

不均一ビット幅データパスに対するテスト容易化設計法

山口 賢一・原田 和幸*

A Design-for-Testability method for RTL datapath with various bit width

Ken'ichi Yamaguchi and Kazuyuki Harada*

近年大規模論理回路のテスト生成を容易にするため様々なテスト容易化設計法が提案されている。従来のテスト容易化設計手法では、均一なビット幅を持つデータパスを対象としていた。そこで本稿で提案する手法は、その対象範囲を従来のデータパスから不均一なビット幅を持つデータパスへと拡張し、更に外部入出力端子数の抑制を目指す。

1. 序論

近年の情報技術の普及に伴い、デジタルシステムの信頼性の向上は重要な課題となりつつある。集積回路に関しては、出荷テストを行うことによってその信頼性を確保している。しかしながら出荷テストは、今日の集積化技術の向上により、次第に困難となりつつある。そこで今日ではテスト容易化設計（以下、DFT：Design For Testability）と呼ばれる、テスト時の負担を抑制する設計手法が考案されている。

現在主流なDFTであるスキャン設計は、回路規模の大規模化に伴い、テストパターンの計算時間が増大するという問題点がある。スキャン設計の問題点を解消するため、DFHT（Design For Hierarchical Testability）¹⁾と呼ばれるDFT手法が提案された。DFHTは回路の構成要素である各モジュールに対し、そのモジュールを外部からテストするためのデータの転送経路を構築する。DFHTではDFTのためのアルゴリズムは複雑になるが、一定の回路面積を持つモジュール単位でテストパターンを求めればよいので、回路規模が増大してもテストパターンの計算時間は増大しないという利点がある。しかしながらDFHTは回路の持つビット幅に対する制約が厳しく、全信号線が同じビット幅を持つ回路を仮定している。従ってDFHTは不均一なビット幅を持つ回路に直接適用することはできない。

不均一なビット幅を持つ回路に対応したDFT手法として文献²⁾や文献³⁾のような手法が提案されているが、前者はDFHTではなくスキャン設計に基づく手法であるため、テストパターンの計算時間が増大する。また後者はDFHT

に基づく手法であるがアルゴリズムが論文で明確に定式化されていないため、手法の有効性を調査することができない。また入出力端子数が大幅に増大するという問題点がある。

以上のように、不均一なビット幅を持つ回路を対象としたDFTには未だ有効な手法が存在しないというのが実情である。そこで本稿では、従来のDFHTにおけるビット幅の制約をなくし、更にDFTによって増加する外部入出力の抑制を試みる。

2. DFHT手法

Design For Hierarchical Testability（以下、DFHT）¹⁾は、Ghoshらによって提案されたDFTである。現在ではDFHTを改良した手法として文献⁴⁾、文献⁵⁾に挙げる手法などが提案されている。DFHTはレジスタ転送レベル（以下、RTL：Register Transfer Level）の段階における回路記述を対象としている。RTLはシステムの設計抽象度の1つであり、動作レベルよりも下位で、ゲートレベルよりも上位の抽象度に位置する。RTLの詳細は3.1節で説明する。DFHTではまず、RTL内に存在する各モジュール（回路要素）に外部入力からテストパターンを入力し、テストパターンに対する応答を外部出力で観測できるようなデータの転送経路（以下、経路）を構築する。次に各モジュールのテストパターンをテスト生成アルゴリズムによって求め、構築しておいた経路を利用してテストパターンの入力、及び応答の観測を実現する。RTLにおける各モジュールは一定の面積を持つ小規模な回路であるため、テストパターンは容易に求めることができる。

*本校専攻科電子情報工学専攻学生

3. 定義

3.1 レジスタ転送レベル回路 (RTL)

DFHTの対象となるRTLは、データに対して演算を行うデータパス部と、そのデータパス部を制御するコントローラ部から成る。本節では本稿で提案する手法が対象とする不均一ビット幅データパス（以下、VDP: Variouse bit width Data path）を数学的に定義する。

