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

2007－ARC－172 (8)
2007－HPC－109 (8)
2007／3／1

# 投機的実行の深さに着目した低消費電力化手法

小林 良太郎† 志村 一樹† 島田 俊夫†

本稿では，投機的実行の深さに着目して，消費電力を削減する手法を提案する．投機的実行の深さとは，プロセッサ中に存在する，予測結果の判明していない分岐命令の数である．提案機構では，与えられた負荷に応じて，投機的実行の深さを制御する．これにより，目標とするスループットを満足しつつ，消費電力を削減することができる．評価した結果，提案手法は，目標とするスループットを60%～90% の範囲で変化させた場合，目標とするスループットとの誤差を平均で 1.0% ポイント以下に抑制しつつ，消費電力を 41.0%～10.8% の範囲で削減できることが分かった．

# Power Consumption Reduction Scheme Focusing on the Depth of Speculative Execution

Ryotaro Kobayashi,† Kazuki Shimura† and Toshio Shimada†

In this paper, we propose a mechanism that reduces power consumption focusing on the Speculative Execution Depth (SED). SED is the number of branch instructions whose results are not yet known in the processor. Our mechanism controls the SED depending on the current workload. The evaluation results show that the proposed scheme can reduce power consumption by 41.0-10.8% when the target throughput ranges from 60-90%, suppressing the average error margin with the target throughput to 1.0% or less.

## 1. はじめに

近年のマイクロプロセッサは，電力密度が増大し，電力によって発生する熱が無視できないレベルに達している．そのため，モバイル・プロセッサだけでなく，従来の高性能プロセッサの設計においても，消費電力の削減が非常に重要な要素の１つとなってきている[3]．

消費電力を削減するアプローチの１つとして，動作周波数と電源電圧を動的に制御する DVFS (Dynamic Voltage and Frequency Scaling) と呼ばれる手法が用いられている．DVFS を導入したプロセッサにおいては，与えられた負荷に応じて，プロセッサの動作周波数と電源電圧を低下させることで，消費電力を削減することができる．しかし，プロセスの微細化が進行するにつれ，閾値電圧のスケーリングの速度はより緩やかになっていくと考えられるため，閾値電圧によって下限が決まる電源電圧は，増減の余地が失われていく．したがって，今後，DVFS の効果は減少していくといえる．

そこで我々は，動作周波数や電源電圧を制御するのではなく，アーキテクチャレベルで，目標とするスループットを満足しつつ，消費電力を抑制する方策を考える．具体的には，投機的実行に着目し，プロセッサ中に存在する，予測結果の判明していない分岐命令の数（投機的実行の深さ）を制御する．投機的実行の深さを減少させると，プロセッサ内で，単位時間あたりに処理される命令の数が減少する．これにより，各ノードの平均スイッチング確率が減り，消費電力が減少する．

2 章では投機的実行の深さと消費電力の関係について説明する．3 章では投機的実行の深さを制御する手法を提案する．4 章では提案機構の評価を行う．5 章で関連研究について述べ，6 章でまとめる．

## 2. 投機的実行の深さと消費電力

近年の高性能プロセッサは命令レベルの並列性を引き出し性能を向上させている．分岐命令は，命令レベル並列性を利用するプロセッサの性能を厳しく制限する．この影響を緩和するため，投機的実行と呼ばれる手法が一般的に用いられている．この手法では，分岐命令を実行する前に，その分岐方向を予測し，後続命令を投機的にプロセッサのパイプラインで処理する．分岐予測が正しければ，分岐命令が性能に与える影響を緩和し，プロセッサのスループットを向上させることができる．誤っていた場合，分岐命令の悪影響を緩和できない．また，誤った分岐予測に基づいてフェッ

† 名古屋大学大学院工学研究科
Graduate School of Engineering, Nagoya University

チされた命令（誤ったパス上の命令）を消去し，そらの命令によって変更されたプロセッサ状態を元に戻した後，正しい分岐結果にしたがって，パイプライン処理を再開する必要がある．つまり，投機的実行を導入すると，スループットが向上すると同時に，誤ったパス上の命令をパイプライン処理するようになる．

