

動的再構成可能なプロセッサの自己テストに関する考察

藤井 昂志[†] 市原 英行^{††} 井上 智生^{††}

[†] 広島市立大学大学院情報科学研究科

^{††} 広島市立大学情報科学部

〒 731-3194 広島県広島市安佐南区大塚東 3-4-1

E-mail: fujii@dsgn.im.hiroshima-cu.ac.jp, [††{ichihara,tomoo}@im.hiroshima-cu.ac.jp](mailto:ichihara,tomoo@im.hiroshima-cu.ac.jp)

あらまし マルチコンテキスト型の動的再構成可能なプロセッサでは、1つのタスクの実行中に複数のコンテキストを切り替えることで高い面積効率を実現している。動的再構成可能なプロセッサは従来の再構成可能デバイスと構成が異なるため、そのテストには新たな手法が必要である。本論文では、テストフレームを用いた動的再構成可能なプロセッサの自己テスト手法を提案する。テストフレームの構成はコンテキストによって表現されるため、提案手法ではハードウェアオーバーヘッドがない。テストフレームは Processing Element (PE) を複数用いて構成したテストパターン生成器、応答解析器とテスト対象 PE から成り、これを切り替えることでプロセッサ全体のテストを実行する。テストフレームの構成が異なれば、テストの実行に必要なコンテキスト数が異なり、また、サイクルタイムも異なるため総テスト実行時間も変化する。ここでは、テストフレームの構成例を示しながら、テストフレームの構成とテスト実行時間、コンテキスト数の関係を考察する。テストの対象となるデバイスによりコンテキスト数の制約やコンテキスト数最小、テスト実行時間最小などのテストの方針も異なる。本手法により、コンテキスト数制約などのプロセッサの特性に応じて適切なテストフレーム構成により、テスト実行時間を最小にできる。

キーワード 動的再構成可能なプロセッサ, 自己テスト, コンテキスト数, テスト実行時間, テストフレーム

A Self-Test of Dynamically Reconfigurable Processors

Takashi FUJII[†], Hideyuki ICHIHARA^{††}, and Tomoo INOUE^{††}

[†] Graduate School of Information Sciences, Hiroshima City University

^{††} Faculty of Information Sciences, Hiroshima City University

3-4-1 Ozuka-higashi, Asaminami-ku, Hiroshima 731-3194

E-mail: fujii@dsgn.im.hiroshima-cu.ac.jp, [††{ichihara,tomoo}@im.hiroshima-cu.ac.jp](mailto:ichihara,tomoo@im.hiroshima-cu.ac.jp)

Abstract Dynamically Reconfigurable Processor (DRP), which can execute a task with multiple hardware contexts so as to achieve high area-efficiency, needs a new test methodology because of its distinctive architecture. In this paper, we propose a self-test method of DRPs without area overhead. This method constructs a test flame of processor elements (PEs) such that it consists of test pattern generators, response analyzers and PEs under test, and switches several test flames dynamically so as to test all the PEs. Since the structure of a test flame decides the number of contexts and test application time, we design some test flames with different structures and discuss the relationship of the structures to the number of contexts and test application time. Based on this discussion, we can construct the best test flame according to a given test environment.

Key words Dynamically reconfigurable processors, self-test, number of contexts, test application time, test-frames.

1. ま え が き

FPGA などの再構成可能デバイスはその柔軟性と開発期間の短さから様々な分野での利用が進んでいる。そのなかでも動的再構成可能なプロセッサに注目が集まっている。動的再構成可能なプロセッサは、構成要素である Processing Element (PE) の

機能と相互接続を動的に変更することで、高い面積効率を実現可能であり、広い分野での応用が提案されている [1]~[3]。動的再構成可能なプロセッサでは、従来の FPGA などの再構成可能デバイスと異なり、動的に回路の再構成が可能のため、動的再構成に適したテスト手法が必要である。