データパスは回路要素と呼ばれる機能単位のモジュールを信号線を用いて接続することによって構成する。RTLの回路要素は大きく分けて、外部入出力、組合せ回路要素、順序回路要素、頂点の4種類がある。各回路要素は種類ごとに固有の数の入出力端子を持つ。また信号線は、ある回路要素の出力端子と、ある回路要素の入力端子を接続する。データパス内部の全回路要素の集合を V とする。このとき $V=Vpio \cup Vcmb \cup Vseq \cup Vpoint$ とする。

3.2 外部入出力集合 $Vpio$

外部入出力には外部入力と外部出力の2種類がある。外部入力は1個の出力端子のみを持つ。外部出力は1個の入力端子のみを持つ。

ここで、

- Vpi : 外部入力の集合.
- Vpo : 外部出力の集合.

としたとき、 $Vpio = Vpi \cup Vpo$ である。

3.3 組合せ回路要素集合 $Vcmb$

組合せ回路要素には、マルチプレクサ（以下、MUX: MUltipleXer）、演算器の2種類がある。組合せ回路要素はいずれも2個の入力端子と1個の出力端子を持つ。ここで、

- $Vmux$: MUXの集合.
 - Vn_mux : 通常のMUXの集合.
 - $Vtdic_mux$: TDIC用のMUXの集合.
 - $Vtdro_mux$: TDRO用のMUXの集合.
- Vop : 演算器の集合.

とする。 $Vtdic_mux$ と $Vtdro_mux$ は後節にて説明する。このとき $Vmux = Vn_mux \cup Vtdic_mux \cup Vtdro_mux$ であり、また $Vcmb = Vmux \cup Vop$ である。

3.4 順序回路要素集合 $Vseq$

順序回路要素には、ホールド機能無しレジスタとホールド機能付きレジスタの2種類がある。順序回路要素はいずれも1個の入力端子と1個の出力端子を持つ。ホールド機能無しレジスタに入力されたデータは、1クロックだけ保持し、次のクロックで出力する。ホールド機能付きレ

ジスタはホールド機能により、以前のデータを保持することができる。ここで、

- Vn_reg : ホールド機能無しレジスタの集合.
 - Vh_reg : ホールド機能付きレジスタの集合.
- としたとき、 $Vseq = Vn_reg \cup Vh_reg$ である。

3.5 頂点集合 $Vpoint$

頂点には、セパレート点、バインド点、ファンアウト点、ターミナル点がある。セパレート点は1個の入力端子と2個の出力端子を持ち、入力をビット単位で2つの出力に分割する。従って入力ビット幅は、2つの出力ビット幅の和となる。バインド点は2個の入力端子と1個の出力端子を持ち、2つの入力をビット単位で出力に統合する。従って出力ビット幅は、2つの入力ビット幅の和となる。ファンアウト点は1個の入力端子と2個の出力端子を持ち、信号線の分岐を意味する。すなわち、入力と同じデータを2つの出力端子に出力する。従って入力ビット幅とそれぞれの出力ビット幅は同じ値となる。ターミナル点は1個の入力端子と0個の出力端子を持ち、信号線の終端を意味する。ここで、

- $Vseparate$: セパレート点の集合
- $Vbind$: バインド点の集合
- $Vfanout$: ファンアウト点の集合
- $Vterminal$: ターミナル点の集合

としたとき、 $Vpoint = Vseparate \cup Vbind \cup Vfanout \cup Vterminal$ である。

4. ビット幅圧縮・ビット幅拡張

本節では提案するテスト容易化設計 (DFT) で使用する、ビット幅を圧縮・拡張するアイデアについて説明する。

4.1 時分割応答観測 (先行手法)