投機的実行の持つこれらの特徴は，どちらも消費電力を増加させる．一般に，消費電力 $P$ は以下の式で表される：

$$P = \alpha \cdot f \cdot C \cdot V^2$$

上記の式において，$\alpha$，$f$，$C$，$V^2$ は，それぞれ，ノードのスイッチング確率の平均，動作周波数，ノードの全容量，電源電圧である．投機的実行を導入すると，スループットが向上するだけでなく，誤ったパス上の命令がパイプライン処理されるようになる．これらはいずれも，$\alpha$ を増加させるため，プロセッサの消費電力は増加する．

ここで，予測結果の判明していない分岐命令の数（投機的実行の深さ）に着目する．投機的実行の深さは実行時に動的に変化する値である．投機的実行の深さが大きい程，より多くの投機を行うと考えることができる．より多くの投機を行えば，プロセッサのスループットは向上し，誤ったパス上の命令がより多くパイプライン処理されるようになる．その結果，各ノードのスイッチング確率が高くなり，消費電力が増加する．なお，これ以降の章では，投機的実行の深さを，SED（Speculative Execution Depth）と略記する．

## 3. 投機的実行の深さを制御する手法

前章で述べた投機的実行の性質から，我々は，プロセッサに要求されるスループット（目標スループット）に応じてSEDを制御することにより，アーキテクチャレベルで消費電力を削減する手法を提案する．なお，目標スループットはOSから与えられるものとする．

### 3.1 概　　要

まず，SEDを制御する手法として，目標スループットに応じて，SEDの上限（$D_{limit}$）を静的に一意に決めるという手法を提案する．

しかし，$D_{limit}$ に対応するスループットと消費電力は離散的なものとなるため，この手法では，目標とするスループットと実現可能なスループットとの差（スループットの誤差）が大きくなる可能性がある．

そこで，2つ目の制御手法として，$D_{limit}$ を動的に決定する手法を提案する．この手法では，目標とするスループットと現在のスループットとの差分を見ながら，$D_{limit}$ を動的に変更する．これにより，目標スループットとの差を小さくすることができる．なお，$D_{limit}$ の最小値は0，最大値はスループットが最大となる値とする．

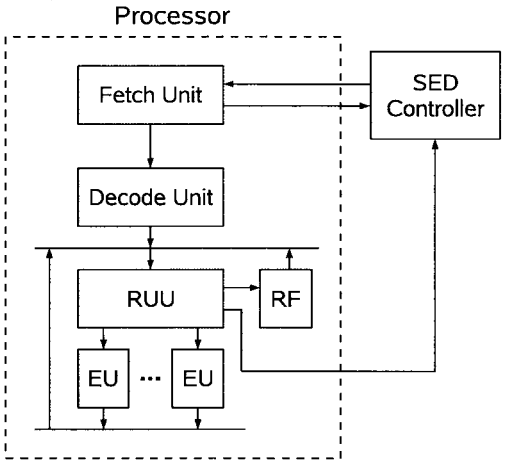

図1　提案機構の全体図

### 3.2 提案手法を実現するハードウェア

**図1**に，提案手法のハードウェア構成を示す．図において，点線で囲われた部分はプロセッサを示す．プロセッサの内部は簡略化して示す．図において，RUUはレジスタ・アップデート・ユニット，RFはレジスタ・ファイル，EUは実行ユニットを意味する．プロセッサの右側は，SEDを制御する機構を示す．

まず，SED制御機構を用いて，SEDを，静的，あるいは，動的に定められた $D_{limit}$ 以下になるように制御する方法を示す．

プロセッサ中に存在する分岐命令数を保持するカウンタを用意し，カウンタの値に，フェッチした分岐命令数を加え，カウンタの値から，コミットした分岐命令数を引く．

カウンタの保持する分岐命令数が，$D_{limit}$ 以下の場合，SEDが $D_{limit}$ を超えることはないので，通常のプロセッサと同様にして命令をフェッチすることを許可する．そうでない場合，SEDが $D_{limit}$ を超える可能性があるため，フェッチ・ユニットをストールさせ，新たな命令フェッチを停止する．以上により，投機的実行の深さの上限が，$D_{limit}$ を超えることは無くなる．

次に，$D_{limit}$ を動的に定める方法を示す．

サンプリング間隔を $SP$ サイクルとする．そして，$SP$ サイクル毎に，目標とするスループット $TP_{target}$ と現在のスループット $TP_{current}$ を比較する．その結果，$TP_{target}$ よりも $TP_{current}$ が大きければ $D_{limit}$ を減少させ，そうでなければ $D_{limit}$ を増加させる．$D_{limit}$ の最小値は0，最大値はスループットが最大となる値とする．

