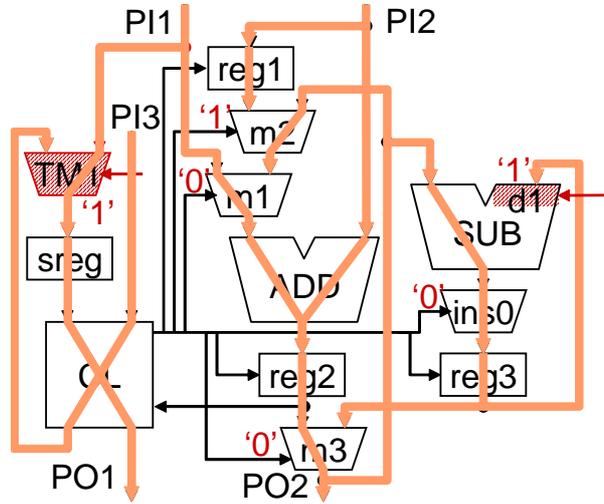


図 5.4 制御林の例

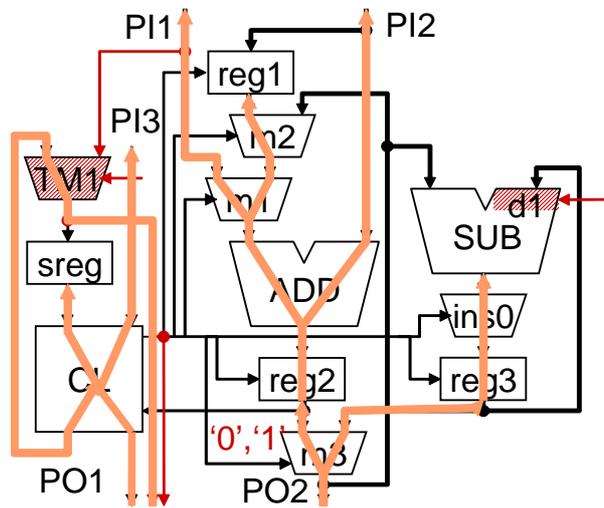


図 5.5 観測林の例

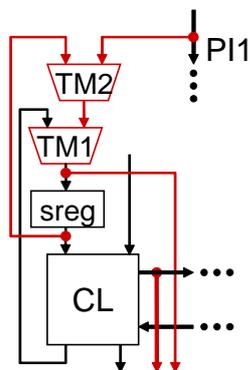


図 5.6 ホールド機能の付加

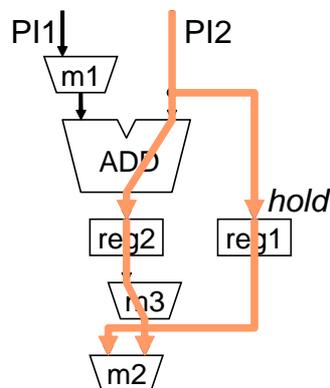


図 5.7 m2 の制御林

ではスルー機能の制御信号となる。図 5.4 の例では、テストコントローラに必要な出力は  $m1, m2, m3, ins0, d1, TM1$  に対して、それぞれ '0', '0', '0', '0', '1', '1' となる。

### 5.3.2 観測林の決定

各回路要素に対して、PO から各データ出力端子までの誤りを伝搬するための経路（観測経路）を決定する。この観測経路の集合を観測林とする。観測林は、PO から各回路要素のデータ出力端子に対して、最小の順序深度で到達する経路の集合として決定する。その際、制御林と観測林が共有する経路が極大となるように決定することで、スルー機能の付加箇所を少なくしている。

図 5.4 の RTL 回路に対する観測林及び、DFT 結果を図 5.5 に示す。コントローラに対しては、CL のデータ出力端子及び全ての制御出力端子に対して、PO への観測経路を付加する。このとき、4.3.1 で付加した TM1 のテストを容易にするために、TM1 のデータ出力端子に接続する経路に対して観測経路を付加する。

観測林上での値の伝搬に必要なデータパス中の各回路要素に対する制御信号は、テストコントローラを用いて印加する。決定した観測林において、PO からの観測林上の順序深度において、回路要素  $m_j$  から PO までの観測林上に、 $m_j$  と順序深度の異なる回路要素  $m_i$  が存在するとする。このとき、観測林上における値域の

