社団法人　情報処理学会　研究報告
IPSJ SIG Technical Reports

2004－SLDM－117（24）
2004／12／2

ビット幅調整機能を用いたデータパスのテスト容易化設計法

村田　優†　大竹　哲史†　藤原　秀雄†

† 奈良先端科学技術大学院大学情報科学研究科　〒 630–0192 けいはんな学研都市
E-mail: †{yuu-m,ohtake,fujiwara}@is.naist.jp

あらまし　本稿では，ビット幅不均一レジスタ転送レベルデータパスを対象とした完全故障検出効率を保証するテスト容易化設計法を提案する．提案するテスト容易化設計法は，ビット幅の均一なデータパスに対して提案された直交スキャン設計を拡張したもので，組合せ回路用テスト生成ツールによるテスト生成を可能としている．実験結果より提案手法は，組合せ回路用テスト生成を利用する従来法の完全スキャン設計法に比べ，ハードウェアオーバーヘッドが小さく，テスト実行時間が短い．さらに，本稿で提案するビット幅調整機能は，ビット幅の均一なデータパスに対して提案された階層テストに基づく手法をビット幅の不均一なデータパスに対しても適用可能にするものである．

キーワード　レジスタ転送レベル，テスト容易化設計，ビット幅不均一データパス，ビット幅調整機能，完全故障検出効率

A method of DFT for data paths using bit-match function

Yuu MURATA†, Satoshi OHTAKE†, and Hideo FUJIWARA†

† Nara Institute of Science and Technology Kansai Science City, 630–0192, Japan
E-mail: †{yuu-m,ohtake,fujiwara}@is.naist.jp

Abstract　In this paper, we propose a method of design-for-testability(DFT) which guarantees complete fault efficiency for register-transfer level data paths with irregular bit width. The proposed DFT method is an extension of the orthogonal scan method which was proposed for data paths with even bit width. The proposed method employs a combinational automatic test pattern generation(ATPG) tool. From the experimental results, the hardware overhead of the proposed method is smaller than that of full scan design which is a typical technique and allows combinational ATPG. The test application time of the proposed method is also shorter than that of full scan design. Moreover, the bit-match function proposed in this paper makes a method based on hierarchical testing for data paths with even bit width applicable to data paths with irregullar bit width.

Key words　Register-transfer level, design for testability, data paths with irregular bit-width, bit-match function, complete fault efficiency

1. まえがき

近年，VLSI の大規模化，高集積化に伴い，テスト費用の削減およびテストの質の向上はますます重要になっている．テスト費用は，テスト生成時間やテスト実行時間で評価され，テストの質は故障検出率や故障検出効率で評価される．組合せ回路のテスト生成に関しては，効率の良いテスト生成アルゴリズム [1] が提案されており，実用的なテスト生成時間で単一縮退故障に対して完全（100%）故障検出効率を達成することができる．しかし，順序回路に対するテスト生成は，一般に膨大な時間がかかり，完全故障検出効率を得るのは困難である．そのため，単一縮退故障に対して高い故障検出効率を達成できるよう，順序回路の設計を変更するテスト容易化設計（DFT）が行われている．

代表的な DFT として完全スキャン設計 [1] がある．完全スキャン設計では，ゲートレベルにおいて，回路中の全てのフリップフロップ（FF）をスキャン FF に置き換えることで，FF の値を外部から制御および観測可能としている．したがって，FF を外部入出力端子で置き換えて得られる組合せ回路（核回路）に対してテスト生成を行うことができ，組合せテスト生成アルゴリズムが完全故障検出効率を達成することができる．しかし，完全スキャン設計には以下の問題がある．