$TP_{target}$ と $TP_{current}$ を比較する回路について述べ

る．値の比較は減算器があれば実現できる．$TP_{target}$ は定数として与えることができる．$TP_{current}$ は，比較の度に計算する必要があり，以下の式で表される：

$$TP_{current} = IPC_{current} \cdot f = \frac{I_{current}}{C_{current}} \cdot f$$

ここで，$IPC_{current}$ は現在までの IPC，$f$ は動作周波数，$I_{current}$ は現在までにコミットした命令数の合計，$C_{current}$ は現在のサイクル数である．$I_{current}$ は，命令のコミット時に，コミットした命令数を足し合わせていくことで，$C_{current}$ はサイクルをカウントすることで得られる．$f$ は定数である．したがって，$TP_{current}$ を求めるためには，加算器の他に，乗除算器が必要となる．

しかし，実際には，$TP_{target}$ と $TP_{current}$ の大小関係が分かれば良いので，$TP_{current}$ を直接求める必要はなく，減算器と加算器でスループット比較回路を実現できる．以下においてその理由を説明する．

値の比較は，減算した結果が正（上位ビットが 0）かどうかを調べることで実現できる．そこでまず，以下のようにして，$TP_{target}$ と $TP_{current}$ の減算式を変形する．

$$\begin{aligned} TP_{target} - TP_{current} &= IPC_{target} \cdot f - \frac{I_{current}}{C_{current}} \cdot f \\ &\propto IPC_{target} - \frac{I_{current}}{C_{current}} \\ &\propto IPC_{target} \cdot C_{current} - I_{current} \\ &= I_{target} - I_{current} \end{aligned}$$

ここで，$IPC_{target}$ は，$TP_{target}$ を $f$ で割って得られる IPC（目標とする IPC）であり，$I_{target}$ は，$IPC_{target}$ と $C_{current}$ の積で表される命令数（目標とするコミット命令数）である．この式より，スループットの比較では，$I_{target}$ と $I_{current}$ を比較すればよいことが分かる．

$I_{current}$ は加算によって得られる値である．一方，$I_{target}$ は定数である $IPC_{target}$ と，サイクル毎にインクリメントされる $C_{current}$ の乗算によって得られる値である．そこでつぎに，以下のようにして，$I_{target}$ の計算式を変形する．

$$\begin{aligned} I_{target} &= IPC_{target} \cdot C_{current} \\ &= IPC_{target} \cdot SP \cdot n \\ &= IPC_{target} \cdot SP \cdot (n-1) + IPC_{target} \cdot SP \end{aligned}$$

ここで，$n$ は現在までのサンプリング回数を示す．この式より，$I_{target}$ は，サンプリング毎に，$IPC_{target}$ と $SP$ の積（定数）を足し合わせていくことで得られることが分かる．

以上より，スループット比較回路は，減算器と加算器で実現できることが分かる．

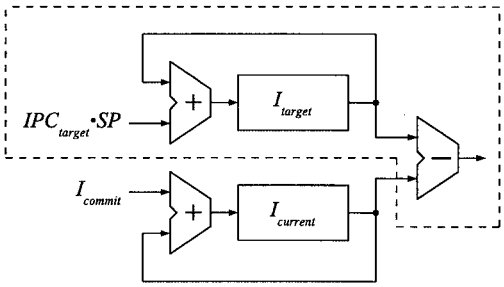


**図 2** スループット比較回路

**表 1** 測定条件

| | |
|---|---|
| フェッチ幅 | 4 命令 |
| 発行幅 | 4 命令 |
| RUU | 64 エントリ |
| 機能ユニット数 | iALU 4, iMULT/DIV 2, fpALU 3, fpMULT/DIV/SQRT 2 |
| 命令キャッシュ | 8KB, 2 ウェイ, 32B ライン, ミスペナルティ 6 サイクル |
| データキャッシュ | 32KB, 2 ウェイ, 32B ライン, 2 ポート, ミスペナルティ 6 サイクル |
| 2 次キャッシュ | 統合, 2MB, 4 ウェイ, 64B ライン, ミスペナルティ 100 サイクル |
| 分岐予測機構 | gshare 6 ビット履歴, 8K エントリ PHT, 予測ミスペナルティ 5 サイクル |

