

演算器の強可検査性を保証するテスト容易化高位合成

和田 弘樹[†] 増澤 利光^{††} 藤原 秀雄[†]

High Level Synthesis for Strong Testability of Operational Modules

Hiroki WADA[†], Toshimitsu MASUZAWA^{††}, and Hideo FUJIWARA[†]

あらまし 本論文では回路面積や動作速度だけでなく演算器の強可検査性を保証したデータバスのテスト容易化高位合成法について考察する。演算器の強可検査性は、データバス上の任意の演算器に対して、外部入力から任意の値を伝達でき、その応答を外部出力で観測できることを保証する。よって単一縮退故障に対して高い故障検出効率が得られる。本論文では、次の条件のもとで任意の高位合成系によって生成されるデータバスがこのテスト容易性を満たすことを示す。(1)データフローグラフが無閉路 (2)全レジスタがホールド機能を有する (3)全演算器で各入力端子ごとに出力端子との間に全単射が実現できる。

キーワード テスト容易化高位合成, レジスタ転送レベルデータバス, 強可検査性

1. ま え が き

半導体技術の発達によって製造可能な論理回路が大規模化している。しかし論理回路の大規模化に伴う設計費用及びテスト費用の増大が問題となっている。

設計費用を削減する目的で高位合成技術が提案されている。高位合成系は設計する回路が実現すべき機能の記述(動作記述)及び回路が満たすべき制約から、動作記述で指定された機能をもち制約を満たすレジスタ転送レベル(RTL)回路を合成する。動作記述の有する抽象性と記述性の高さが設計工数の削減を可能にしている[1]。

高位合成技術は現在も活発に研究が行われており、近年においては面積及び速度等の従来からの制約に加えてテスト容易性に関する制約を同時に満足するテスト容易化高位合成系も提案されている[2], [3]。テスト容易性を考慮せずに合成されたRTL回路に対してテスト容易化設計を施す場合に比べて、テスト容易化高位合成ではより低いハードウェアオーバーヘッドで同等のテスト容易性をもつ回路が得られる。また高位合成

時に得られた回路に関する情報を利用してテスト生成費用も削減できる。

本論文では演算器の強可検査性[4]を保証するテスト容易性を制約とするテスト容易化高位合成法について考察する。演算器の強可検査性は、データバス上の任意の演算器に対して外部入力から任意の値を伝達し、その応答を外部出力で観測することが可能であることを保証する。このような条件を満たす回路に対して単一縮退故障集合に対するテスト生成を行った場合、高い故障検出効率が得られる[4]。

本論文では以下の三つの条件のもとで任意の高位合成手続きが生成するRTL回路上の演算器が強可検査性を満たすことを示す。

- (1) データフローグラフが無閉路
- (2) すべてのレジスタがホールド機能を有する
- (3) 全演算器で各入力端子ごとに出力端子との間に全単射が実現できる

上記の条件は高位合成手続きに関しては制約をもたない。つまり無閉路データフローグラフに対しては、条件(2)(3)を満たすレジスタや演算器を使用するだけで、既存のあるいは将来提案されるであろう任意の高位合成手続きをそのまま用いて、生成されるデータバスのテスト容易性を向上させることができるという利点をもつ。

また加算器や乗算器、減算器といった通常の設計で用いられる一般的な演算器の多くは条件(3)を満たす

[†] 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 and Technology, 8916-5 Takayamacho, Ikoma-shi,
630-0101 Japan

^{††} 大阪大学大学院基礎工学研究科, 豊中市
Graduate School of Engineering Science, Osaka University,
1-3 Machikaneyamacho, Toyonaka-shi, 560-8531 Japan

ので、演算器にはハードウェアオーバーヘッドが生じない。従来の高位合成系で生成されるデータパス上の多くのレジスタはホールド機能を有するため、すべてのレジスタがホールド機能をもつという条件はハードウェアオーバーヘッドをほとんど増加させない。

以下、2. では任意の高位合成系に現れる共通の特徴のみを用いてデータパスの高位合成系をモデル化する。3. では任意の高位合成系で生成されるデータパスがテスト容易となる条件を示し、その条件のもとでデータパスがテスト容易となることを証明する。4. では3. の条件のもとで生成されるデータパスのテスト方法について述べ、5. で全体をまとめる。