動的再構成可能なプロセッサのテストは、FPGA などの再構

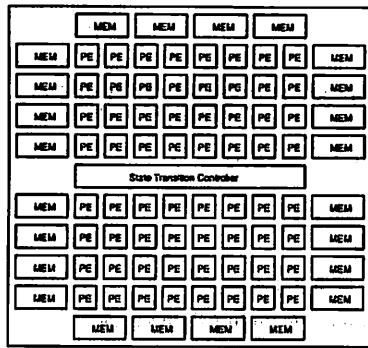


図1 DRPのタイル

成可能デバイスと回路構成が類似する部分があり、FPGAのテスト手法[4]~[7]を応用できると考えられる。現在、動的再構成可能なプロセッサの自己テストのためにPEにDFTとして回路を付加する手法が提案されている[8],[9]。文献[8],[9]では、テスト容易化設計として、PEの一部をLFSR、MISR化することで自己テストを行っている。この手法ではPEに回路を付加しているため、ハードウェアオーバーヘッドが存在する。

本論文では、ハードウェアオーバーヘッドを必要としない動的再構成可能なプロセッサの自己テスト法を提案する。これは、設計変更や回路の付加を必要としないFPGAの自己テスト法[7]を応用したものである。提案手法では、テストパターン生成器(TPG)、応答解析器(RA)、テスト対象PE(PEUT)からなるテストフレームを構成し、複数のテストフレームを用いてテストを実行する。テストフレームの構成方法により、テストの実行に必要なコンテキスト数が増減し、テスト実行時間も変化する。テストフレームの構成方法が異なる設計例からテストフレームの構成とテスト実行時間、コンテキスト数の関係を考察する。なお、今回は、単一PE故障を対象としている。

提案手法では、構成の異なるテストフレームの中から、コンテキスト数などの動的再構成可能なプロセッサの制約やテストの条件に適したテストフレームを選択し、テストを実行できる。

本論文の構成を以下に示す。2節では、テスト対象となる動的再構成可能なプロセッサについて述べ、そのテスト方法と提案手法を適用するモデルについて述べる。3節では、提案手法とテストフレームの構成について述べる。4節では、テストフレームの構成が異なる例から、テストフレームの構成とテスト実行時間、コンテキスト数の関係について考察する。最後に5節にてまとめを述べる。

2. 動的再構成可能なプロセッサ

現在、動的再構成可能なプロセッサは多数提案されている[10]~[12]。動的再構成の実現方式は、命令/構成データ配送方式とマルチコンテキスト方式に大きく分けられる[10]。本論文では、マルチコンテキスト型動的再構成可能なプロセッサについて考える。

マルチコンテキスト型の動的再構成可能なプロセッサとし

て、DRP[10]を例にあげる。DRPのコアには、タイルと呼ばれる再構成可能ユニットがアレイ状に配置され、中央にチップ全体の状態遷移を一括管理するCSTC(Central State Transition Controller)が配置されている。

図1に再構成ユニットであるタイルの構成を示す。図1のように、Processing Element(PE)が8×8のアレイ状に配置されており、中央に状態遷移を行うSTC(State Transition Controller)が配置されている。周囲にはワード構成可変なメモリが配置されている。

PEは演算ユニットであるALUとDMU、レジスタファイル、バスセクタ、命令メモリなどから構成される。DMUはシフト命令、マスク命令、ビット操作命令などを実行する演算ユニットである。PEでは、STCから命令ポインタを受け取り、指定された命令を実行する。命令メモリに格納されている命令コードはPEの機能を決定するオペレーションコードのほか、PE間の接続関係の情報も含まれている。

STCからの命令ポインタによって、個々のPEの機能が決まりPEの機能を組み合わせることで目的の回路を構成する。命令ポインタが変化すると、PEの機能が変更され回路が動的に再構成される。命令ポインタによって指定される構成データをコンテキストと呼ぶ。コンテキストが切り替わると、PEの機能は変わるが、レジスタファイルの内容は保持されるので、コンテキスト間のデータの交換はレジスタファイルを用いて行う。内部に保存可能なコンテキスト数には上限が存在し、上限数を超えるコンテキストを使うためには、外部からコンテキストを読み込む必要がある。

