

スルー演算を用いた非スキャン方式による
データパスのテスト容易化設計

高畠 勝之[†] 井上美智子^{††} 増澤 利光^{††} 藤原 秀雄^{††}

Non-Scan Design for Testable Data Paths Using Thru Operation

Katsuyuki TAKABATAKE[†], Michiko INOUE^{††}, Toshimitsu MASUZAWA^{††},
and Hideo FUJIWARA^{††}

あらまし 本論文では、レジスタ転送レベルのデータパスを対象に、スルー演算を用いた非スキャン方式によるテスト容易化設計法を提案する。まずデータパスの弱可検査性を定義し、小さいハードウェアオーバーヘッドで与えられたデータパスを弱可検査にする問題について考察する。この問題のNP困難性を示し、この問題に対する発見的方法を提案する。次にレジスタを制御、観測するのに必要なクロック数を与える尺度を提案し、テスト生成時間との相関を考察する。最後に、実験結果により本手法および提案した尺度の有効性を示す。

キーワード テスト容易化設計, 非スキャン方式, RT レベルデータパス, スルー演算

1. まえがき

近年のデジタルLSIの大規模複雑化に伴い、回路のテストはますます困難で費用のかかる問題となっている。この問題に対する一つの有効なアプローチとして、与えられた回路をテスト容易にするために設計変更を施すテスト容易化設計がある。

レジスタ転送 (Register Transfer: RT) レベルのデータパスに対するテスト容易化設計としては、レジスタを外入出力より自由に制御、観測できるように設計するスキャン設計方式に基づく手法[1]~[3]が主に提案されている。スキャン方式ではテスト生成の複雑度が飛躍的に改善されるが、一般に付加ハードウェアのオーバーヘッドが大きいという問題点を抱えている。しかしながら最大の問題点は、通常の動作モード以外にレジスタを直列のシフトレジスタとして動作させるスキャン操作モードを併用するため、回路の通常の動作速度 (at the operational speed: at-speed) でのテスト実行が不可能であるということである。文献[4]では、縮退故障のテストパターン集合を at-speed で印

加した場合に、それより遅い速度で同じテストパターン集合を印加したときよりも多くの故障を検出したと報告している。また、スキャン方式ではテスト実行の際、スキャン操作に伴う多くの時間を要する可能性がある。最近、RT レベルデータパスに対して配線とマルチプレクサを追加することにより外部入出力までの経路を確保し、信号線の可制御性、可観測性を高める非スキャン方式によるテスト容易化設計法[5]が提案された。この手法では、同程度の故障検出率を得るスキャン設計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。

本論文では、RT レベルデータパスに対して、演算モジュールのスルー演算機能を用いた非スキャン方式による一つのテスト容易化設計法を提案する。一般に演算モジュールの出力値はすべての入力値に依存し、信号線の制御、観測は制御すべきこれらの入力信号線数に依存して困難となる場合が多い。従って、演算モジュールのスルー演算機能、すなわち少なくとも一つの入力値を他の入力値に依存せず出力する機能を用いることにより信号線の制御、観測が容易になると考えられる。図1(a)は算術演算モジュールALUの実現例を示したものである。この例では、制御信号 s_0, s_1 およびキャリー入力 c_{in} をすべて0とすることにより入力 X の値を入力 Y の値に依存せず出力することができる。図1(b)のような加算モジュールの例では、AND

[†] NTT 北陸マルチメディア推進本部, 金沢市
NTT Hokuriku Multimedia Service Promotion Headquarters,
Kanazawa-shi, 920 Japan

^{††} 奈良先端科学技術大学院大学 情報科学研究科, 生駒市
Graduate School of Information Science, Nara Institute of Science
& Technology, Ikoma-shi, 630-01 Japan

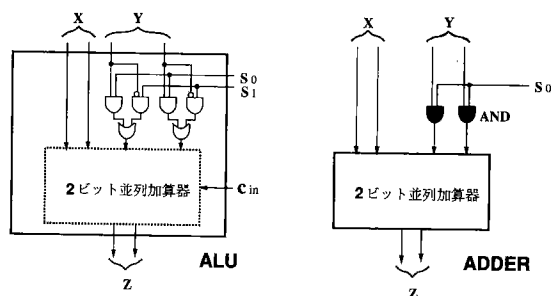


図1 スルー演算モジュールの設計例: (a) ALUにおけるスルー演算機能の実現, (b) 加算モジュールにおけるスルー演算機能の実現

Fig.1 An example of thru module: (a) realization of thru on ALU, (b) realization of thru on adder.

