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

2007－ARC－172 (7)
2007－HPC－109 (7)
2007／3／1

パイプラインステージ統合のオンチップ制御機構

間所峻洋† 小林良太郎† 島田俊夫†

本論文では，パイプラインステージ統合 (PSU: Pipeline Stage Unification) を用いて消費電力を削減するための，2つのオンチップ制御機構を提案する．第1の機構は，与えられた目標性能を達成しつつ消費電力を削減するために，PI 制御を利用する．第2の機構は，PSU における性能と消費電力の特性に着目し，パイプラインステージ統合状態のフィードバックを加える．以上2つの PSU 制御機構を評価した結果，実際のプロセッサで使用されている DVFS に比べて，それぞれ，最大 11.5%，最大 15.6%消費電力を削減できることが示せた．

On-Chip Control Mechanisms for Pipeline Stage Unification

Takahiro Madokoro,† Ryotaro Kobayashi† and Toshio Shimada†

In this paper, we propose two on-chip control mechanisms for Pipeline Stage Unification (PSU) which reduce power consumption. The first mechanism uses PI control to reduce power consumption satisfying the performance target. The second mechanism uses feedback of the state of pipeline unification. The evaluation results show that our mechanisms reduce power consumption by 11.5% and 15.6% compared to DVFS.

1. はじめに

近年，プロセッサには，低消費電力と高性能の両立が求められている．この要求を満たすために，DVFS(Dynamic Voltage and Frequency Scaling) と呼ばれる手法が用いられている．DVFS では，与えられた負荷が低い場合にクロック周波数を低下させ，それによって延長されたクロックサイクル時間に信号の遅延を合わせて電源電圧を低下させることで，消費電力を削減する．しかしながら，プロセス技術の進歩による閾値電圧やリーク電流の制約のために，電源電圧の低下が制限されるため，今後，DVFS の効果は減少していくと言える．

これに対して，パイプラインステージ統合 (PSU: Pipeline Stage Unification)[1] は，デバイス技術に依存しないため，その効果が持続するという利点がある．しかしながら，PSU はパイプラインステージ数の変更により消費電力を削減するため，それだけでは離散的な性能しか実現できない．したがって，PSU で任意の性能を実現するためには動的制御機構が必要である．これまでは，制御のために SMT を使用してきたが[2]，本論文では，PSU の制御をオンチップで実現する2種類の機構を提案する．1つは PI 制御であり，もう1つは PSU における性能と消費電力の特性に着目した制御アルゴリズムである．

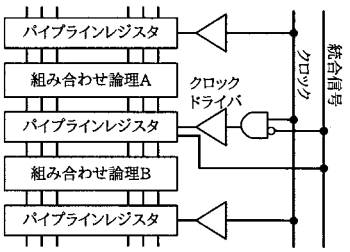

図1 PSU の実装 (バイパス回路は省略してある)

論文は次のような構成になっている．まず，2章で PSU のための回路について述べる．続いて，3章で本論文で提案する，PSU におけるスループット制御機構について述べる．4章で評価環境を示し，5章で評価結果を示す．6章で関連研究を示し，7章で本論文をまとめる．

2. PSU のための回路

図1に示すように，PSU では統合信号と呼ばれる信号線を追加し，パイプラインステージの統合を指示する．統合信号が0の場合，図1の隣接する組み合わせ論理回路 A と B は，それらの回路の間のパイプラインレジスタが動作しているため，異なったステージ

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

表 1　PI 制御を用いたスループット制御機構におけるパイプラインステージ数の変更

$TP_{avg}(i)$	$TP(i)$	次の SI
TP_{target} 以上	TP_{target} 以上	1 段階統合
TP_{target} 以上	TP_{target} 以下	変えない
TP_{target} 以下	TP_{target} 以上	変えない
TP_{target} 以下	TP_{target} 以下	1 段階統合解除

として動作する．一方，統合信号が 1 の場合，組み合わせ論理回路 A と B の間のパイプラインレジスタへのクロックが入力されなくなり，このパイプラインレジスタに入力される信号は次のステージへバイパスされる．したがって，このパイプラインレジスタは動作せず，2 つの組み合わせ論理回路は 1 つのステージとして動作する．