2.1 動的再構成可能なプロセッサのテスト

動的再構成可能なプロセッサをテストするためには、PE、コントローラ、メモリなどのテストを考える必要がある。ここでは、PE部の機能テストについて考える。

PEの機能テストを行う場合、PEは複数の機能を持つため、PEのテストためには、すべての機能をテストする必要がある。また、動的再構成可能なプロセッサは多数のPEから構成されており、そのPE部をテストするためには、各PEのすべての機能をテストする必要がある。

2.2 モデル

提案手法が対象とする動的再構成可能なプロセッサのモデルを説明する。文献[9],[10],[13]を参考にモデルを作成した。

タイルではN個のPEがアレイ状に配置され、PEはSTCにより制御される。STCから出力される命令ポインタはすべてのPEに入力されるものとする。このため、一部のPEの機能を変更するとすべてのPEの命令ポインタが変更されることになる。命令ポインタによって指定される構成データをコンテキストとするので、PEの機能を変更するとコンテキストが増加する。

PEのモデルを図2に示す。図2に示すように、PEは8ビット入力2本、8ビット出力1本、フラグ入力、フラグ出力を持つものとした。PE内部には8ビットALU、シフトやビット操作を行うDMU、命令メモリ、レジスタファイルを持つ。PEの持つ機能は、加減算、AND、OR、EXOR、NOT、多ビット論理シフト、算術シフト、巡回シフトなどがある。ただし、提案す

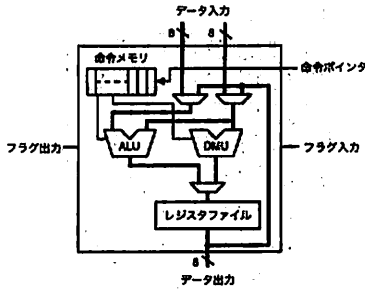


図2 PEのモデル

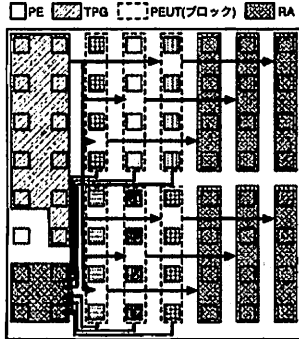


図3 テストフレーム

る手法では、LFSRによる疑似乱数テストを想定しているため機能は限定しない。

コンテキスト間のデータ交換、結果の保存などでレジスタファイルを使用するが、レジスタファイルのサイズ1分大きいものと仮定し、レジスタファイルのサイズによる制約はないものとした。

PEを複数用いて回路を構成すると、構成内容などによって遅延が異なる[13]。本論文では、議論を簡単にするために、PEの機能に関係なく1つのPEの処理にかかる時間を一定とし、単位時間とする。

3. 動的再構成可能なプロセッサの自己テスト手法

3.1 提案手法の概要

本論文では動的再構成可能なプロセッサの自己テスト手法を提案する。提案手法では、テストフレームを用いて自己テストを行う。テストフレームは、複数のPEを組み合わせて構成された、テストパターン生成器 (TPG)、応答解析器 (RA) とテスト対象PE (PEUT) から構成される。

PE部のテストに用いるテストフレームを図3に示す。図3で構成したテストフレームで対象とする故障は、単一PE故障とする。故障を検出するために、TPGではLFSRを用いてランダムパターンを生成し、生成パターンをテストパターンとしてすべてのPEUTに入力する。PEUTの被テスト機能が同じならばPEUTの出力が一致するため、比較器で構成されたRAで出力を比較することで故障の有無を判断できる。図3は、11個

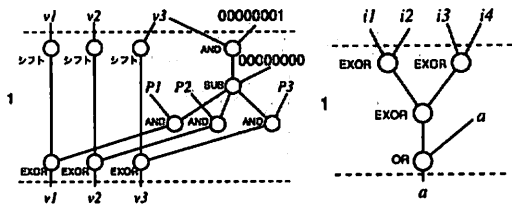


図4 TPGのデータフローグラフ 図5 RAのデータフローグラフ

のPEで構成されたTPGが1つ、4個のPEで構成された7個のRA、4つのPEUTからなるブロックが6個で構成されている。1つのブロックは同じRAで比較されるPEUTで構成し、ブロック単位でPEUTの同じ機能をテストする。これは、故障検出はRAの単位で実行しているため、同じRAで比較できるPEUTの出力が一致していれば故障の検出が可能なのである。