ゲートをビット幅分だけ付加することによってスルー演算機能を実現できる。このように、演算モジュールによってはもともとスルー演算機能を有するものもあり、そうでない場合でもわずかなハードウェアオーバーヘッドでスルー演算機能をもたせることが可能である。制御部からデータバスへの制御信号は文献[6]の手法により自由に制御可能であると仮定する。

以下では、まずデータバスに対して弱可検査性を定義し、小さいハードウェアオーバーヘッドで与えられたデータバスを弱可検査にする問題について考察する。この問題のNP困難性を示し、この問題を解く発見的方法を提案する。次に新しい可検査性尺度を提案する。この尺度はレジスタを制御、観測するために必要なクロック数を示すものであり、この尺度とテスト生成時間との相関関係について考察する。最後に、本手法および提案した尺度の有効性を実験結果により示す。

2. スルー演算を用いたテスト容易化設計

2.1 ハードウェアモデル

データバスは外部入力、外部出力、レジスタ、マルチプレクサ、演算モジュールからなるハードウェア要素とこれらを接続する接続信号線から構成されるものとする。

演算モジュール M に対し、 M が実現する関数の集合を \mathcal{F}_M と表す。演算モジュール M が以下を満たすとき、 M をスルー演算モジュールと呼ぶ。

$$\exists f_{th} \in \mathcal{F}_M, \exists i, f_{th}(X_1, \dots, X_i, \dots, X_n) = X_i$$

また、上式の入力 X_i をスルー可能入力と言う。

2.2 諸定義

各ハードウェア要素について弱可制御性、弱可観測

性およびデータバスについて弱可検査性をそれぞれ定義する。

演算モジュール M の入力集合を \mathcal{IN}_M と表し、入力 X がスルー可能入力であれば \hat{X} と表記する。ハードウェア要素 H_1 の出力とハードウェア要素 H_2 の入力 X 間に接続信号線があることを、 $H_1 \rightarrow X$ と表記する。但し、特に混乱のない場合、 $H_1 \rightarrow H_2$ と表記することもある。

ハードウェア要素 H に対して、その出力に何らかの値を設定できるとき、その性質を弱可制御性と言い、 H の出力に外部入力から（直接または他のハードウェア要素を介して）値を設定することを H を弱制御と言う。外部入力は弱可制御である。レジスタについてはその入力が弱可制御であれば弱可制御となり、マルチプレクサについてはある入力が弱可制御であれば弱可制御となる。演算モジュールについてはすべての入力が弱可制御、あるいは、あるスルー可能入力が弱可制御ならば弱可制御となる。

[定義 1] 弱可制御性

以下の条件を満たすハードウェア要素の極小集合を \mathcal{H}_{wc} とする。

- (1) 任意の外部入力 I に対し、 $I \in \mathcal{H}_{wc}$ 。
- (2) 任意のレジスタまたはマルチプレクサ H に対し、 $\exists H_{wc} \in \mathcal{H}_{wc} [H_{wc} \rightarrow H] \Rightarrow H \in \mathcal{H}_{wc}$ 。
- (3) 任意の演算モジュール M に対し、

$$\forall X \in \mathcal{IN}_M [\exists H_{wc} \in \mathcal{H}_{wc} [H_{wc} \rightarrow X]] \vee \exists \hat{X} \in \mathcal{IN}_M [\exists H_{wc} \in \mathcal{H}_{wc} [H_{wc} \rightarrow \hat{X}]] \Rightarrow M \in \mathcal{H}_{wc}$$

このとき、 $H \in \mathcal{H}_{wc}$ なるハードウェア要素 H は弱可制御である。

演算モジュール M の非スルー可能入力 X 以外の入力をすべて弱制御することをここでは弱い意味で X の値を M の出力に伝搬と言う。また、レジスタの入力、マルチプレクサの各入力、演算モジュールのスルー可能入力の値はその出力に伝搬できる。このときハードウェア要素 H に対して、その出力値を外部出力まで（弱い意味で）伝搬できるとき、その性質を弱可観測性と言い、 H の出力値を外部出力まで（弱い意味で）伝搬することを H を弱可観測と言う。

[定義 2] 弱可観測性

以下の条件を満たすハードウェア要素の極小集合を \mathcal{H}_{wo} とする。

- (1) 任意の外部出力 U に対し、 $U \in \mathcal{H}_{wo}$ 。

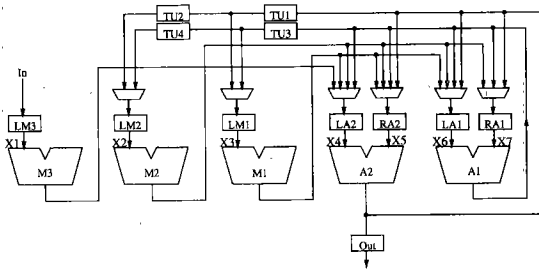


図2 4th Order IIR Cascade Filter の RT レベルデータパス (注1)

Fig.2 RT-level data path of 4th order IIR cascade filter.

