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

2007－ARC－171 (9)
2007－EMB－ 3 (9)
2007／1／22

# 省エネルギー指向マルチコアプロセッサにおける値予測機構の影響

舟木 敏正† 千代延 昭宏† 佐藤 寿倫‡

† 九州工業大学 情報工学部 知能情報工学科
‡ 九州大学 システムＬＳＩ研究センター

携帯情報端末用プロセッサでは，高性能かつ省エネルギーなプロセッサが要求されており，近年需要が急速に増えている．この要求に対して，我々はコントレイルプロセッサを提案している．コントレイルプロセッサはマルチコアプロセッサである．値予測を用い投機的マルチスレッド実行することで，エネルギー消費量を削減する．これまでは最終値予測機構を用いコントレイルプロセッサの評価を行ってきた．しかし他の予測機構でのコントレイルプロセッサの評価を行っていない．そこで，本稿ではコントレイルプロセッサにおける値予測機構の影響について調査する．

# Evaluating Value Predictors in an Energy-Efficent MultiCore Processor

TOSHIMASA FUNAKI† AKIHIRO CHIYONOBU† TOSHINORI SATO‡

†Departoment of Artificial Intelligence, Kyushu Institute of Technology
‡Sytem LSI Research Center, Kyushu University

For portable devices, high-performance and low-power processors are required, and the requirement has increased rapidly in recent years. Considering the trend, we are proposing Contrail processor, which is built on MultiCore processors. Its energy consumption is reduced by speculative multithreading based on value prediction. While we evaluated Contrail processor that utilizes last value predictor, we have not considered the use of other value predictors. In this paper, we conduct our research on evaluating how several value predictors affect Contrail processor performance and its energy efficiency.

## 1. はじめに

近年，フルブラウザや音楽・動画再生などの高度なアプリケーションを携帯情報端末上で動作することが求められており，より高性能なプロセッサが求められている．しかし，高性能なプロセッサはエネルギー消費量の増加という問題を引き起こしている．特に携帯情報端末では，バッテリーの容量がエネルギー消費量を制限している．この問題に答えるために，最近では汎用プロセッサ・組み込みプロセッサの両方で，低消費電力なマルチコアプロセッサ[6),8)]が研究・開発されている．携帯情報端末用プロセッサでは，高性能かつ省エネルギーなプロセッサの需要が急速に増えている．

プロセッサのエネルギー消費量は，消費電力とプログラム実行時間の積で求められる．したがって，プロセッサのエネルギー消費量を削減するためには，プログラム実行時間を増加させずに消費電力を削減する，もしくは消費電力を増加させずにプログラム実行時間を削減しなければならない．この目標を達成するために，我々はコントレイルプロセッサ[4),5),9)]を検討している．コントレイルプロセッサは，予測を用いることで[4)]命令ストリームを分割し，複数スレッドに分割された命令ストリームをマルチコアプロセッサ上で並列に投機実行する．予測が成功していれば，その検証を低速に実行しても性能に悪影響を及ぼさないので，クロック周波数や電源電圧を抑えて実行することが可能になる．つまりコントレイルプロセッサでは，値予測に基づく投機実行によってプログラム実行時間を削減し，値予測の検証を低速で実行することによって消費電力を削減している．

値予測とは命令の実行結果を予測する手法である．プログラム中に出現する値の局所性を利用し，命令の実行結果を過去の実行結果から予測する．予測値を用いることでデータ依存関係を解消し，投機的に命令を実行できる．予測精度が高いほど投機実行が成功するので，コントレイルプロセッサのエネルギー消費量削減効果は高くなる．値予測には最終値予測，ストライド値予測，コンテキストベース値予測，ハイブリッド値予測などの方式がある[7)]．これまでは最終値予測機構を使いコントレイルプロセッサの評価を行ってきた．しかし他の予測機構でのコントレイルプロセッサの評価を行っていない．

本稿では，コントレイルプロセッサにおける値予測機構の影響について評価を行う．2節でコントレイルプロセッサを，3節では値予測機構を説明する．4節で評価結果を紹介する．5節はまとめである．

