社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2006－SLDM－127 (12)
2006／11／28

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

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

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 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なため，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

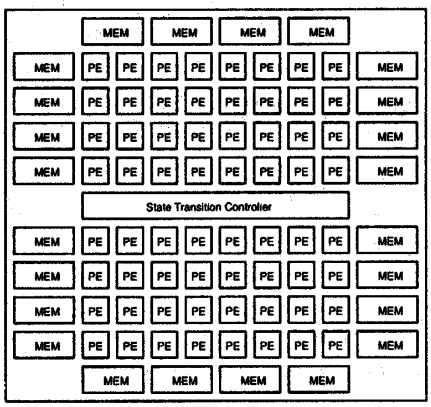


図1 DRP のタイル

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 化と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変化し，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，バスセレクタ，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に，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

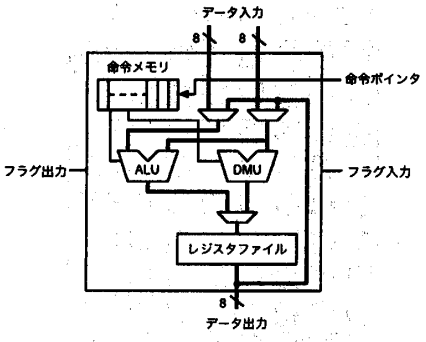


図2　PE のモデル

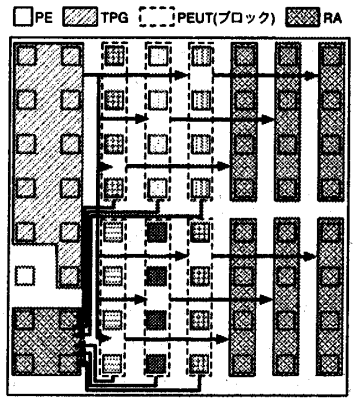


図3　テストフレーム

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十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．

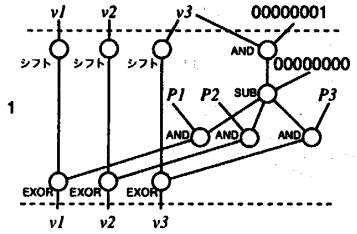
PE を複数用いて回路を構成すると，構成内容などによって遅延が異なる [13]．本論文では，議論を簡単にするために，PE の機能に関係なく 1 つの PE の処理にかかる時間を一定とし，単位時間とする．

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



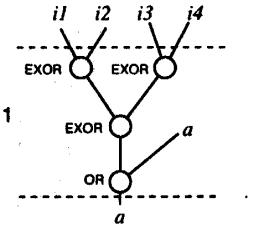


図4　TPG のデータフローグラフ　図5　RA のデータフローグラフ

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ので，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ためである．

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 $v1$，$v2$，$v3$，$P1$，$P2$，$P3$ は 8 ビットの値を表している．$v1$，$v2$，$v3$ は LFSR のフリップフロップの状態を表し，$P1$，$P2$，$P3$ は LFSR の特性方程式となっている．$v1$，$v2$，$v3$ 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，$v1$，$v2$，$v3$ 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．

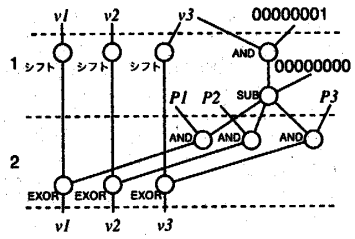


図6 TPGのデータフローグラフ(レーテンシが2のとき)

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$i1$，$i2$，$i3$，$i4$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Fa，Fb，Fcとする．

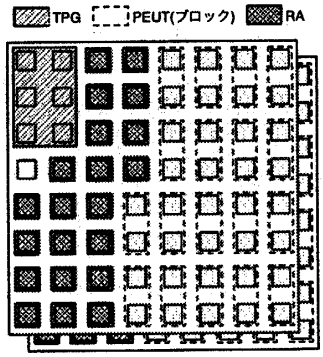


図8 テストフレームの実現例(テストレーテンシが2のとき)

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Fa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FB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Fc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．

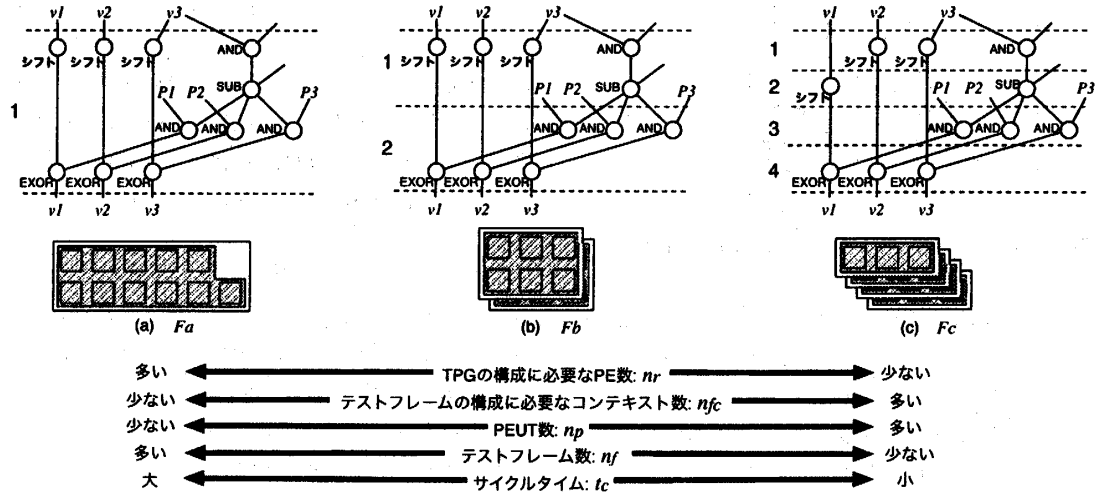


図7　テストフレームの構成例

表1　適用結果

テストフレーム (テストレーテンシ)	n_t	n_r	n_p	n_f	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_t，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_t，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

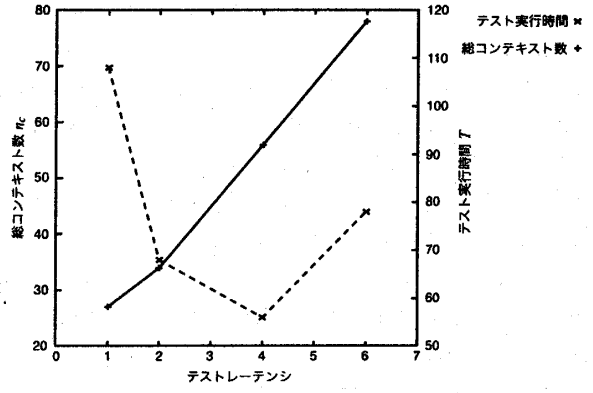


図9　テストフレームとコンテキスト数，テスト実行時間の関係

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のテーストフレームでは，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_c は1で変化せず，n_f は14から13と変化量が少ないため，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_c 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_c は変化せず，n_p も増加量が少ないため，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，黒瀧，大野，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] 黒瀧，鈴木，中盛，奥乃，大野，"動的リコンフィギャラブルデバイス DRP を用いた音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，"粗粒度動的再構成可能デバイスの PE 部のテストのための DFT"，信学技報，DC2006，Vol106，No4，pp.19-24，2006 年 4 月

[10] 末吉，大野，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 Im plemented with IPFlex's DAPDNA Tecnology," 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] 大野，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．