

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

岩田 浩幸[†] 米田 友和[†] 大竹 哲史[†] 藤原 秀雄[†]

A DFT Method Based on Partially Strong Testability of RTL Data Paths to Guarantee Complete Fault Efficiency

Hiroyuki IWATA[†], Tomokazu YONEDA[†], Satoshi OHTAKE[†], and Hideo FUJIWARA[†]

あらまし 本論文では、レジスタ転送レベルデータパスを対象とし、完全故障検出効率を保証する非スキャン方式に基づくテスト容易化設計法を提案する。この手法では、レジスタ転送レベルデータパスの性質として部分強可検査性を定義し、それに基づくデータパスのテスト容易化設計法及びテスト生成法を提案する。提案したテスト容易化設計法は、強可検査性に基づくテスト容易化設計と比較して、大幅に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ができる。また、回路の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する。

キーワード テスト容易化設計, データパス, 強可検査, 部分強可検査, 完全故障検出効率

1. ま え が き

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。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。故障検出効率は, 回路のテスト生成の対象となる全故障数に対する, テスト生成アルゴリズムによって生成されたテスト系列が検出可能な故障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。ここで冗長とは, 回路中の故障に対するテスト系列が存在しないことをいう。

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, 検出可能な故障をすべ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 数の指数のオーダ以上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ことになる。しかし, 閉路のない順序回路 (無閉路順序回路) に対しては, 順序深度の時間展開数ですべての検出可能な故障を検出可能である。順序深度程度の時間展開数では, 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。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], [2] がある。この手法では, 順序回路のすべての FF を外部から直接制御及び観測可能なスキャン FF に置き換える。スキャン FF を擬似外部入力及び擬似外部出力とみなした組合せ回路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, この手法では, すべての 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くなるという問題がある。また, テスト実行の際

[†] 奈良先端科学技術大学院大学, 生駒市
Graduate School of Information, Nara Institute of Science and Technology, Ikoma-shi, 630-0192 Japan

に、回路をスキャンモードに設定し、スキャンシフト動作させるため、テスト実行時間が長くなるという問題もある。更に、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、実動作速度でのテスト実行が困難であるという問題がある。

完全スキャン設計法での問題点を解消する手法として強可検査性に基づくテスト容易化設計法 [4] や固定制御可検査性に基づくテスト容易化設計法 [5] がある。これらの手法では、データパスの強可検査性を利用している。強可検査性とは、すべての回路要素に対して、任意の値の印加・観測を可能とするテストプラン（制御入力の時系列）の存在を保证する性質である。強可検査性を満たすために、レジスタへのホールド機能の付加を行い、すべての演算回路にスルー機能をもたせる。スルー機能は、回路要素の入力に印加した値を変化させることなく回路要素の出力へ伝搬させる機能である。これにより、各回路要素ごと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と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。これらの手法では、実動作速度でのテストが可能であり、完全スキャン設計と比較して面積オーバーヘッドの削減が可能となっている。しかし、すべての演算モジュールにスルー機能をもたせずとも任意の値が伝搬できる回路要素が存在する。また、完全故障検出効率を達成するためには、すべての回路要素に対して任意の値を伝搬する必要はなく、外部入力から正当化可能なすべての値の集合（値域）の任意の値が印加できればよい（値域に関しては [定義 2] を参照）。したがって、強可検査性の性質を緩和することで面積オーバーヘッドを削減できる余地があると考えられる。

本研究では、強可検査性に基づくテスト容易化設計法と同様に、対象とする故障モデルを単一縮退故障とし、レジスタ転送レベルデータパスをテスト容易化設計の対象とする。強可検査のテスト容易性を失うことなく、面積オーバーヘッドを更に削減するために、強可検査性の性質を緩和した部分強可検査性を新たに導入し、部分強可検査性に基づくデータパスのテスト容易化設計法及びテスト生成法を提案する。強可検査性がすべての回路要素に対して任意の値の印加を保证するのに対し、部分強可検査性では回路要素に対して値域の任意の値の印加を保证する。また、提案するテスト容易化設計法では、データパスを部分強可検査にする