任意の値の伝搬の保証のために、 $m_i$  の制御信号に対して観測林上での値の伝搬に必要な制御信号がテストコントローラに必要となる。図 5.5 の例では、 $m_3$  は PO からの順序深度の異なる回路要素の観測林上にある。従って、テストコントローラの出力として、 $m_3$  の制御信号に対する '0' 及び '1' が必要となる。

### 5.3.3 依存関係の解消

PI からある回路要素  $m$  の各データ入力端子までの制御林か又は観測林上における経路が、 $m$  で再収斂する再収斂経路であり、その順序深度が等しいとき、 $m$  のデータ入力端子間には依存関係が存在する。制御林及び観測林上における値域の値の伝搬の保証のために、その依存関係を解消を行う。依存関係の解消のために、ホールド機能の付加及び、ホールド機能を用いるサイクル数の決定を行う。できるだけ HR を用いて依存関係の解消を行うことにより、ホールド機能の付加箇所を削減する。

4.3.1 で決定する制御林において、コントローラ中の SR に対しては、データパスのある PI からの付加経路を用いて正当化を行う。そのため、データパス中の制御信号を持つ回路要素において、データパスのその PI からデータ入力端子到達する経路と、その PI からコントローラの CL を通って制御入力端子に到達する経路が再収斂経路となり、依存関係が発生する。この依存関係を解消するために、SR に対してホールド機能を付加する。ここでは、通常動作時における付加回路による遅延の影響を小さくするために、図 5.6 のようにホールド機能のための MUX (TM2) を付加する。

図 5.7 に  $m_2$  の各入力端子に対する制御林の例を示す。 $m_2$  の制御林上に PI2 から等しい順序深度で到達するレジスタが存在するので、 $m_2$  の依存関係を解消するために、1 サイクル  $reg1$  のホールド機能を用いる。

### 5.3.4 テストコントローラの生成

決定した制御林、観測林及び依存関係の解消において、各回路要素に必要な制御信号を発生するテストコントローラ (TC) を生成する。TC から制御信号を発

生させることにより，各回路要素に対する値域の値の印加及び観測が可能となり，各回路要素に対するテストを保証する．TCはCLと付加観測経路間に挿入し，付加DFTの制御信号を観測するために観測経路を付加する．TCは，組合せ回路で構成し，制御出力の切替は付加外部入力から行う．TCの出力パターン数を $n$ とすると，TCに必要な付加外部ピン数は $\lceil \log_2 n \rceil$ である．また，TCによってコントローラ/データパス間の信号線上に挿入されるのはANDゲート及びORゲートの高々2ゲートである．

TCの出力パターンは，SRの正当化，制御林の制御，観測林の制御，RTL回路のテスト，TM1のテスト及び通常動作で構成される．SRの正当化では，SRに対する任意の値を設定するために，TM1，TM2に対してそれぞれ'1'，'1'を出力する．RTL回路のテストは，ロードの制御信号を出力し，データパスの各組合せ回路要素への制御信号はCLの制御出力端子の値を出力する．制御林の制御では，データパス中における依存関係解消のためのホールド機能の制御信号と制御林上の値の伝搬に必要な制御信号の組合せの制御信号を出力する．通常動作は，コントローラの制御出力信号の値をそのまま出力する．

図5.7のRTL回路に対して生成したテストコントローラの出力パターン例を図5.8に示す．この例では，SREGの正当化及び観測林の制御信号が，1行目のようにまとめられる．2，3行は制御林，4行はRTL回路のテスト，5行は付加MUXのテスト，6行目は通常動作のための制御出力である．TCの出力パターン数が6なので，TCに必要な付加外部ピンは3となる．

図5.9に部分強可検査RTL回路を示す．付加観測経路はデータパスのPOを用いて観測できるようにMUX(OMUX)を追加する．OMUXの制御は付加外部ピンによって行う．