2. 諸定義

一般にデータパスの高位合成系は次の手順で動作記述からRTLデータパスを生成する。

(1) フロントエンド：動作記述からデータフローグラフを抽出する。データフローグラフは動作記述に現れる演算の依存関係を表した有向グラフで、動作記述上の演算を頂点とする。演算 f の引数となる変数は f に対応するデータフローグラフ上の頂点 n_f への流入辺として表され、演算 f の結果を受け入れる変数は n_f の流出辺として表される。

(2) アロケーション：RTLデータパスで使用されるレジスタ及び演算器の種類と個数を決定する。

(3) スケジューリング：データフローグラフ上の各演算をRTL回路のクロックに同期した制御ステップに割り当てる。

(4) バインディング：スケジュール済データフローグラフ上の演算の演算器への割当て、変数のレジスタへの割当てを決定し、転送回路でレジスタと演算器を接続する。転送回路はある演算が割り当てられた演算器と演算の流入辺及び流出辺にあたる変数が割り当てられたレジスタ間のデータ転送を行うデータパス上の部分回路である。本論文では2入力のマルチプレクサを適切に組み合わせて転送回路とする。

本章では本論文で取り上げるバインディング系とその入力であるスケジュール済データフローグラフ、出力であるデータパス及び若干の記号を定義する。

2.1 スケジュール済データフローグラフ

問題の簡単化のためにデータフローグラフ上の各演算は一つまたは二つの入力をもち1出力をもつものと仮定する^(注1)。

スケジュール済データフローグラフを、頂点集合 V

と辺集合 E からなる有向グラフ DFG として次のように定義する。

$DFG = (V, E)$ ただし

$$V = \{(op, t_s, t_e) | op \in OP, t_s \in T, t_e \in T\}$$

$$E = \{(v_t, v_h, p) | v_t, v_h \in V, p \in \{0, 1\}\}$$

頂点 $v = (op, t_s, t_e)$ は時刻 t_s に実行を開始し、時刻 t_e に実行を終了する op という種類の演算に対応する。 OP は高位合成系で利用可能な演算の集合、 T は時刻（自然数）の集合とする。以下では v の演算の種類、実行開始時刻及び実行終了時刻をそれぞれ $v.op$ 、 $v.t_s$ 及び $v.t_e$ で表す。 $\forall v \in V (v.t_s \leq v.t_e)$ である。有向辺 $e = (v_t, v_h, p)$ は演算 v_t の結果であり、かつ演算 v_h の p 番目の引数である変数に対応する。以下では e の始点となる演算を $e.v_t$ 、 e の終点となる演算を $e.v_h$ 、 e が $e.v_h$ のどの入力であるかを $e.p$ で表す。

回路外部との入出力は、特別な演算である外部入力演算及び外部出力演算として表現する。外部入力演算は流出辺のみをもつ頂点で、回路外部から取り込んだ値を流出辺に対応する変数に格納する。外部出力演算はただ一つの流入辺のみをもつ頂点で、流入辺に対応する変数に格納されている値を回路外部に出力する。また外部入力演算に対応する頂点の流出辺に対応する変数を外部入力変数と呼び、外部出力演算に対応する頂点の流入辺に対応する変数を外部出力変数と呼ぶ。以下では特に混乱がない限り V の要素を演算として取り扱い、 E の要素を変数として取り扱う。

次に変数集合 E 上の2項関係 \rightsquigarrow_u 及び \rightsquigarrow を次のように定める。

$$e_1 \rightsquigarrow_u e_2 \stackrel{\text{def}}{=} (e_1 = e_2 \vee \exists v \in V (e_1.v_h = v \wedge e_2.v_t = v))$$

$$e_1 \rightsquigarrow e_2 \stackrel{\text{def}}{=} (e_1 \rightsquigarrow_u e_2 \vee \exists e \in E (e \neq e_1 \wedge e_1 \rightsquigarrow_u e \wedge e \rightsquigarrow e_2))$$

つまり DFG 上で変数 e_1 から e_2 へ至る経路が存在する場合かつその場合に限り $e_1 \rightsquigarrow e_2$ は真である。

本論文では DFG が無閉路であること、つまり命題 $\forall e_1, e_2 \in E (e_1 \neq e_2 \Rightarrow \overline{e_1 \rightsquigarrow e_2} \vee \overline{e_2 \rightsquigarrow e_1})$ が恒真であることを仮定する。よって $\forall e \in E (e.v_t.t_e < e.v_h.t_s)$ である。

DFG 上の各変数 $e \in E$ に対して関係 \rightsquigarrow を用いて支配変数集合を $D(e) = \{e' \in E | e' \rightsquigarrow e\}$ と定める。