ために一部の回路要素に対してのみ、ハードウェアを付加する。そのため、強可検査性に基づくテスト容易化設計に比べて、面積オーバーヘッドを大幅に削減できる。更に、提案するテスト生成法では、時間展開モデル [7] を部分強可検査データパス用に拡張する。各回路要素に対するテスト生成に必要な時間展開数は、ある外部入力からその回路要素を通りある外部出力に至るまでの単純経路の順序深度程度となり、強可検査性におけるテストプラン長と同程度である。したがって、たかだか順序深度程度の時間展開モデルに対しては、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と考える。また、実動作速度でのテスト実行が可能であり、テスト実行時間が短いという利点がある。

以下、2. では、本論文で対象となるデータパスを述べる。3. では、部分強可検査性を定義し、4. では、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について述べる。5. では、ベンチマーク回路を用いた実験により、提案手法の有効性を示す。

2. 諸 定 義

2.1 データパス

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。回路要素には、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 (CM)、観測モジュール (OM) に分類される。それぞれの回路要素は、たかだか 2 個のデータ入力端子、たかだか 1 個の制御端子、たかだか 1 個の状態端子、たかだか 1 個のデータ出力端子をもつ。CM、OM 及び MUX を組合せ回路要素と呼ぶ。ロード制御のための制御端子をもつレジスタを HR とし、もたないものを LR とする。また、データ出力端子をもたず状態端子をもつ回路要素を OM とする。CM は更に CMA と CMB の二つの種類に分類する。CMA は、任意の値を入力に印加したとき任意の値を出力に設定できる CM とし、それ以外を CMB とする。具体的には、CMA はデータ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。データパスの制約として、すべての回路要素のデータ端子のビット幅は等しいものとする。信号線は、データ信号線、制御信号線、状態信号線に

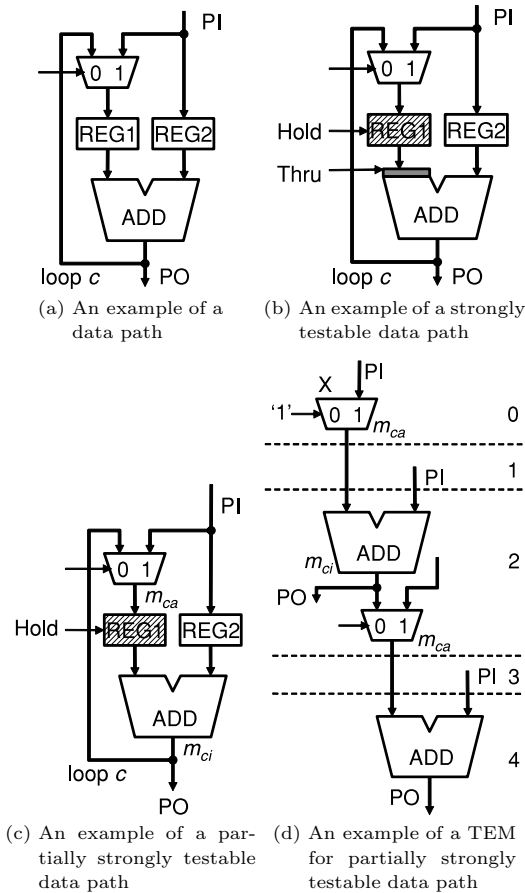


図 1 部分強可検査性
Fig. 1 Partially strong testability.

分類できる。データ信号線は、回路要素のデータ入力端子及びデータ出力端子を接続する。入力端子に接続可能なデータ信号線は 1 本とし、出力端子には複数のデータ信号線を接続可能とする。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の制御端子に接続される。状態信号線は、データパスの観測端子からコントローラに接続される。制御信号線の値は任意の時刻に任意の値が設定可能であり、状態信号線の値は任意の時刻に任意の値が観測可能であるとする。データパスの例を図 1 (a) に示す。

データパスの回路要素を e_i 、データ信号線を l_i と記述する。 e_i のデータ出力端子と e_{i+1} のデータ入力端子をデータ信号線 l_i が接続するとき、その経路 p を $p = (e_i, l_i, e_{i+1})$ と表す。 e_1 を始点とし、 e_k を終点とする経路 $p = (e_1, l_1, e_2, l_2, \dots, e_{k-1}, l_{k-1}, e_k)$ について考える。経路 p のすべての e_i が異なるとき、 p を単