5.3.1，5.3.2，5.3.3，5.3.4より，決定した制御林及び観測林上での値域の値の伝搬を保証することで，RTL回路中の任意の閉路に対して，依存関係のない任意の値の設定及び値域の任意の値の観測が可能である．従って，部分強可検査性の条件を満たす．

| m1 | m2 | m3 | ins0 | reg1 | reg2 | reg3 | d1 | TM1 | TM2 |
|----|----|----|------|------|------|------|----|-----|-----|
| X  | X  | 1  | X    | X    | X    | X    | X  | 1   | 1   |
| 0  | 0  | 0  | 0    | 1    | 1    | 1    | 1  | 1   | 0   |
| 0  | 0  | 0  | 0    | 0    | 1    | 1    | 1  | 1   | 0   |
| T  | T  | T  | T    | 1    | 1    | 1    | 0  | 0   | 0   |
| T  | T  | T  | T    | 1    | 1    | 1    | 0  | 1   | 0   |
| T  | T  | T  | T    | T    | T    | T    | 0  | 0   | X   |

T : CL の制御出力端子の値を出力

X : don't care

図 5.8 テストコントローラの出カパターン例

## 5.4. 部分強可検査 RTL 回路に対するテスト生成法

部分強可検査 RTL 回路に対するテスト生成では、単一の時間展開モデル (TEM) を生成し、その TEM 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。TEM は以下の規則に従って生成する。

- PO から順に回路要素を TEM に追加する。
- ある回路要素が初めて TEM に追加される場合は、全ての入力端子に接続する回路要素を TEM に追加する。TEM に追加されたことがある回路要素が再度追加される場合は、制御林上のデータ入力端子に接続する回路要素及び制御入力端子に接続する回路要素を TEM に追加する。
- 回路要素のデータ入力端子にレジスタが接続するとき、レジスタに対して 5.2 で計算したホールドのサイクル数に対応する信号線を TEM に追加する。
- 追加した回路要素から PO までの観測林上に存在する回路要素を TEM に追加する。

図 5.9 に対する TEM の生成結果を図 5.10 に示す。

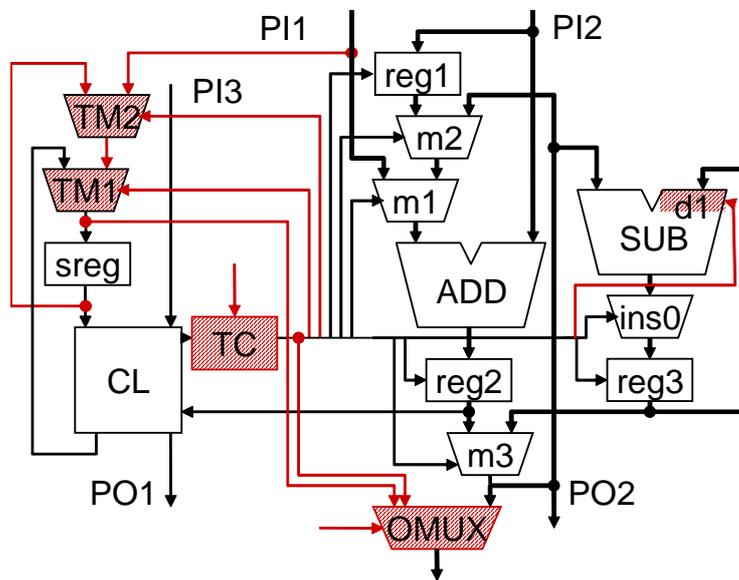
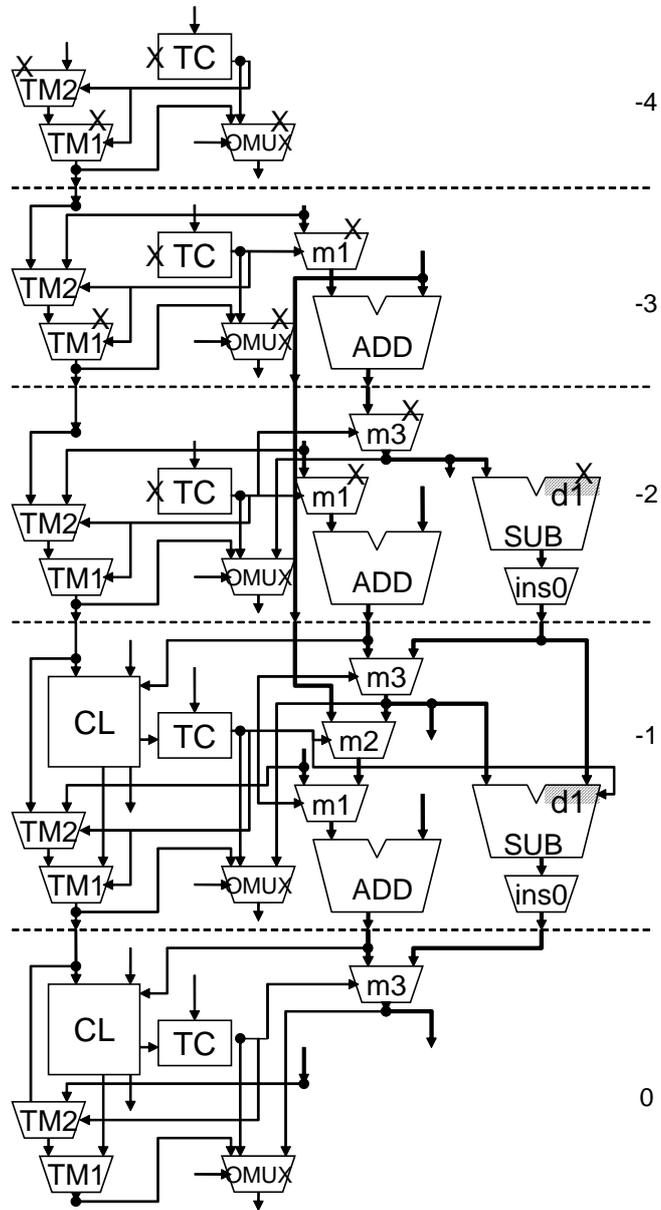