（1）　論理合成後の回路を変更するので，タイミング等の合成時の制約を損なう．

（2）　ハードウェアオーバーヘッドが大きい．

（3）　テスト系列長が大きい．

（4） at-speed テストができない．

上記の問題を解決するために，ゲートレベル回路に合成される前のレジスタ転送レベル（RTL）設計を対象としたDFTが提案されている[3]～[5]．回路はRTLではデータを処理するデータパスとデータパスの動作を制御するコントローラの2つの部分回路で構成さる．ここで，データパスはレジスタやマルチプレクサ，演算器などの回路要素の相互接続で表現される．これまで，上記の問題点の解消のため，RTLデータパスを対象としたDFT法が提案されている[3]～[6]．RTL回路に対してDFTを行うことにより，論理合成後の回路へのDFTが不要となるため，問題点（1）は解消される．問題点（2）～（4）を解消する目的で，データパスが通常動作で使用するデータ転送経路を用いてレジスタの値の制御および観測を行うDFTとして直交スキャン設計[2],[3]や階層テスト生成に基づくDFT[4]が提案されている．これらの手法は，ハードウェアオーバヘッドが完全スキャン設計より小さく，テスト実行時間も短いことが実験により示されている．したがって，問題点（2),(3）は解消される．一般に，完全スキャン設計では，回路を通常動作のクロック周波数で動作させて行うテスト（at-speedテスト）ができない．上述の直交スキャン設計や階層テスト生成に基づくDFTでは，テスト実行時のテストパターンや応答の伝達にデータパスの通常動作時のデータ転送経路を利用するので，通常動作のクロック周波数でテストが可能であるので，問題点（4）を解消している．しかし，これらの手法は，ビット幅が均一なデータパスに対して考えられたもので，ビット幅が不均一なデータパスに対してはそのまま適用することはできない．

不均一なビット幅をもつデータパスに対する手法として，上述の階層テスト生成に基づくDFT[4]を拡張した手法が提案されている[5]．この手法では，通常動作で使用するデータ転送経路では制御および観測できないレジスタの一部のビットに対して，それぞれ外部入力から新たにテスト用の経路を付加して制御，および，外部出力へ新たにテスト用の経路を付加して観測することにより，文献[4]の手法をビット幅が不均一なデータパスに対して適用できるようにしている．しかし，回路によっては多くの外部入出力を追加する必要があり，実用上は問題が起こると考えられる．

また，不均一なビット幅をもつデータパスに対して，テスト容易化設計に頼らず，RTLの情報を用いてテスト生成のみで解決しようとする方法も提案されている[6]～[8]．この場合，ハードウェアオーバヘッドはかからないが，ゲートレベルのテスト生成よりもテスト生成時間がかかる場合や，完全故障検出効率を達成できない場合がある．

本稿では完全故障検出効率を保証し，完全スキャン設計の問題点（1）～（4）を解決する不均一なビット幅をもつRTLデータパスに対するDFT手法を提案する．具体的には，直交スキャン設計法を拡張し，文献[2]の均一なビット幅の直交スキャンパスを，不均一なビット幅に対応する．提案法では，異なるビット幅のレジスタ間で任意の値を伝搬するために，ビット幅調整機能（bit-match function）[9]を用いる．この機能は，元々システム・オン・チップのテストの際に，コアの入力ビット幅と出力ビット幅の差を解消するために提案されたもので，新たにフリップフロップなどの記憶素子を付加することで実現するため，ハードウェアオーバーヘッドが大きくなる．本稿では，データパスの元々持っている機能を用いてビット幅調整機能を実現する方法を提案する．本稿で提案するビット幅調整機能は，直交スキャン設計をビット幅の不均一なデータパスに適用可能にするだけでなく，階層テスト生成に基づく手法などにも応用可能である．

本稿では，ベンチマーク回路を用いた実験により，従来の完全スキャン設計と比べて，ハードウェアオーバヘッドおよびテスト実行時間を削減できることを示す．また，ビット幅の変化する箇所に，本稿で提案する万能なビット幅調整機能を付加して直交スキャン設計[2]を適用した場合と，提案法を用いた場合を比較し，提案法が適切なビット幅調整機能を選ぶことができ，前者よりもハードウェアオーバヘッドおよびテスト実行時間が短くなることを示す．

2. 諸 定 義

本稿で扱うRTLデータパスおよびDFT要素について述べる．

2.1 データパス

データパスは信号線と回路要素で構成される．信号線は，制御信号線，状態信号線，データ信号線に分類される．制御信号線はコントローラからデータパスの回路要素へ制御信号を伝達し，状態信号線はデータパスの回路要素からコントローラへ状態信号を伝達する．

本稿では，テスト実行時にすべての制御信号線に対して任意の系列を回路外部から設定可能で，かつ，すべての状態信号線上の任意の系列を回路外部から観測可能であると仮定する．