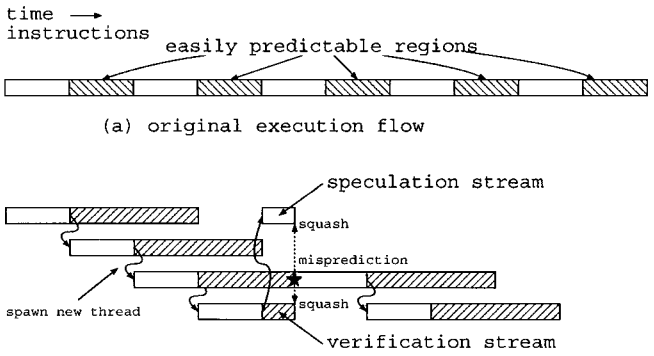


**図 1** コントレイルプロセッサ上での実行例

## 2. コントレイルプロセッサ

コントレイルプロセッサはエネルギー利用効率を改善するために，アプリケーションプログラムの実行を 2 種類の命令ストリームに分割する．このうちプログラム実行の主要部分を担う命令ストリームを投機流 (speculation stream) と呼び，もう一方を検証流 (verification stream) と呼ぶ．コントレイルプロセッサでは，投機流で値予測が用いられている[4]．値予測を用いてプログラム中の予測可能な命令を実行命令列から取り除き，投機的に命令ストリームを実行する．一方，検証流ではその投機流で実行された値予測の検証を行う．この投機流と検証流をスレッドとしてマルチコアプロセッサ上の複数のプロセッサコアによって並列に実行される．

コントレイルアーキテクチャの特徴は値予測を用いることによってプログラム中のクリティカルな命令列を非クリティカルに変換し，投機流から検証流へと移動させることである．投機流では予測された命令が検証流へと移されるため，実行される命令は元のプログラムに比べて少なくなる．検証流では，予測した値が正しければ，性能に悪影響を与えることなく検証を低速に行うことが可能である．これにより値予測によって実行命令数が少なくなる投機流ではプログラムの実行時間を削減できるので消費電力が削減され，検証流では検証を低速に，つまり電源電圧とクロック周波数を下げて実行することで電力消費を削減することが可能である．コントレイルプロセッサ上でのプログラム実行例を図 1 に示す．

図 1(a) に示す元の命令列は予測可能な命令列とそれ以外に分けることができる．この例では値予測によって投機流から取り除かれる命令列はプログラム中に均等に存在し，それら全てに値予測が成功した場合を表している．すなわち全命令の半分が予測対象になると仮定している．図 1(b) は同じ命令列をコントレイルプロセッサ上で実行している．投機流で予測の実行によって取り除かれた命令列を検証流として低速に実行している．この例では検証流のクロック周波数を投機流の半分としている．投機流と検証流は複数のプロセッサコアによって分散的に実行される．最も若い投機流を実行しているプロセッサコアで予測可能な命令列が検出されると，この命令列の先の命令から空いているプロセッサコアで投機流の実行が開始される．各プロセッサコアは予測された命令列を検証流として実行するときにクロック周波数や電源電圧を変更する．値予測に失敗しない理想的なケースでは，検証流が完了するとスレッドは消滅し，そのプロセッサコアは開放される．予測に失敗した場合は，図 1(b) のように予測失敗時以降の投機流と検証流は破棄され，検証流の結果を利用して投機流の実行が再開される．

## 3. 値 予 測

値予測とは命令の実行結果を予測する手法である．予測された値を用いて投機的に命令を実行することが可能になるので，プロセッサの性能向上を困難にする要因であるデータ依存関係を解消することができる．成立・不成立の 2 パターンしか存在しない分岐予測とは異なり，実行結果を予測することは非常に困難に思われるかもしれない．しかし，プログラム中で出現する値の局所性を利用することで，過去の実行結果から命令の実行結果の予測を行うことは容易である．様々なアルゴリズムで履歴から値を予測する方式があり，最終値予測，ストライド値予測，コンテキスト・ベース値予測，ハイブリッド値予測などの予測方式がある[7]．