図 5.9 部分強可検査 RTL 回路

生成した TEM 上において，部分強可検査 RTL 回路の各回路要素する値域の任意の値の印加及び値域の任意の値の観測が可能である．従って，部分強可検査 RTL 回路における故障の検出を生成した TEM 上においても保証できる．TEM 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し，得られたテストパターンを元の順序回路に対するテスト系列へと変換する．

部分強可検査 RTL 回路中の任意の回路要素  $m$  は，生成した TEM 上において， $m$  の入力端子に対する値域の任意の値の印加及び観測が保証される．また，生成した TEM の時間展開数は，高々  $d_{pc} + n_{REG}$  である．従って，この TEM に対するテスト生成では，実用的なテスト生成時間で，完全故障検出効率が達成可能であるとえられる．



X: don't care

図 5.10 時間展開モデルの例

表 5.1 RTL 回路全体の面積

| 回路     | 面積 [#units] |
|--------|-------------|
| GCD    | 1127.0      |
| LWF    | 1413.3      |
| JWF    | 4322.5      |
| Paulin | 4430.6      |
| RISC   | 40827.9     |
| MPEG   | 52169.5     |

表 5.2 コントローラの特徴

| 回路     | #PI | #PO | #FF | #Status | #Control | Area[#units] |
|--------|-----|-----|-----|---------|----------|--------------|
| GCD    | 1   | 1   | 2   | 3       | 7        | 116.3        |
| LWF    | 1   | 0   | 2   | 0       | 8        | 49.7         |
| JWF    | 1   | 0   | 3   | 0       | 38       | 172.0        |
| Paulin | 1   | 0   | 3   | 0       | 16       | 107.6        |
| RISC   | 1   | 2   | 4   | 54      | 62       | 1463.9       |
| MPEG   | 6   | 0   | 8   | 0       | 271      | 3459.8       |

## 5.5. 実験結果

提案手法の有効性を実験により評価した．実験に使用した回路は，GCD，LWF，JWF，PAULIN [14] 及び，より実用的で大規模な回路である RISC と MPEG<sup>1</sup> [14] である．これらのベンチマーク回路の特性を表 5.1，5.2，5.3 に示す．表における「#PI」，「#PO」は外部入力数，外部出力数を表す．コントローラにおける「#FF」，「#Status」，「#Control」は，それぞれ，FF 数，ステータス入力数，制御出力数を表す．データパスにおける「bit」，「#Reg」，「#MUX」，「#MOD」は，それぞれデータパスのビット幅，レジスタ数，マルチプレクサ数，演算モジュール及び観測モ

<sup>1</sup>半導体理工学研究センター（STARC）との共同研究（1997-2001）において，提供を受けたものである．

表 5.3 データパスの特性

| 回路     | #PI | #PO | bit | #Reg. | #Mod. | Area[#units] |
|--------|-----|-----|-----|-------|-------|--------------|
| GCD    | 32  | 16  | 16  | 3     | 8     | 1127.0       |
| LWF    | 32  | 32  | 16  | 5     | 8     | 1363.6       |
| JWF    | 80  | 80  | 16  | 14    | 28    | 4150.5       |
| Paulin | 32  | 32  | 16  | 7     | 15    | 4323.0       |
| RISC   | 32  | 96  | 32  | 40    | 107   | 39364.0      |
| MPEG   | 56  | 148 | 8   | 241   | 368   | 47883.9      |