テスト実行は複数のテストフレームを切り替えて行う。つまり、PEUTの位置やPEUTの機能の異なるテストフレームを複数用意し、テストフレームを切り替えることで、すべてのPEのすべての機能をテストする。

すべてのPEのすべての被テスト機能をテストするためのテストの実行に必要なテストフレームの総数 n_f は、PE数 N とPEのテストの対象となる機能数 M 、テストフレームに構成された、PEUT数 n_p で決まる。 N 個のPEの1つの機能をテストするためには、1テストフレーム当たり n_p 個のPEがテスト可能なので、 $\lceil N/n_p \rceil$ 個のテストフレームが必要となる。よって、総テストフレーム数は

$$n_f = \lceil MN/n_p \rceil$$

と表される。

図3のテストフレームは単一PE故障を対象として構成しているが、対象とする故障に合わせて、テストフレームの構成要素であるTPGやRAの構成、PEUTとの接続関係を変更することで他の故障のモデルに対するテストができると考えられる。

3.2 テストフレームとテスト実行時間、コンテキスト数の関係

図3で示したテストフレームにおけるTPGとRAのデータフローグラフを図4、図5に示す。

図4のデータフローグラフはTPGである24ビットLFSRを表している。データフローグラフ中の演算をPEで実行することで回路を構成している。図3では、PEの入力が8ビット2本、フラグの入力が1ビット1本の計17ビットとなっているので、17ビット以上のLFSRを構成すれば、PEのテストパターンが生成できる。図4の $v_1, v_2, v_3, P_1, P_2, P_3$ は8ビットの値を表している。 v_1, v_2, v_3 はLFSRのフリップフロップの状態を表し、 P_1, P_2, P_3 はLFSRの特性方程式となっている。 v_1, v_2, v_3 を1ビット右巡回シフトして、シフトアウトした値と特性方程式からフィードバックループを作成し、 v_1, v_2, v_3 とEXORをとることによって、次状態を生成している。生成された値は次時刻の入力として用いる。生成した24ビットのうち、17ビットをテストパターンとして使用する。

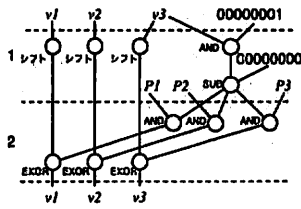


図6 TPG のデータフローグラフ (レーテンシが2のとき)

図5では、比較器として構成したRAのデータフローグラフを示している。PEUTからの8ビット入力*i*₁, *i*₂, *i*₃, *i*₄をEXORツリーで比較している。最後にシグネチャを保存している。時刻*t*に保存されるシグネチャは時刻*t*-1の出力*a*とEXORツリーの出力のORとなる。PEの出力をEXORツリーで比較すると、正常時は0、故障検出時は1となる。1つ前の出力とORをとることでシグネチャを保存するため、故障を検出した場合はシグネチャが1となる。このため、テスト終了時にシグネチャを観測することで、故障検出の有無を判断できる。

図4、図5では、TPGのレーテンシを1として回路を構成している。レーテンシを増やすことで、時刻の変化とともに、PEの機能を動的に変更することで、少ないPEで同じ回路を構成することができる。レーテンシを2とした場合のTPGのデータフローグラフを図6に示す。

図6では、TPGのレーテンシを2として、24ビットLFSRを構成している。TPGのレーテンシを2としたことにより、1時刻あたりに必要な演算器数の最大値が6となる。演算はPEの機能を用いて実行するため、図6のTPGの構成に必要なPE数は6となる。時刻1の構成を時刻2の構成に動的に再構成すること6個のPEで回路を構成する。このTPGでは2サイクルで1つのテストパターンを生成できる。