**図 2** に，スループット比較回路を示す．点線で囲われた部分では，$SP$ サイクル毎に，$I_{target}$ と $I_{current}$ の比較を行い，$I_{target}$ を更新する．具体的には，減算器を用いて $I_{target}$ から $I_{current}$ を減算し，減算結果の符号ビットが 0 であれば $I_{target}$ の方が大きく，そうでなければ $I_{current}$ の方が大きいと判定する．また，加算器を用いて $IPC_{target}$ と $SP$ の積（定数）を足し合わせていくことで，$I_{target}$ を更新する．一方，点線で囲われていない部分では，命令のコミット時に，コミットした命令数を足し合わせていくことで，$I_{current}$ を更新する．

## 4. 評　　価

まず，評価環境について述べる．次に，$D_{limit}$ が性能と電力に与える影響について示す．その後，$TP_{target}$ に応じて，$D_{limit}$ を静的，動的に定めた場合の評価結果を示す．

### 4.1 評価環境

シミュレータには，アーキテクチャレベルで消費電力を評価できる Wattch[2] を用い，提案手法を組み込んで評価した．命令セットには MIPS R10000 を拡張した SimpleScalar/PISA[1] を用いた．ベンチマー

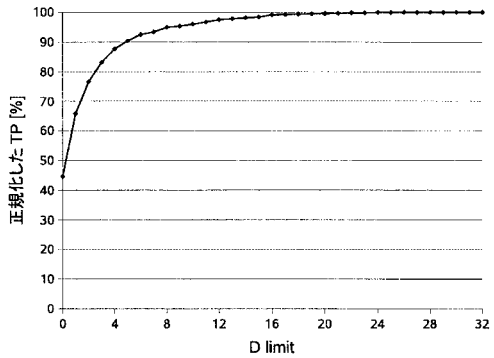


図 3　$D_{limit}$ と TP の関係

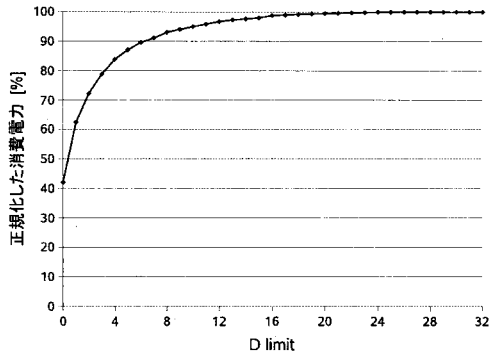


図 4　$D_{limit}$ と消費電力の関係

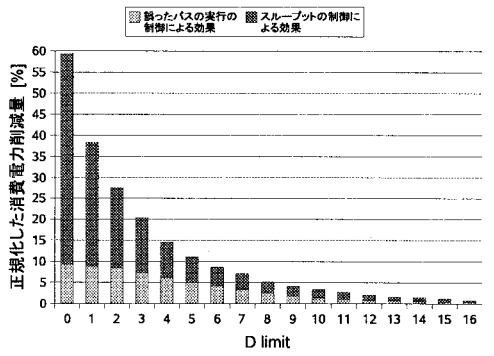


図 5　削減した消費電力の内訳

ク・プログラムは，SPECint2000 の gzip，vpr，gcc，mcf，parser，perlbmk，vortex，bzip2 の 8 本を使用し，20 億命令をスキップした後，1000 万命令を実行した．測定条件として**表 1** を用いた．

### 4.2　$D_{limit}$ が性能と電力に与える影響

この節では，$D_{limit}$ が性能，あるいは，電力に与える影響に着目するため，$D_{limit}$ を連続的に変化させ，

表 2　静的に定めた $D_{limit}$

| ベンチマーク | $TP_{taryet}$ 40% | 50% | 60% | 70% | 80% | 90% |
|---|---|---|---|---|---|---|
| gzip | 0 | 0 | 1 | 1 | 2 | 3 |
| vpr | 0 | 0 | 1 | 2 | 3 | 4 |
| gcc | 0* | 0 | 0 | 0 | 1 | 1 |
| mcf | 1 | 2 | 5 | 8 | 11 | 15 |
| pars. | 0 | 1 | 1 | 2 | 3 | 4 |
| perl. | 0* | 0* | 0 | 0 | 1 | 1 |
| vort. | 0* | 0 | 0 | 1 | 1 | 3 |
| bzip2 | 0 | 1 | 1 | 2 | 3 | 5 |