ジュールの数を表す．論理合成ツールには AutoLogicII ( MentorGraphics ) を使用した．NOT ゲートの面積を 1 としたときの，各回路の面積を表 4.1 の「面積」に示す．提案手法との比較対象は，テスト容易化設計前，完全スキャン設計法，強可検査法 [14] とした．

面積オーバーヘッドの結果を表 5.4 及び 5.5 に示す．完全スキャン法における面積オーバーヘッドは，元の RTL 回路に対する，全ての FF に 1 ビット MUX を付加することによって増加した面積の割合となる．1 ビット MUX は，AutoLogic II を用いて合成を行った．表 5.4 の「C」, 「DP」, 「TC」, 「MUX」は，それぞれ，コントローラの付加ハードウェア，データパスの付加ハードウェア，テストコントローラ及び付加 MUX のテスト容易化設計前の回路面積に対する面積比を表す．また，表 5.4 の「TC」及び「OMUX」は，それぞれ TC 及び OMUX に必要な外部ピン数を示す．強可検査法では，完全スキャン設計法と比較して面積オーバーヘッドが大きくなっていたのに対し，提案手法では大幅に面積オーバーヘッドを削減できた．強可検査法と比較して，提案手法における面積オーバーヘッドの削減は，主に TC と MUX の面積を削減できたことに起因する．提案手法では，TC を組み合わせ回路で構成しデータパス/コントローラ間に MUX を付加しないために，面積オーバーヘッドを抑えることができた．また，提案手法では，データパス/コントローラ間に多くとも 2 ゲート挿入するだけなので，強可検査法と比べて遅延オーバーヘッドが小さい．外部ピンオーバーヘッドについて，提案手法では，RISC 及び MPEG の外部ピンオーバーヘッドが他の回路よりも多くなった．RISC

表 5.4 面積オーバーヘッド

| 回路     | 完全<br>スキャン | 強可検査法 [14] |     |      |      | 提案手法 |     |     |     |     |     |
|--------|------------|------------|-----|------|------|------|-----|-----|-----|-----|-----|
|        |            | C          | DP  | TC   | MUX  | C    | DP  | TC  | MUX |     |     |
| GCD    | 26.6       | 39.7       | 1.1 | 2.6  | 23.2 | 12.8 | 8.6 | 1.4 | 0.0 | 4.3 | 2.8 |
| LWF    | 26.7       | 37.1       | 0.4 | 5.2  | 21.9 | 9.7  | 6.3 | 1.1 | 0.0 | 2.7 | 2.5 |
| JWF    | 33.4       | 48.6       | 0.8 | 18.1 | 21.1 | 8.6  | 6.7 | 0.5 | 0.0 | 2.9 | 3.3 |
| Paulin | 20.8       | 31.7       | 1.1 | 3.4  | 19.4 | 7.8  | 5.7 | 0.5 | 1.7 | 1.7 | 1.7 |
| RISC   | 16.7       | 27.3       | 0.1 | 10.9 | 12.5 | 3.6  | 3.3 | 0.1 | 0.0 | 2.2 | 1.1 |
| MPEG   | 19.7       | 24.9       | 0.2 | 4.0  | 13.0 | 7.2  | 5.1 | 0.2 | 0.6 | 2.4 | 1.9 |

では、ホールド機能の制御パターン数が多いために TC に必要な外部ピンが 4 本となった。また、RISC 及び MPEG では、データパスの外部出力のビット幅よりも観測が必要な信号線のビット幅が大きいため、OMUX が 2 段以上必要となった。