データ信号線（以下，信号線という）はデータ出力端子とデータ入力端子を接続する．データ出力端子には複数の信号線が接続できる（ファンアウト可能）が，データ入力端子に接続できる信号線は1つとする．

回路要素は，外部入力，外部出力，レジスタ，マルチプレクサ，演算モジュール，観測モジュール，分割ノード，収束ノード，終端ノードに分類され，各種信号線が接続される端子を持つ．データ信号線が接続される端子をデータ端子，制御信号線が接続される端子を制御端子，状態信号線が接続される端子を状態端子と呼ぶ．データ端子は回路要素の入力であるデータ入力端子（以下，入力端子という）と，出力であるデータ出力端子（以下，出力端子という）に分類される．各データ端子は，それぞれ任意のビット幅を持つ．

便宜上，外部入力は1つの出力端子のみを，外部出力は1つの入力端子のみをもつ回路要素として扱う．レジスタは，1つの入力端子と1つの出力端子をもつ．また，各レジスタの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，出力端子からは少なくとも1つの外部出力に到達可能とする．マルチプレクサは2つ以上の入力端子と1つの出力端子，1つの制御端子をもち，制御端子の値によって，いずれかの入力端子の値を出力する．演算モジュールは1つ以上の入力端子，1つの出力端子，高々1つの制御端子，高々1つの状態端子をもつ．観測

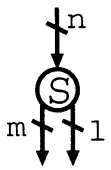

図 1　分割ノード

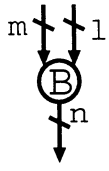

図 2　収束ノード

図 3　終端ノード

モジュールは 1 つ以上の入力端子，1 つの状態端子，高々1 つの制御端子をもち，出力端子をもたない．観測モジュールは比較器等のモデルである．

分割ノードは 1 つの入力端子と 2 つ以上の出力端子をもち，信号線のビット分割を表す．例えば入力 X（24 ビット）と 2 つの出力 Y（16 ビット），Z（8 ビット）をもつ分割ノード S は，以下のように表すことができる．

$$S(X[23:8] \rightarrow Y[15:0], X[7:0] \rightarrow Z[7:0])$$

収束ノードは 1 つ以上の入力端子と 1 つの出力端子をもち，信号線の収束を表す．例えば 2 つの入力 X（8 ビット），Y（16 ビット）と出力 Z（24 ビット）をもつ収束ノード B は，以下のように表すことができる．

$$B(X[7:0] \rightarrow Z[7:0], Y[15:0] \rightarrow Z[23:8]$$

終端ノードは 1 つの入力端子のみをもち，信号線の終端を表す．分割ノード，収束ノード，終端ノードの回路表記を図 1，2，3 にそれぞれ示す．図 1，2 において n, m, l の関係は，$n = m + l$ である．

2.2　レジスタ連結グラフ

本稿では直交スキャンパスの構成を決めるために，レジスタ連結グラフを用いる．レジスタ連結グラフは重み付き有向グラフ $G = (V, E, w)$ で表す．ここで，頂点集合 $V = V_1 \cup V_2$ はデータパス上のすべてのレジスタの集合 V_1 およびすべての外部入出力の集合 V_2 からなる．また，有向辺 $(v_i, v_j) \in E$ はレジスタ（外部入出力）v_i，$v_j \in V$ の間に組合せ回路部分のみを通って転送できる経路が存在することを示す．重み $w(v_i, v_j)$ は，v_i に対応するレジスタから v_j に対応するレジスタへ任意の値を伝搬するために必要な DFT 要素のハードウェア量とする．DFT 要素については次節以降で述べる．

図 4 に示すデータパスに対応するレジスタ連結グラフを図 5 に示す．

2.3　使用する DFT 要素

2.3.1　レジスタのホールド機能

レジスタの値を任意の時間保持するための機能である．レジスタの直前にマルチプレクサを追加し自己ループを構成することによって実現する．このときのハードウェアオーバーヘッドはレジスタのビット幅を n とすると，n ビット分のマルチプレクサとなる．