(注1): 付録参照。

2.2 バインディング系

バインディング系は DFG の演算の演算器への割当て、変数のレジスタへの割当てを決定し、 DFG からデータパス DP を生成する。演算器割当て、レジスタ割当てはそれぞれ、 DFG の演算集合 V の分割 $B_{op} = \{V_1, V_2, \dots, V_k\}$ 、及び辺集合 E の分割 $B_r = \{E_1, E_2, \dots, E_l\}$ として表される。ここで、 $B_{op} = \{V_1, V_2, \dots, V_k\}$ は V の分割なので、

- 各 $i(1 \leq i \leq k)$ について $V_i \neq \phi$
- 各 $i, j(1 \leq i < j \leq k)$ について $V_i \cap V_j = \phi$
- $\bigcup_{i=1}^k V_i = V$

が成り立つ。同様のことが B_r についても成り立つ。

演算集合 V の分割 B_{op} は、各演算集合 $V_i \in B_{op}$ の演算を同じ演算器に割り当て、異なる演算集合の演算を異なる演算器に割り当てることを表している。変数集合の分割も同様である。このことから、同じ演算集合に属する演算の実行時刻は相異なる必要がある。同様に同じ変数集合に属する変数のうち、始点である演算が異なるものの生存時間は相異なる必要がある。

よって演算の分割 B_{op} において 1 以上 k 以下の任意の整数 i に対して命題 $\forall u, v \in V_i (u \neq v \Rightarrow (u.t_e < v.t_s \vee v.t_e < u.t_s))$ が恒真であり、変数の分割 B_r において 1 以上 l 以下の任意の整数 i に対して次の命題が恒真であると仮定する。

$$\begin{aligned} \forall e, e' \in E_i (e.v_t \neq e'.v_t \\ \Rightarrow e.v_t.t_e < e'.v_h.t_s \vee e'.v_t.t_e < e.v_h.t_s) \end{aligned}$$

以下では $V_i \in B_{op}$ なる演算集合 V_i と V_i 上の演算が割り当てられる演算器を同一視する。また変数集合 $E_i \in B_r$ と E_i 上の変数が割り当てられるレジスタを同一視する。更に B_{op} をデータパス上の演算器集合と同一視し、 B_r をデータパス上のレジスタ集合と同一視する。ここで、2 入力演算器 M に割り当てられているすべての 2 入力演算について二つの入力変数の始点が同じ演算である場合、 M に割り当てられている 2 入力演算は 1 入力演算に変換することができる。このような場合、 M に割り当てられているすべての演算を 1 入力演算に変換し、 M に対応する演算器も 1 入力とする。

次にデータパス $DP = (B_{op}, B_r, MMX)$ を次のように定義する。

B_{op} はデータパス上の演算器集合、 B_r はデータパス上のレジスタ集合であり、 MMX は転送回路である。

本論文では簡単のためデータパス上に現れるすべての回路要素（外部入力、外部出力、演算器、レジスタ及びマルチプレクサ）のデータ入力端子及びデータ出力端子の定義域及び値域が等しいものとする。以下ではデータ入力端子 x の定義域を $DOM(x)$ 、データ出力端子 z の値域を $RNG(z)$ とする。

転送回路は 2 入力のマルチプレクサからなるフィードバックをもたないデータパス上の部分回路で、 B_{op} に含まれるすべての演算器と B_r に含まれるすべてのレジスタの出力を入力とし、 B_{op} に含まれるすべての演算器と B_r に含まれるすべてのレジスタの入力を入力とする。

$v.t_s = t$ である演算 v に対して v が割り当てられている演算器を M 、 $e.p = 0$ 及び $e.p = 1$ となる v の流入辺 e （変数）が割り当てられているレジスタをそれぞれ r_0, r_1 とする。 MMX 上に r_0 及び r_1 から M に至る経路が存在して、時刻 t においてこれらの経路に沿って任意の値が伝達されるように各マルチプレクサが制御されるものとする。同様に $v'.t_e = t$ である演算 v' に対して v' が割り当てられている演算器を M' 、 v' の流出辺（変数）が割り当てられているレジスタを r' とすると、 MMX 上に M' から r' に至る経路が存在して、時刻 t においてこれらの経路に沿って任意の値が伝達されるように各マルチプレクサが制御されるものとする。