純経路という。経路 p の始点 e_1 と終点 e_k が同じ回路要素であり、 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。経路 p に現れるレジスタ数を p の順序深度という。異なる二つの回路要素 e_1, e_2 に対して、 e_1 を始点とし、 e_2 を終点とする任意の異なる単純経路の組を再収れん経路の組という。

2.2 強可検査

[定義 1] (データパスの強可検査性 [4]) 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可能、かつ m の出力ポートから任意の値を外部出力へ伝搬可能なとき、 DP は強可検査であるという。□

データパス DP が強可検査であるとき、回路要素 m に対してテストプラン TP が存在する。組合せ回路要素に対しては、実用的なテスト生成時間で完全故障検出効率を達成できると考えられる。したがって、組合せ回路要素 m に対して組合せテスト生成を行い、それらのテストパターンをテストプランによって印加・観測することで完全故障検出効率が高くなる。図 1 (a) のデータパスに対して、強可検査性に基づくテスト容易化設計法 [4] を適用した例を図 1 (b) に示す。図 1 (a) では、外部入力から $ADD1$ のデータ入力端子までの経路は、 $ADD1$ で等しい順序深度で再収れんする。そのため、ある時刻において、 $ADD1$ のデータ入力端子間には同時に任意の値が設定不可能であり、依存関係をもつ (依存関係については [定義 3] 参照)。テストパターンの正当化及び出力応答の観測のために、図 1 (b) のように $REG1$ へのホールド機能の付加、及び ADD へのスルー機能の付加が行われる。

PI から回路要素 m までの単純経路のうち、順序深度が最大となる経路を p_{mi} とし、その順序深度を d_{mi} とする。また、 m から PO までの単純経路のうち、順序深度が最大となる経路を p_{mo} とし、その順序深度を d_{mo} とする。強可検査性に基づくテスト容易化設計法では、 m への値の正当化、 m の応答の伝搬、 m のテスト、 m への依存関係の解消により、 m のテストプラン長はたかだか $d_{mi} + d_{mo} + 2$ となる。

3. 部分強可検査性

本論文では、データパス中の回路要素 m に対して、強可検査性におけるテストプラン長と同程度の時間展開数で、 m に値域の任意の値が正当化及び観測可能となるための性質として、データパスの部分強可検査性

を定義する．

[定義 2](信号線の値域) 外部入力からデータ信号線 l に正当化可能なすべての値の集合を l の値域という． l に接続するデータ入力/出力端子の値域は l の値域で定義する． □

[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 は依存関係をもつという．同一の回路要素のデータ入力端子に接続する信号線 l_i, l_j は時刻 t において、各々の値域の任意の値が設定不可能であるとき、 t における信号線 l_i, l_j は依存関係をもつ．データ入出力端子における依存関係は、データ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． □

[定義 4](データパスの部分強可検査性) データパス DP に存在する閉路の集合を L とする． L 中の任意の閉路を c とし、 c に存在する回路要素の集合を M_c とする． L に属するすべての要素 c が以下の条件をすべて満たすとき、 DP は部分強可検査性であるという．

(1) ある回路要素 $m_{ca} \in M_c$ に対して、PI から単純経路を通り m_{ca} のデータ出力端子に任意の値が設定可能でありその値が m_{ca} から単純経路を通り PO で観測可能．

(2) 各回路要素 $m_{ci} \in M_c$ に対して、 c 上を通る(1)における m_{ca} から m_{ci} までの経路の順序深度を d_{ci} とする．時刻 t における m_{ca} のデータ出力端子と、時刻 $t + d_{ci}$ における m_{ci} の閉路外のデータ入力端子は依存関係をもたない． □

図 1(c) に部分強可検査データパスの例を示す．この回路は図 1(a) のデータパスに対し、REG1 にホールド機能を追加したものである．閉路 c に対しては、MUX の出力において PI からの単純経路で任意の値が設定可能である．また、 c 中に存在する ADD のデータ入力端子には、REG1 のホールド機能が存在するので、依存関係をもたない．強可検査データパスは、単純経路を用いて任意の値の伝搬を行うのに対し、部分強可検査データパスは、複数の経路を用いて任意の値の伝搬を行う．ADD は CMA であり、図 1(c) の部分強可検査データパスにおいて ADD のすべてのデータ入力端子には任意の値が印加できるため、ADD にはスルー機能の付加を行わない．したがって、強可検査データパスと比較して、部分強可検査データパスは、少ない面積オーバーヘッドで実現可能である．