(2) \mathcal{M} を演算モジュールの集合とすると、任意のハードウェア要素 H に対し、

$$\begin{aligned} & \exists H_{wo} \in \mathcal{H}_{wo} - \mathcal{M} [H \rightarrow H_{wo}] \vee \\ & \exists M_{wo} \in \mathcal{H}_{wo} \cap \mathcal{M} [\\ & \quad \exists X \in \mathcal{IN}_{M_{wo}} [H \rightarrow X \wedge \forall X' \in \mathcal{IN}_{M_{wo}} - \{X\} \\ & \quad \quad [H \rightarrow X' \vee H_{wc} \in \mathcal{H}_{wc} [H_{wc} \rightarrow X']]] \vee \\ & \quad \exists \hat{X} \in \mathcal{IN}_{M_{wo}} [H \rightarrow \hat{X}]] \\ & \Rightarrow H \in \mathcal{H}_{wo}. \end{aligned}$$

このとき、 $H \in \mathcal{H}_{wo}$ なるハードウェア要素 H は弱可観測である。

[定義 3] 弱可検査性

データパス中のすべてのレジスタが弱可制御、かつ、弱可観測であるとき、そのデータパスは弱可検査である。

データパス中のすべてのレジスタが弱可制御なら、明らかに、すべてのレジスタは弱可観測となり、すなわちそのデータパスは弱可検査である。

(例) 図2は4th Order IIR Cascade Filter のRTレベルデータパスである。この例では、外部入力 In, レジスタ LM3, 演算モジュール M3, レジスタ LA2 は弱可制御となるが、他のハードウェア要素は弱可制御でない。従って、このデータパスは弱可検査ではない。そこで、入力 X4 をスルー可能入力とすれば弱可検査となる。

3. スルー可能入力の最小化問題

弱可検査なるデータパスを得るために追加する最小個のスルー可能入力を求める問題について考察する。

3.1 NP 困難性

データパスの演算モジュールの入力集合に対し、ある入力集合をスルー可能入力にすると、そのデータパス

が弱可検査になるとき、その入力集合をスルー可能入力集合 (Thru Input Set: TIS) と呼ぶ。

[問題] 最小スルー可能入力集合 (Minimum Thru Input Set: MTIS) 問題

RT レベルデータパスが与えられたとき、最小のスルー可能入力集合を求める。

以下では、MTIS 問題が NP 困難であることを示す。そのため、NP 困難である有向グラフの最小フィードバック頂点集合 (MFVS) 問題を多項式時間で MTIS 問題に還元可能であることを示す。

有向グラフ $G = (V, E)$ に対し、ある頂点集合を G から削除すると、 G が有向閉路を含まなくなるとき、その頂点集合をフィードバック頂点集合 (Feedback Vertex Set: FVS) と呼ぶ。

[問題] 最小フィードバック頂点集合 (Minimum Feedback Vertex Set: MFVS) 問題

有向グラフ $G = (V, E)$ が与えられたとき、最小のフィードバック頂点集合を求める。

MFVS 問題は各頂点の入次数を2以下に限定した2-MFVS問題も NP 困難 [8] である。以下では、各頂点の入次数を2以下に制限した有向グラフ $G = (V, E)$ を $G' = (V', E')$ に変換し、更に G' をデータパス DP' に変換する。

[$G = (V, E)$ から $G' = (V', E')$ への変換]

まず、有向グラフ $G = (V, E)$ に次の変換規則を適用する変換規則がなくなるまで繰り返し適用し、図3のような各頂点の入次数がちょうど2である有向グラフ $G' = (V', E')$ に変換する。

T1: 頂点 v が自己閉路をもつならば、 v を削除し、 v を返す。

T2: 頂点 v の入次数, 出次数のどちらかが0であれば、 v を削除する。

T3: 頂点 v の入次数が1であり、かつ、自己閉路をもたないならば、 $(u, v) \in E$ である各頂点 u から、 $(v, w) \in E$ であるすべての頂点 w に有向枝を加え、 v を削除する。

G' は上記の変換順序に依存せず一意に決まり、また、有向グラフ G' の MFVS 問題の解はもとの有向グラフ G の MFVS 問題の解と T1 で返された頂点の和集合であることが保証されている [9]。

(注1): 定数入力は演算モジュールの論理に含まれるものとし、ここでは省略する。

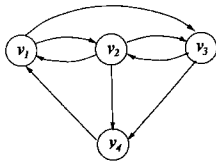


図3 有向グラフ G' の例
Fig.3 An example of directed graph G' .

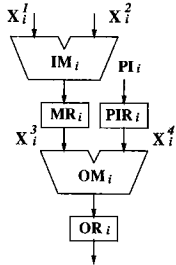


図4 有向グラフ G' の頂点 v_i に対応するハードウェア構成
Fig.4 Hardware structure corresponding to a vertex v_i of directed graph G' .