次にデータパス上の経路 $p = \dots, r_i, M_i, r_{i+1}, \dots$ を演算器とレジスタが交互に現れる系列として定義する。ただし p 上で隣接関係にある任意のレジスタ r_i, r_{i+1} 及び演算器 M_i からなる組に対して二つの条件

$$\begin{aligned} \exists u \in M_i (\exists e \in r_i (e.v_h = u)) \quad \text{及び} \\ \exists v \in M_i (\exists e' \in r_{i+1} (e'.v_t = v)) \end{aligned}$$

が成り立つものとする。これらの条件は、 MMX 上に r_i から M_i へ至る経路と M_i から r_{i+1} へ至る経路が存在することを保証する。

更にデータパス上の任意の回路要素 M において M がコントローラと直接接続している入力端子（制御端子）は回路外部から任意の時刻で任意の値に設定可能であり M がコントローラと直接接続している出力端子（状態端子）は任意の時刻で回路外部から観測可能であるものと仮定する。

3. DP のテスト容易性に関する十分条件

本論文ではデータパス DP のテスト容易性として、

「 DP 上の各演算器が強可検査 [4] である」という条件を用いる．ここで、回路要素 M が以下の二つの条件を満たすとき M が強可検査であるという．

- 強可制御性：外部入力から任意の値の組を M の二つのデータ入力端子まで伝達可能
- 強可観測性： M のデータ出力端子に現れる任意の値を外部出力まで伝達可能

[定理 1] テスト容易データパス生成の十分条件

以下の 3 条件のもとで、任意のバインディング系によって DFG から生成される任意の DP 上の任意の演算器が強可検査である．

- DFG が無閉路
- すべてのレジスタがホールド機能を有する
- 演算器において各入力端子と出力端子の間に全単射が成り立つ

(証明) 次節の補題 2 及び補題 3 より証明される．

□

定理 1 において「演算器の各入力端子と出力端子の間に全単射が成り立つ」とは、演算器が実現する関数族にある関数 f (または g) が存在して次の条件を満たすことである．

- 演算器が 1 入力の場合：データ入力端子を x , データ出力端子を z とした場合、次式が恒真．

$$\forall u, v \in \text{DOM}(x)(u \neq v \Leftrightarrow f(u) \neq f(v))$$

- 演算器が 2 入力の場合：データ入力端子を x , y , データ出力端子を z とした場合、次の二つの式が恒真．

$$\forall u, v \in \text{DOM}(x)(\exists a \in \text{DOM}(y)$$

$$(u \neq v \Leftrightarrow f(u, a) \neq f(v, a))) \text{ 及び}$$

$$\forall u, v \in \text{DOM}(y)(\exists a \in \text{DOM}(x)$$

$$(u \neq v \Leftrightarrow g(a, u) \neq g(a, v)))$$

つまり、1 入力演算器では入出力間の全単射 f を実現できることを意味する．また 2 入力演算器では、一方の入力に定数を印加 (例えば加算器の場合、定数 0 を印加) することで、他方の入力と出力の間に全単射 f (または g) が実現できることを意味する．この全単射 f (または g) を用いれば、演算器の出力を一つの入力端子で制御可能であり、また演算器の入力を出力端子で観測可能である．以下では関数 f (または g) を演算器 M の全単射関数と呼ぶ．

次節では定理 1 の条件のもとで任意の演算器が強可検査であることを示す．

3.1 証明

[定義 1] レジスタの依存関係

レジスタ集合 B_r 上の 2 項関係 $Rdep()$ を次のように定義する．

$$Rdep(r_1, r_2) \stackrel{\text{def}}{=} (\forall e \in r_2(\exists e' \in r_1(e' \in D(e))))$$

□

[補題 1] レジスタの強可制御性

DP 上のすべてのレジスタに任意の値が設定可能である．

(証明) 定義 1 より、 $Rdep(r_1, r_2)$ が偽の場合、 r_1 に割り当てられているどの変数からも DFG 上で到達可能でない変数 e が r_2 に存在する．つまり変数 e の支配集合 $D(e)$ に r_1 に割り当てられた変数は現れない．したがって $D(e)$ 上の変数のみを使用して e に任意の値を設定できることから、 r_1 をホールドしたままでも外部入力から r_2 に任意の値を設定できる．