[定理 1] 部分強可検査データパスを DP とする． DP 中の回路要素 m に存在する故障 f_m に対して、 f_m を検出可能な DP への入力系列が存在するとき、時間展開数がたかだか $d_{mi} + d_{mo} + 1 + n_{CM}$ で、 f_m が検出可能な時間展開モデルが生成可能である．ここで、 n_{CM} を DP 中の CM の総数とする． □

(証明) 部分強可検査性の条件 1 より、閉路 c 中のある回路要素 m_{ca} の出力には単純経路を通り任意の値が設定可能である． m_{ca} のデータ出力端子と接続する c 上の回路要素を m_{cb} とする． m_{cb} が 2 入力 CM であるとき、部分強可検査の条件 2 より、 m_{cb} のデータ入力端子間は依存関係をもたない．したがって、 m_{cb} のデータ出力端子には、値域の任意の値が設定可能となり、 c 中の m_{cb} のデータ出力端子と接続する回路要素に対しても、そのデータ入力端子間に依存関係をもたない．同様に、 c 中の任意の回路要素 m は、単純経路を通り値域の任意の値を m のデータ出力端子に設定可能である．また、PI から回路要素 m のデータ入力端子に至るまでが無閉路構造であるとき、 m のデータ出力端子に対しては、単純経路を通り値域の任意の値が設定可能である．以上より、 DP の任意の回路要素 m のデータ入力端子に対して、PI から単純経路 p_{mi} を通り、値域の任意の値が印加可能である．また、 m のデータ出力端子から単純経路 p_{mo} を通り、値域の任意の値が PO で観測可能である．したがって、 f_m は検出可能である．同様に、 m に存在する故障に対して、その故障を検出可能な DP への入力系列が存在しないとき、時間展開モデル上でも検出不可能である．ここで、 p_{mi} 及び p_{mo} 上に依存関係をもつ回路要素が存在するとき、それらの依存関係をすべて解消するためにたかだか n_{CM} サイクルが必要である．したがって、たかだか $d_{mi} + d_{mo} + 1 + n_{CM}$ の時間展開数で、 f_m が検出可能な時間展開モデルが生成可能である． □

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られる．また、順序深度程度の時間展開数の時間展開モデルに対しても、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．これらの前提のもとでは、定理 1 から、部分強可検査データパス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．すなわち、閉路のある順序回路に対して、 $d_{mi} + d_{mo}$ を順序深度と考えると、定理 1 より、データパスが部分強可検査性を満たすとき、データパ

ス中の任意の回路要素 m に存在する検出可能な故障を検出するのに必要な時間展開数は、順序深度程度となる。したがって、先の前提から、検出可能な故障を検出するのに必要な時間展開数が順序深度程度の時間展開数であるため、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といえる。

図 1(c) に対する時間展開モデルの例を図 1(d) に示す。PI から MUX の左のデータ入力端子までの経路 p_{mi} の順序深度 d_{mi} は 1 である。また、MUX から PO までの経路 p_{mo} の順序深度 d_{mo} は 1 である。また、 p_{mi} 及び p_{mo} 上に ADD が存在し、ADD は PI からの順序深度の時刻において、データ入力端子間に依存関係をもつ。依存関係を解消するために REG1 のホールド機能を 1 サイクル用いる必要がある。したがって、MUX に対する時間展開モデルの時間展開数は 5 となり、この時間展開モデルで MUX に対する値域の任意の値の印加及び観測が可能となる。

4. データパスのテスト容易化設計法

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もつ場合は、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。したがって、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 テスト容易化設計概要

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 3 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) 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もつかどうかを調べる。依存関係をもつ場合、各回路要素に対して依存関係を解消できるレジスタ及び、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。

(3) 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。その際、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もち、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。

4.3 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) すべ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。
- m_t が CMA であり、 m_t のすべ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} 中のすべてのレジスタを 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 in_{1m} とする。 m のデータ出力端子を out_m とする。 m が以下の条件のいずれかを満たすとき、 in_{1m} 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 で再収れんする再収れん経路であり、その再収れん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 またはスルー機能を付加した回路要素である場合、 in_{1m} から out_m までの経路を制御林に決

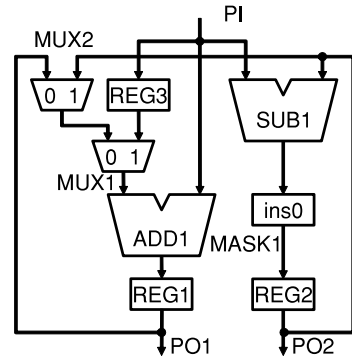


図2 データパスの例
Fig. 2 An example of a data path.