### 3.1 最終値予測

最終値予測は前回の実行結果を予測値として返す方式である．図 2 に最終値予測器の構成を示す．

最終値予測は命令アドレスをインデックスとする値履歴表 (VHT:Value History Table) を持つ．VHT の各エントリは命令を識別するタグ，予測の信頼性を示す飽和型アップダウンカウンタ，対象命令が最後に出力した値から構成される．まず命令アドレスの下位 i

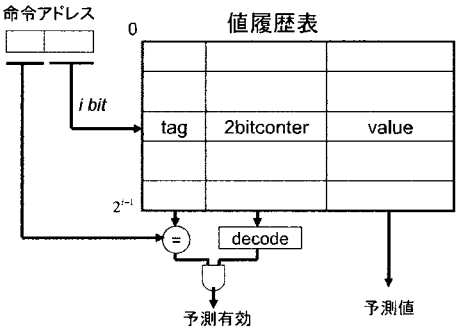


図 2　最終値予測器

ビットを使って VHT を検索する．対応するエントリのタグと命令アドレスを比較してエントリが有効であるかを調べる．さらに飽和型アップダウンカウンタの値が閾値以上であれば，予測は有効となりエントリに記憶された最終値を予測値として出力する．

VHT のエントリ数を 4096 とした場合，tag は 18 ビット，2bc は 2 ビット，value は 32 ビットとなり，1 エントリの容量は 52 ビットとなる．最終値予測は，最もシンプルであるため，ハードウェア量は他の値予測機構より少なくて済む．

### 3.2　ストライド値予測

ストライド値予測は，値がループ変数のように一定の差分を持つと予測する方式である．図 3 にストライド値予測器の構成を示す．ストライド値予測器も最終

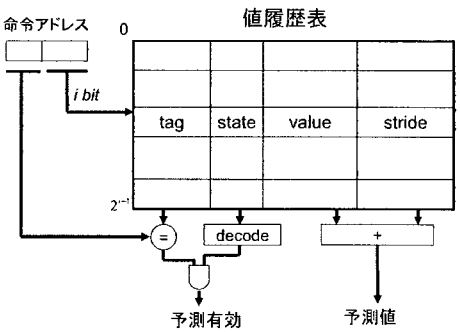


図 3　ストライド値予測器

値予測器と同様に VHT を持ち，VHT の各エントリはタグ，ステート，最終値，差分値から構成され，最終値に差分を加えたものが予測値として出力される．ステートは init，transient，steady の 3 状態のいずれかを取る．ステートはエントリの登録時に init をセットし，次回そのエントリ参照した時に実行結果から差分を取り，transient に移る．再びそのエントリを参照した時は，前回と差分が一致すれば steady へと移行する．差分が一致しない場合は差分を更新し，transient になる．図 4 にストライド値予測におけるステートの状態遷移図を示す．

ストライド値予測は，最終値予測を改良した予測機

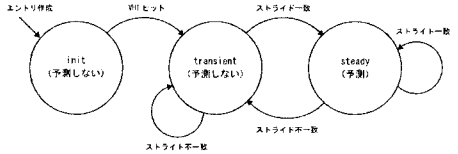


図 4　ストライド値予測器におけるステート

構と考えることもできる．ループの多いプログラムでは，ストライド値予測は最終値予測より予測精度が高くなると期待される．

VHT のエントリ数を 4096 とした場合，tag が 18 ビット，state が 2 ビット，value が 32 ビット，stride が 32 ビットとなり，1 エントリの容量は 84 ビットとなる．実行結果だけでなくストライド値も保持しておかなければならないため，最終値予測よりも多くのハードウェア量を必要とする．

### 3.3　コンテキスト・ベース値予測

コンテキスト・ベース値予測は過去の実行結果を複数記録し，過去の値の出現パターンから予測する値を決定する方式である．図 5 に代表的なコンテキスト・ベース値予測方式の 2 レベル値予測器を示す．2 レベル値