テスト生成結果を表 5.5 に示す。テスト生成ツール (ATPG) は TestGen (Synopsys)、計算機は SunBlade2000 を使用した。提案手法のテスト生成では、多重故障を扱う必要がある。TestGen は多重故障を扱うことができないので、文献 [21] の手法を用いて多重故障を単一縮退故障として扱える回路に変換し、TestGen を適用した。テスト生成時間には、テスト容易化設計に要した時間は含んでいない。それぞれの手法におけるテスト容易化設計にかかる時間は 1 秒以下となり、無視できる位に小さい。故障検出効率は、ベンチマーク回路においては、完全スキャン設計法、強可検査法及び提案手法は完全故障検出効率を達成した。RISC に関しては、完全スキャン設計及び提案手法では、一部の観測モジュールに対してテストベクトルを生成できなかった。提案手法におけるテスト生成時間は、スキャン設計法や強可検査法と比べて大きくなっているが、実用的なテスト生成時間が実現できた。RISC において提案手法では生成した時間展開モデルの時間展開数が小さいために比較的短いテスト生成時間になったのに対し、MPEG では回路の順序深度が非常に大きいために時間展開数が大きくなり、完全スキャン設計と比較してテスト生成時間が長くなった。RISC の制御信号線には多数のファンア

表 5.5 外部ピンオーバーヘッド

| 完全スキャン | 強可検査法 | 提案手法 |      |   |
|--------|-------|------|------|---|
|        |       | TC   | OMUX |   |
| 3      | 5     | 4    | 3    | 1 |
| 3      | 5     | 4    | 3    | 1 |
| 3      | 5     | 4    | 3    | 1 |
| 3      | 5     | 4    | 3    | 1 |
| 3      | 6     | 6    | 4    | 2 |
| 3      | 7     | 6    | 3    | 3 |

表 5.6 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    | 83.39 | 100.00 | 100.00 | 100.00 |
| LWF    | 99.05 | 100.00 | 100.00 | 100.00 |
| JWF    | 96.16 | 100.00 | 100.00 | 100.00 |
| Paulin | 96.55 | 100.00 | 100.00 | 100.00 |
| RISC   | 63.95 | 99.97  | 100.00 | 99.97  |
| MPEG   | 74.48 | 100.00 | 100.00 | 100.00 |

ウトが存在しており、データパス中の複数の回路要素に同一の制御信号が印加される。提案手法では、テストコントローラによりデータパスの各回路要素に個別に制御できるように制御信号線を付加し、全ての制御信号線を観測可能としている。このため、RISCにおける提案手法のテスト生成時間は完全スキャン設計と比較して大幅に削減できたと考えられる。提案手法と強可検査法は、スキャンシフト操作を必要としないので完全スキャン設計法よりも大幅にテスト実行時間を短縮できた。また、提案手法と強可検査法は実動作速度での実行が可能であり、ほとんどの回路において、提案手法は強可検査法に比べてテスト実行時間を大幅に削減することができた。特に、RISCにおいては、強可検査法の約 1/2、MPEG

表 5.7 テスト生成時間

| 回路     | DFT 前     | 完全スキャン   | 強可検査法  | 提案手法    |
|--------|-----------|----------|--------|---------|
| GCD    | 3070.07   | 0.27     | 1.13   | 1.72    |
| LWF    | 85.45     | 0.17     | 0.89   | 1.01    |
| JWF    | 2873.34   | 0.88     | 1.22   | 5.91    |
| Paulin | 4290.51   | 0.53     | 1.60   | 8.07    |
| RISC   | 156808.67 | 98870.71 | 105.26 | 166.88  |
| MPEG   | 195260.82 | 55.72    | 17.64  | 1208.90 |

表 5.8 テスト実行時間

| 回路     | DFT 前 | 完全スキャン  | 強可検査法  | 提案手法 |
|--------|-------|---------|--------|------|
| GCD    | 421   | 4232    | 456    | 588  |
| LWF    | 392   | 2904    | 295    | 108  |
| JWF    | 412   | 20975   | 1000   | 675  |
| Paulin | 201   | 6147    | 1136   | 798  |
| RISC   | 6928  | 1233859 | 7914   | 4345 |
| MPEG   | 148   | 462942  | 150019 | 8515 |

においては、強可検査法の約 1/20 に削減できた。強可検査法が回路要素毎にテスト生成をするのに対し、提案手法では回路全体を対象にテスト生成を行う。このため、故障シミュレーションにより効率よく故障が検出でき、系列長が短くなったためであると考えられる。テスト生成は各回路に対して一度実行するだけで済むのに対し、テスト系列長は製造した回路毎のテスト実行時間に影響することから、提案手法はより有用である。