また DFG が無閉路なので $\forall r_1, r_2 \in B_r(\overline{Rdep(r_1, r_2)} \vee \overline{Rdep(r_2, r_1)})$ が常に成り立つ． $Rdep()$ の定義より $\forall r_1, r_2, r_3 \in B_r(Rdep(r_1, r_2) \wedge Rdep(r_2, r_3) \Rightarrow Rdep(r_1, r_3))$ も常に成り立つ．よって $Rdep(r_1, r_2) \Rightarrow r_2 \geq r_1$ であるように r_1, r_2 の大小関係 \geq を定めると、 \geq は B_r 上の半順序関係をなす．ゆえに \geq を拡張して得られる任意の全順序関係において、最も大きいレジスタ r から順に r に値を設定してから r をホールドすることで、 DP 上のすべてのレジスタに任意の値を設定することができる．□

[補題 2] 演算器の強可制御性

DP 上の任意の演算器 M は強可制御である．

(証明) 転送回路 MMX の定義から M の各入力端子に対して、値を供給するレジスタ r_1, r_2 (ただし $r_1 \neq r_2$) が存在する．補題 1 より全レジスタに対して任意の値が設定できるので、 r_1, r_2 に値を設定した後、 MMX に対して適当な制御信号を与えることで、任意の値を M の入力端子に伝達することができる． M が 1 入力の場合も、同様に証明できる．□

[定義 2] レジスタ間の隣接関係

DP 上の 2 入力演算器 M に対して、レジスタ集合 B_r 上の 3 項関係 $Adj_M()$ を次のように定義する．

$$\begin{aligned} Adj_M(r_1, r_2, r_3) &\stackrel{\text{def}}{=} \exists e_1 \in r_1, \exists e_2 \in r_2, \exists e_3 \in r_3 \\ &(e_1.v_h \in \mathbf{M} \wedge e_2.v_h \in \mathbf{M} \wedge e_3.v_t \in \mathbf{M} \wedge e_1.p \\ &= 1 - (e_2.p)) \end{aligned}$$

□

つまり $Adj_M(r_1, r_2, r_3)$ が真であれば演算器 M の二つのデータ入力端子にレジスタ r_1, r_2 がそれぞれ転送回路 MMX を通して接続しており, M の出力端子にレジスタ r_3 が MMX を通して接続している.

[定義3] 隣接レジスタ間の伝達関係

演算器 M を介して隣接するレジスタ間の関係 $store_M()$ を次のように定義する.

- 演算器 M が1入力の場合:

$$store_M(r_1, r_2) \stackrel{\text{def}}{=} \exists e \in r_1 (\exists e' \in r_2 (e.v_h \in M \wedge e'.v_t \in M))$$

- 演算器 M が2入力の場合:

$$store_M(r_1, r_2) \stackrel{\text{def}}{=} \exists r_3 \in \mathbf{B}_r (Adj_M(r_1, r_3, r_2) \wedge \overline{Rdep(r_1, r_3)})$$

□

M が2入力の場合, $store_M(r_1, r_2)$ が真であれば r_1 をホールドしたままで r_3 に任意の値を設定することができる. そこで r_1 をホールドしたまま r_3 に $r_1 \rightarrow r_2$ 間で全単射を実現するために必要な値を設定した後で M が全単射関数を実現するような制御信号を与えることで r_1 上の任意の値を M を介して r_2 に伝達することができる.

M が1入力の場合は M が全単射関数を実現するような制御信号を与えるだけで r_1 上の任意の値を M を介して r_2 に伝達することができる.

[定義4] レジスタ間の伝達関係

レジスタ集合 \mathbf{B}_r 上の2項関係 $store^*()$ を次のように定義する.

$$store^*(r_1, r_2) \stackrel{\text{def}}{=} \exists M \in \mathbf{B}_{op} (store_M(r_1, r_2) \vee \exists r \in \mathbf{B}_r (store^*(r_1, r) \wedge store_M(r, r_2)))$$

□

$store^*(r_1, r_2)$ が真であれば, DP 上にある経路 $p = r_1, \dots, r_2$ が存在して p 上の任意の部分経路 r, M, r' において $store_M(r, r')$ が真である. よって各部分経路ごとに r から r' へ値を伝達できるので, r_1 の値を r_2 に伝達可能である.

[補題3] 演算器の強可観測性

DP 上の任意の演算器 M は強可観測である.