テストフレームはTPG, RA, PEUTから構成されるため、テストフレームのレーテンシは構成要素のレーテンシの最大値と等しくなる。テストフレームのレーテンシをテストレーテンシと呼ぶ。TPGによるテストパターン生成やRAによるPEUTの出力結果の比較はテストレーテンシ内で実行される。1時刻の処理を1コンテキストで構成するため、テストフレームの構成に必要なコンテキスト数*n_{fc}*はテストレーテンシと等しくなる。

テストフレームの構成により、*n_p*やサイクルタイム、テストレーテンシが決まるため、テストの実行に必要なコンテキストの総数*n_c*やテスト実行時間*T*に大きく影響する。

テストレーテンシを増やすことで、PEの機能を動的に変更することで面積効率を上げ、TPGやRAを構成するために必要なPE数を減らすことができ、その結果、*n_p*が増加する。一方、テストレーテンシが増加し、*n_{fc}*が増加することになる。テストレーテンシを変えた場合のTPG, RAの構成に必要なPE数や*n_p*数の変化などを図7に示す。

図7では、(a)にTPGのレーテンシを1とした場合のTPGのデータフローグラフ、PEを用いたTPGの構成、TPGに必要なPE数、*n_p*などの関係を示している。同様に、(b)ではTPGのレーテンシを2とした場合、(c)ではTPGのレーテンシを3

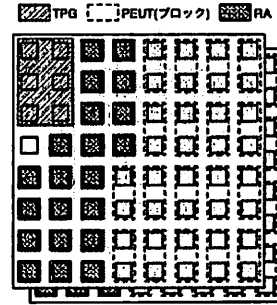


図8 テストフレームの実現例 (テストレーテンシが2のとき)

にした場合をそれぞれ示している。(a), (b), (c)のTPGを用いて構成したテストフレームを*F_a*, *F_b*, *F_c*とする。

図7(a)のデータフローグラフでは、レーテンシを1としてTPGを構成している。この例では、TPGの構成に必要なPE数*n_t*は11で、レーテンシは1、サイクルタイム*t_c*は4となる。このTPGを用いたテストフレーム*F_a*が図3となる。

図3はテストレーテンシが1のテストフレームを構成している。同時に4つのPEUTが比較できるRAを4つのPEで構成している。RAの*t_c*は3となる。TPGとRAの構成が決定すると、*N*, *n_t*, RAの構成に必要なPE数より*n_p*が決まる。図3では*n_p*は24、テストフレーム内でRAとして構成したPEの総数*n_r*は26となる。テストフレームの構成に必要なコンテキスト数*n_{fc}*はテストフレームを1サイクルで実行するためコンテキストを切り替える必要がないので1となる。

図7(b)では、TPGのレーテンシを2として構成している。時刻1で使用したPEを機能を動的に変更して時刻2でも使用することにより*n_t*を小さくできる。TPGのレーテンシが2の場合では、*n_t*が6となり、図3と比較して半分程度になっている。また*t_c*は2となる。処理を2サイクルかけて実行するTPGを用いたテストフレーム*F_b*を図8示す。

図8では、同時に2つのPEUTを比較可能なRAを2つのPEで構成している。RAの*t_c*は1、レーテンシは2となる。TPG, RAともにレーテンシは2となるため、テストレーテンシは2となる。また、テストレーテンシと*n_{fc}*は等しいため*n_{fc}*は2となる。RAの*t_c*は1であるが、TPGの*t_c*が2のため、テストフレームの*t_c*は2となる。図8では、*n_t*=6, *n_r*=21, *n_p*=36となっており、図3のテストフレームの例と比較して*n_t*, *n_r*が減少し、*n_p*が増加している。*n_p*が増加したことで、総テストフレーム数*n_f*が小さくなる。*t_c*と*n_{fc}*に注目すると、図3に比べ*t_c*は小さくなっているが、*n_{fc}*は増加している。

同様に、図7(c)について考える。TPGのレーテンシが4となり、*n_t*=3, *t_c*=1となる。テストレーテンシが4のテストフレーム*F_c*では、*n_{fc}*=4, *n_t*=3, *n_r*=14, *n_p*=46となる。