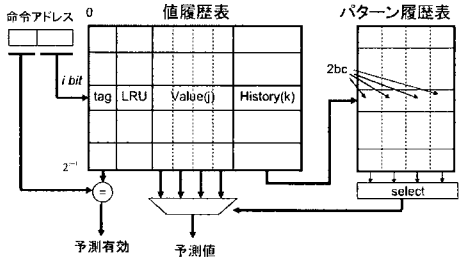


図 5　コンテキスト・ベース値予測器

予測器は，命令アドレスをインデックスとする VHT とエンコードされた履歴パターンをインデックスとするパターン履歴表 (PHT: pattern history table) から構成される．VHT の各エントリはタグ，対象のエントリにある最も過去に現れた値，最近の異なる j 個の実行結果，最近 k 個の履歴パターンから構成される．PHT の各エントリは j 個の 2 ビットカウンタから構成され，履歴パターンに対して VHT に格納された値のどれが次に出現するかの頻度を記録する．予測は VHT から命令に対応するエントリを検索し，エントリが存在すれば履歴パターンの値を用いて PHT を検索する．そして PHT のエントリ内の各フィールドのカウンタ値を比較し，PHT の比較結果から最近の実行結果の内どの値を予測結果とするか選択する．選択された値が閾値以上であれば予測値として出力する．

VHT のエントリ数を 4096，j を 4，k を 6 とした場合，tag が 18 ビット，LRU が 2 ビット× 4，value が 32 ビット× 4，history(k) が 2 ビット× 6 となり 1 エントリの容量は 166 ビットとなる．PHT は 2bc

が2ビット×4となり，1エントリの容量は8ビットとなる．ハードウェア量が制限されてない場合，高い予測成功率を期待できる．しかしながら最終値予測やストライド値予測に比べ，はるかに大きなハードウェア量を必要とし，ハードウェア量を制限すると，予測精度が低下すると言われている．

### 3.4 ハイブリッド値予測

一般に値の出現パターンによって予測可能な方式は異なる．例えばループ内で一定の値の増減がある命令は，ストライド値予測で予測可能であるが最終値予測での予測は難しい．そこで複数の予測方式を組み合わせて状況に応じて最適な予測器を用い予測を行う方式がある．これはハイブリッド値予測と呼ばれる．本稿ではストライド値予測とコンテキスト・ベース値予測の組み合わせを使用する．コンテキスト・ベース値予測が予測可能なら，コンテキスト・ベース値予測を用いて値予測を行い，コンテキスト・ベース値予測が予測しない時，ストライド値予測を用い値予測を行う．

VHTのエントリ数を4096，jを4，kを6とした場合，tagが18ビット，stateが2ビット，strideが32ビット，LRUが2ビット×4，valueが32ビット×4，history(k)が2ビット×6となり，1エントリの容量は200ビットとなる．PHTは2bcが2×4ビットとなり，1エントリの容量は8ビットとなる．

2つの値予測機構を組み合わせるために高い予測成功率を期待できる．しかし，非常に大きなハードウェア量を必要とし，ハードウェア量を制限すると，予測精度が低下してしまう．

## 4. 評　　価

### 4.1 評価環境

評価にはスーパースカラプロセッサシミュレータであるMASE/SimpleScalar[2]を基に作られたコントレイルプロセッサシミュレータを用いる．命令セットにはMIPSの命令セットを拡張したPISAを用いる．各プロセッサコア構成は表1の通りである．ベンチマークプログラムにはSPEC2000のCINTとCFPより11種類，MediaBenchより10種類，MiBenchより3種類使用した．SPEC2000では先頭より5億命令をシミュレーションした．MediaBench，MiBenchでは全ての命令をシミュレーションした．

命令キャッシュ(I$)とデータキャッシュ(D$)は理想的とし，それぞれ常にヒットするものとする．分岐予測も常に正しく予測できると仮定する．さらにメモリを介した曖昧な依存関係も理想的に解消可能とする．スレッドの分割には固定間隔分割法[1]を選んだ．なお本研究では命令列を32命令間隔に分割し，同時実行可能なスレッド数は4としている．新しいスレッドの起動時には8サイクルが必要と仮定した．

各値予測機構の構成を表2に示す．コンテキスト・ベース値予測器とハイブリッド値予測器のjとkには，

**表1** プロセッサの構成

| | |
|---|---|
| 命令フェッチ幅 | 8命令 |
| 命令デコード幅 | 2命令 |
| 命令発行幅 | 2命令 |
| 命令完了幅 | 2命令 |
| 命令キュー | 16エントリ |
| リオーダーバッファ | 16エントリ |
| ロードストアキュー | 8エントリ |
| 整数ALU | 2 |
| 整数乗算器 | 1 |
| D$ポート | 1 |
| 浮動小数点ALU | 2 |
| 浮動小数点乗算器 | 1 |

