

実速度スキャンテストにおける キャプチャ時の低消費電力テスト生成手法について

福澤 友晶[†] 宮瀬 紘平[†] 大和 勇太[†] 古川 寛[†] 温 暁青[†] 梶原 誠司[†]

[†]九州工業大学情報工学部 〒820-8502 福岡県飯塚市川津 680-4

E-mail: [†] {fukuzawa, yamato}@aries30.cse.kyutech.ac.jp, {k_miyase, wen, kajihara}@cse.kyutech.ac.jp

あらまし 実速度スキャンテストにおいて、キャプチャ時に多くのフリップフロップで論理値の遷移が起こると過度のIRドロップが生じる可能性がある。過度のIRドロップはゲート遅延を増大させ、正常回路を不良と判定する誤テストを引き起こし、歩留り低下の原因となる。本論文では、遷移故障を対象として、通常の故障検出のみならず、実速度スキャンテストのキャプチャ時の消費電力を考慮したテスト生成手法を提案する。一般に低消費電力テストではテストパターン数が増加する傾向にあるが、提案手法では少ないテストベクトル数の増加で、低消費電力のテストを実現する。ベンチマーク回路に対する実験により提案手法の有効性を示す。

キーワード キャプチャ時の低消費電力, ATPG, X 割当, 実速度スキャンテスト

A Transition Delay Test Generation Method for Capture Power Reduction during At-Speed Scan Testing

Tomoaki FUKUZAWA[†] Kohei MIYASE[†] Yuta YAMATO[†]

Hiroshi FURUKAWA[†] Xiaoqing WEN[†] and Seiji KAJIHARA[†]

[†] Dept of CSE, Kyushu Institute of Technology, 680-4 Kawazu, Iizuka-shi 820-8502, Japan

E-mail: [†] {fukuzawa, yamato}@aries30.cse.kyutech.ac.jp, {k_miyase, wen, kajihara}@cse.kyutech.ac.jp

Abstract High power dissipation can occur when a response to the test vector is captured by flip-flops in at-speed scan testing, resulting in excessive IR drop. Excessive IR drop may cause larger gate delays which may cause good chips to fail tests. As a result, significant capture-induced yield loss may occur in the deep submicron era. This paper addresses this serious problem with a new transition delay test generation method, featuring a unique algorithm that deterministically generates test cubes not only for fault detection but also for capture power reduction. In general, the number of test patterns tends to increase in low power testing, but the new method achieves capture power reduction with less test set inflation. Experimental results show its effectiveness.

Keyword Low Capture Power, ATPG, X-filling, At-speed scan testing

1. はじめに

同期式順序回路のテストにおいて、フルスキャン設計と組合せ回路用 ATPG(automatic test pattern generation)に基づくスキャンテストは広く用いられており、特に大規模回路では必要不可欠になっている。フルスキャン順序回路では、すべてのフリップフロップ(F/F)がシフトレジスタ機構を付加したスキャン F/F に置き換えられ、それらはシフトとキャプチャの2つのモードで動作する。シフトモードでは、スキャン F/F はテスターから直接制御・観測可能なシフトレジスタへ接続される。このモードはシフトインによるテストベクトルの印加、またはシフトアウトによるテスト応答の観測に使われる。キャプチャモードでは、スキャン F/F は通常の F/F として動作し、組合せ回路部分の応答を取り込む。その結果、フルスキャン順序回路のテストは、組合せ回路用の ATPG により組合せ回路部分に対してのみ

テストベクトルを生成することになるため、組合せ回路のテストと同様に扱うことができる[1]。

一方で、スキャンテストには多くの利点があるにも関わらず、テスト実施において、テストデータ量やテスト実行時間の増加、消費電力増加による発熱、さらにはテストに起因する歩留り低下といった問題がある。テストデータ量やテスト実行時間の増加は、回路の大規模化によるゲート数や F/F 数の増加、スキャンチェーンの長大化、遷移遅延故障やパス遅延故障のような複雑な故障モデルへの対応などが原因である。これらはディープサブミクロン回路において必然的に起こる問題である。これらの問題を解決するために、テスト圧縮、多重キャプチャクロックなどの多くの手法が提案されている[2]。

一般にテスト時の消費電力は通常動作時の消費電力よりも大きいことが知られている[3]。これは通常動作時では複数の機能ブロックは相互に機能的に依存しているため、同時に複数の