以上では 2 ステージ統合の場合について述べたが，さらに多くのステージの統合への拡張が可能である．

3. PSU におけるスループット制御機構

本章では，2 種類のスループット制御機構を提案する．これらのスループット制御機構は，オンチップで実現する．

本論文では，文献[2] と同様に，OS から処理に必要なスループットの目標値が与えられるとする．ここで，本論文では，性能として単位時間当たりの命令数であるスループットを用いる．また，以降，目標スループットのことを TP_{target} と呼ぶ．

本章で説明を行う 2 種類の制御機構は，TP_{target} を達成するために，実行中に定期的にパイプラインステージ数を変更する．そのために，プログラムの実行を一定時間ごとに区切る．この区切られた区間をサンプリング区間 (SI: Sampling Interval) と呼ぶ．各 SI の間は一定のパイプラインステージ数で実行する．

3.1 PI 制御を用いたスループット制御機構

本節では，PI 制御を用いたスループット制御機構について説明する．まず，実行開始から数えて i 番目の SI(第 i SI) のスループットを $TP(i)$，実行開始から第 i SI までの平均スループットを $TP_{avg}(i)$ とする．現在，第 i SI の実行が完了したとすると，表 1 にしたがってパイプラインステージ数を増減させる．表 1 によるパイプラインステージ数の変更方法は，PI 制御を利用し，現時点のスループットの値と実行開始から現時点までのスループットの平均値を用いている．

以上のアルゴリズムでは，第 i SI の終了時に，$TP_{avg}(i-1)$ から $TP_{avg}(i)$ への更新を，次の式によって行う．

$$TP_{avg}(i) = \frac{(i-1)\cdot TP_{avg}(i-1) + TP(i)}{i} \quad (1)$$

式 (1) に示すように，平均スループット $TP_{avg}(i)$ を更新するためには，加算器のほかに乗除算器が必要となる．しかしながら，実際には表 1 に示すように，$TP_{avg}(i)$ と TP_{target}，及び，$TP(i)$ と TP_{target} の大小関係がわかればよいため，乗除算器は不要である．以下でその理由を説明する．

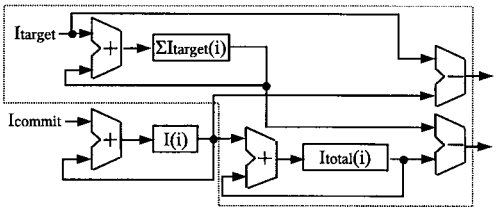

図 2　PI 制御を用いたスループット制御機構

まず，$TP_{avg}(i)$ と TP_{target} の比較について説明する．値の比較は，減算した結果の正負 (最上位ビット) を調べることで実現できる．スループットが単位時間に実行される命令数であることから，以下のようにして，$TP_{avg}(i)$ と TP_{target} の減算式を変形する．

$$TP_{target} - TP_{avg}(i) = \frac{\sum I_{target}(i)}{i \times T_{SI}} - \frac{I_{total}(i)}{i \times T_{SI}}$$

$$\propto \sum I_{target}(i) - I_{total}(i) \quad (2)$$

ここで，T_{SI} はサンプリング時間で定数，$I_{total}(i)$ は実行開始から第 i SI までの総実行命令数，$\sum I_{target}(i)$ は実行開始から第 i SI までの総目標実行命令数で，1 SI 当たりの目標実行命令数 I_{target} を i 回加えたものである．I_{target} は，TP_{target} と T_{SI} の積で与えられる定数となる．また，$TP(i)$ と TP_{target} の比較も同様にして，次のように簡単化できる．

$$TP_{target} - TP(i) \propto I_{target} - I(i) \quad (3)$$

ここで $I(i)$ は第 i SI の実行命令数である．

以上より，PI 制御を用いたスループット制御回路は，減算器と加算器で実現できることがわかる．図 2 に PI 制御を用いたスループット制御機構を示す．制御機構では，有効命令数のみをカウントするため，実行命令数としてコミット命令数 I_{commit} を用いる．

図 2 の点線で囲まれている部分は各 SI の終了時にのみ動作し，それ以外の部分は命令のコミット時に動作する．以下，具体的な動作を説明する．まず，各 SI の終了時に，減算器を用いて $\sum I_{target}(i) - I_{total}(i)$，及び，$I_{target} - I(i)$ の計算を行う．それぞれの計算結果について，符号ビットが 0 であれば $\sum I_{target}(i)$，及び，I_{target} の方が大きいと判定し，そうでなければ $I_{total}(i)$，及び，$I(i)$ の方が大きいと判定する．さらに，加算器を用いて $I(i)$ を足し合わせることで $I_{total}(i)$ を更新し，I_{target} を足し合わせることで $\sum I_{target}(i)$ を更新する．また，命令のコミット時に，コミットした命令数を足し合わせることで，$I(i)$ を更新し，各 SI の開始時に $I(i)$ を 0 に初期化する．