(証明) 転送回路 MMX の定義から, M の出力端子の値を入力とするレジスタ r が存在する. DFG に関する仮定より r を始点, ある出力変数が割り当てら

れているレジスタ r' を終点とする DP 上の単純経路 p が DP 上に存在する. 補題4で示すとおり, 次の命題は真である.

$$\forall M \in \mathbf{B}_{op} (\forall r_1, r_2, r_3 \in \mathbf{B}_r (Adj_M(r_1, r_2, r_3) \Rightarrow store^*(r_1, r_3) \wedge store^*(r_2, r_3)))$$

よって p 上の任意の演算器 M に対してその前後にあるレジスタを含んだ p 上の任意の部分経路 r, M, r' において r から r' へ値が伝達可能であるので補題は成り立つ. □

[補題4] 命題(隣接伝達関係)の恒真性

次の命題は恒真.

$$\forall M \in \mathbf{B}_{op} (\forall r_1, r_2, r_3 \in \mathbf{B}_r (Adj_M(r_1, r_2, r_3) \Rightarrow store^*(r_1, r_3) \wedge store^*(r_2, r_3)))$$

(証明) $Adj_M(r_1, r_2, r_3)$ が真の場合を考える. まず DFG が無閉路であることから命題 $\forall r_1, r_2 \in \mathbf{B}_r (\overline{Rdep(r_1, r_2)} \vee \overline{Rdep(r_2, r_1)})$ は真. ここで $Rdep(r_1, r_2)$ と $Rdep(r_2, r_1)$ の真偽について以下の2通りに場合分けが可能.

- $Rdep(r_1, r_2)$ と $Rdep(r_2, r_1)$ がともに偽の場合:

2項関係 $store_M()$ の定義より $store_M(r_1, r_3)$ 及び $store_M(r_2, r_3)$ はともに真である. よって命題は真

- $Rdep(r_1, r_2); Rdep(r_2, r_1)$ の一方が真で他方が偽の場合:

一般性を失うことなく $Rdep(r_1, r_2)$ が偽, $Rdep(r_2, r_1)$ が真であると仮定する. 2項関係 $store_M()$ の定義より $store_M(r_1, r_3)$ は真である. よって $store^*(r_1, r_3)$ は真.

また $store^*(r_2, r_3)$ については補題5で示すとおり, $Rdep(r_2, r_1) \Rightarrow store^*(r_2, r_1)$ が成り立つので, $store^*(r_2, r_1)$ は真. $store_M(r_1, r_3)$ が真であることより $store^*(r_2, r_3)$ も真である. よって命題は真.

以上より補題は成り立つ. □

[補題5] 命題(従属伝達関係)の恒真性

次の命題は恒真.

$$\forall r, r' \in \mathbf{B}_r (Rdep(r, r') \Rightarrow store^*(r, r'))$$

(証明) 命題の否定 $Rdep(r, r') \wedge \overline{store^*(r, r')}$ が真であると仮定する.

$Rdep(r, r')$ より $e \rightsquigarrow e'$ を満たす変数の組 (e, e') (ただし, $e \in r, e' \in r'$) が存在する. ここで e を始点,

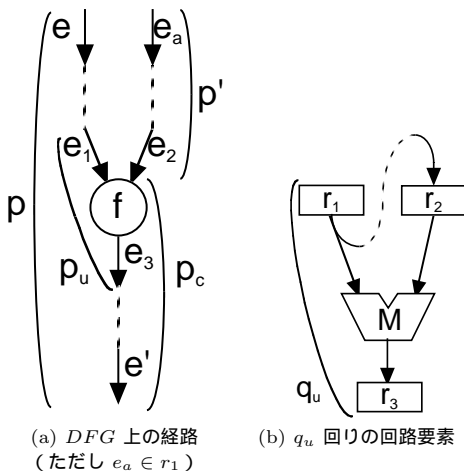
e' を終点とする DFG 上の経路を p , p に対応する DP 上の経路を q とする. すなわち $p = a_1, a_2, \dots, a_n$, $q = b_1, b_2, \dots, b_n$ において各整数 $i(1 \leq i \leq n)$ に対して $a_i \in b_i$ である.

$store^*(r, r')$ よりレジスタ r が保持している値を経路 q に沿ってレジスタ r' まで伝達することはできない. よって q 上に部分経路 r_1, M, r_3 が少なくとも一つ存在して, $store^*(r_1, r_3)$ である. このような条件を満たす q 上の部分経路のうち対応する DFG 上の部分経路 p_u ($p_u = e_1, f, e_3$ とする)(図 1(a)) が p の終点に最も近いものを q_u ($q_u = r_1, M, r_3$ とする) とする(図 1(b)). このとき, p_u 上の演算 f を始点とし, p の終点 e' を終点とする p 上の部分経路を p_c とし, p_c に対応する q 上の部分経路を q_c とすると, q_c の始点である演算器の出力を q_c の終点であるレジスタ r' まで伝達できることを意味する.