ブロックは動作しないが、テスト生成やテストクロックは、相互間の機能的な依存性を無視するためである。

CMOS 論理回路の消費電力は漏洩電流による静的消費電力と信号値変化による動的消費電力から成る。スキャンテストの動的消費電力には、シフト時とキャプチャ時の両方が存在する。シフト時の消費電力は、テストベクトルがスキャン回路のスキャンチェーンに印加される時に発生する。このシフト時の消費電力の増大は、蓄積的に過度の熱によって回路を破損させる事や、信頼性を低下させる可能性がある[4]。一方、キャプチャ時の消費電力は、テストベクトルによる組合せ回路部分の出力応答が F/F へ取り込まれる時に、その F/F が保持している値と取り込まれる値が異なる場合に発生する。このキャプチャ時の消費電力の増大は、瞬間的な IR ドロップを引き起こす。その IR ドロップによって F/F の誤動作や回路素子の遅延の増加などが起こる。その結果、テスト時に正常な回路を不良品と誤って判定してしまう誤テストが起こり、歩留り低下の原因となる[5]。

また、近年の VLSI の設計製造技術の進歩により、微細化・高速化が進んでいる。これに伴い、タイミングに影響する欠陥が増加しているため、実速度テストの重要性が高まっている。しかしながら、実速度テストは主に 2 つの要因により誤テストによる歩留り低下を生じやすい。1 つは、テスト時には多数の信号値変化が起こり、過度の IR ドロップが引き起こされるためである。もう 1 つは、テスト応答のキャプチャサイクルが短く、IR ドロップによる遅延の影響を受けやすいためである。それゆえ、テスト時の消費電力の削減は、実速度テストの重要な課題となっており、シフト消費電力とキャプチャ消費電力の削減が求められている。

シフト消費電力の削減を目的とした手法は多く提案されており[6]、それらは主にテストスケジューリング、テストベクトル操作、回路操作、そしてスキャンチェーン操作の 4 つのアプローチに分類される。テストスケジューリングは、IP コアベース設計において、同時にテストするコアを消費電力を考慮して選択することで電力量を一定値以下に制御する。テストベクトル操作は、低消費電力 ATPG、静的圧縮、テストベクトル変更、テストベクトル順序変更、テストベクトル圧縮、符号化などである。回路操作は、遷移ブロック、クロックゲーティング、マルチデューティスキャンなどである。スキャンチェーン操作は、スキャンチェーン順変更やスキャンチェーン分割などである。

シフト消費電力の削減と比べ、キャプチャ消費電力の削減は容易でなく、あまり多くの研究はなされていない。キャプチャ消費電力の削減は通常、テストベクトル操作を基にした手法が必要となる。従来手法の多くは、未定値 X を含むテストベクトル（以降、テストキューブという）中の X ビットに 0 と 1 を適切に割り当てることですべての値が決定したテストベクトルを得るという X 割当手法[7,8]に基づいている。しかし、これらの方法では、テストキューブ中の論理値が 0 か 1 に割り当て済みのビット（以降、特定値ビットという）は故障検出のために変更できないため、キャプチャ消費電力の削減が限定的になるという問題がある。

テストベクトル操作のために必要なテストキューブの生成

には 2 つの方法がある。1 つは ATPG による方法で、限られたわずかな対象故障の検出目的だけにテストベクトル中の論理値を決定し、その他の割当てが不要なビットは X として残す手法である。その結果、テストキューブのみが生成される[1]。このような手法は、テスト生成時間とテストパターン数の増加をもたらす。もう 1 つは ATPG 後に適用する手法で、すべての論理値が決定されたテストベクトルが与えられたとき、故障検出に影響がないビットを見つけ、X ビットに変更する[7,9]。しかし、この X ビット特定の際にはキャプチャ消費電力の削減の有効性は考慮されていない。つまり、テストキューブ中の特定値ビットはキャプチャ消費電力とは無関係に決定されるため、X 割当による全体のキャプチャ消費電力の削減効果は、X ビットに関連する削減のみなので不十分となる可能性がある。

本論文では、実速度スキャンテストにおいて、遷移遅延故障を対象とし、故障検出のみではなくキャプチャ消費電力についても考慮したテストキューブ生成アルゴリズムを提案する。提案するアルゴリズムはキャプチャ消費電力削減を考慮に入れたバックトラックと、故障検出率を保障するための復元含意スタックを新しく PODEM ベースの ATPG に適用する。生成されたテストキューブに X 割当を用いると全体のキャプチャ消費電力について、より良い効果が得られる。

論文構成は以下の通りである。2 節で研究背景を述べる。3 節でテストキューブ生成手法と X 割当に基づくキャプチャ時消費電力削減のための新しい ATPG 手法を説明する。4 節で実験結果を示し、最後に 5 節で本論文のまとめを行う。