時分割応答観測 (以下、TDRO: Time Division Response Observation)⁶⁾は、外部出力よりも大きなビット幅を持つ回路要素の出力端子からの応答データを、外部出力で時分割に観測することを目的としたDFTアイデアである。TDROは文献⁷⁾で提案されたビット幅調整回路 (bit match circuit) に基礎を置いている。TDROの例を次ページFig. 1に示す。Fig. 1では入力をセパレート点 (Sと書かれた円) を用いて分割し、それぞれ2段に挿入したMUX (MUXと書かれた台形) に入力している。テスト時にはMUXの入力を適宜選択することにより、データを時分割に観測することができる。また本稿ではTDROを実現する際に挿入されるセパレート点、MUXをまとめてTDRO-Unitと呼ぶ。

4.2 時分割入力制御（提案手法）

提案手法である時分割入力制御（以下、TDIC：Time Division Input Control）は、TDROの時分割でビット幅調整を行うという考え方を入力に応用したDFT アイデアである。TDICを用いることにより、外部入力よりも大きなビット幅を持つ回路要素の入力端子にテストパターンを入力することができる。TDICの例をFig. 2に示す。Fig. 2では入力を2つのファンアウト点（Fと書かれた円）で分岐し、2つのホールド機能無しレジスタ（REGと書かれた長方形）に分割して保持し、最後にそれらをバインド点（Bと書かれた円）により統合する。テスト時にはテストパターンを上位6ビットから6ビットずつ順番に1クロックごとに入力すれば、元のテストパターンを再現できる。TDIC-Unitと同様に、TDICを実現するために挿入されるTDIC、ファンアウト点、バインド点をまとめてTDIC-Unitと呼ぶ。

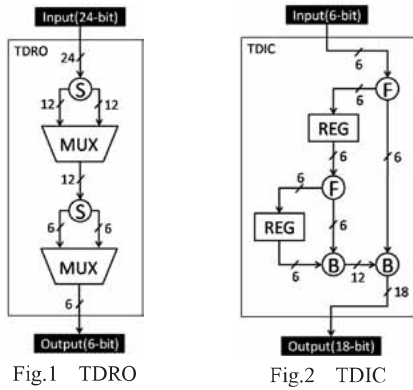


Fig.1 TDRO

Fig.2 TDIC

5. データパスの定義の拡張

本節では提案するDFT アルゴリズムを記述するため、データパスの定義を拡張する。

5.1 回路要素から入出力端子への写像

VDP内の全回路要素の入力端子全体の集合を IN 、出力端子全体の集合を OUT とする。このとき回路要素ごとの入出力数に応じて、 V と IN 、もしくは V と OUT との対応を写像として定義する。

5.1.1 1入力回路要素

VDP内の1入力回路要素を $V_1_input = V_{po} \cup V_{seq} \cup V_{separate} \cup V_{fanout} \cup V_{terminal}$ とする。このとき、 $v \in V_1_input$ とその入力端子 $a \in IN$ との対応を、写像 $Input$ として下式のように定義する。

$$Input = \{(v, a) \in V_1_input \times IN \mid a \text{ は } v \text{ の入力端子}\}$$

5.1.2 1出力回路要素

VDP内の1出力回路要素を $V_1_output = V_{pi} \cup V_{cmb} \cup V_{seq} \cup$

V_{bind} とする。このとき、 $v \in V_1_output$ とその出力端子 $c \in OUT$ との対応を、写像 $Output$ として下式のように定義する。
 $Output = \{(v, c) \in V_1_output \times OUT \mid c \text{ は } v \text{ の出力端子}\}$

5.1.3 2入力回路要素

VDP内の2入力回路要素を $V_2_input = V_{cmb} \cup V_{bind}$ とする。また、 $v \in V_2_input$ の2つの入力端子をそれぞれ $a, b (\in IN)$ とする。このとき、 v と a, b のペア (a, b) との対応を、写像 $InputPair$ として下式のように定義する。

$$InputPair = \{(v, (a, b)) \mid a, b \text{ は } v \text{ の入力端子かつ } a \neq b\}$$

5.1.4 2出力回路要素

VDP内の2出力回路要素を $V_2_output = V_{fanout} \cup V_{separate}$ とする。また、 $v \in V_2_output$ の2つの出力端子をそれぞれ $c, d (\in OUT)$ とする。このとき、 v と c, d のペア (c, d) との対応を、写像 $OutputPair$ として下式のように定義する。

$$OutputPair = \{(v, (c, d)) \mid c, d \text{ は } v \text{ の出力端子かつ } c \neq d\}$$