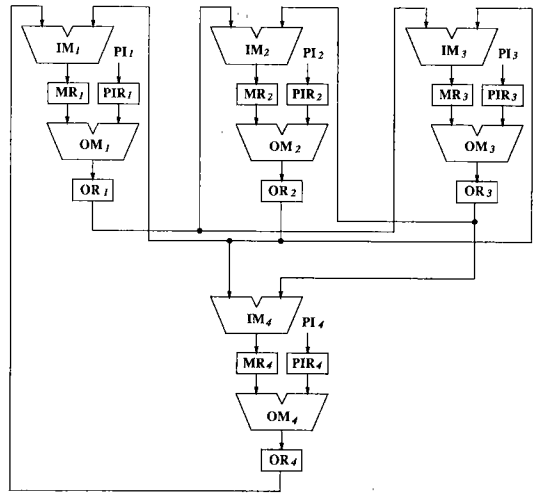


図5 図3に対応するデータパス
Fig.5 Data path corresponding to Fig.3.

[$G' = (V', E')$ から DP' への変換]

$V' = \{v_1, v_2, \dots, v_n\}$ とする。各頂点 v_i を以下のハードウェア要素と接続信号線に変換する。 v_i に対応して、2入力1出力演算モジュール IM_i, OM_i とレジスタ PIR_i, MR_i, OR_i および外部入力 PI_i からなるハードウェア要素を次のように接続する (図4参照)。但し、 X_i^1, X_i^2 および X_i^3, X_i^4 はそれぞれ IM_i, OM_i の二つの入力を表す。

- $IM_i \rightarrow MR_i, MR_i \rightarrow X_i^3$
- $PI_i \rightarrow PIR_i, PIR_i \rightarrow X_i^4$
- $OM_i \rightarrow OR_i$

次に各頂点 v_i に入射する2辺 $(v_{i1}, v_i), (v_{i2}, v_i) \in E'$ を接続信号線 $OR_{ip} \rightarrow X_i^p (p=1, 2)$ に変換する。図3のグラフ G' から得られるデータパスを図5に示す。

この変換は G' の大きさ $(|V'| + |E'|)$ の多項式時間で行える。

このように変換された G' と DP' に対し、以下の補題が成り立つ。

[補題1] $T' = \{T_1, T_2, \dots, T_m\}$ を DP' の TIS とする。 $F' = \{v_i \mid \exists k [X_i^k \in T']\}$ は G' の FVS である。

(証明) F' が G' の FVS でないとする。すると、 F'

の要素を削除しても G' に閉路 $(v_{f1}, v_{f2}, \dots, v_{fp}, v_{f1})$ が存在する。この閉路に対応した DP' の閉路 $OR_{f1} \rightarrow IM_{f2} \rightarrow MR_{f2} \rightarrow OM_{f2} \rightarrow OR_{f2} \rightarrow \dots \rightarrow OM_{fp} \rightarrow OR_{fp} \rightarrow IM_{f1} \rightarrow MR_{f1} \rightarrow OM_{f1} \rightarrow OR_{f1}$ はスルー演算モジュールを通らず、この閉路上の各 MR_i は非弱可制御である。これは T' が DP' の TIS であることに矛盾する。 □

[補題2] $F' = \{v_1, v_2, \dots, v_m\}$ を G' の FVS とする。 $T' = \{X_i^4 \mid v_i \in F'\}$ は DP' の TIS である。

(証明) T' が DP' の TIS でないとする。すると、 T' の要素をスルー可能入力としても DP' は弱可検査でない。このとき、非弱可制御なるレジスタが存在する。任意の PIR_i は弱可制御である。また、すべての OR_i が弱可制御であれば、すべての MR_i も弱可制御となるので、ある非弱可制御なるレジスタ OR_i が存在する。 OR_i が非弱可制御であれば、 OM_i の入力 X_i^4 はスルー可能入力ではなく、かつ、 MR_i は非弱可制御である。また、 MR_i が非弱可制御であれば、 T' の定義より、 IM_i はスルー演算モジュールでないので、ある非弱可制御なる OR_j が存在し、 $OR_j \rightarrow IM_i$ である。 DP' のハードウェア要素数は有限であるから DP' には閉路 $OR_a \rightarrow IM_b \rightarrow MR_b \rightarrow OM_b \rightarrow OR_b \rightarrow IM_c \rightarrow \dots \rightarrow OR_a$ が存在し、各 OM_i の入力 X_i^4 はスルー可能入力ではない。これは F' が G' の FVS であることに矛盾する。 □

[補題3] DP' の MTIS には、各 $i (1 \leq i \leq n)$ につ

いて、 $X_i^1, X_i^2, X_i^3, X_i^4$ のうちたかだか一つが属する。
 (証明) T を DP' の MTIS とし、 $X_i^p, X_i^q \in T$ ($p \neq q$) と仮定する。補題 1 より、 $F' = \{v_i \mid \exists k [X_i^k \in T]\}$ は G' の FVS であり、補題 2 より、 $T' = \{X_i^4 \mid v_i \in F'\} = \{X_i^4 \mid \exists k [X_i^k \in T]\}$ は DP' の TIS である。ここで、 $|T'| < |T|$ となるので、 T が DP' の MTIS であることに矛盾する。 □