2. 準備

2.1. Launch-off-Capture (LoC) 方式

図 1 は Launch-off-Capture 方式のクロックスキームを示している。LoC 方式は、2 つのキャプチャ用クロックパルスを使用することが特徴である。SE はスキャンテストにおけるシフトモード ($SE=1$) とキャプチャモード ($SE=0$) の切り替え制御を行い、CE はキャプチャモードにおける実速度クロック発生器 PLL (Phase-Locked Loop) の動作を制御する。信号値遷移は最後にシフトパルス S_L でスキャン入力された入力パターンと 1 番目のキャプチャパルス C_1 で得られる回路応答値の差分によって発生させられる。この信号値遷移に対する回路応答は、2 番目のキャプチャパルス (C_2) で得られる。 T は回路の実クロック間隔なので、実速度計算が再現される。これによって遅延テストを行うことができる。 S_L から C_1 の T の間隔で SE を変化させさえすればよく、実クロック間隔 T の間隔で SE を変化させる必要がないため、制御を行いやすいという利点がある。

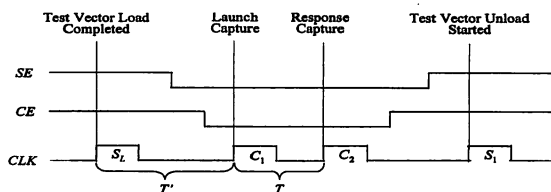


図 1 Launch-off-Capture 方式

2.2. キャプチャ時消費電力の削減

一般的なフルスキャン順序回路の概略を図2に示す。 v はテストベクトルを示す。 v の外部入力 PI と擬似外部入力 PPI のビットはそれぞれ $\langle v:PI \rangle$, $\langle v:PPI \rangle$ とする。組合せ回路部分の論理関数 F , v に対する応答を $F(v)$ とする。 $F(v)$ の外部出力 PO と擬似外部出力 PPO のビットをそれぞれ $\langle F(v):PO \rangle$, $\langle F(v):PPO \rangle$ で示す。

あるスキャン F/F で $\langle v:PPI \rangle$ のあるビット a がそれに対応する $\langle F(v):PPO \rangle$ のビット b と反対の論理値をとる場合、スキャンテストのキャプチャ時にスキャン F/F の出力において論理値の遷移が起こる。それをキャプチャ遷移という。 v に対するキャプチャ遷移の数を $CT(v)$ とする。

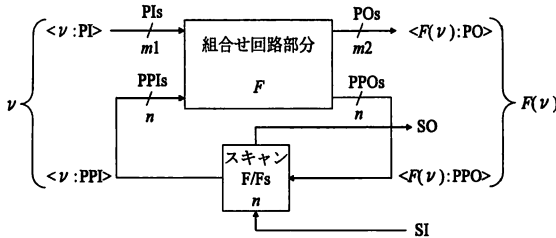


図2 一般的なフルスキャン回路

もし v のすべてのビットに論理値が割り当てられているならば、 $CT(v) = |\langle v:PPI \rangle \oplus \langle F(v):PPO \rangle|$ である。 $CT(v)$ はテストベクトル v によって引き起こされる回路の信号値変化の数と密接に関係していることが知られている[7]。 $CT(v)$ が大きい場合、 $\langle F(v):PPO \rangle$ が取り込まれた時に、組合せ回路部分で多くの信号値変化が起こり、結果としてIRドロップが生じ、キャプチャによる誤動作につながる可能性が高くなる[10]。

キャプチャによる歩留まり低下を減らすには、テストベクトル v のキャプチャ遷移数 $CT(v)$ を小さくするべきである。これは $CT(v)$ が前もって決められた制限値以下になるような v を生成することで達成できる。それゆえLCP(Low Capture Power)テスト生成は、ある正の整数 c_limit を前もって決められたキャプチャ遷移数のリミットとして、 $CT(v) < c_limit$ となるようにテストベクトル v を生成することで達成できる。

2.3. X 割当手法

従来のLCPテスト生成手法の多くはX割当に基づいており、テストキューブ v' 中のXに0と1を適切に割り当てることで、得られるテストベクトル v の $CT(v)$ を制御する。しかしながら、元のテストキューブ中の論理値0,1は、故障検出のためだけに決定されており、キャプチャ消費電力の観点からは望ましくない値である場合もある。

図3のように、テストキューブ $v' = \langle a_1, a_2, a_3, a_4, a_5 \rangle = \langle 1, X, X, 0, 1 \rangle$ とする。特定値ビットは故障 f を検出するためのものである。 $\langle v':PPI \rangle$ 中の特定値ビットは $a_4=0$, $a_5=1$ 、また、 $b_4=1$, $b_5=0$ とする。この場合、 a_4/b_4 と a_5/b_5 においてキャプチャ遷移が生じる。Xビット a_3 に0を割り当てることで a_5/b_5 に関してはキャプチャ遷移を回避することができるが[7,8]、X割当では特定値ビットである a_4 と a_5 でのキャプチャ遷移を回避するこ