各 $D_{limit}$ ごとにベンチマーク・プログラムを実行して評価を行う．

なお，$TP_{target}$ に応じて $D_{limit}$ を静的に定める場合については，次節で評価する．

**図 3** に，各 $D_{limit}$ 毎のスループット（TP）をベンチマーク平均で示す．縦軸は，SED を制御しない場合で正規化した TP を示し，横軸は $D_{limit}$ を示す．提案機構では，SED を制御しても周波数は変化しないので，正規化した TP は，正規化した IPC と等しくなる．$D_{limit}$ が 0 の場合は，投機的実行を全く行わないことを意味する．

図より，TP は，$D_{limit}$ を増加させるにつれて増加し，$D_{limit}$ が 16 で，ほぼ上限に達することが分かる．また，$D_{limit}$ を制御することによって，TP を 44.6% から 100% の範囲で制御できることが分かる．

**図 4** に，各 $D_{limit}$ 毎の消費電力をベンチマーク平均で示す．縦軸は，SED を制御しない場合で正規化した消費電力を示し，横軸は $D_{limit}$ を示す．

図より，$D_{limit}$ を変化させた場合，消費電力は，TP と同様に変化し，消費電力の削減率は，IPC の削減率に近いことが分かる．また，$D_{limit}$ を制御することによって，消費電力を 42.0% から 100% の範囲で制御できることが分かる．

**図 5** に，消費電力の削減量の内訳を示す．縦軸は，SED を制御しない場合で正規化した消費電力の削減量を示し，横軸は $D_{limit}$ を示す．棒グラフは 2 つの部分からなり，上部が，TP を制御したことによる削減量を示し，下部が，誤ったパスの実行を制御したことによる削減量を示す．

図より，誤ったパスの実行を制御する効果よりも，TP を制御する効果の方が大きいことが分かる．また，TP を制御する効果は $D_{limit}$ が小さいほど大きくなり，$D_{limit}$ が 0 の場合，誤ったパスの実行を制御して得られる削減量が 9.3% であるのに対し，TP を抑制して得られる削減量は 50.2% にもなる．

### 4.3　静的な $D_{limit}$ を用いた SED の制御

$TP_{target}$ に応じて $D_{limit}$ を静的に一意に定め，SED

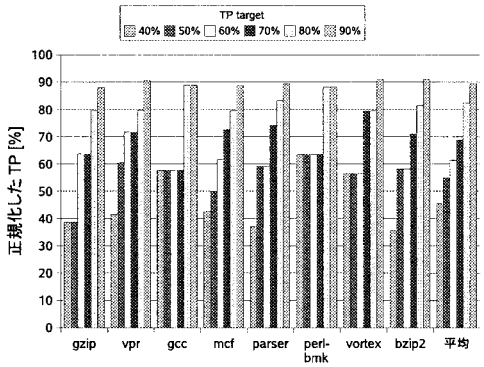


図 6　TP（静的 $D_{limit}$）

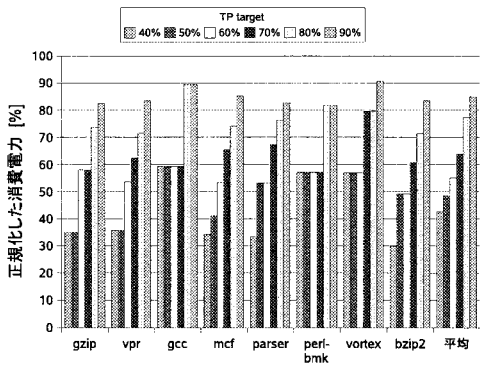


図 7　消費電力（静的 $D_{limit}$）

を制御した場合の評価結果を示す．評価において，$TP_{target}$ は，6 通り（40%, 50%, 60%, 70%, 80%, 90%）とする．$D_{limit}$ は，$TP_{target}$ と TP との差（TP の誤差）が最も小さくなるように静的に定める．**表 2** に，各 $TP_{target}$ に対し，ベンチマーク毎に定めた静的な $D_{limit}$ を示す．表において，「*」を付した数字は，$D_{limit}$ が下限（0）に達したために TP も下限に達し，$TP_{target}$ との差が 10% ポイント以上になることを示す．