5.2 入出力端子から回路要素への写像

入出力端子と回路要素の対応を写像として定義する。入力端子と出力端子の集合 $IN \cup OUT$ からその回路要素 V への写像 $Module$ を下式のように定義する。

$$Module = \{(x, v) \mid x \text{ は } v \text{ の入力端子または } v \text{ の出力端子}\}$$

5.3 信号線

回路要素間を接続する信号線は、出力端子 $out \in OUT$ と入力端子 $in \in IN$ を接続する。ここでは信号線を表現する写像として $Next$ と $Prev$ を定める。VDP内の入力端子を $x \in IN$ 、出力端子を $y \in OUT$ としたとき、 $Next$ 及び $Prev$ を下式のように定義する。

$$Next = \{(y, x) \mid y \text{ から } x \text{ への信号線がVDPに存在する}\}$$

$$Prev = \{(y, x) \mid y \text{ から } x \text{ への信号線がVDPに存在する}\}$$

5.4 ビット幅

VDP内の信号線はそれぞれ固有のビット幅を持つ。ここでは入出力端子 $IN \cup OUT$ からそのビット幅への写像 u を定義する。VDP内の入出力端子を $x (\in IN \cup OUT)$ 、その入出力端子が接続している信号線のビット幅を $k (\in N)$ とする。このとき x と k の対応を意味する写像 u を下式のように定義する。

$$u = \{(x, k) \in (IN \cup OUT) \times N \mid x \text{ は } k \text{ のビット幅を持つ}\}$$

6. 単一制御可検査不均一ビット幅データパス

本稿でDFTの対象とするのはデータパスである。コン

トローラではなくデータバスを対象とするのは、一般的にコントローラよりもデータバスの方が面積が大きいため、DFTの影響がより表面化するからである。データバスのテストの容易性を評価するために、そのデータバスが満たすべき様々な条件が定義されている。単一制御可検査性⁹⁾は、テスト容易なデータバスが満たすべき条件である。

しかしながら単一制御可検査性はビット幅を考慮しないデータバスに対して定義されたものであるため、不均一ビット幅データバス (VDP) のテスト容易性を評価する際には定義を拡張する必要がある。そこで、単一制御可検査データバスを拡張し、ビット幅を考慮した単一制御可検査性を新たに定義する。またそれを満たすVDPを単一制御可検査不均一ビット幅データバス (以下、SC VDP) とする。

データバスがSC VDPであるためには、データバスの各組合せ回路要素 $M \in V$ が以下の3条件を満たす必要がある。

- 外部入力から M の両入力端子の各ビットに、任意の値を伝搬可能
- M の出力端子の各ビットから外部出力に任意の値を伝搬可能
- 上記2条件の達成にTDIC, TDROの使用を許可

次節以降では、与えられたデータバスをSC VDPに変換するための提案アルゴリズムを説明する。

7. DFTの前処理

ここでは前処理として、与えられたVDPに対し、TDIC-Unit及びTDRO-Unitの付加を行う。

7.1 TDIC-Unitの付加

VDPの全外部入力 $pi \in V_{PI}$ のビット幅の和を S_{PI} とする。すなわち、

$$S_{PI} = \sum_{pi \in V_{PI}} u(output(pi))$$

また、各組合せ回路要素 $v \in V_{cmb}$ の入力端子 a, b (ここで $(a, b) = InputPair(v)$)のビット幅の和 $u(a) + u(b)$ のうち、データバス中最大の値を M_{IN} とする。すなわち、

$$M_{IN} = \max(u(a) + u(b))$$

もし対象としているデータバスが $S_{PI} < M_{IN}$ ならば、ある回路要素の入力端子にテストパターンを供給するための外部入力不足しているため、データバスはSC VDPではない。従って、TDIC-Unitを付加して外部入力のビッ