とはできない。

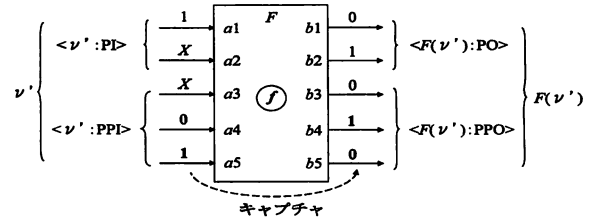


図3 テストキューブ中の特定値ビットの影響

2.4. 方針

図3の例は、十分なキャプチャ消費電力削減を達成するためには、テストキューブ中のXビットについてX割当によるものだけでは不十分となる場合を表している。つまり、キャプチャ消費電力削減がテストキューブの0,1を決定する時にも考慮されると、より大きな消費電力削減効果が得られる可能性がある。そのようなテストキューブの生成を考えるときに注目すべき事実として、一般に故障は複数のテストキューブによって検出され、そのテストキューブの中には特定値ビットに関連するキャプチャ遷移が少ないものも存在することが挙げられる。つまり同じ故障を検出する場合でも、キャプチャ遷移を少なくすることは可能である。図3の例で、テストキューブ $v' = \langle a_1, a_2, a_3, a_4, a_5 \rangle = \langle 0, X, X, 1, 0 \rangle$ も故障 f を検出し、 v' の応答が $\langle b_1, b_2, b_3, b_4, b_5 \rangle = \langle 0, 0, 0, 1, 0 \rangle$ であるとする。この場合、 v' 中の特定値ビットはキャプチャ遷移を引き起こさないようにできる。

これらを考慮し、本論文ではキャプチャ遷移を考慮したテストキューブ生成アルゴリズムを提案する。これはLCPテスト生成のための新しい手法であり、全体のキャプチャ消費電力削減は、本アルゴリズムとX割当手法を組み合わせることによって実現する。

3. 新しいLCPテスト生成手法

3.1. 概要

提案する新たなLCPテスト生成手法は、次の3つのフェーズで構成される。

フェーズ1: LoC方式による実速度スキャンテストを行うために、回路の2時刻分の動作を空間的に展開する。

フェーズ2: 従来の検出重視のATPGにより、十分な検出率をもつ、テスト集合 T をテスト圧縮の技術を用いて生成する。

フェーズ3: T をキャプチャ遷移によって分類する。キャプチャ遷移が多いテストベクトル v に対し、 v を $CT(v) < c_limit$ となるような新しいテストベクトル v'' に置き換える。

LCPテスト生成のための提案手法の処理の流れを図4に示す。フェーズ1では、回路の2時刻分の動作を空間的に展開する。フェーズ2の②の初期テスト集合 T は従来の検出重視のATPGによって生成されたテスト集合である。フェーズ3では、③で T から $CT(v) > c_limit$ という条件で v を選択し、対象テストベクトル T_{tar} を得る。続いて、④では T_{tar} の各テストベクトル v について、 v の代わりに再生成されるテストベクトルで検出す

べき故障のリスト $Ftar(v)$ を求める。 $Ttar$ のすべてのテストベクトルに対して $Ftar(v)$ の全故障が検出された場合、故障検出率の低下は起こらない。 次に、⑤でキャプチャ遷移を考慮したテスト生成で $Ftar(v)$ の故障を検出するための新しいテストキューブ v' を生成する。この処理は $Ftar(v)$ のすべての故障が検出されるまで繰り返す。その後、⑥で v' 中の X ビットに LCP X 割当を行い、すべての論理値が決定したテストベクトル v'' を得る。最後に、⑦で T 中の対象テストベクトル v を v'' に置き換えられることにより新しいテスト集合 T' が得られる。明らかに、 T' は T と同じ故障検出率を保持するが、 T' はキャプチャ消費電力が低いものとなる。