ここで DFG 上の f を終点とする e_1 以外の変数を e_2 , e_2 が割り当てられているレジスタを r_2 とする. このとき $store_M(r_1, r_3)$ より $Rdep(r_1, r_2)$ が真. よって $\exists e_a \in r_1(e_a \rightsquigarrow e_2)$ も真である.

DFG 上の e_a を始点, e_2 を終点とする経路を p' とする. DFG が無閉路であることより, p' は p_c と共通部分をもたない(注2). また $store_M(r_2, r_3)$ より, e_2 の値は e_3 に伝達できる. したがって, e_2, f, e_3, \dots, e' で任意の値を伝達できる.

また p' に対して回路要素 r'_1, r'_2, r'_3, M' 及び p'



(a) DFG 上の経路 (ただし $e_a \in r_1$) (b) q_u 回りの回路要素

図 1 演算 f と対応する演算器 M

Fig. 1 Operation f and corresponding operational module M .

上の部分経路 p'_c を p に対する回路要素 r_1, r_2, r_3 , M 及び部分経路 p_c と同様に定めることができる.

ここで p'_c を p に連結した経路を新たに p_c とし, r'_1, r'_2 を新たに r_1, r_2 とすると, 上記の手続きを繰り返すことで任意の p_c に対してその経路上に現れる演算の個数がいくらかでも大きな経路を生成することができる. これはデータフローグラフが有限であることに矛盾する. よって補題は示された. □

4. テストプラン生成手続き

前章で示した条件のもとで任意の高位合成系を用いて合成されたデータパス上の各回路要素を適切に制御することで, 外部入力から任意の値の組を演算器の二つのデータ入力端子まで伝達することが可能であり, また演算器のデータ出力端子に現れる任意の値を外部出力まで伝達することも可能である. このときに必要な制御信号の時系列(テストプラン)は, 次のように容易に生成できる.

補題 1, 2 の条件に従って任意の演算器を強可制御とするデータパスへの制御信号の時系列が生成できる. また補題 5 の証明から $Rdep(r, r')$ であるようなレジスタの組 (r, r') に対して r を始点 r' を終点とする DP 上のある経路 q が存在して, q 上の任意の部分経路 r_1, M, r_2 において $store_M(r_1, r_2)$ であることがわかる. よって, $store_M(r_1, r_2)$ が真となる経路 r_1, M, r_2 のみを用いて外部出力変数が割り当てられているレジスタを根とする DP 上の林 G を生成することができて, この林には DP 上のすべてのレジスタが含まれる. そこで任意のレジスタに対して外部出力変数が割り当てられているレジスタに至る G 上の経路を選択することができて, この経路上の各演算器において定義 3 に従ってデータパスを制御することで任意のレジスタの値をデータパスの外部出力まで伝達することができる.

5. むすび

本論文では強可検査性に基づくテスト容易性を満たすデータパスを任意の高位合成手続きで生成するための条件を提案した. 提案した条件に基づいて生成され

(注2): p' の終点は e_2 であり p_c の始点は f である. p と p' が共有点 o (演算でも変数でもない) をもったとすると p' 上で o から e_2 へ至る経路が存在し, p 上で f から o へ至る経路が存在する. ここで $e_2.v_h = f$ であるので o から o へ至る経路が DFG 上に存在することになるがこれは DFG が無閉路という仮定に反する.

るデータベースはテスト容易であるだけでなくテスト容易性の改善に伴うハードウェアオーバーヘッドが小さい。また高位合成手続きそのものに対しては何らの制約を課さないで従来の高位合成手続きや将来にわたって提案される種々の特徴をもった高位合成手続きに対して、それらに何らの変更を加えることなく生成されるデータベースのテスト容易性を改善することができる。

今後の課題としては、閉路をもつデータフローグラフへの対応等が挙げられる。

謝辞 本研究に際し、多くの貴重な意見を頂いた本学の情報論理学講座の諸氏及び広島市立大学の井上智生助教授に深く感謝します。本研究は一部、日本学術振興会・科学研究費補助金・基盤研究 B(2) (課題番号 09480054) の研究助成、及び (株) 半導体理工学研究センター (STARC) との共同研究 (研究番号 973) による。