[補題 4] T を DP' の MTIS とする。 $F = \{v_i \mid \exists k [X_i^k \in T]\}$ は G' の MFVS である。

(証明) 補題 3 より、 $|T| = |F|$ である。補題 1 より、 F は G' の FVS である。 F が G' の MFVS でないと仮定する。このとき、 $|F'| < |F|$ なる G' の FVS F' が存在する。 F' は G' の FVS であるので、補題 2 より、 $T' = \{X_i^4 \mid v_i \in F'\}$ は DP' の TIS であり、かつ、 $|F'| = |T'|$ である。よって、 $|T'| = |F'| < |F| = |T|$ より $|T'| < |T|$ となり、 T が MTIS であることに矛盾する。 □

これらの補題より、以下の定理が成り立つ。ここで、用いる演算モジュールの入力数を 2 に制限した MTIS 問題を 2-MTIS 問題と呼ぶ。

[定理 1] MTIS 問題および 2-MTIS 問題はともに NP 困難である。

(証明) 有向グラフ G の MFVS 問題を 2-MTIS 問題に多項式時間で還元できることを示す。

有向グラフ G の MFVS を以下のように求める。有向グラフ G を前述の変換規則によって、有向グラフ G' 、データパス DP' に順に変換する。この変換は多項式時間で可能である。補題 4 より、 DP' の MTIS から G' の MFVS が求まる。また、 G の MFVS は G' の MFVS より多項式時間で求まる。

以上より、MTIS 問題および 2-MTIS 問題は NP 困難である。 □

3.2 発見的方法

データパスが与えられたとき、その TIS を求める発見的方法を提案する。この方法では、データパス中のハードウェア要素、および演算モジュールの入力で弱可制御であるものを弱可制御性の定義に従い外部入力から順にマークしていく。マークできないハードウェア要素または演算モジュールの入力が残った場合、一つ以上の入力マークされ、かつ自身はマークされていない演算モジュールを選び、その 1 入力をスルー可能入力とし、新たに弱可制御となるハードウェア要素、および演算モジュールの入力をマークしていく。この操作をすべてのハードウェア要素、および演算モ

ジュールの入力がマークされるまで繰り返す。この方法では、弱可制御性の定義に従いハードウェア要素および演算モジュール入力をマークするので、すべてがマークされたとき、データパスは弱可検査となる。以下にアルゴリズムを示す。

[TIS を求める発見的方法]

(Step 1) すべての外部入力をキューに入れる。

(Step 2) 以下の操作をキューが空になるまで繰り返す。キューから一つの要素 a を取り出し、 a をマークする。

(2.1) a が演算モジュールの入力でなければ、その出力と接続するレジスタ、マルチプレクサ、演算モジュールの入力のうち、マークされていないものをキューに入れる。

(2.2) a が演算モジュール M の入力で、かつ M のすべての入力がマークされている場合、あるいは a がスルー可能入力の場合、 M がマークされていなければ M をマークしてキューに入れる。また、 M がスルー演算モジュール候補リストに入っていれば、そのリストから外す。

(2.3) a が演算モジュール M の入力であり、 M がマークされていない入力を持ち、かつ、 M がマークされていなければ、 M をスルー演算モジュール候補リストに入れる。

(Step 3) すべてのハードウェア要素、および演算モジュールの入力がマークされていれば、Step 4 を行う。そうでない場合、スルー演算モジュール候補リストに入っている演算モジュールのうち、その出力に接続するレジスタ、マルチプレクサ、演算モジュールの入力でマークされていないものの個数が最大な任意の 1 演算モジュール M を選び、スルー演算モジュール候補リストから外す。 M の入力でマークされている任意の 1 入力をスルー可能入力リストに加える。 M をキューに入れ、Step 2 に戻る。

(Step 4) スルー可能入力リストを返す。

このアルゴリズムは、与えられたデータパスの大きさに比例した時間で解を得ることができる。提案した発見的方法は必ずしも解の最小性を保証しないが、5. で述べる実験に用いた二つのデータパスについて、ともに最小個のスルー可能入力を求めている。

4. 弱可検査性尺度

順序回路のテスト生成は、組合せ回路部を時間展開して行われることが多いため、データパスを論理合成

して得られる順序回路のテスト生成時間はレジスタの制御、観測に必要なクロック数に依存すると考えられる。本節では、弱可検査なる RT レベルデータパスが与えられたとき、レジスタを弱制御、弱観測するために必要なクロック数を評価する尺度を導入する。