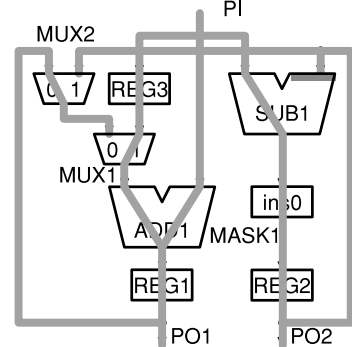


図3 制御林の例
Fig. 3 An example of a control forest.

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3に示す。図2のADD1、SUB1及びins0はCMA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で初期化し、手順(2)で C_C の要素がPIのみであるので、PIから探索を開始する。PIと接続するADD1及びSUB1を C_{CM} に追加し、REG3を C_{REG} に追加する。また、PIからADD1、SUB1及びREG3までの経路を制御林に決定し、 C_C からPIを削除する。次に、 C_C が空となるため、REG3を C_{REG} から削除し C_C に追加する。以降、同様に手順(2)~(4)を行うと、手順(5)において、SUB1は C_{CM} に属するので、 C_{CM} から削除し C_C に追加し、同様に手順(2)~(4)を行う。手順(6)において、SUB1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。

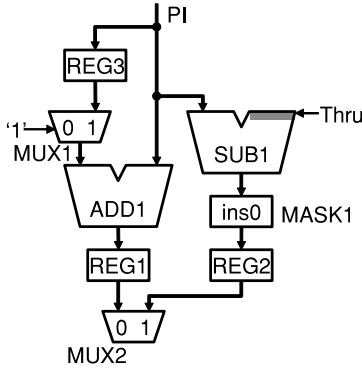


図 4 MUX2 の制御経路
Fig. 4 Control paths for MUX2.

4.4 依存関係の解消

ある 2 入力回路要素 m の制御経路が, m で再取れんする再取れん経路であり, その再取れん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には依存関係をもつ.

データ入力端子間に依存関係をも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 2 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 3 の制御経路から, MUX2 に対する制御経路のみを抜き出したものを図 4 に示す. MUX2 の左側のデータ入力端子には, PI からの制御経路を通り順序深度が 1 と 2 で到達する. MUX2 の右側のデータ入力端子には, PI からの制御経路を通り順序深度が 1 で到達する. したがって, MUX2 のデータ入力端子間には依存関係をもつ時刻が存在する. MUX2 は閉路上に存在するので, 部分強可検査性の定義の条件 2 を満た

すために, MUX2 のデータ入力端子間の依存関係を解消する必要がある. ここで, MUX2 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 REG1 のホールド機能を用いて MUX2 の依存関係を解消する. REG2 では依存関係を解消するのに必要なホールド回数が 2 回であるのに対し, REG1 ではホールド回数が 1 回で済む.

4.3 では, 閉路に任意の値を伝搬するためのスルー機能の付加及び制御経路を決定した. 更に, 4.4 では, データパス中の閉路に存在する各回路要素のデータ入力端子間に対して, 制御経路を用いたとき, 依存関係をもたないことを保証した. したがって, 4.3 及び 4.4 の DFT により, 閉路中のある回路要素の出力に任意の値が設定可能である. また, 閉路中の任意の回路要素は, 制御経路を用いたときに回路要素のデータ入力端子間に依存関係をもたない.