**表2** 値予測機構の構成

| | 値履歴表 | パターン履歴表 |
|---|---|---|
| 最終値予測 | 4096エントリ | |
| ストライド値予測 | 4096エントリ | |
| コンテキスト・ベース値予測 | 4096エントリ | 4096エントリ |
| ハイブリッド値予測 | 4096エントリ | 4096エントリ |

それぞれ4と6を用いる．値予測は命令単位で行っている．予測に失敗した場合のペナルティは，次の投機流スレッドの起動が現在の検証流の完了まで遅れるとする．

供給電圧とクロック周波数には，サムソン社のARMプロセッサの値，0.7V時に600MHzと1.1V時に1.2GHz[3]を用いる．リーク電流と値予測機構が消費する電力は考慮していない．

以上の仮定の下で，各値予測機構の値予測精度，プログラム実行サイクル数，消費電力，エネルギー消費量を調査する．

予測精度以外の評価項目についてはコントレイルプロセッサを構成するプロセッサコア1台の値を基準としている．シミュレーション結果は，全ベンチマークプログラムの平均結果を表す．

### 4.2 評価結果

#### 4.2.1 値予測精度

図6に値予測精度の結果を示す．横軸はベンチマークプログラムと値予測機構を表す．縦軸はVHT総参照回数と予測が成功した命令の割合を表している．図中は下から，予測が成功した割合(correct)，予測が失敗した割合(incorrect)，予測対象命令中の予測しない命令の割合(non-speculated)，VHTにヒットしなかった割合(VHT miss)を表す．

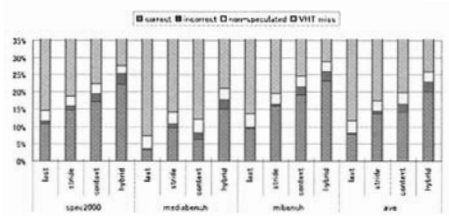


**図6** 値予測精度

各値予測機構の予測率について述べる．予測率は

VHT 総参照回数に対して値予測を行った命令数の割合である．図中では correct と miss-predicited の和である．より多くの命令を予測する方が，投機実行が増えるのでエネルギー削減効果が高くなると期待される．そのため予測率は高い方が好ましい．各ベンチマークの平均の結果では，最も高いものはハイブリッド値予測機構で 23%，最も低いものは最終値予測機構で 8%であった．ハイブリッド値予測機構はストライド値予測機構とコンテキスト・ベース値予測機構の二つの値予測機構の組み合わせで，様々なプログラムに対応できるため予測率が高くなっている．

続いて各値予測機構の予測精度について述べる．予測精度とは値予測の成功率を示す．図中で correct と incorrect の和に対する correct の割合である．予測ミスが発生するとサイクルペナルティを被ってしまい，実行時間が増加する．また，投機実行から予測ミス判明までの実行結果が破棄されてしまい，無駄な実行を行ってしまうため，エネルギー消費量が増加してしまう．そのため予測精度が高い値予測機構が好ましい．各ベンチマークの平均の結果では，最も高いもので最終値予測機構の 93%，最も低いものでコンテキスト・ベース値予測機構の 88%であった．コンテキスト・ベース値予測機構では積極的に値予測を行っているため，予測ミス回数が増加したと思われる．コンテキスト・ベース値予測機構の予測ミス回数が多いため，ハイブリッド値予測機構でも予測ミス回数が多くなっている．