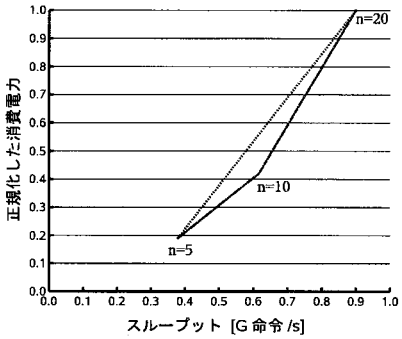

図 3 PSU におけるスループットと消費電力の関係

表 2 パイプラインステージ統合状態のフィードバックを行う制御機構におけるパイプラインステージ数の変更

$U(i+1)$ の値	ステージ数 n
$R \leq U(i+1)$	20
$-R < U(i+1) < R$	10
$U(i+1) \leq -R$	5

3.2 パイプラインステージ統合状態のフィードバックを行う制御機構

本節では，PSU におけるスループットと消費電力の特性に着目し，パイプラインステージ統合状態のフィードバックを行う制御機構について説明する．この制御機構は，PI 制御より多くの消費電力の削減を目指している．

本節で説明する制御機構において，パイプラインステージ統合状態のフィードバックを行う理由を説明する．なお，以下の説明で，パイプラインステージ数を n とする．ここでは，プロセッサのパイプラインステージ数を 20 段とし，パイプラインステージの統合によって $n = 20, 10, 5$ の 3 状態が選択可能であるとして説明を行う．

説明のために，各パイプラインステージ数を選択した場合に達成できるスループットと，その消費電力の関係を図 3 に示す．横軸はスループット，縦軸は $n = 20$ の消費電力で正規化した消費電力である．グラフ上の各点は，4 章の評価環境のもとで，各パイプラインステージ数を選択した場合の評価を行い，算出した値をプロットした．ここで，制御手法を用いて，通常どおり TP_{target} を与えてパイプラインステージ数を変更しながら実行したときの，スループットと消費電力の関係を図 3 のグラフ上にプロットした場合，その点の位置は，実行中に各パイプラインステージ数を選択した割合によって決まり，グラフの三角形の辺または内側になる．

$n = 20$ と $n = 10$ を結ぶ線分より，$n = 10$ と $n = 5$ を結ぶ線分の方が傾きが小さいという性質は，PSU の持つ性質であるため，プロセッサのコンフィギュレーションには依存しない．これは，$n = 20$ を $n = 10$ に変更した場合より，$n = 10$ を $n = 5$ に変更した場合の方がパイプラインステージ数の減少量が小さいため，$n = 10$ を $n = 5$ に変更した場合の方がスループットの低下に対して消費電力削減効果が小さくなるためである．以上は例として $n = 20, 10, 5$ の 3 状態が選択可能であるとして説明を行ったが，小さい n 同士を結んだ線分ほど傾きが小さくなるため，以上の性質は選択可能な n を追加しても成立する．

グラフからわかるように，点線は実線を常に上回っている．これは，ある TP_{target} を達成する場合，$n = 20$, $n = 5$ のみを選択して達成するよりも，その TP_{target} が $n = 20$，$n = 10$ のスループットの間にある場合は $n = 20$，$n = 10$ のみを，その TP_{target} が $n = 10$, $n = 5$ のスループットの間にある場合は $n = 10$，$n = 5$ のみを選択して達成した方が，消費電力が小さくなることを示している．

以上の PSU の性質より，$n = 20, 5$ のみが多く選択されないようにすることで，より多くの消費電力を削減できる．そこで，スループットが TP_{target} から大きく外れないようにしながら，$n = 10$ をできるだけ多く選択するよう，パイプラインステージ統合状態のフィードバックを行い，パイプラインステージ数を変更する．