**図 6** と**図 7** に，それぞれ，静的な $D_{limit}$ を用いた場合の TP と消費電力を示す．図 6 の縦軸は，SED を制御しない場合で正規化した TP を示す．一方，図 7 の縦軸は，SED を制御しない場合で正規化した消費電力を示す．いずれにおいても，6 本で組になった棒グラフは，左から順に，$TP_{target}$ が，40%, 50%, 60%, 70%, 80%, 90% の場合である．

図より，どのベンチマークにおいても，TP に応じて，消費電力を削減できることが分かる．しかし，TP の誤差が大きいことが分かる．図 6 における TP の誤差の平均値と最大値を，**表 3** に示す．

**表 3**　静的 $D_{limit}$ を用いた場合の TP の誤差（%ポイント）

| | $TP_{target}$ | | | | | |
|---|---|---|---|---|---|---|
| | 40% | 50% | 60% | 70% | 80% | 90% |
| 平均 | 8.8 | 8.3 | 3.6 | 5.5 | 2.9 | 1.2 |
| 最大 | 23.5 | 13.5 | 11.6 | 12.4 | 8.8 | 2.0 |

この表からも分かるように，$TP_{target}$ が小さくなる程，TP の誤差が大きくなる傾向にあることが分かる．この理由は 2 つある．第 1 の理由は，表 2 に示されているように，一部のベンチマークにおいて，$TP_{target}$ が 50% 以下になると，制御可能な TP の下限との差が 10% ポイント以上になってしまうことにある．第 2 の理由は，TP が離散的な値となることに加え，図 3 に示されているように，TP の変化量は，$D_{limit}$ が小さい程，大きくなることにある．

### 4.4　動的な $D_{limit}$ を用いた SED の制御

この節では，$TP_{target}$ に応じて $D_{limit}$ を動的に定め，SED を制御した場合の評価結果を示す．

まず，適切な $SP$ の値を知るため，$SP$ が TP の誤差に与える影響を調べる．

**表 4** に，$SP$ を 100〜1M サイクルの間で変化させて，TP の誤差を測定した結果を，平均値と最大値に分けて示す．

表より，$TP_{target}$ が 60% 以上の場合，$SP$ が 10K サイクルあれば，平均誤差，最大誤差を，それぞれ，1.0% ポイント以下，4.8% ポイント以下に抑制できることが分かる．

一方，$TP_{target}$ が 50% 以下の場合，最大誤差は $SP$ にほとんど依存することなく，10% ポイントを超える値となってしまう．この原因は，表 2 に示されているように，一部のベンチマークでは，$TP_{target}$ が 50% 以下になると，制御可能な TP の下限との差が 10% ポイント以上になってしまうことにある．これら一部のベンチマークを除外して考えると，$TP_{target}$ が 50% 以下の場合も，$SP$ が 10K サイクルあれば，TP の誤差を十分抑制することができる．

そこで，これ以降の評価では，$SP$ を 10K サイクルに固定する．

**図 8** と**図 9** に，それぞれ，動的な $D_{limit}$ を用いた場合の TP と消費電力を示す．図 6 の縦軸は，SED を制御しない場合で正規化した TP を示す．一方，図 7 の縦軸は，SED を制御しない場合で正規化した消費電力を示す．いずれにおいても，6 本で組になった棒グラフは，左から順に，$TP_{target}$ が，40%, 50%, 60%, 70%, 80%, 90% の場合である．

図より，どのベンチマークにおいても，TP に応じて，

表 4 動的 $D_{limit}$ を用いた場合の TP の誤差（%ポイント）

(a) 平均誤差

| $SP$ | $TP_{target}$ | | | | | |
|---|---|---|---|---|---|---|
| (cycle) | 40% | 50% | 60% | 70% | 80% | 90% |
| 100 | 8.3 | 3.8 | 0.9 | 0.1 | 0.3 | 0.4 |
| 1K | 8.3 | 3.9 | 0.9 | 0.1 | 0.2 | 0.4 |
| 10K | 8.8 | 4.2 | 1.0 | 0.1 | 0.3 | 0.3 |
| 100K | 10.5 | 7.3 | 5.0 | 2.4 | 4.1 | 2.7 |
| 1M | 9.4 | 6.4 | 5.3 | 5.6 | 2.6 | 6.0 |