文 献

- [1] P. Michel, U. Lauthner, and P. Duzy, The Synthesis Approach to Digital System Design, Kluwer Academic Publishers Group, Dordrecht, 1992.
- [2] I. Ghosh, A. Raghunathan, and N.K. Jha, "Design for hierarchical testability of RTL circuits obtained by behavioral synthesis," IEEE Trans. CAD, vol.16, no.9, pp.1001-1014, Sept. 1997.
- [3] M. Inoue, T. Higashimura, K. Noda, T. Masuzawa, and H. Fujiwara, "A high-level synthesis method for weakly testable data paths," Proc. IEEE 7th Asian Test Symposium, pp.44-45, 1998.
- [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證するレジスタ転送レベルデータベースの非スキャンテスト容易化設計法," 信学論 (D-I), vol.J82-D-I, no.7, pp.843-851, July 1999.

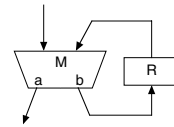
付 録

1. 演算に対する条件の緩和

2. で導入したデータフローグラフでは、各演算は 1 または 2 個の入力をもち 1 出力をもつものと仮定した。演算がこの仮定を満たさない場合、つまり入力が 3 個以上の演算あるいは出力が 2 個以上の演算について、本論文で提案したテスト容易性を満たすための条件を述べる。

1.1 入力が 3 個以上の演算

n 入力 1 出力の演算 f が n 入力 1 出力の演算器 M に割り当てられている場合を考える。マルチプレクサのみを介して M の入力に接続しているレジスタの集合を R_{in} とする。 R_{in} 中の任意のレジスタの値を M の出力に伝達可能ならば、定理 1 が成立する。



M の出力 b の値は M を介して出力 a に伝達

図 A.1 多出力演算器 M が強可検査とならない例
Fig.A.1 Example of the operational module M without strong testability.

R_{in} は補題 1 で定義した大小関係のもとで半順序をなすので R_{in} の極大な要素からなる集合を R_{max} とする。補題 5 と同様に R_{in} 中の任意のレジスタの値を R_{max} 中のいずれかのレジスタに伝達可能であり、かつ補題 1 と同様に R_{max} の要素であるレジスタ r の値を保持したまま $R_{in} - \{r\}$ 中のすべてのレジスタの値を任意の値に変更可能である。以上より多入力演算 f を有するデータフローグラフで定理 1 が成立する十分条件は、 M の入力端子ごとに残りの入力端子に適当な定数を印加することで出力端子との間に全単射を実現する関数が、 M が実現する関数群に含まれることである。

1.2 出力が 2 個以上の演算

多出力の演算器 M を用いてデータベースを構成した場合、 M の応答を M を介して観測することしかできないことがある (図 A.1)。この場合 M は強可検査とはならない [4]。またデータベースを構成するすべての回路要素が 1 出力であればテスト対象の回路要素を介した応答の観測を要しないので、全単射関数とホールド機能を適切なレジスタ及び演算器に割り当てることでデータベースを強可検査とすることができる。よって高位合成によって得られるデータベースが 1 出力の回路要素のみを用いて構成できればよい。

よってデータベースを 1 出力の回路要素のみを用いて構成する目的で、 m 入力 n 出力の演算 f を m 入力 1 出力の演算 f_1, \dots, f_n で置き換え、個々の 1 出力関数を割当て可能な 1 出力演算器を高位合成時に用いる回路要素ライブラリに登録すると、定理 1 が成立する。

(平成 12 年 8 月 9 日受付, 12 月 11 日再受付)



和田 弘樹 (学生員)

平 8 阪大・工・通信卒。平 10 奈良先端科学技術大学院大学博士前期課程了。現在、奈良先端科学技術大学院大学博士後期課程に在学中。現在、テスト容易化設計の研究に従事。



増澤 利光 (正員)

昭 57 阪大・基礎工・情報卒。昭 62 同大学院博士後期課程了。同年同大情報処理教育センター助手。同大基礎工助教授を経て、平 6 奈良先端科学技術大学院大学情報科学研究科助教授。平 12 大阪大学大学院基礎工学研究科教授、現在に至る。平 5 コーネル大客員準教授(文部省在外研究員)。分散アルゴリズム、並列アルゴリズム、テスト容易化設計、テスト容易化高位合成に関する研究に従事。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 46 同大学院博士後期課程了。阪大工学部助手、明治大理工学部教授を経て、現在、奈良先端科学技術大学院大学情報科学研究科教授。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、高信頼設計、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”(The MIT Press)など。大川出版賞受賞。工博。IEEE、情報処理学会各会員、IEEE Fellow、IEEE Golden Core Member。