2.3.2　モジュールのスルー機能

2 つの入力端子 in_1（l ビット），in_2（m ビット）と 1 つの出力端子 out（n ビット）をもつ演算モジュール M を考える．

in_1 の $min(l,n)$ ビットの値を変化させることなく out の $min(l,n)$ ビットへ伝搬する機能をモジュール M の in_1 のスルー機能といい，以下のように記述する．

$$Thru(out[min(l,n):0], in_1[min(l,n):0])$$

1 入力モジュール，3 入力以上のモジュールについても同様に記述することができる．この機能を実現する方法としてマスク素子を用いたものとマルチプレクサを用いたものがある．

M が加算器，減算器，乗算器などのような単純な演算モジュールである場合には，マスク素子を用いることができる．これは，モジュール自身の機能を用いてスルーを実現するためである．例えば，先に考えた演算モジュール M が加算器であり，in_1 のスルー機能を実現することを考える．M の機能は加算であるため in_2 の値を“0”とすることで in_1 の下位 $min(l,n)$ ビットの値を変化させることなく out の下位 $min(l,n)$ ビットへと伝搬できる．そこで，in_2 に強制的に値を“0”とするマスク素子（この場合は m 個の AND ゲート）を付加することでスルー機能を実現することができる．

マルチプレクサを用いた場合は，スルーする回路要素の入力と回路要素の出力をマルチプレクサで接続することで実現する．マスク素子を用いた場合と異なり回路要素の機能に影響されないためいかなる回路要素に対しても適用することができる．スルー可能なビット幅は，マスク素子を用いた場合と同じであるが，入力の任意のビットを出力の任意のビットにスルーするといった複雑なスルー機能も実現することができる．しかし，ハードウェアオーバヘッドは $min(l,n)$ 個の 1 ビット 2 入力マルチプレクサとなり，マスク素子を用いた場合よりも一般的に大きくなる．

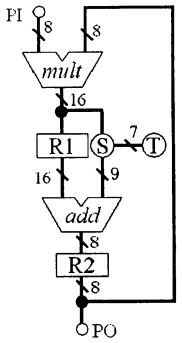

図 4　データパス

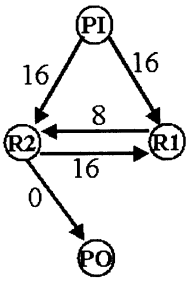

図 5　レジスタ連結グラフ

3.　ビット幅調整機能

本稿で提案する DFT 要素であるビット幅調整機能について述べる．ビット幅調整機能として，システム *cdot* オン *cdot* チップのテストの際に，コアの入力ビット幅と出力ビット幅の差をうめるために提案されたものがある [9]．これは，新たにフリップフロップなどの記憶素子を付加するのでハードウェアオーバーヘッド大きくなる．本稿では，データパス内部に存在するレジスタを利用してこの機能を実現する方法を提案する．そして，スルー機能およびホールド機能だけでは完全可制御・完全可観測が得られないレジスタに対してこの機能を付加することで完全可制御・完全可観測に必要なビットの制御性と観測性を

補う..

図6にスルー機能およびホールド機能だけではレジスタに対して完全可制御・完全可観測が得られない例を示す．M_1 の入力ビット幅 m_1 と M_2 の出力ビット幅 m_2，Reg のビット幅 n の関係は $m_1 < m_2 < n$ であることから，M_1, M_2 にどの様なスルー機能を付加しても Reg は完全可制御・完全可観測を得られない．このため，Reg を完全可制御・完全可観測とするために，Reg に対してビット幅調整機能を付加する必要がある．

ビット幅調整機能を実現する方法として，循環シフトレジスタを用いる方法とシフトレジスタを用いる方法を考える．

循環 k ビットシフトレジスタを用いた手法

循環 k ビットシフトレジスタとは，1クロックにつき k ビットずつレジスタの内容を左（または右）シフトし，上位（または下位）からシフトアウトされた k ビットを下位（または上位）からシフトインするものである．この手法は，ビット幅調整機能を必要とするレジスタを循環 k ビットシフトレジスタに置き換えることで実現できる．