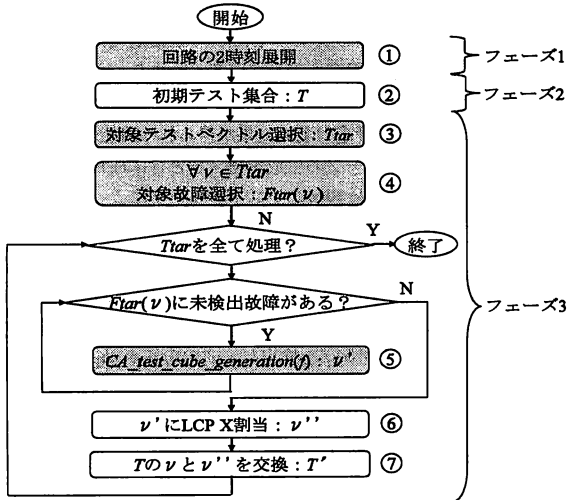


図4 LCPテスト生成のための提案手法の処理の流れ

図4のグレイになっている手順①、③～⑤が、LCPテスト生成のための提案手法に固有のものである。手順①、③～⑤の詳細を3.2～3.5でそれぞれ説明する。

3.2. 回路の2時刻展開

LoC方式による実速度スキャンテストを行うため、本論文では、フルスキャン順序回路の2時刻分の動作を空間的に展開して得られる組合せ回路を用いる。1時刻目の擬似外部出力値を2時刻目の擬似外部入力値に設定できる。また、実速度スキャンテストであるため、1時刻目と2時刻目で外部入力値に遷移を発生させないと仮定する。そこで、時間展開した回路において、外部入力を1時刻目と2時刻目で連結する。2時刻展開後の回路のイメージを図5に示す。

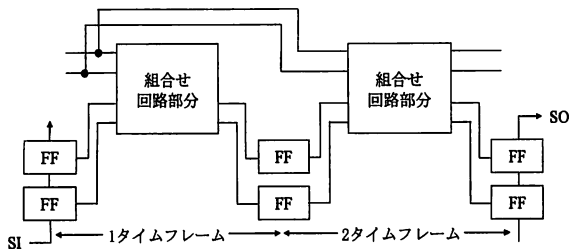


図5 2時刻展開後の回路モデル

3.3. 対象テストベクトル選択

図4の③の処理は、 T からキャプチャ遷移が多いテストベクトル集合を選択するために行い、選択されたテストベクトル集合を $Ttar$ とする。この処理の目的は、元々キャプチャ遷移が少ないテストベクトルは変更しないことにより、無駄な計算を避けることである。テストベクトル選択には電力解析を用いることが最も正確であるが、電力解析の使用は時間を浪費し、またこの段階で電力解析に必要なレイアウト情報は一般に利用可能ではない。そこで、本研究では、 $CT(v) > c_limit$ となるテストベクトル v を選択する。

3.4. 対象故障選択

キャプチャ遷移が多いテストベクトル集合 $Ttar$ を選択した後、 $Ttar$ のテストベクトルを置き換えるために、キャプチャ遷移の少ないテストベクトルを生成する。テスト生成を行う際に、故障検出率を低下させないために、テスト生成で対象となる故障 $Ftar(v)$ の故障リストを作成する。選択される故障は、 $Ttar$ 中のテストベクトルによってのみ検出され、 $T-Ttar$ のテストベクトルでは検出されない故障をすべて含む。これにより、得られるテスト集合に T からの故障検出率の低下は起こらない。

図4の④の対象故障選択の例を図6に示す。図6では元のテスト集合は $T = \{v_1, v_2, v_3, v_4, v_5\}$ である。 v_1, v_4, v_5 がキャプチャ遷移が多いもので、 $Ttar = \{v_1, v_4, v_5\}$ と仮定する。故障シミュレーションにより、12個の故障が検出されたとする。ここで、故障 f_{11} と f_{12} は、 $Ttar$ に含まれない v_2 と v_3 によっても検出されている。 $Ttar$ 中のテストベクトルだけで検出される故障は、 $TA = \{f_1, f_4 \sim f_{10}\}$ である。 TA のすべての故障が、 $Ttar$ と置き換えられる新しいテスト集合によって検出される場合、故障検出率は低下しない。

TA で検出される故障は2つのグループに分類される。 $Ttar$ 中の1つのテストベクトルでのみ検出され、 T の他のテストベクトルでは検出されない故障をベクトル必須故障と呼ぶことにする。図6の丸印で記されたものがベクトル必須故障となる。一方、 $Ttar$ 中の複数のテストベクトルで検出され、 $(T-Ttar)$ のテストベクトルでは検出できない故障を集合必須故障と呼ぶことにする。図6の四角印で記されたものが集合必須故障となる。

	高遷移数	故障分類	対象故障 $Ftar(v_i)$
v_1	Y	(f_1, f_6, f_9, f_{10})	f_1, f_6, f_9, f_{10}
v_2	N	f_2, f_{11}	
v_3	N	f_3, f_{12}	
v_4	Y	$(f_4, f_7, f_8, f_9, f_{12})$	f_4, f_7, f_8
v_5	Y	(f_5, f_{10}, f_{11})	f_5