#### 4.2.2 プログラム実行サイクル数

図 7 にプログラム実行サイクル数の結果を示す．縦軸はプロセッサコア 1 台での実行を基準とし，コントレイルプロセッサのプログラム実行サイクル数の割合を表す．横軸はベンチマークプログラムと値予測機構を表す． 平均で最終値予測機構では 12%，ストライド値予測機構では 18%，コンテキスト・ベース値予測機構では 14%，ハイブリッド値予測機構では 14%プログラム実行サイクル数を削減できた．値予測精度とプログラム実行サイクル数を比較する．ハイブリッド値予測機構では予測率，予測成功数は他の値予測機構より高いが，プログラム実行サイクル数は他の値予測機構に比べあまり削減されていない．これは，予測ミス命令数が他の値予測機構に比べ多いためである．予測ミスが判明すると，予測ミス時以降の投機流と検証流が破棄されてしまう．このとき，予測ミスを発見した検証流の実行サイクル数が予測ミスペナルティとなる．この予測ミスペナルティがプログラム実行サイクル数削減の妨げとなっている．最終値予測機構では他の値予測機構に比べ予測精度が最も高いが，プログラム実行サイクル数削減は他の値予測機構に比べ最も小さい．これは予測率が他の値予測機構に比べ低いためである．予測率が低いと，多くの命令列を投機実行できず，プログラム実行サイクル数は削減されない．予測精度が高いが予測率が低い場合，プログラム実行サイクル数は削減されない．予測率が高いが予測精度が低い場合，予測ミスのサイクルペナルティを被ってしまい，プログラム実行サイクル数は削減されない．予測精度と予測率の両方が良いことが，プログラム実行サイクル数の削減に要求される．予測率と予測精度のバランスが最も良かったストライド値予測機構が，最もプログラム実行サイクル数を削減できたと言える．

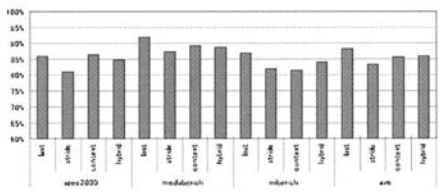

**図 7** プログラム実行サイクル数

#### 4.2.3 消費電力

図 7 に消費電力の結果を示す．縦軸はプロセッサコア 1 台での実行を基準とし，コントレイルプロセッサの消費電力の割合を表す．横軸はベンチマークプログラムと値予測機構を表す． 平均で最終値予測では 8%，ストライド値予測では 13%，コンテキスト・ベース値予測では 12%，ハイブリッド値予測では 19%消費電力が増加した．値予測精度，プログラム実行サイクル数と消費電力を比較する．ハイブリッド値予測機構は最も消費電力が高い．値予測精度を見るとハイブリッド値予測機構は予測成功数が高い．このことより，ハイブリッド値予測では投機実行が増え，並列に実行される検証流が増えたためであると思われる．同時に最大 4 台のプロセッサコアが動作するため，必然的に消費電力は増加する．また消費電力の増加は，投機実行の増加とみなすことができ，並列性の増加を意味する．並列性が増えればプログラム実行サイクル数を削減でき，エネルギー消費量を削減することができる．このとき，検証流を実行するコアは低速・低電力で動作しているため，消費電力の増加を最小限に抑えている．

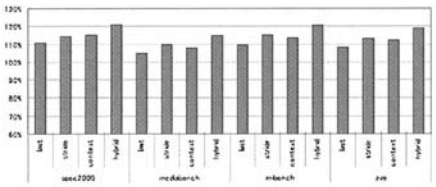

**図 8** 消費電力

#### 4.2.4 エネルギー消費量

図 9 に消費エネルギーの結果を示す．縦軸はプロセッサコア 1 台での実行を基準とし，コントレイルプロセッサの消費エネルギーの割合を表す．横軸はベンチマークプログラムと値予測機構を表す． 平均で最終値予測では 5%，ストライド値予測では 6%，コンテ