ト幅を拡張する。なお $S_{PI} \geq M_{IN}$ のときは、TDIC-Unitの付加は必要ない。

付加するTDIC-Unitは S_{PI} の入力ビット幅と M_{IN} の出力ビット幅を持つ。

7.2 TDRO-Unitの付加

TDIC-Unitと同様にして、TDRO-Unitも予め付加しておく。データバスの全外部出力 $po \in V_{PO}$ のビット幅の和 S_{PO} は以下ようになる。

$$S_{PO} = \sum_{po \in V_{PO}} u(input(po))$$

各組合せ回路要素 $v \in V_{cmb}$ の出力端子 $c = Output(v)$ のビット幅のうち、データバス中最大の値を M_{OUT} とする。すなわち、

$$M_{OUT} = \max(u(c))$$

もし対象としているデータバスが $S_{PO} < M_{OUT}$ ならば、ある回路要素の出力端子から応答を観測するための外部出力が不足しているため、データバスはSC VDPではない。従って、出力端子の信号をTDRO-Unitを用いて圧縮する必要がある。なお、 $S_{PO} \geq M_{OUT}$ のときは、TDRO-Unitの付加は必要ない。

付加するTDRO-Unitは M_{OUT} の入力ビット幅と S_{PO} の出力ビット幅を持つ。

8. テストパターン供給の実現

VDPをSC VDPに変換するためには、各組合せ回路要素 M にテストパターンが供給できる必要がある。 M へのテストパターン供給は、 M の入力端子とTDIC-Unitの出力をMUXを用いて接続することによって実現できる。MUXを介して入力する理由は、データバスの通常動作時の挙動に影響を及ぼさないためである。このMUXをTDIC MUXと呼ぶ。

ところで付加するTDIC MUXは、既に付加されているTDIC MUXと共有することができる。すなわち M のテストパターン供給の実現のために挿入されるTDIC MUXを可能な限り共有することで、DFTの面積増加を抑制することができる。そこで、 M の入力にTDIC MUXを挿入する際に、TDIC MUXの探索を行う。

8.1 TDIC MUX探索アルゴリズム

TDIC MUXは、 M の入力端子を始点とし、信号の流れと逆の方向 (入力端子から出力端子) に探索する。ここで M の2つの入力端子をそれぞれ a, b ($(a, b) = InputPair(M)$) とする。また、 $u(a) \geq u(b)$ である。探索はまず a を始点に探索し、その後 b を始点に探索する。

8.1.1 入力端子 a を始点とした探索

a と信号線を介した回路要素を v とする. すなわち $v = Module(Prev(a))$ である. v の種類によって以下のような操作を施す.

- $v \in Vpi \cup Vtdic_mux$ の場合: v がPI, もしくはTDIC MUX の場合は, M のテストパターン供給用に新たにTDIC MUX を付加する必要がない. 従ってTDIC MUX を付加せずに探索を終了する.
- $v \in Vn_mux$ の場合: v が通常のMUX の場合は, そのMUX から探索を再開する. しかし b が同じ経路を使用してしまうと, SC VDP を満たさなくなる. そこで a が使用した経路を区別するために, ここで v にチェックを付けておく. その後 $(x, y) = InputPair(v)$ としたとき, $max(u(x), u(y))$ のビット幅を持つ入力端子を新たに a とする. そして a と信号線を介して接続する回路要素を新たに $v = Module(Prev(a))$ として探索を続行する.
- $v \in Vop \cup Vbind$ の場合: v が演算器, もしくはバインド点の場合は, TDIC MUX を付加して探索終了となる. v の出力端子 $Output(v) = Prev(a)$ とその時点での a の間に, $u(a)$ のビット幅を持つTDIC MUX を挿入する.
- $v \in Vseparate \cup Vseq$ の場合: v がセパレート点, もしくは順序回路要素の場合は, $Input(v)$ を新たに a とし, $Module(Prev(a))$ を新たに v として探索を続行する.
- $v \in Vfanout$ の場合: v がファンアウト点の場合は, MUX と同様に v にチェックをつけた後, $Input(v)$ を新たに a とし, $Module(Prev(a))$ を新たに v として探索を再開する.