図6 対象故障選択例

v のすべてのベクトル必須故障は $Ftar(v)$ に含まれる。例えば、 f_1 と f_6 は $Ftar(v_1)$ に含まれなければならない。一方、 v の集合必須故障は $Ftar(v)$ または $Ttar$ 中のその故障を検出する他のテストベクトルの対象故障リストに含めることもでき、そうすることで故障検出率の低下を避けることができる。例えば、 f_9 は v_1 と v_4 によって検出された集合必須故障であるので、 f_9 は $Ftar(v_1)$ か $Ftar(v_4)$ に含める必要がある。本研究では、集合必須故障は再生

成の順番が早いテストベクトルの対象故障リストに含める。しかし、キャプチャ遷移を考慮してテストキューブ生成を行うため、同時に故障を検出できない場合がある。この場合は、同時に検出できない故障を、他のテストベクトルに対する対象故障リストに含めるようにする。他のテストベクトルの生成の際に検出ができれば、パターン数の増加を防ぐことができる。図6では、集合必須故障 f_9 と f_{10} は $Ftar(v_1)$ に含め、もし同時に検出できなくなれば、 $Ftar(v_4)$ に含める例を示している。

3.5. キャプチャ遷移を考慮したテストキューブ生成

あるテストベクトル v に対し、検出すべき対象故障リスト $Ftar(v)$ が得られたら、 $Ftar(v)$ のすべての故障を検出するキャプチャ遷移の少ないテストキューブを生成する。図4中の $CA_test_cube_generation(f)$ は、この目的の為に実行される。そのフローを図7に示す。

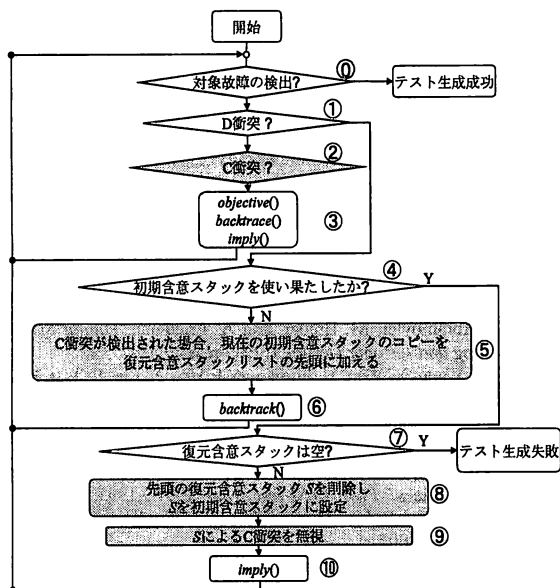


図7 キャプチャ遷移を考慮したテストキューブ生成処理手順

$CA_test_cube_generation(f)$ はPODEM[11]に基づくアルゴリズムであり、グレイで記されたいくつかの追加処理がある。これらの部分は、C衝突と復元含意スタックの2つの新しい考え方に基づいている。これらは、故障 f を検出すると同時に、可能な限りテストキューブ中の特定値ビットのキャプチャ遷移数を削減する。詳細を以下に述べる。

従来のPODEMベースのテスト生成では、バックトラックは、故障の顕在化に失敗したとき、および、X-path-checkでD衝突を見つけたとき(D-frontiersが空になったとき)のみで、故障検出のためのパスを完全に活性化した後には生じない[11]。

$CA_test_cube_generation(f)$ では、C衝突という新しいバックトラックの条件を導入する。これはPPIとそれに対応するPPOが反対の論理値に割り当てられた場合で、1キャプチャ目でのキャプチャ遷移が生じることを意味する。仮に n 個のスキャンF/Fがあるとすると、 n 個のC衝突が存在する、図8に表される C_1 ,

$C_2, \dots, C_n$ で、 C_i は i 番目のスキャンF/FについてのPPIとPPOラインでのC衝突である。1キャプチャ目でのキャプチャ遷移を表すC衝突によるバックトラックで、キャプチャ遷移を削減する。これにより、IRドロップによる回路遅延を削減し、2キャプチャ目で誤ったテスト応答が取り込まれるのを防ぐ。