例えば，図6の左の図において，M_1，M_2 がそれぞれ $Thru(C[m_1-1:0], A[m_1-1:0])$，$Thru(F[m_2-1:0], D[m_2-1:0])$ を持っている場合を考える．この時 Reg は $Reg[m_1-1:0]$ が可制御であり，$R[m_2-1:0]$ が可観測となる．ここで，Reg を循環 m_1 ビットシフトレジスタに置き換えると，循環 m_1 ビットシフトしつつ $Reg[m_1-1:0]$ を更新することで Reg のすべてのビットを制御できる．また，循環 m_1 ビットシフトしつつ $Reg[m_2-1:0]$ を観測することで Reg のすべてのビットを観測できる．よって，Reg は完全可制御・完全可観測となる．このときのハードウェアオーバヘッドは n 個のマルチプレクサと制御信号線2本である．

循環1ビットシフトレジスタは，どのようにビット幅に過不足がある場合でも用いることができる（万能）ので，これを用いることで，Norwood らによって提案された直交スキャン設計[2]に変更を加えることなくビット幅が不均一なデータパスに対しても適用することが可能となる．

k ビットシフトレジスタを用いた手法

k ビットシフトレジスタとは，1クロックにつき k ビットずつレジスタの内容を左（または右）にシフトするものである．上位（または下位）からシフトアウトされた k ビットは破棄される．

k ビットシフトレジスタを用いた場合の実現方法を図6に示す．図6においてDFTを施す前の状態では，M_1, M_2 にスルー機能があったとしても Reg に m_1 ビットしか任意の値を制御でず，m_2 ビットしか任意の値を観測できない．しかし，DFT後の状態では，Reg の値を左シフトしつつ $Reg[m_1-1:0]$ を更新することで n ビットすべてを制御可能にしている．観測に関しても同様に左シフトしつつ $Reg[n:m_1]$ を観測することで n ビットすべてを観測可能としている．このとき n ビット制御するために必要なハードウェアオーバーヘッドは $(n-m_1)$ 個のマルチプレクサと制御信号線1本となる．

ここで，注意しなければならないのはレジスタの前後の回路要素において共にビット幅が変化した場合，k ビットシフトレジスタ単独ではレジスタに対して完全可制御・完全可観測が得られないということである．このような場合は，レジスタの前後の回路要素に対して上位と下位の逆側をスルーするようにスルー機能を付加することで完全可制御・完全可観測を得る．例えば，図6において M_1 のスルー機能を下位ビットスルー機能，すなわち $Thru(B[m_1-1:0], A[m_1-1:0])$ としたならば，M_2 のスルー機能は上位ビットスルー機能，すなわち $Thru(F[m_2-1:0], D[n-1:n-m_2])$ としなければならない．

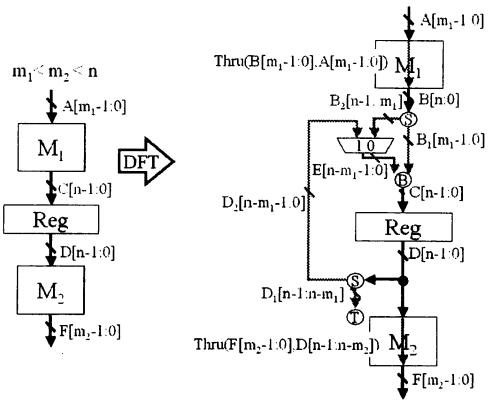

図6 ビット幅調整機能

これらの実現方法は，レジスタのシフト機能を有する回路構造をもつデータパスであれば，その部分を利用して実現可能である．しかし，一般的にそのような回路構造を見分けることは難しいが，そのような回路構造を見分けることができるならばハードウェアオーバーヘッドは，より小さくすることが可能である．

4. テスト容易化設計

4.1 拡張直交スキャン設計

本稿で提案する拡張直交スキャン設計について述べる．拡張直交スキャン設計は，直交スキャン設計と同様に拡張直交スキャンパスを用いてデータパス中のすべてのレジスタを同時に完全可制御・完全可観測とする．本稿で扱う拡張直交スキャンパスを以下に定義する．