(b) 最大誤差

| $SP$ | $TP_{target}$ | | | | | |
|---|---|---|---|---|---|---|
| (cycle) | 40% | 50% | 60% | 70% | 80% | 90% |
| 100 | 23.9 | 14.1 | 4.4 | 0.4 | 0.9 | 1.8 |
| 1K | 23.9 | 14.1 | 4.4 | 0.4 | 0.9 | 1.8 |
| 10K | 24.3 | 14.5 | 4.8 | 0.4 | 1.0 | 1.9 |
| 100K | 24.5 | 14.8 | 16.1 | 14.7 | 17.5 | 7.9 |
| 1M | 24.5 | 14.5 | 18.9 | 12.8 | 8.0 | 12.0 |

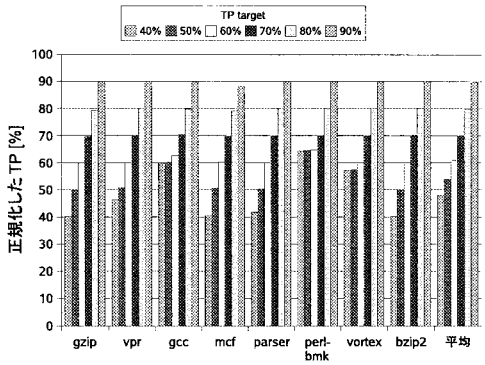


図 8 TP（動的 $D_{limit}$）

消費電力を削減できることが分かる．また，$TP_{target}$ が 60%～90% の場合，TP の誤差を平均で 1.0% ポイント以下に抑制しつつ，消費電力を 41.0%～10.8% の範囲で削減できることが分かる．なお，一部の場合において TP の誤差が大きくなる原因は，先程述べたように，$TP_{target}$ が，制御可能な TP の下限を大きく下回ることにある．

また，図 6 を図 8 と比較すると，3 章で述べたとおり，動的な $D_{limit}$ を使用する方が，TP の誤差を小さくできることが分かる．

## 5. 関連研究

DVS，PSU[5]，VSP[6] では，消費電力を削減するために，周波数を制御する．一方，我々の機構では，消費電力を削減するために SED を制御する点が異なる．

Manne らは，成功する可能性の低い投機的実行を抑制することで，無駄に消費されるエネルギーを削減する手法を提案している[4]．本論文とは，誤ったパス上の命令の実行を削減するという点が似ている．しかし，我々の機構とは異なり，与えられた負荷に応じてスループットを制御し，消費電力を制御することはできない．

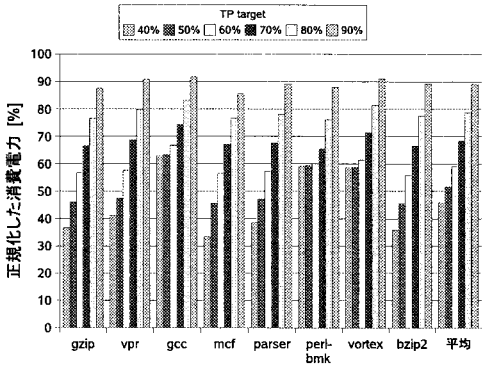


図 9 消費電力（動的 $D_{limit}$）

## 6. まとめ

今回我々は，投機的実行の深さに着目して，消費電力を削減する手法を提案した．提案機構では，与えられた負荷に応じて，投機的実行の深さを制御することができるため，目標とするスループットを満足しつつ，消費電力を削減することができる．

**謝辞** 本研究の一部は，株式会社半導体理工学センターとの共同研究により行った．

## 参考文献

[1] D. Burger, et al., "The Simplescalar Tool Set Version 2.0," Technical Report 1342, Department of Computer Sciences, University of Wisconsin-Madison, June 1997.

[2] D. Brooks, et al., "Wattch: a framework for architectural-level power analysis and optimizations," In *Proc. ISCA-27*, June 1999.

[3] F. Pollack, "New Microarchitecture Challenges in the Coming Generations of CMOS Process Technologies," Keynote Presentation to MICRO-32, Nov. 1999.

[4] S. Manne, et al., "Pipeline Gating: Speculative Control For Energy Reduction," In *Proc. ISCA-25*, June 1998.

[5] 嶋田ほか，"低消費電力化のための可変パイプライン，" 情報処理学会研究報告，Vol. 2001-ARC-145, 2001.

[6] 市川ほか，"可変パイプラインを用いた低消費エネルギープロセッサの設計と評価，" 情報処理学会論文誌，Vol.47, SIG 7 (ACS 14), 2006.