## 5.6. むすび

本論文では，RTL 回路の性質として部分強可検査性を定義し，部分強可検査性に基づくテスト容易化設計法及びテスト生成法を提案した．提案手法は，組合せ回路用のテスト生成アルゴリズムが適用可能であり，実用的なテスト生成時間で完全故障検出効率を達成した．また，実動作速度テストが可能であり，完全スキャン設計法と比較して面積オーバーヘッド及びテスト実行時間を大幅に削減できることを示した．

## 第6章

### 結論

近年，VLSIの大規模化に伴い，従来のテスト容易化設計法である完全スキャン設計法では，テスト実行時間が非常に長いという問題や，より信頼性を保証する上で重要な実動作速度テストが困難といった問題がでてきている．そのため，実動作速度テストが実行可能であり，短いテスト実行時間を実現するテスト容易化設計法が求められている．同時に付加回路の面積オーバーヘッドや外部ピンオーバーヘッドをできるだけ小さく抑えることが必要である．

本論文では，完全スキャン設計法と比較して，面積オーバーヘッド及びテスト実行時間を大幅に削減し，完全故障検出効率が達成可能な手法として，時間展開モデルに基づくレジスタ転送レベル（RTL）回路のテスト容易化設計法を提案した．また，時間展開モデルを用いたテスト生成が容易になるRTL回路の性質として，部分強可検査性を提案した．

4章では，データパスのみを対象とし，部分強可検査性に基づくテスト容易化設計法を提案した．提案手法では，閉路に対して任意の値の印加及び任意の値を保証することで，検出可能故障を検出するのに必要な時間展開数が閉路の順序深度及び回路中のレジスタの個数程度となる．従って，実用的なテスト生成時間で完全故障検出効率を達成するテスト生成が可能であると考えられる．実験結果では，実用的なテスト生成時間で完全故障検出効率が達成可能であることを示した．また，文献 [16] の手法と比較して，面積オーバーヘッド及びテスト実行時間を削減することに成功した．

5章では，RTL回路全体を対象とし，部分強可検査性に基づくテスト容易化設計法を提案した．コントローラからの制御信号を考慮することで，テストコントローラに必要な出力パターン数は少なくなる．従って，文献[14]で問題となっていたテストコントローラの面積を削減することが期待できる．実験結果では，完全スキャン設計法と比較して，面積オーバーヘッドを大幅に削減することに成功した．また，実用的なテスト生成時間でほとんどの回路で完全故障検出効率を達成し，完全スキャン設計法と比較してテスト実行時間を大幅に削減することに成功した．テスト生成は各回路に対して一度実行するだけで済むのに対し，テスト系列長は製造した回路毎のテスト実行時間に影響する．従って，提案手法ではテスト系列長を大幅に削減できているため，より有用性が高い．本手法の今後の課題としては，データパスのビット幅が異なる回路に対する拡張や，コントローラ／データパス以外の回路に対する拡張が挙げられる．

# 謝辞

本研究の全過程を通して、絶えず適切な御指導、貴重な御助言、御討論、御配慮を賜りました藤原秀雄教授に深く感謝の意を表します。

本研究に際して、有益な御指導と御助言を頂きました松本健一教授に深く感謝の意を表します。

本研究に際して、貴重な御意見及び問題点の指摘を頂きました広島市立大学情報科学部の井上智生教授に深く感謝の意を表します。

本研究に際して、日頃から有益な御指導を頂きました井上美智子助教授に深く感謝の意を表します。

本研究の全過程を通じて、様々な面でお世話になり、御助言、御協力頂きました大竹哲史助手に深く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多くの貴重な御助言ならびに御討論を頂きました米田智和助手に深く感謝の意を表します。