[定義1] 拡張直交スキャンパス

以下の条件を満たすRTLのパス P を拡張直交スキャンパスと呼ぶ．

- P は外部入力から始まり外部出力で終わる．
- P 上のすべてのレジスタの値を，P の始点から P 上の回路要素のホールド，スルー，ビット幅調整機能のみを用いて同時に任意の値に設定できる．
- P 上のすべてのレジスタに同時に存在する任意の値を，P の終点へ P 上の回路要素のホールド，スルー，ビット幅調整機能のみを用いてすべて伝搬できる． □

[定義2] 完全拡張直交スキャン設計

与えられたデータパスに対して，以下のすべての条件を満たすように設計変更することを完全拡張直交スキャン設計と呼ぶ．

- 各レジスタは少なくとも1つの拡張直交スキャンパスに含まれる．
- 複数の拡張直交スキャンパスがある場合，すべてのレジスタはホールド機能をもつ． □

4.2 基本方針

拡張直交スキャンパスは1つのデータパスに対して何通りも構成することが可能であり，その構成によってハードウェアオーバーヘッドは異なる．そこで，本稿では，レジスタ連結グラフ $G=(V,E,w)$ を用い，なるべく小さいハードウェアオーバーヘッドで完全拡張直交スキャンデータパスを実現することを考える．

レジスタ連結グラフ G における重み $w(v_i,v_j)$ は，以下のように計算する．頂点 v_i に対応する回路要素から頂点 v_j に対応する回路要素への各組合せパス P_k で任意の値を伝搬するためのハードウェアオーバーヘッドの最大値を C_k とする．すなわち，C_k はスルー機能（マルチプレクサ実現）に必要なマルチプレクサのビット幅と，ビット幅調整機能（シフトレジスタ実現）に必要なマルチプレクサのビット幅の和とする．C_k のうち最も小さい値を $w(v_i,v_j)$ とする．

提案するDFTアルゴリズムは2つのステップからなる．

（1） データパス中のすべてのレジスタを制御・観測するための経路として拡張直交スキャンパスの選択

（2） 選択された拡張直交スキャンパス上で任意の値を伝搬できるようにDFT要素の追加

4.3 拡張直交スキャンパスの選択

レジスタ連結グラフ $G=(V,E,w)$ において，重みが最小となる拡張直交スキャンパスの選択は，完全有向グラフ $G^*=(V,E^*,w^*)$ 上で巡回セールスパーソン問題（TSP, Traveling Salesperson Problem）[10]を解くことと同じである．ここで，G^* の頂点集合は G と同じ V である．G^* の重み $w^*(v_i,v_j)$ は，G に外部出力に対応する任意の頂点から外部入力に対応する任意の頂点への重み0の辺を加えたグラフ上で，v_i，v_j 間の最短経路長とする．TSPは，NP困難問題であるが，これまでに，TSPを解く多くのヒューリスティックアルゴリズムが提案されているので，それらを用いて拡張直交スキャンパスを求めることができる．

4.4 値の伝搬の保証

各拡張直交スキャンパスについて，まずパス上の演算モジュールに対してスルー機能を付加する．スルー機能の付加は以下のように行う．

ステップ1 データパス中においてビット幅が変化している部分パス P を考える．P の始点，終点は外部入力，レジスタ，または外部出力とする．P 上の演算モジュールについて，レジスタ段数に関するパリティを割当てる．

ステップ2 割当てられたパリティが同じモジュールに対して同一のスルー機能を与える．スルー機能の割当ては2通りあり，パリティが1のモジュールに上位スルーを与える場合と下位スルーを与える場合のそれぞれについてスルーを付加したときのハードウェアオーバーヘッドを計算し，ハードウェアオーバーヘッドが小さい方のスルー機能を実現する．

次に，完全可制御・完全可観測が得られていないレジスタ $R[n:0]$ に対してビット幅調整機能を付加する．ビット幅調整機能には，2通りの実現方法があるため R の制御可能ビット位置によって以下のように使い分ける．