以下，アルゴリズムの詳細を述べながら，パイプラインステージ統合状態をフィードバックすることで $n = 10$ を多く選択できる理由を説明する．現在第 i SI の実行が終了したとすると，その時点で次の計算を行う．

$$U(i+1) = \frac{TP_{target} - TP_{avg}(i)}{TP_{target}} + P(i) \quad (4)$$

ここで，$U(i+1)$ は第 $(i+1)$ SI のパイプラインステージ数を決定するための値で，第 $(i+1)$ SI のパイプラインステージ数は，表 2 にしたがって決定する．なお，表 2 の R はシステムに応じて決定される定数である．また，$P(i)$ は第 i フェーズのパイプラインステージ数に対応する値であり，システムに応じて決定される定数 $k(>0)$ を用いて，次のように決定する．

$$P(i) = \begin{cases} -k & (\text{ステージ数 } n = 20) \\ 0 & (\text{ステージ数 } n = 10) \\ k & (\text{ステージ数 } n = 5) \end{cases} \quad (5)$$

式 (4) では，平均スループット $TP_{avg}(i)$ と TP_{target} の差が大きい場合，第 1 項目が支配的となり，$TP_{avg}(i)$ を TP_{target} に近づけようとする．逆に，平均スループット $TP_{avg}(i)$ と TP_{target} の差が小さくなった場合，第 2 項目のパイプラインステージ数のフィードバックが支配的となり，なるべく $n = 10$ を選択するように

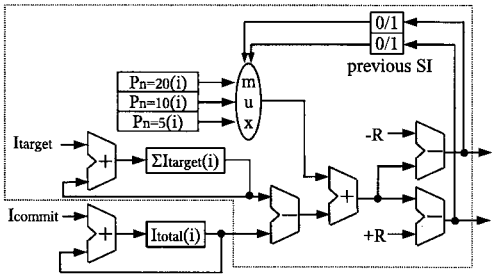

図 4　パイプラインステージ統合状態のフィードバックを行うスループット制御機構

制御する．

ここで，式 (4) の 2 項目で $P(i)$ を加えることにより，次の SI で $n = 10$ が選択されやすくなる理由を説明する．通常，実行中における平均スループット $TP_{avg}(i)$ の変化は SI の長さに対して緩やかであるため，$P(i)$ を加えない状態では，連続した SI で同じパイプラインステージ数を選択する場合が多い．例えば，直前の SI で $n = 20$ を選択した場合は，次の SI でも式 (4) の第 1 項目が正となり，$n = 20$ を選択する可能性が高い．したがって，その場合は，$P(i) < 0$ として $U(i+1)$ を 0 に近づけ，次の SI で $n = 10$ が選択されやすいようにする．逆に $n = 5$ の場合は，$P(i) > 0$ として $U(i+1)$ を 0 に近づける．

本制御手法の実装は，前節で述べた PI 制御を用いた制御機構と同様の方針で行う．まず，式 (4) の第 1 項目の，TP_{target} による正規化の操作を省略すると，単に $TP_{target} - TP_{avg}(i)$ の計算を行えばよいため，式 (2) と同様の変形により，$\sum I_{target}(i) - I_{total}(i)$ の演算に置き換えることが可能である．さらに，この命令数の差 $\sum I_{target}(i) - I_{total}(i)$ に対してパイプラインステージ数のフィードバックを加える．したがって，式 (4) の代わりに，以下の計算により $U(i+1)$ を算出する．

$$U(i+1) = \sum I_{target}(i) - I_{total}(i) + P(i) \quad (6)$$

以上より，パイプラインステージ統合状態をフィードバックするスループット制御機構も，乗除算器が不要である．図 4 にパイプラインステージ統合状態のフィードバックを行うスループット制御機構を示す．

図 4 の点線で囲まれている部分は各 SI の終了時にのみ動作し，それ以外の部分は命令のコミット時に動作する．具体的には，各 SI の終了時に，減算器を用いて $\sum I_{target}(i) - I_{total}(i)$ の計算を行い，それに，マルチプレクサで選択された $P(i)$ の値を加える．その後，$-R$，及び，R との大小比較を行うことで，表 2 に対応する判定を行う．また，各 SI の終了時に，I_{target} を足し合わせることで $\sum I_{target}(i)$ を更新し，命令のコミット時に，コミットした命令数を足し合わせることで，$I_{total}(i)$ を更新する．

表 3　プロセッサの構成

プロセッサコア	発行幅 8, RUU 64 エントリ, LSQ 32 エントリ, int ALU 8, int mult/div 4, fp ALU 8, fp mult/div 4, メモリポート 8
分岐予測	PHT 8K エントリ/履歴長 6 の gshare, BTB 2K エントリ, RAS 16 エントリ
キャッシュ	L1 命令/データ 64KB/ 32B ライン/2-way, L2 統合 2MB/64B ライン/4-way
メモリ	初期参照 64 サイクル, 転送間隔 2 サイクル
TLB	命令 16 エントリ, データ 32 エントリ, ミスレイテンシ 128 サイクル