本研究に際し、日頃より多くの貴重な御意見、御討論を頂いた本学のコンピュータ設計学講座の諸氏に感謝の意を表します。

そして、これまでの学生生活や研究生活を支え、励まし、協力し続けてくれた両親に感謝致します。

## 参考文献

- [1] H. Fujiwara, *Logic Testing and Design for Testability*, The MIT press, 1985.
- [2] T. Inoue, T. Hosokawa, T. Mihara, and H. Fujiwara, "An optimal time expansion model based on combinational ATPG for RT level circuits," *Proc. IEEE the 7th Asian Test Symp.*, pp. 190-197, Dec. 1998.
- [3] T. Inoue, D. K. Das, T. Mihara, C. Sano and H. Fujiwara, "Test generation for acyclic sequential circuits with hold registers," *International Conference on Computer Aided Design*, pp.550-556, Nov. 2000.
- [4] M. Abramovici, M. A. Breuer and A.D.Friedman, *Digital Systems Testing and Testable Design*, Computer Science Press, 1990.
- [5] S. Dey and M. Potkonjak, "Non-scan design-for-testability of RT-level data paths," in *Proc. Int. Conf. Computer-Aided Design*, pp.640-645, Nov.1994.
- [6] S. Dey and M. Potkonjak, "Nonscan design-for-testability techniques using RT-level design information," *IEEE Trans. on CAD of Integrated Circuits and Systems*, pp.1488-1506, Dec. 1997.
- [7] M. L. Flottes, D. Hammad and B. Rouzeyre, "Improving Testability of Non-Scan Designs during BehavioralSynthesis," *Journal of Electronic Testing: Theory and Applications 11*, pp.29-42, 1997
- [8] M. L. Flottes, B. Rouzeyre and L. Volpe, "Improving Datapath Testability by Modifying Controller Specification," *VLSI Design Volume 15 (2002)*, pp.491-498, 2002

- [9] R. B. Norwood and E. J. McCluskey, "Orthogonal scan:Low overhead scan for data Paths," *Proc.1996 Int.Test Conf.*,pp.659-668(1996)
- [10] R. B. Norwood and E. J. McCluskey, "High-level synthesis for orthogonal scan," *Proc. 15th VLSI Test Symp*, pp370-375, 1997
- [11] I. Ghosh, A. Raghunathan, and N. K. Jha, "Design for hierarchical testability of RTL circuits obtained by behavioral synthesis," *Proc. IEEE. Int. Conf. on Computer Design*, pp.173-179, 1995
- [12] I. Ghosh, A. Raghunathan, and N. K. Jha, "A Design for Testability Techique for RTL Circuits Using Control/Data Flow Extraction," *IEEE Trans. on Computer Aided Design*, Vol.17, pp.706-723, Aug.1998.
- [13] S. Nagai, S. Ohtake, and H. Fujiwara, "A Design for Hierarchical Testability for RTL Data Paths," *Digest of Papers, IEEE 2001 Workshop on RTL ATPG and DFT*, pp. 128-133, Nov. 2001.
- [14] S. Ohtake, H. Wada, T. Masuzawa and H. Fujiwara, "A non-scan DFT method at register-transfer level to achieve 100% fault efficiency," *Journal of IPSJ*, Vol.44, No.5, pp.1266-1275, May 2003.
- [15] S. Ohtake, T. Masuzawa and H. Fujiwara, "A non-scan DFT method for controllers to achieve complete fault efficiency," *in Proc. of the 7th Asian test symposium*, pp.204-211,1998.
- [16] H. Wada, T. Masuzawa, K. K. Saluja and H. Fujiwara, "Design for strong testablility of RTL data paths to provide complete fault efficiency," *Proc. of 13th International Conf.on VLSI Design*, pp.300-305, Jan. 2000.
- [17] Michiko Inoue, Kazuhiro Suzuki, Hiroyuki Okamoto and Hideo Fujiwara, "Test synthesis for datapaths using datapath-controller functions," *Digest of Papers of European Test Workshop (ETW2003)*, pp.207-208, May 2003.

- [18] Hiroyuki Iwata, Tomokazu Yoneda, Satoshi Ohtake, and Hideo Fujiwara, "A DFT Method for RTL Data Paths Based on Partially Strong Testability to Guarantee Complete Fault Efficiency," in *Proc. of the 14th Asian Test Symposium*, pp. 306-311, Dec. 2005.
- [19] B. T. Murray and J. H. Hayes, "Hierarchical test generation using precomputed tests for modules," in *Proc. of IEEE Transaction on CAD*, Vol.9, No.6, pp.594-603, June 1990.
- [20] J. Lee and J.H. Patel, "Hierarchical test generation under architectural level functional constraints," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, no. 9, pp. 1144-1151, Sept. 1996.
- [21] H. Ichihara, and T. Inoue, "A method of test generation for acyclic sequential circuits using single stuck-at fault combinational ATPG," *IEICE Trans. Fundamentals*, Vol. E86-A, No. 12, pp. 3072-3078, Dec. 2003.