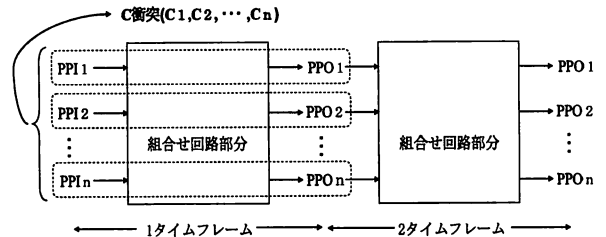


図8 C衝突

図7の $CA_test_cube_generation(f)$ のバックトラック⑥は、①でD衝突または②でC衝突のどちらかが起きたときである。D衝突とC衝突は次に述べる理由で基本的に異なる。もしD衝突のためだけに検索領域が使い果たされた場合、テスト生成は失敗である。しかしながら、検索領域が使い果たされる前に少なくとも1つのC衝突がある場合、テスト生成はC衝突が無視されていたなら、テスト生成に成功するかもしれない。C衝突を無視して生成されたテストキューブは対象故障を検出できるが、対応するスキャンF/Fにおけるキャプチャ遷移を避けることはできない。

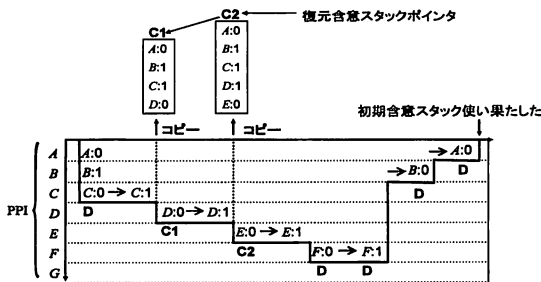
キャプチャ遷移を減らすためにC衝突を考慮することは有益であるが、故障検出のためのテストキューブ生成を妨げる可能性もあるためC衝突を無視することも、状況によって必要である。そこで、2つのタイプの含意スタックを導入する。初期含意スタック(以下PS)は従来のPODEMに基づいたATPGで使われるのと同じもので、検索領域を管理するために使われる。復元含意スタック(以下RS)はC衝突が起こった時に得られるPSのコピーである。複数のC衝突が起こる可能性があるため、複数のRSが存在し、これらのスタックはリストの中に置かれ、RSリストと呼ぶ。

PSが④において使い果たされたとき、⑦においてRSリストが空でないかどうかチェックする。リストが空でないことは、少なくとも1つはC衝突が起こったことを意味し、テスト生成の現在のパスの失敗に起因する。この場合、RSリストの中の先頭あるいは最新のスタックSがリストから削除され、⑧において、PSとして復元される。加えて、⑨の段階でC衝突チェックを進めるので、スタックSに対応するC衝突は無視される。その後、テスト生成を再開する。この方法は、対象故障を検出すると同時に可能な限りキャプチャ遷移数を削減し、テストキューブを生成することができる。

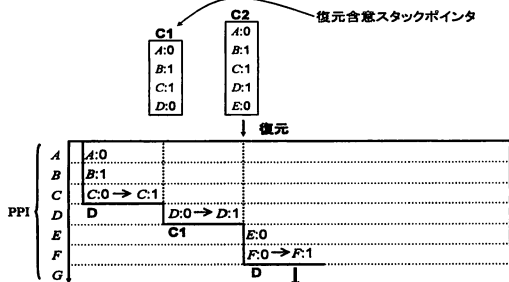
$CA_test_cube_generation(f)$ によるテストキューブ生成の例を図9に示す。 $A, B, \dots, G$ はPPIを示す。 $backtrace()$ はAからGの検索順で、テスト生成中にこれらのPPIの論理値を決定すると仮定する。図9(a)の例では、 $PS = \langle 4:0, B:1, C:0 \rangle$ である時、Dによって示されるD衝突が起こっているとする。バックトラッ

クによって、論理値 1 が C に割り当てられ、*backtrace()*はさらに D に論理値 0 を決定する。PS=<A:0, B:1, C:1, D:0>の時 C 衝突 C1 が起こる。この場合、PS のコピーは、C1 で示すように、RS リストに置かれる。バックトラックにより、D に論理値 1 を割り当てる。同様に PS=<A:0, B:1, C:1, D:1, E:0>の時、C 衝突 C2 が起こり、C2 で示すように PS のコピーは、RS リストに置かれる。最終的に、PS は D 衝突の発生が多すぎたために使用果たされる。

図 9(b)では、RS リスト中の先頭のスタック C2 が、PS として復元され、C 衝突 C2 が無視されつつ、テスト生成が再開される。結果として、テストキューブは PS=<A, B, C, D, E, F, G>=<0, 1, 1, 1, 0, 1, X>となり、対象故障を検出するとともに、C 衝突 C1 に対するキャプチャ遷移を回避できる。