テストの実行に必要な総コンテキスト数*n_c*は、*n_f*と*n_{fc}*の積で表すことができるため

$$n_c = n_f n_{fc}$$

となる。

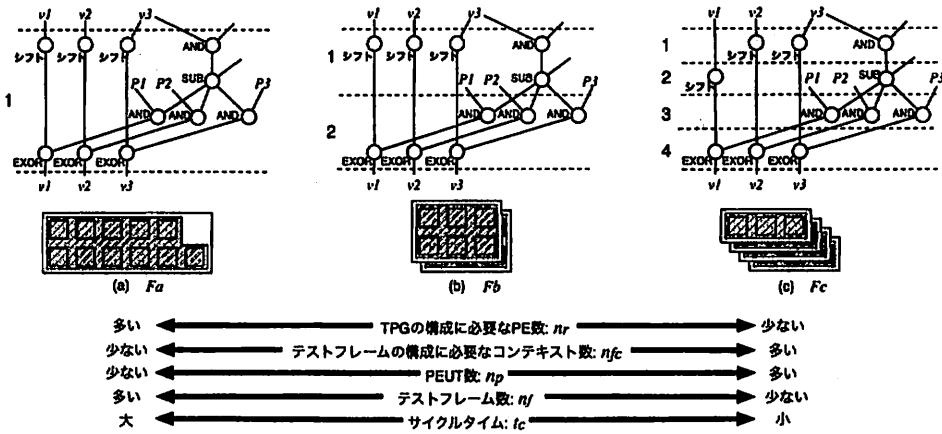


図7 テストフレームの構成例

表1 適用結果

テストフレーム (テストレーテンシ)	n_f	n_r	n_p	n_j	n_{fc}	n_c	t_c	T
Fa (1)	11	26	24	27	1	27	4	$108N_{TP}$
Fb (2)	6	21	36	17	2	34	2	$68N_{TP}$
Fc (4)	3	14	46	14	4	56	1	$56N_{TP}$
Fd (6)	2	12	50	13	6	78	1	$78N_{TP}$

※ Fa, Fb, Fc は図7の (a), (b), (c) に対応。Fd は未記載。

テスト実行時間 T は、 t_c と n_c 、テストパターン数 N_{TP} の積で表すことができるため

$$T = N_{TP} t_c n_c$$

となる。PEの機能の違いによるテストパターン数 N_{TP} の違いや、動的再構成にかかる時間は無視できるものとする。

図7の各TPGを用いて構成したテストフレームの総コンテキスト数 n_c について考える。 n_c は、 n_f と n_{fc} の積で表されるが、 n_f と n_{fc} の間にはトレードオフの関係があり、テストレーテンシが大きくなると n_{fc} は減少し、 n_f は増加する。 T は、 N_{TP} 、 t_c 、 n_c によって決まるため、テストフレームの構成により T が変化し、テスト実行時間 T を最小化する最適なテストフレームの構成が存在することがわかる。

次節では、テストレーテンシの異なるテストフレームの例を示し、テストフレームの構成とテスト実行時間、コンテキスト数の関係を考察する。

4. 適用例

表1に、テストレーテンシの異なるテストフレームのTPGの構成に必要なPE数 n_r 、RAの構成に必要なPE数 n_r 、PEUT数 n_p などを示す。

表1を見ると、テストレーテンシが増えるにしたがって、 n_r 、 n_r 、総テストフレーム数 n_f は減少し、 n_p 、1つのテストフレームの構成に必要なコンテキスト数 n_{fc} は増加することがわかる。テストレーテンシが4のとき、テストフレームのサイクルタイム t_c は最小値の1となるので、テストレーテンシを6にしても