- $R[n:m]$ もしくは，$R[m:0]$ のみ制御可能の場合は，シフトレジスタを用いる．
- 上記以外の場合は循環シフトレジスタを用いる．

最後に，複数クロック値を保持する必要があるレジスタに対してホールド機能を付加する．

例として図4に示すデータパスに対してDFT要素を付加することを考える．ここで，拡張直交スキャンパスとして $PI \rightarrow R_1 \rightarrow R_2 \rightarrow PO$ が選ばれたとするとDFT対象となる回路要素は，*mult*，R_1，*add*，R_2 である．まず，*mult*，*add* にそれぞれスルー機能を付加する．ここで，ビット幅が変化している部分パスを抜き出す．パスのビット幅の変化は，PI（8ビット）$\rightarrow R_1$（16ビット）$\rightarrow R_2$（8ビット）$\rightarrow PO$（8ビット）であるから PI から R_2 までが抜き出され，*mult* と *add* にパリティが割当てられる．*mult* のパリティを0とすると *add* は1となる．スルー機能の付加の仕方によるハードウェアオーバーヘッドをそれぞれ計算すると，パリティ0のモジュール（*mult*）に上位スルーを付加した場合は，*mult* の上位をスルーするためのマルチプレクサ8ビットとパリティ1のモジュール（*add*）の下位スルーのためのマスク素子9ビットとなる．一方，*mult* に下位スルーを付加した場合は，マルチプレクサ8ビットとマスク素子8ビットとなる．よって，パリティ0のモジュール（*mult*）に下位スルーを，パリティ1のモジュール（*add*）に上位スルーをそれぞれ付加する．次に，R_1 に8ビット左シフトレジスタを用いたビット幅調整機能を付加する．最後に，R_2 にホールド機能を付加する．

5. 提案したビット幅調整機能の応用

ビット幅調整機能を用いることで階層テストに基づくDFT[4]をビット幅が不均一なデータパスに対しても適用することができる．

具体的には，ビット幅の変化によって任意の値が正当化（観測）できないモジュールの直前（直後）のレジスタに対してビット幅調整機能を付加する．これによって，データパス中の各モジュールの入力には，任意の値を外部入力から正当化することができ，レジスタに取り込まれた任意の応答を外部出力で観測することができる．

6. 実験結果

実験では，完全故障検出効率を保証するテスト容易化設計法として，完全スキャン設計法，ビット幅調整機能に循環1ビットシフトレジスタを使用した拡張直交スキャン設計法（以下，*RS-EOS*）（従来の直交スキャン設計と万能なビット幅調整機能を組み合わせたものと考えることができる）とビット幅調整機能に多ビットシフトレジスタを使用した拡張直交スキャン設計

表1 データパスの特性

回路名	外部入力数（ビット数）	外部出力数（bit 数）	面積	レジスタ数（FF 数）
b04	1 (8)	1 (8)	2540	8 (64)
b11	2 (7)	1 (6)	1766	2 (15)
Paulin*	2 (16)	2 (12)	5224	7 (56)
LWF*	2 (12)	2 (32)	2569	5 (60)

法（以下，*S-EOS*）を比較する．表1に示す4種類のデータパスに対して，ハードウェアオーバーヘッド，テスト系列長の2項目を評価した．表1中の「b04」,「b11」はITC'99のベンチマーク[11]で,「Paulin*」,「LWF*」は，それぞれRTLのテスト容易化設計のベンチマークとしてよく用いられるビット幅の均一な「Paulin」と「LWF」のデータパスに対して,(1) 乗算器の出力のビット幅を入力のビット幅の総和にする.(2) ファンアウトを分割ノードに置き換える．といった変更を加えたものである．面積はDesign Compiler（Synopsys）で合成して得られたユニット数を示している．

ハードウェアオーバーヘッドに関する実験結果を表2に，テスト系列長に関する実験結果を表3にそれぞれ示す．

表2中の数値は，DFT後のデータパスの面積に対するDFT要素の占める割合をパーセントで表している．ここで，付加DFT要素の面積はそれぞれDesign Compilerで求め，1ビット2入力1出力マルチプレクサは9ユニット，ANDゲート，ORゲートは4ユニット，NOTゲートは2ユニットとして計算した．

表3において，各データパスの組合せ回路部分に対するテストベクタ数をnとし，テスト系列長はn個のテストベクタの印加と，その応答の観測に必要なクロック数を示している．