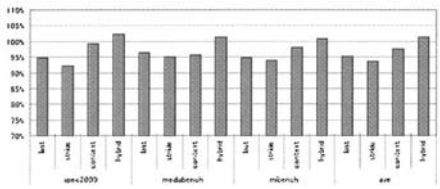

図 9 エネルギー消費量

キスト・ベース値予測では 2%エネルギー消費量を削減でき, ハイブリッド値予測では 1%エネルギー消費量が増加した. ハイブリッド値予測機構におけるエネルギー消費量増加は, 予測ミスによるサイクルミスペナルティのためであると思われる. エネルギー消費量は消費電力とプログラム実行サイクル数との積で表される. ハイブリッド値予測機構では, 予測ミス回数が多いためプログラム実行サイクル数が予測成功回数に比べ削減されていない. そのため, ハイブリッド値予測機構ではエネルギー消費量が増加したと思われる. 最もエネルギー消費量を削減できたのは, ストライド値予測機構であった. ストライド値予測機構は, 最もプログラム実行サイクル数を削減している. さらに, 消費電力も最小限に抑えているので, 最もエネルギー消費量を削減できたと思われる. 予測率や予測成功回数が多くても、予測ミス回数が多ければ, プログラム実行サイクル数が削減されず, 結果としてエネルギー消費量もあまり削減されない. プログラム実行サイクル数の削減がエネルギー消費量削減に大きな影響を及ぼすことがわかる.

以上の評価結果より結論を述べる. エネルギー消費量削減効果は 4 つの値予測機構の中でストライド値予測機構が最も高くなっている. 今回の評価では値予測機構の消費電力を考慮していないが, 複雑な予測器はその消費電力が無視できない. 3 節で述べた値予測機構のハードウェア量を見ると, ストライド値予測機構は 1 エントリ当たり 84 ビットと最終値予測の次に少ない. そのため, ストライド値予測が最も好ましい.

## 5. ま と め

携帯情報端末用プロセッサでは, 高性能かつ省エネルギーなプロセッサが要求されており, 近年需要が急速に増えている. この要求に対して, 我々は省エネルギー指向マルチコアプロセッサであるコントレイルプロセッサを提案している. これまで最終値予測機構を用いてコントレイルプロセッサの評価を行ってきた. 本稿では, 未評価であった値予測機構のコントレイルプロセッサへの影響について調査した. シミュレーションの結果、エネルギー消費量で最終値予測で 5%, ストライド値予測で 6%, コンテキスト・ベース値予測で 2%エネルギー消費量を削減でき, ハイブリッド値予測では 1%エネルギー消費量が増加した. エネルギー消費量削減効果とハードウェア量を考慮すると, 今回評価した予測機構の中では, コントレイルプロセッサにはストライド値予測が最も好ましいことがわかった.

**謝　　辞**

本研究の一部は, 文部科学省科学研究費補助金 (No. 16300019, No.176549) の援助によるものです.

## 参 考 文 献

1) L. Codrescu, D. S. Wills, On Dynamic Speculative Thread Partitioning and the MEM-Slicing Algorithm, International Conference on Parallel Architectures and Compilation Techniques, October 1999.
2) E. Larson, S, Chatterjee, T. Austin, MASE: A Novel Infrastructure for Detailed Microarchitectural Modeling, International Symposium on Performance Analysis of System and Software, November 2001.
3) M. Levy, Samsung Twists ARM Past 1GHz, Microprocessor Report, 16(10), October 2002.
4) T. Sato, I. Arita, Contrail Processors for Converting High-Performance into Energy-Efficiency, 10th International Conference on Parallel Architectures and Compilation Techniques, Work-in-progress session, September 2001.
5) T. Sato, Y. Tanaka, H. Sato, T. Funaki, T. Koushiro, A Chiyonobu, Improving Energy Efficiency via Speculative Multithreading on MultiCore Processors, 16th International Workshop on Power and Timing Modeling, Optimization and Simulation, September 2006.
6) S. Torii, S. Suzuki, H. Tomonaga, T. Tokue, J. Sakai, N. Suzuki, K. Murakami, T. Hiraga, K. Shigemoto, Y. Tatebe, E. Obuchi, N. Kayama, E. Edahiro, T. Kusano, N. Nishi, A 600MIPS 120mW 70A Leakage Triple-CPU Mobile Application Processor Chip, International Solid State Circuits Conference, February 2005.
7) K. Wang, M. Franklin, Highly Accurate Data Value Prediction using Hybrid Predictors, 30th International Symposium on Microarchitecture, December 1997.
8) 亀井達也, SH-X3:低消費電力マルチプロセッサシステム向け SuperH 強化コア, フォールマイクロプロセッサフォーラムジャパン, November 2006.
9) 舟木敏正, 千代延昭宏, 佐藤寿倫, 省エネルギーな投機的マルチスレッド・マルチコアプロセッサの評価, 先進的計算基盤システムシンポジウム, May 2006.