4.5 時間展開モデルの生成

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すべての回路要素を 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. 更に, 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 $m \in M_R$ が依存関係をもつとき、 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} のすべてのレジスタを C_{REG} から削除し、 C_{REG} のすべ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 2 に対する時間展開モデルの例を図 5 に示す。図 2 のデータパスでは、MUX1 と MUX2 のデータ入力端子はそれぞれ依存関係をもつ。図 5 の例では、手順 (2) で時刻 -1 に MUX1 と MUX2 を追加したとき、依存関係を解消するために時刻 -1 で REG1 (ADD1 と MUX2 間に存在) のホールド機能を 1 サイクル使用し、時刻 -3 に ADD1 を追加する。また、手順 (5) において、PO2 は時刻 -1 、 -2 で現れているため、PO2 からの操作は行わない。

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も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及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、もと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、すべ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

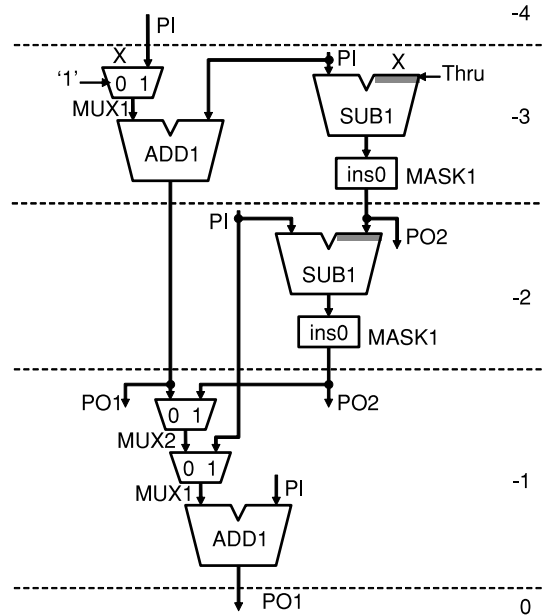


図 5 時間展開モデルの例
Fig. 5 An example of a time expansion model.

に追加し、2 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すべての回路要素及び、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もたない。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。したがって、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 のすべ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すべての故障は、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、順序深度程度となる。したがって、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、完全故障検出効率率が達成可能であると考えられる。

5. 実験結果

提案手法の有効性を実験により評価した。実験に使用したベンチマーク回路は、典型的なベンチマー

表 1 データパスの特性
Table 1 Data paths characteristics.

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

表 2 面積オーバーヘッド
Table 2 Hardware overheads.

Circuits	Hardware overhead[%]		
	FS	ST	PST
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	7.1	0.0
Mpeg	14.0	8.0	0.8

ク回路である GCD, LWF, JWF, PAULIN [4], [5] 及び, より実用的で大規模な回路である RISC と MPEG^(注1)[9] である. これらのベンチマーク回路の特性を表 1 に示す. 「bit」, 「#PI」, 「#PO」, 「#Reg」, 「#MUX」, 「#OP」, 「#OB」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 1 としたときの, 各ベンチマークの回路面積を表 1 の「area」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 [10] 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.

面積オーバーヘッド及びテスト生成結果を表 2 及び表 3 に示す. 「org」, 「FS」, 「ST」, 「PST」は, それぞれ, テスト容易化設計を行う前の回路, 完全スキャン設計法, 強可検査法 [4] 及び提案手法の結果を指す. 表 2 の面積オーバーヘッドは, テスト容易化設計前の回路の面積に対する付加ハードウェアの面積の比を示す. 表 3 の「Fault efficiency」にある括弧内の数値は, 故障検出率を示す. 「Test generation time」は, テスト生成時

間を示し, テスト容易化設計に要した時間は含まない. 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, 無視できるくらいに小さい. 「Test application time」は, テスト実行時間を示す. FS は, すべての 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. FS における面積オーバーヘッドは, もとのデータパスに対する, すべての 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. また, FS 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. ST は, [4] の方法をもとに実験を行った. ST のテスト実行時間は, すべての「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.

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, 提案手法は更に大幅に削減ができています. 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が 0 となった.

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. また, 完全スキャン設計法, 強可検査法及び提案手法は, 同程度のテスト生成時間となった. 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. 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. これは, 強可検査法が回路要素ごと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

(注1): 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 3 テスト生成結果
Table 3 Test generation results.