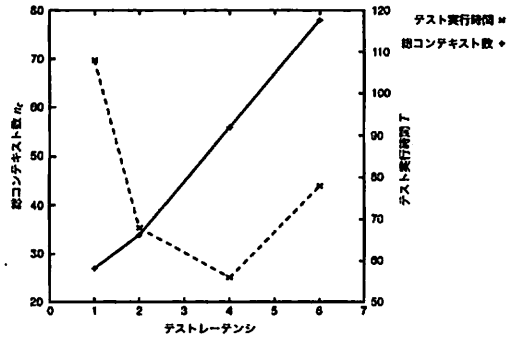


図9 テストフレームとコンテキスト数、テスト実行時間の関係

t_c はこれ以上減少しない。

4.1 考察

図9にテストレーテンシとコンテキスト数、テスト実行時間の関係をグラフ化したものを示す。

図9より、テストレーテンシが増加するにしたがってテスト実行に必要な総コンテキスト数 n_c が増加することがわかる。表1より、テストレーテンシが1に比べてテストレーテンシが2のテストフレームでは、 n_{fc} は2倍に増加するが、 n_f は2/3倍程度のため、テストレーテンシが2のテストフレームでは n_c が増加している。テストレーテンシが2とテストレーテンシが4のテストフレームを比較すると、 n_{fc} が2倍に増加するのに対し、 n_f は減少量が少ないので、 n_c の増加量は大きくなる。このように、テストレーテンシが増加したときの n_{fc} の増加量に比べ n_f の減少量が少ないため、テストレーテンシが増加すると n_c も増加する。

図9においてテスト実行時間は、テストレーテンシが4のときまでは減少し、テストレーテンシが6のとき増加する。これは、テストレーテンシが4のときまでは n_{fc} と t_c が反比例の関係にあり、 n_f の減少がテスト実行時間の減少に影響するためである。テストレーテンシが4と6のテストフレームを比較する

表2 テスト効率

テストフレーム (テストレーテンシ)	t_e
Fa (1)	0.15
Fb (2)	0.3214
Fc (4)	0.6389
Fd (6)	0.5952

※ Fa, Fb, Fc は図7の (a), (b), (c) に対応。Fd は未記載。

と、 n_{fc} が4から6に増加するが、 t_e は1で変化せず、 n_f は14から13と変化量が小さいため、テスト実行時間が増加する。

PEのテストの効率を表す指標として、テストフレームでTPG, RAを構成するPEが単位時間あたりに何個のPEをテスト可能かを表すテスト効率 t_e を考える。未使用PEもTPG, RAを構成するPEとして計算し、 t_e は次式で表す。

$$t_e = n_p / ((N - n_p)n_{fc}t_c)$$

表1に示した各テストレーテンシに対するテスト効率 t_e を表2に示す。

表2では、テストレーテンシが4のときまではテストレーテンシの増加とともに t_e が増加する。このとき、 n_{fc} と t_e は反比例の関係にあり、 n_p の増加が t_e に影響するためである。図8と図3を比較すると、少ないPE数で構成されたTPG, RAで多くのPEをテストしていることがわかる。テストレーテンシが4とテストレーテンシが6のテストフレームを比較すると、 n_{fc} は増加するが、 t_e は変化せず、 n_p も増加量が小さいため、テストレーテンシが6のテストフレームでは、テストレーテンシが4のテストフレームに比べて t_e が減少した。

ここで、テスト実行時間を最小化するためのテストフレームの選択例を示す。コンテキスト数の制約が40の動的再構成可能なプロセッサに提案手法を適用する場合、図9より、テストレーテンシが2のテストフレームを構成すると、テスト実行時間が小さくなることがわかる。一方、外部からのコンテキストの読み込みが十分高速で外部からのコンテキストの読み込みが処理に影響しない場合、言い換えると、コンテキスト数の制約がない場合では、図9より、テストレーテンシが4のテストフレームがテスト実行時間が最小になることがわかる。このように、提案手法では、構成の異なるテストフレームの中からコンテキスト数などの動的再構成可能なプロセッサの制約やテストの条件に適したテストフレームを選択し、テストを実行できる。

5. まとめ

動的再構成可能なプロセッサの自己テスト手法を提案した。提案手法では、テストパターン生成器 (TPG)、応答解析器 (RA)、テスト対象PE (PEUT) からなるテストフレームを構成し、テストフレームを複数用いてPEのテストを行った。