表2，3よりすべての回路に対して，*RS-EOS*と*S-EOS*は共にハードウェアオーバーヘッド，テスト系列長がそれぞれ*FS*よりも小さくなっている．このことより，本稿で提案した拡張直交スキャン設計法は，直交スキャン設計法の利点を失うことなくビット幅が不均一なデータパスに対して完全故障検出効率を得ることが出来ると確認できる．また，*S-EOS*のハードウェアオーバーヘッドがすべての回路に対して*RS-EOS*よりも小さくなっているのは，レジスタの前後のモジュールに対するスルー機能を調整することでビット幅調整機能に必要なマルチプレクサの数を減らすことに成功したためである．

*S-EOS*のテスト系列長が*RS-EOS*の系列長より短くなっているのは，ビット幅調整機能に多ビットシフトレジスタを用いたためである．*RS-EOS*に関してもビット幅調整機能に循環多ビットシフトレジスタを用いればテスト系列長はさらに短くすることは可能である．しかし，ビット幅調整を行うレジスタが同じであれば*RS-EOS*のテスト系列長が*S-EOS*のテスト系列長より短くなることはない．

7. むすび

本稿では，与えられたレジスタ転送レベルデータパスに対して，完全故障検出効率を保証するテスト容易化設計法を示した．

表2 ハードウェアオーバーヘッド（%）

回路名	*FS*	*RS-EOS*	*S-EOS*
b04	22.7	10.7	6.0
b11	7.6	4.6	1.5
Paulin*	9.6	8.3	4.8
LWF*	21.0	20.6	16.3

表3 テスト系列長（単位：クロック）

回路名	*FS*	*RS-EOS*	*S-EOS*
b04	$65n+64$	$21n$	$19n$
b11	$16n+15$	$12n$	$7n$
Paulin*	$57n+57$	$43n$	$11n$
LWF*	$61n+60$	$36n$	$13n$

提案したテスト容易化設計法では，ビット幅調整機能を用いることによってビット幅が不均一なデータパスに対しても直交スキャンを適用できるようにした．また，完全故障検出効率を保証するテスト容易化設計法として完全スキャン設計と提案法の比較実験をベンチマーク回路に対して行い，ハードウェアオーバーヘッド，テスト系列長の2点で，提案法が完全スキャン設計より優れていることを示した．

今後の課題としては，階層テスト生成に基づくDFTへの適用と評価，ハードウェアオーバーヘッドとテスト実行時間の相互最適化を行える拡張直交スキャンパス選択アルゴリズムを考案することなどが挙げられる．

謝辞 本研究は一部，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, Cambridge, 1985.

[2] R. B. Norwood and E. J. McCluskey, "High-Level synthesis for orthogonal scan," *in Proc. of VTS*, pp.370–375, 1997.

[3] R. B. Norwood and E. J. McCluskey, "Orthogonal scan: low overhead scan for data paths," *in Proc. of ITC*, pp.659–668, 1996.

[4] 和田弘樹，増澤利光，K. K. Saluja，藤原秀雄，"完全故障検出効率を保証するデータパスの非スキャンテスト容易化設計法，"電子情報通信学会誌 D-I，Vol.J82-D-I No.7，pp.843–851，1999.

[5] H. Date, T. Hosokawa, M. Miyazaki and M. Muraoka, "A non-scan DFT method for RTL circuits of irregular data path," *Digest of papers, WRTLT*, pp32–37, 2002.

[6] L. Lingappan, S. Ravi, N. K. Jha, "Test generation for non-separable RTL controller-datapath circuits using a satisfiability based approach,"*in Proc. of ICCD*, pp187–193, 2003.

[7] I. Ghosh, M. Fujita, "Automatic test pattern generation for functional RTL circuits using assignment decision diagrams," *in Proc. of DAC*, pp43–48, 2000.

[8] L. Zhang, I. Ghosh and M. Hisao, "Efficient sequential ATPG for functional RTL circuits,"*in Proc. of ITC*, pp290–298, 2003.

[9] M. Nourani and C. A. Papachristou, "Structural fault testing of embedded cores using pipelining," *JETTA vol.15*, pp.129–144, 1999.

[10] M. R. Garey and D. S. Johnson, *Computer and intractability*, W. H. Freeman and Company, 1979.

[11] F. Corno, M. S. Reorda and G. Squillero, "RT-level ITC'99 benchmarks and first ATPG results," *IEEE Design & Test of Computers*, pp. 44-53, 2000.