Circuits	Fault efficiency[%] (Fault coverage)				Test generation time[s]				Test application time[clock]			
	org	FS	ST	PST	org	FS	ST	PST	org	FS	ST	PST
GCD	100.00 (100.00)	100.00 (100.00)	100.00 (100.00)	100.00 (100.00)	3.18	0.20	0.83	0.39	133	2351	387	197
LWF	100.00 (99.90)	100.00 (99.90)	100.00 (100.00)	100.00 (99.90)	0.33	0.21	0.62	0.24	56	2674	250	70
JWF	100.00 (99.95)	100.00 (99.95)	100.00 (100.00)	100.00 (99.95)	3.63	0.60	0.62	1.74	244	14849	769	608
Paulin	99.23 (99.20)	100.00 (100.00)	100.00 (100.00)	100.00 (100.00)	297.50	0.83	0.86	4.69	147	2824	875	526
Tseng	99.01 (98.96)	100.00 (100.00)	100.00 (100.00)	100.00 (100.00)	703.90	0.34	1.08	0.87	590	4752	633	523
Risc	99.37 (99.34)	100.00 (99.97)	100.00 (100.00)	100.00 (99.69)	12210.81	55.17	76.56	60.02	2271	621284	5520	3420
Mpeg	88.30 (87.16)	100.00 (99.45)	100.00 (99.51)	100.00 (99.45)	68947.90	1.84	1.18	13.34	1216	185183	107359	8448

で、故障シミュレーションにより効率良く故障が検出でき、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。また、結果としてテスト実行時間を強可検査法より更に短縮するのに成功している。

6. 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、テスト系列長が短くなることを示した。

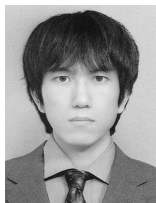
本研究では、レジスタ転送レベル回路のデータパスのみを対象としており、コントローラの内容を考慮していない。今後の課題としては、コントローラの内容を考慮した場合での適用が挙げられる。

謝辞 本研究に際し、多くの貴重な意見を頂いた本学の井上美智子助教授、並びにコンピュータ設計学講座の諸氏に深く感謝します。本研究は一部、21世紀COEプログラム(研究拠点形成費補助金)、及び、日本学術振興会科学技術研究費補助金・基盤研究B(2)(課題番号15300018)の研究助成による。

文 献

[1] H. Fujiwara, *Logic Testing and Design for Testability*, The MIT Press, 1985.

- [2] M. Abramovici, M.A. Breuer, and A.D. Friedman, *Digital Systems Testing and Testable Design*, Computer Science Press, 1990.
- [3] R.B. Norwood and E.J. McCluskey, "Orthogonal scan: Low overhead scan for data paths," *Proc. 1996 Int. Test Conf.*, pp.659-668, 1996.
- [4] H. Wada, T. Masuzawa, K.K. Saluja, and H. Fujiwara, "Design for strong testability of RTL data paths to provide complete fault efficiency," *Proc. 13th International Conf. on VLSI Design*, pp.300-305, Jan. 2000.
- [5] 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," *Proc. ASP-DAC*, pp.331-334, 2001.
- [6] J. Lee and J.H. Patel, "Hierarchical test generation under architectural level functional constraints," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol.15, no.9, pp.1144-1151, Sept. 1996.
- [7] 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.
- [8] 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.
- [9] S. Ohtake, H. Wada, T. Masuzawa, and H. Fujiwara, "A non-scan DFT method at register-transfer level to achieve 100% fault efficiency," *Trans. IPSJ*, vol.44, no.5, pp.1266-1275, May 2003.
- [10] 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. (平成17年10月27日受付, 18年3月1日再受付)



岩田 浩幸 (学生員)

平 15 横浜国大・工・電子情報卒。平 17 奈良先端科学技術大学院大博士前期課程了。現在、同大博士後期課程に在学中。VLSI CAD, テスト容易化設計に関する研究に従事。



米田 友和 (正員)

平 10 阪大・工・情報システム卒。平 13 奈良先端科学技術大学院大博士前期課程了。平 14 同大博士後期課程了。現在奈良先端大・情報科学研究科助手。VLSI CAD, テスト容易化設計, システムオンチップのテストアーキテクチャ及びスケジューリング

に関する研究に従事。



大竹 哲史 (正員)

平 7 電通大・電通・情報卒。平 11 奈良先端科学技術大学院大博士後期課程了。現在奈良先端大・情報科学研究科助手。平 10 日本学術振興会特別研究員。VLSI CAD, テスト容易化設計, テスト生成アルゴリズム

に関する研究に従事。平 13 本会情報システムソサイエティ論文賞など受賞。IEEE Computer Society, 情報処理学会各会員。



藤原 秀雄 (正員)

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