4.1 弱制御費

ハードウェア要素 H を弱制御するために必要なクロック数を評価する尺度を H の弱制御費と言い、 $wcc(H)$ で表す。

[定義 4] 弱制御費 $wcc(H)$

(1) 外部入力 I について

$$wcc(I) = 0$$

(2) レジスタ R について、 H を $H \rightarrow R^{(注2)}$ なるハードウェア要素とすると

$$wcc(R) = wcc(H) + 1$$

(3) マルチプレクサ S について

$$wcc(S) = \min\{wcc(H) \mid H \rightarrow S\}$$

(4) 演算モジュール M について

(a) M がスルー演算モジュールでない場合

$$wcc(M) = \max\{wcc(H) \mid X \in \mathcal{IN}_M [H \rightarrow X]\}$$

(b) M がスルー演算モジュールの場合

$$wcc(M) = \min\{wcc(H) \mid \hat{X} \in \mathcal{IN}_M [H \rightarrow \hat{X}]\}$$

4.2 弱観測費

ハードウェア要素 H を弱観測するために必要なクロック数を評価する尺度を H の弱観測費と言い、 $woc(H)$ で表す。ハードウェア要素 H を弱観測するのに必要なクロック数は、 H から外部出力までの伝搬経路 P 中のレジスタの段数 (弱観測費 I) と P 中の演算モジュールの入力 X を弱制御するのに必要なレジスタの段数 (弱観測費 II) の和を各伝搬経路について求めたうちの最小値である。ここで、伝搬経路 P に対する H の弱観測費 II とは P 中で弱制御する必要のある各入力 X について X に接続するハードウェア要素の弱制御費と、 H から入力 X をもつ演算モジュールまでのレジスタの段数の差 (負のときは 0 とする) を求めたうちの最大値である。次定義中の $\mathcal{L}(X)$ は、各ハードウェア要素の各入力 X について X から外部出力に至る経路に対する X の弱観測費 I (d)、弱観測費 II (p) の組の集合を表す。

[定義 5] 弱観測費 $woc(H)$

(1) 外部出力 U と入力 X_U について

$$\begin{cases} woc(U) = 0 \\ \mathcal{L}(X_U) = \{(0, 0)\} \end{cases}$$

(2) レジスタ R と入力 X_R について

$$\begin{cases} woc(R) = \min\{(d+p) \mid (d,p) \in \bigcup_{R \rightarrow X} \mathcal{L}(X)\} \\ \mathcal{L}(X_R) = \{(d+1, \max(p-1, 0)) \mid (d,p) \in \bigcup_{R \rightarrow X} \mathcal{L}(X)\} \end{cases}$$

(3) マルチプレクサ S と入力 X_S について

$$\begin{cases} woc(S) = \min\{(d+p) \mid (d,p) \in \bigcup_{S \rightarrow X} \mathcal{L}(X)\} \\ \mathcal{L}(X_S) = \{(d,p) \mid (d,p) \in \bigcup_{S \rightarrow X} \mathcal{L}(X)\} \end{cases}$$

(4) 演算モジュール M と入力 $X_M \in \mathcal{IN}_M$ について

$$woc(M) = \min\{(d+p) \mid (d,p) \in \bigcup_{M \rightarrow X} \mathcal{L}(X)\}$$

ここで、入力 X_M については

(a) X_M がスルー可能入力でない場合

$mcc(X_M) = \max\{wcc(X) \mid X \in \mathcal{IN}_M - \{X_M\}\}$ とすると

$$\begin{aligned} \mathcal{L}(X_M) = \{(d, \max(mcc(X_M), p)) \mid (d,p) \in \bigcup_{M \rightarrow X} \mathcal{L}(X)\} \end{aligned}$$

(b) X_M がスルー可能入力の場合

$$\mathcal{L}(X_M) = \{(d,p) \mid (d,p) \in \bigcup_{M \rightarrow X} \mathcal{L}(X)\}$$

4.3 弱検査費

データパスを論理合成して得られる順序回路のテスト生成時間は、各レジスタの弱制御費、弱観測費に依存すると考えられる。レジスタの可検査性はその弱制御費と弱観測費の和で評価できると考えられ、データパス全体の可検査性はすべてのレジスタの可検査性の平均と考えることができる。データパス DP に対して、弱検査費 $wtc(DP)$ を DP のすべてのレジスタの弱制御費と弱観測費の和で定義する。但し、以下の \mathcal{R} は DP のレジスタの集合を表す。

(注2): レジスタは1入力であるから、このような H は一つしか存在しない。

[定義 6] 弱検査費 $wtc(DP)$

$$wtc(DP) = \sum_{R \in R} (wcc(R) + woc(R))$$

5. 実験結果

弱可検査なるデータパスのテスト容易性を評価するため、デジタル信号処理プロセッサの RT レベルデータパスに対して実験を行った結果を示す。用いたデータパスは図 2 に示す 4th Order IIR Cascade Filter (4th

IIR) と図 6 に示す 5th Order Wave Digital Elliptical Filter (5th EWF) である。それぞれのデータパスのハードウェア資源を表 1 に示す。 $bits$ はワード長を表し、 $\#add, \#mult, \#reg$ はそれぞれ加算モジュール数、乗算モジュール数、レジスタ数を表す。