3.3　スループット制御機構のオーバヘッド

スループット制御機構の追加によるオーバヘッドを見積もるため，VerilogHDL を用いて，2 つのスループット制御機構の設計を行った．なお，制御機構のビット幅はオーバフローが発生しないよう十分大きくとり，64bit とした．設計した回路は Synopsys Design Compiler を用いて，ROHM 0.35μm CMOS テクノロジを用いて論理合成し，Synopsys Apollo を用いて配置配線を行った．その結果，PI 制御を用いたスループット制御機構は 24k トランジスタ，統合状態をフィードバックするスループット制御機構は 30k トランジスタで実現できることがわかった．現在の汎用プロセッサのトランジスタ数は 20M を超える[3] ことから，追加するスループット制御機構の規模は十分に小さい．さらに，制御機構の大部分のトランジスタは，各 SI の終了時にのみ動作するため，動作頻度が低い．したがって，追加した制御機構における消費電力のオーバヘッドは十分に小さいといえる．

また，図 4 に示すように，提案する制御機構は 4 段の加減算器が直列に接続されているため，比較結果から 1 サイクルで統合信号が生成できるとすると，制御機構の動作に必要なサイクル数は 5 サイクル程度であると考えられる．

4.　評価環境

4.1　シミュレーション環境

SimpleScalar Tool Set 中の out-of-order 実行シミュレータを用いて，パイプラインのステージ数を変化させ，IPC を測定した．命令セットは SimpleScalar PISA である．ベンチマークプログラムは，SPECint2000 から，bzip2, gcc, gzip, mcf, parser, perlbmk, vortex, vpr の 8 本を用いた．なお，gcc では 1G 命令をスキップした後，587M 命令を実行し，その他では 2G 命令をスキップした後，1G 命令を実

行した．

表3に，プロセッサ構成を示す．また，PSUパイプラインの仮定は文献[1),2)]と同様とした．すなわち，Pentium4[3)]のステージ数20のパイプラインをベースとし，選択可能なパイプラインステージ数nは，$n=20$，10，5の3状態であり，それぞれ最大クロック周波数の100%，50%，25%で動作するとした．

ただし，$n=5$で達成できるスループットより低いTP_{target}が与えられた場合，スループットをTP_{target}まで低下させることができなくなる．そこで，$n=5$で達成できるスループットより低いスループットは，$n=5$のままクロック周波数のみを12.5%，6.25%，3.125%と1/2倍ずつ低下させることで実現する．そのために，3章で述べた制御手法を次のようにわずかに拡張した．パイプラインステージ数$n=5$を連続して選択した場合，最初はクロック周波数25%を選択し，それに続くSIでは12.5%，6.25%，3.125%と1段階ずつ低下させ，3.125%に達したら3.125%のままで実行する．パイプラインステージ数$n=5$の連続選択が終了するまで以上を続ける．

評価における，DVFSの各クロック周波数における電源電圧の値は，文献[2)]で用いられている値を用いた．また，最大クロック周波数は1GHzとした．

4.2 消費電力の計算

実行開始から終了までの平均消費電力は，実行開始から終了までに，パイプラインステージ数とクロック周波数の各組み合わせが選択された回数を記録し，それにもとづいて算出する．また，パイプラインステージ数とクロック周波数の各組み合わせの消費電力の相対値pを，以下の式によって求める．

$$p = f \times V^2 \times r \tag{7}$$

$$r = 1 - \frac{20-n}{20} \times m \tag{8}$$

ここで，fは選択したクロック周波数，Vは電源電圧，rはステージ統合によってパイプラインレジスタを停止させる前と停止させた後の消費電力比で，式(8)に示すように，パイプラインステージ数nとプロセッサの全消費電力に対するクロックネットワーク最終段のクロックドライバの消費電力の割合mによって決まる．本論文では，プロセッサの全消費電力に対するクロックネットワークの消費電力の割合を，文献[1),2)]と同様に30%とし，クロックネットワークの全消費電力に対するクロックネットワーク最終段のクロックドライバの消費電力の割合を88%[4)]とした．したがって，mは30%と88%の積である26.4%となる．なお，近年，プロセッサの静的消費電力が改善され，プロセッサの全消費電力に対する静的消費電力の割合はわずか10%程度に抑えることが可能であるため[5)]，ここでは消費電力削減において重要な動的消費電力について評価を行う．