テストフレームの構成法により、コンテキスト数やテスト実行時間が変化することに着目し、複数のテストフレームの例から、コンテキスト数やテスト実行時間の関係を考察した。提案手法では、構成の異なるテストフレームの中から、コンテキスト数の制約やテストの方針に適したテストフレームを選択する

ことで、テストの対象となるデバイスに適したテストを実行できる。

本論文では、PE部の単一PE故障を対象としたテストフレームを用いたが、対象とする故障に合わせて、テストフレームの構成要素であるTPGやRAの構成、PEUTとの接続関係を変更することで他の故障のモデルに対するテストができると考えられる。

本論文では、PE部のテストについて説明したが、PE間の配線、コントローラ、メモリなどのテストは今後の課題である。例えば、PE間を接続する配線のテストは、FPGAの配線のテスト手法[6]が応用できると考えられる。

今後の課題としては、PE間の配線、コントローラ、メモリなどのテスト手法や被テスト機能に応じたテストフレームの構成法の提案、コンテキスト数とテスト実行時間のトレードオフの解析、テストを効率的に実行するためのアーキテクチャの提案などが挙げられる。

参考文献

- [1] 阿部, 長谷川, 黒瀬, 天野, 安生, 栗島, "リコンフィギャラブルプロセッサDRP-1上でのAES-CBCの実装", 第4回リコンフィギャラブルシステム研究会論文集, pp.239-242, 2004年9月。
- [2] 大塚, 倉田, 北道, 黒田, "動的再構成可能デバイスPCA-2への誤差伝搬方法の実装", 信学技報, RECONF2005, Vol.105, No.43, pp.73-78, 2005年5月。
- [3] 黒瀬, 鈴木, 中森, 奥乃, 天野, "動的リコンフィギャラブルデバイスDRPを用いた資源分離フィルタの実装と評価", 信学技報, RECONF2005, Vol.105, No.43, pp.67-72, 2005年5月。
- [4] T. Inoue, H. Fujiwara, H. Michinishi, T. Yokohira, and T.Okamoto, "Universal Test Complexity of Field-Programmable Gate Arrays," Proc. IEEE Asian Test Symp, pp.259-265, 1995.
- [5] T. Inoue, S. Mitazaki, H. Fujiwara, "Universal Fault Diagnosis for Lookup Table FPGAs," IEEE Design & Test of Computers, jan-mar 1998, Vol.15, Issue.1, pp.39-44,1998.
- [6] M. Renovell, J. Figueras, and Y. zorian, "Test of RAM-Based FPGA: Methodology and Application to Interconnect," Proc. IEEE VLSI Test Symp., pp.230-237, 1997.
- [7] C. Stroud, S. Konala, P. Chen, and M. Abramovici, "Built-In Self-Test of Logic Blocks in FPGAs (Finally, A Free Lunch: BIST Without Overhead!)," Proc. IEEE VLSI Test Symp., pp.387-392, 1996.
- [8] K. Katoh, H. Ito, "Built-In Self-Test for PEs of Coarse Grained Dynamically Reconfigurable Devices," Proc. European Test Symp., '06, pp.69-74, 2006.
- [9] 加藤, 姚, 難波, 伊藤, "粗粒度的動的再構成可能デバイスのPE部のテストのためのDFM", 信学技報, DC2006, Vol106, No4, pp.19-24, 2006年4月
- [10] 末吉, 天野, リコンフィギャラブルシステム, オーム社, 東京, 2005.
- [11] T. Sugawara, K. Ide, and T. Sato, "Dynamically Reconfigurable Processor Implemented with IPFlex's DAPDNA Technology," IEICE TRANS. INF. & SYST., Vol87-D, No.8, pp1997-2003, 2004.
- [12] H. Ito, R. Konishi, H. Nakada, H. Tsuboi, Y. Okuyama, A. Nagoya, "Dyanamically Reconfigurable Logic LSI: PCA-2," IEICE Trans. Inf. & Syst., Vol87-D, No.8, pp2011-2020, AUG 2004.
- [13] 天野, 阿部, 出口, "動的リコンフィギャラブルプロセッサの基本トレードオフの解析", 第4回リコンフィギャラブルシステム研究会論文集, pp.25-32, 2004年9月。