4th IIR, 5th EWF に従来法および本手法を適用した結果をそれぞれ表 2, 3 に示す。表 2, 3 とも、 $orig$ はもとのデータパス、LR は文献 [7] の部分スキャン設計法を適用したデータパス、DP は文献 [5] の非スキャン設計法のうち可検査性が一番高いものを適用した (“0-level testable” な) データパスである。また、表 2 の T1~T6, 表 3 の T1~T9 は、本手法により弱可検査となったデータパスである。もとのデータパスをテスト容易にするために追加したハードウェアを、 $hardware\ overhead$ に示す。ここで、TI はスルー可能

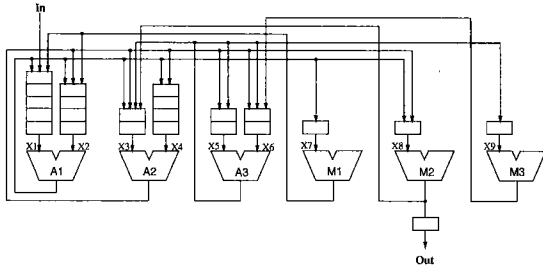


図 6 5th Order Wave Digital Elliptical Filter の RT レベルデータパス

Fig.6 RT-level data path of 5th order wave digital elliptical filter.

表 1 各データパスのハードウェア資源
Table 1 Hardware resources of each data path.

design	bits	#add	#mult	#reg
4th IIR	10	2	3	12
5th EWF	10	3	3	23

表 2 実験結果 1: 4th IIR
Table 2 Experimental result1: 4th IIR.

type	hardware overhead (thru inputs)	#gates	wtc	#faults	test eff. [%]	CPU [sec]	appl. [cycle]
orig	-	5827	-	12536	13.97	>6hr	-
LR	3 scan registers	5947	-	12656	99.83	234	4950
DP	3 multiplexors	5947	-	12776	99.99	72	464
T1	1 TI(X4)	5837	72	12596	99.98	188	426
T2	2 TIs(X4,X5)	5847	70	12656	100.00	87	492
T3	2 TIs(X4,X6)	5847	69	12656	100.00	69	570
T4	3 TIs(X4,X5,X7)	5857	67	12716	100.00	67	602
T5	4 TIs(X4,X5,X6,X7)	5867	64	12776	100.00	64	506
T6	7 TIs(all inputs)	5900	64	12956	100.00	66	704

表 3 実験結果 2: 5th EWF
Table 3 Experimental result2: 5th EWF.

type	hardware overhead (thru inputs)	#gates	wtc	#faults	test eff. [%]	CPU [sec]	appl. [cycle]
orig	-	6576	-	15652	19.52	>8hr	-
LR	15 scan registers	7176	-	16252	99.46	497	19583
DP	6 multiplexors	6816	-	16132	98.12	1428	313
T1	1 TI(X1)	6586	128	15712	97.47	2116	390
T2	2 TIs(X1,X4)	6596	120	15772	98.00	1777	1054
T3	3 TIs(X1,X3,X4)	6606	116	15832	98.00	1567	905
T4	4 TIs(X1,X2,X3,X5)	6616	114	15892	98.11	1519	836
T5	4 TIs(X1,X3,X4,X5)	6616	110	15892	98.08	1512	832
T6	4 TIs(X1,X3,X4,X6)	6616	108	15892	98.16	1417	825
T7	5 TIs(X1,X2,X3,X5,X6)	6626	106	15952	98.17	1404	825
T8	6 TIs(X1,X2,X3,X4,X5,X6)	6636	98	16012	98.18	1347	972
T9	9 TIs(all inputs)	6669	98	16192	99.55	1005	892

入力を表す。表 2, 3 の T1 はそれぞれ 3.2 で述べたアルゴリズムを適用した結果の弱可検査なデータパスであり、追加したスルー可能入力数はともに最小である。

各データパスの VHDL 記述を Mentor Graphics 社の論理合成ツール AutoLogic でゲートレベルネットリストに変換し、Sunrise 社の順序回路テスト生成ツール TestGen を用いてテスト容易性を評価した。#gates は論理合成後のゲート数、#faults は全縮退故障数を表す。test eff., CPU はそれぞれ、テスト効率、Sun SPARCstation10 でのテスト生成時間を表す。また、appl. はテスト実行に必要なクロック数（スキャン操作にかかるクロック数を含む）を表す。