8.1.2 入力端子 b を始点とした探索

入力端子 b の探索は v が通常のMUXの場合とファンアウト点の場合以外は, 入力端子 a の探索と同様に行う.

- $v \in Vn_mux$ の場合: v が通常のMUX の場合は, v にチェックが付いていればTDIC MUX を付加して探索を終了する. チェックが付いていなければ, a の場合と同様に探索を再開する.
- $v \in Vfanout$ の場合: v がファンアウト点の場合は, v にチェックが付いていればTDIC MUX を付加して探索を終了する. 付いていなければ a の場合と同様に探索を再開する.

9. 応答観測の実現

前節で M にテストパターンを供給するための経路を実現した. しかしVDP をSC VDP に変換するためには更に, M の応答観測を実現する必要がある. 提案手法では, 応答観測を実現するために, M の出力端子のビット幅をTDRO-Unitで調整する. TDRO-Unitの入力と M の出力端子を結ぶMUXをTDRO MUXと呼ぶ. TDRO MUXの付加の場合もTDIC MUXのときと同様に, 面積抑制のための探索を行う.

9.1 TDRO MUX 探索アルゴリズム

TDRO MUX は, M の出力端子を始点とし, 信号の流れと同じ方向に探索する. M の出力端子を $c = Output(M)$ とする.

9.1.1 出力端子 c からの探索

c と信号線を介して接続している回路要素を $v = Module(Next(c))$ とする.

- $v \in Vpo \cup Vtdro_mux$ の場合: v がPO, もしくはTDRO MUX の場合は, M の応答観測用に新たにTDRO MUX を付加する必要がない. 従ってTDRO MUX を付加せずに探索を終了する.
- $v \in Vn_mux \cup Vtdic_mux \cup Vseq \cup Vbind$ の場合: v が通常のMUX, TDIC MUX, 順序回路, バインド点の場合は, $Output(v)$ を新たに c とし, $Module(Next(c))$ を新たに v として探索を再開する.
- $v \in Vop$ の場合: v が演算器の場合は, TDRO MUX を付加して探索終了となる. その時点での c と $Next(c)$ の間に $u(c)$ のビット幅を持つTDRO MUX を挿入する.
- $v \in Vseparate \cup Vterminal$ の場合: v がセパレート点かターミナル点の場合は, TDRO MUX を付加して探索終了となる. その時点での c と $Next(c)$ の間に $u(c)$ のビット幅を持つTDRO MUX を挿入する.
- $v \in Vfanout$ の場合: ファンアウトの場合は, 出力端子のペアを $(x, y) = OutputPair(v)$ としたとき, x か y のどちらかを任意に選択して新たに c とし, $Module(Next(c))$ を新たに v として探索を続行する.

10. 提案するDFT手法のまとめ

8. 節と9. 節の操作を経ることにより, 各組合せ回路

要素 $M \in V$ はTDIC-Unitを介してテストパターンの供給が可能となり、かつTDRO-Unitを介して応答観測が実現できる。従って、以上でVDPのSC VDPへの変換は達成できた。

SC VDPに変換したデータパスは、各 $M \in V$ に対するテストパターンが存在すれば信頼性の高いテストを実現できる。そこで次節では提案するDFT手法を実際のVDPに適用して故障シミュレーションを行うことにより、その有効性と欠点を調査する。

11. 提案手法の適用例

提案手法の有効性を評価するために、不均一なビット幅を持つベンチマークデータパスに対して提案手法を適用した。提案手法を適用したデータパスはLWF (3rd Lattice Wave Filter)とWRTである。