(a) C 衝突と D 衝突によるテスト生成失敗



(b) 含意スタック復元によるテスト生成成功

図 9 キャプチャ遷移を考慮したテスト生成例

4. 実験結果

図 4 の LCP テスト生成のための提案手法のフローを実装し、ISCAS'89 ベンチマーク回路に対して実験を行った。実験では、*c_limit* を初期テスト集合の最大遷移数の 70% に設定した。結果を表 1 にまとめる。表中の "Original" に通常の ATPG の結果、"Proposed Method" に提案手法の結果を示す。また、"Test Vec.", "Target Vec.", "Max. Trans." の欄はそれぞれ、テストベクトル数、再生成の対象となるテストベクトル数、各テストベクトルのキャプチャ時に論理値が遷移する F/F 数とゲート数の和の最大値を示す。また、"Inflation Rate", "Reduction Rate", "CPU" の欄はそれぞれ、提案手法におけるテストベクトル数増減率、最大キャプチャ遷移数削減率、CPU 時間を示す。提案手法では最大キャプチャ遷移数を平均 30.5% 削減することができた。しかし、テストベクトル数については増加するものもあったが、

平均 2.1 % の増加に抑えることができた。

表 1 LCP テスト生成の実験結果

Circuit	Original			Proposed Method			
	Test Vec.	Target Vec.	Max. Trans.	Test Vec.	Inflation Rate(%)	Max. Trans.	Reduction Rate (%)
s1238	26	13	33	25	-3.8	22	33.3
s1423	76	22	175	79	3.9	130	25.7
s5378	178	138	322	172	-3.4	224	30.4
s9234	376	83	401	380	1.1	279	30.4
s13207	323	245	849	329	1.9	593	30.2
s15850	221	108	951	227	2.7	664	30.2
s35932	337	5	6235	338	0.3	4165	33.2
s38417	270	250	1648	314	16.3	1150	30.2
s38584	412	4	5286	413	0.2	3637	31.2
							102.6

5. まとめ

本論文では、実速度スキャンテストにおいて、ATPG 中に故障検出だけでなく、C 衝突の概念も追加することで、キャプチャ消費電力削減を可能にしたテストキューブ生成手法を提案した。提案手法をテストキューブに対する LCP X 割当手法と共に用いることで、より効果的な LCP テスト生成が達成でき、誤テストによる歩留り低下を回避できた。またパターン数は平均すると増加したが、少ない増加に抑えることができた。今後の課題は、電力解析ツールによる提案手法の効果の詳細な検証およびテストパターン数増加のさらなる抑制である。

参考文献

- [1] M. Abramovici, M. Breuer, and A. Friedman, *Digital System Testing and Testable Design*, IEEE Press 1990.
- [2] L.-T. Wang, X. Wen, H. Furukawa, F. Hsu, S. Lin, S. Tsai, K. S. Abdel-Hafez, and S. Wu, "Virtual Scan: A New Compressed Scan Technology for Test Cost Reduction," *Proc. Int'l Test Conf.*, pp.916-925,2004.
- [3] Y.Zorian, "A Distributed BIST Control Scheme for Complex VLSI Devices," *Proc. VLSI Test Symp.*, PP.4-9,1993.
- [4] J. Altet and A.Rubio, *Thermal Testing of ICs*, Kluwer Academic Publishers, 2002.
- [5] T. Yoshida and M. Watari, "A new approach for low power scan testing," *Proc. Intl. Test Conf.*, pp. 480-487, 2003.
- [6] P. Girad, "Survey of Low-Power Testing of VLSI Circuits," *IEEE Design & Test of Computers*, Vol.19, No. 3, pp. 82-92, 2002.
- [7] R. Sankaralingam, R. Oruganti, and N. Toubia, "Static Compaction Techniques to Control Scan Vector Power Dissipation," *Proc. VLSI Test Symp.*, pp. 35-40, 2002.
- [8] X. Wen, H. Yamashita, S. Morishima, S. Kajihara, L.-T. Wang, K. Saluja, and K.Kinoshita, "Low-Capture-Power Test Generation for Scan-Based At-Speed Testing," *Proc. Int'l Test Conf.*, pp. 1019-1028, 2005.
- [9] K. Miyase and S.Kajihara, "XID: Don't Care Identification of Test Patterns for Combinational Circuits," *IEEE Trans. Computer-Aided Design*, Vol. 23, No. 2, pp. 321-326, 2004.
- [10] J.Saxena, K. M. Butler, V. B. Jayaram, and S. Kundu, "A Case Study of IR-Drop in Structured At-Speed Testing," *Proc. Intl. Test Conf.*, pp. 1098-1104, 2003.
- [11] P. Goel, "An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits," *IEEE Trans. Computers*, Vol. 30, No. 3, 1981.
- [12] X. Wen, S. Kajihara, et al, "A New ATPG Method for Efficient Capture Power Reduction During Scan Testing," *Proc. VLSI Test Symp.*, pp.58-65, 2006.