表 2 において、もとのデータパスではテスト効率が 13.97% と低い値であるが、3 種類のテスト容易化設計を適用したところ、いずれも 99% 台まで上がった。提案した手法による設計 (T1) ではハードウェアオーバーヘッドが LR, DP に比べてともに小さいことがわかる。テスト実行時間に関しては、非スキャン方式である DP と同程度であり、スキャン方式 LR の約 1/10 となった。表 3 においては、もとのデータパスでは 19.52% しか得られなかったテスト効率を、本手法により 97.47% まで上げることができた。T1 ではテスト効率が二つの従来法よりも低いが、T9 との比較では、LR, DP よりも小さいハードウェアオーバーヘッドで 99% 台のテスト効率を得ている。また、表 2, 3 とともに、4.3 で述べた弱検査費（表中の *wtc*）と、テスト効率、およびテスト生成時間とに相関が認められる。すなわち、弱検査費はテスト効率やテスト生成時間の見積りに利用できると考えられる。

6. むすび

本論文では RT レベルデータパスを対象とした、スルー演算を用いた非スキャン方式による一つのテスト容易化設計法を提案した。スキャン方式によるテスト容易化設計に対し、提案した手法はスキャン操作を必要とせず、at-speed でテスト実行が可能のため、より多くの故障を検出できると考えられる。実験結果によれば、本手法は少ないハードウェアオーバーヘッドでテスト容易なデータパスを得ることができ、テスト実行時間もスキャン設計よりも大幅に少なくすることができた。また、提案した弱可検査性尺度と、テスト効率、テスト生成時間との相関も認められた。しかし、面積オーバーヘッドであるスルー可能入力の個数と、弱検査

費とはトレードオフがあると考えられ、これらを同時に考慮することは今後の課題である。

本論文ではデータパス系のテスト容易化設計法を示したが、そこでは RT レベルの設計が完了した後にスルー演算機能を追加するという手法である。RT レベルデータパスを設計する高位合成の段階でこの種のテスト容易化を考慮することにより、テスト費用も含めた最適な設計が期待できる。従って、今後の課題としては高位合成においてスルー可能入力を考慮する問題等が考えられる。

謝辞 本研究に関し、多くの貴重な意見を頂いた井上智生助手、四浦洋氏はじめ奈良先端科学技術大学院大学情報論理学講座の皆様方に感謝致します。

文 献

- [1] V. Chickermane, J. Lee, and J. H. Patel, "A comparative study of design for testability methods using high-level and gate-level descriptions," Proc. ICCAD, pp.620-624, Nov. 1992.
- [2] H. Harmanani and C. Papachristou, "An improved method for RTL synthesis with testability tradeoffs," Proc. ICCAD, pp.30-35, Nov. 1993.
- [3] R. Gupta and M.A. Breuer, "Partial scan design of register-transfer level circuits," JETTA, vol.7, no.1/2, pp.25-46, Aug./Oct. 1995.
- [4] P.C. Maxwell, R.C. Aitken, V. Johansen, and I. Chiang, "The effect of different test sets on quality level prediction: When is 80% better than 90%?," Proc. ITC, pp.358-364, Oct. 1991.
- [5] S. Dey and M. Potkonjak, "Non-scan design-for-testability of RT-level data paths," Proc. ICCAD, pp.640-645, Nov. 1994.
- [6] V. Chickermane, E. M. Rudnick, P. Banerjee, and J. H. Patel, "Non-scan design-for-testability techniques for sequential circuits," Proc. DAC, pp.236-241, June 1993.
- [7] D.H. Lee and S.M. Reddy, "On determining scan flip-flops in partial-scan design approach," Proc. ICCAD, pp.322-325, Nov. 1990.
- [8] M.R. Garey and D.S. Johnson, "Computers and Intractability — A Guide to the Theory of NP-Completeness—," Freeman, March 1991.
- [9] D.H. Younger, "Minimum feedback vertex sets for directed graphs," IEEE Trans. Circuit Theory, vol.10, pp.238-245, June 1963.

(平成 8 年 3 月 11 日受付, 7 月 11 日再受付)



高島 勝之

平6 阪大・工・情報システム卒。平8 奈良先端大博士前期課程修了。同年 NTT 入社。テスト容易化設計、高位合成の研究に従事。



井上美智子 (正員)

昭62 阪大・基礎工・情報卒。平1 同大大学院博士前期課程了。同年富士通研究所(株)入社。平7 阪大大学院博士後期課程了。現在奈良先端大情報科学研究科助手。分散アルゴリズム、グラフ理論、テスト容易化設計、高位合成の研究に従事。工博。情報処理学会、人工知能学会各会員。



増澤 利光 (正員)

昭57 阪大・基礎工・情報卒。昭62 同大大学院博士課程了。阪大情報処理教育センター助手、阪大基礎工学部助教授を経て、現在奈良先端大情報科学研究科助教授。分散アルゴリズム、並列アルゴリズム、テスト容易化設計の研究に従事。工博。ACM, IEEE, EATCS, 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 同大大学院博士課程了。阪大工学部助手、明治大理工学部教授を経て、現在奈良先端大情報科学研究科教授。論理設計、テスト容易化設計、テスト生成、高位合成、フォールトトレランスの研究に従事。著書「Logic Testing and Design for Testability」(MIT Press) など。工博。IEEE Fellow。