いずれのデータパスもVHDLで記述し、Synopsys社の論理合成ツールDesign Analyzerを用いて回路面積を算出した。ここで、DFT適用前から適用後に増えた面積の比率を面積増加率とする。また、Synopsys社の故障シミュレータTetra Maxを用いて、データパス内の各回路要素の全単一縮退故障を検出可能なテストパターン数を算出した。そして得られたテストパターン数とTDIC及びTDROに要する時間を掛け合わせた結果をテスト実行時間とする。

LWF, WRTに対する提案手法の適用前と適用後の面積、面積増加率、外部入出力数、テスト実行時間をTable 1, Table 2に示す。尚、適用前のテスト実行時間は、TDIC及びTDROを使用しないときの各回路要素のテストパターン数の和を示している。

Table 1 提案手法の適用前と適用後 (LWF, WRT)

	回路面積		面積 増加率	外部入出力数	
	適用前	適用後		適用前	適用後
LWF	1121	1373	22.5%	44	44
WRT	580	696	20.0%	18	18

Table 1によると、DFT適用によるLWFとWRTの外部入出力数の増加は0である。これは外部入出力の付加が困難な状況における、提案手法の有効性を示唆している。

Table 2 提案手法の適用前と適用後 (LWF, WRT)

	テスト実行時間		テスト実行時間 増加率
	適用前	適用後	
LWF	108	292	170%
WRT	76	180	237%

しかしながらテスト実行時間はいずれも2倍~3倍に増加している。これはTDICとTDROの時分割回数に依存し

ている。従って提案手法は、LWFやWRTのように外部入出力のビット幅の和が回路要素のビット幅よりも極端に小さいデータパスに適用する際には、テスト実行時間が大幅に増大するという欠点がある。改善案としては、各回路要素間のテストパターンの共通部分を利用した連続クロックでのテストなどが考えられる。

12. 結論

本稿では不均一なビット幅を持つデータパスに対するテスト容易化設計 (DFT) 手法を提案した。

提案したDFT手法では外部入出力の付加を抑制するため、TDRO及びTDICを用いた。どちらも外部入出力と回路要素のビット幅の差を埋めるために使用する。

提案するテスト容易化設計アルゴリズムでは、付加するマルチプレクサ (MUX) を抑制するために、MUXの探索を行う。新たにMUXを付加する場所へとデータが到達可能な範囲でMUXが既に付加されていたならば、新たにMUXを付加する必要はないからである。

提案手法をベンチマークデータパスに適用したところ、DFTによる外部入出力の増加の抑制を達成した。

参考文献

- 1) I.Ghosh, A.Raghunath and N.K.Jha: "Design for hierarchical testability of RTL circuits obtained by behavioral synthesis," IEEE ICCD '95, pp.173-179
- 2) Y.Murata, S.Ohtake and H.Fujiwara: "A method of DFT for data paths using bit-match function," 情報処理学会研究報告.SLDM, Vol.2004, No.122 (20041201), pp.137-142
- 3) H.Date, T.Hosokawa, M.Miyazaki, and M.Muraoka: "A Nonscan DFT Method for RTL Data Path Circuits with Various Bit Width," IEEE Proceedings of The 3rd Workshop on RTL and High Level Testing, pp.32-37
- 4) T.Masuzawa, M.Idutsu, H.Wada and H.Fujiwara: "Single control testability of RTL data paths for BIST," Ninth Asian Test Symposium (ATS' 00), p.210
- 5) K.Yamaguchi, M.Inoue and H.Fujiwara: "Hierarchical BIST: Test-per-clock BIST with low overhead," Digest of Papers, IEEE 3rd Workshop on WRTL' 02, pp.42-47
- 6) 青山瑠美, 山口賢一: "不均一-bit幅データパスに対する組込み自己テスト法," FIT2005 (情報科学技術フォーラム)
- 7) M.Nourani and C.A.Papachristou: "Structural Fault Testing of Embedded Cores Using Pipelining," JETTA vol.15, pp.129-144