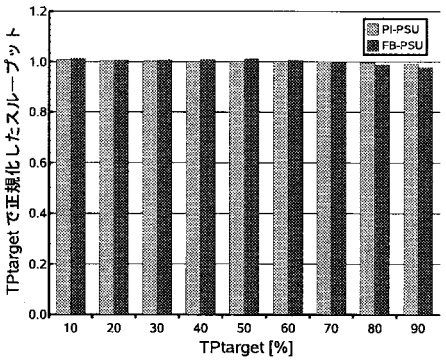

図5 スループット

5. 評価結果

評価において，SIの時間は10msとした．また，TP_{target}は，パイプラインステージ数$n=20$，最大クロック周波数，最大電源電圧で，各ベンチマークを最初から最後まで実行したときの，実行開始から終了までの平均スループットを100%とした割合で表す．

今回の評価では，パイプラインステージ数の変更のためのオーバヘッドは含めていない．これは，パイプラインステージ数の変更に必要な時間が，SIの時間に比べて十分短いためである．まず，3.3節で述べたとおり，制御機構の動作に必要なサイクル数は5サイクル程度である．加えて，パイプラインステージ数変更のためには，フェッチを停止しパイプラインが空になるまで待つ必要があり，20サイクル程度かかると考えられる．すると，パイプラインステージ数変更のために必要な時間は25ns程度であり，SIの時間に比べて十分に短い．

5.1 スループット

本論文で提案している，PI制御を用いたPSUと，パイプラインステージ統合状態のフィードバックを行うPSUのスループットを比較した．なお，以降の説明では，PI制御を用いたPSUをPI-PSU，パイプラインステージ統合状態のフィードバックを行うPSUをFB-PSUと呼ぶ．

図5に評価結果を示す．グラフの横軸はTP_{target}で，縦軸はTP_{target}で正規化したスループットのベンチマーク平均である．したがって，スループットが1に近いほど，TP_{target}に近いことを意味する．

PI-PSUのスループットはFB-PSUに比べてスループットが1に近い．この理由として，FB-PSUはパイプラインステージ統合状態のフィードバックを行っている点が挙げられる．具体的には，式(6)で$P(i)$を加えることにより，FB-PSUはPI-PSUより多くの消

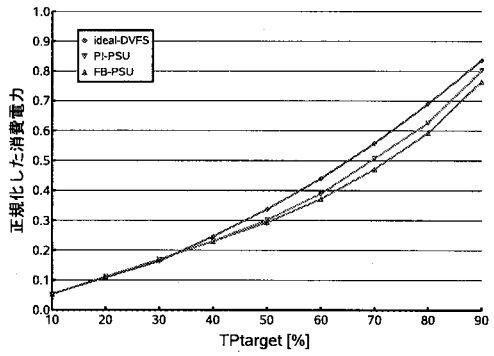

図 6 消費電力

費電力削減を可能とするが，スループットを TP_{target} に近づける $\sum I_{target}(i) - I_{total}(i)$ の効果を妨げる効果があるとも言える．したがって，スループットを TP_{target} に近づけることのみを目指した PI-PSU の方が，FB-PSU よりも，スループットが 1 に近くなる．ただし，FB-PSU において，実行のスループットと TP_{target} の差は最大でも 2.1%($TP_{target} = 90\%$) であり，問題のない程度に収まっている．

5.2 消費電力

本論文で提案している，PI-PSU，FB-PSU の消費電力と，DVFS の消費電力を比較する．ここで，DVFS については，スループット制御を行わず，TP_{target} を満たす理想的な消費電力の値 (ideal-DVFS) を示す．DVFS は PSU とは異なり IPC が向上しないため，TP_{target}[%] が与えられた場合は，実行開始から終了までクロック周波数を TP_{target}[%] に設定し，そのクロック周波数に対応する電源電圧で実行したときの消費電力が最適となり，これを ideal-DVFS とする．

測定結果を図 6 に示す．グラフの横軸は TP_{target} で，縦軸は消費電力のベンチマーク平均である．この消費電力の値は，パイプラインステージ数 $n = 20$，最大クロック周波数，最大電源電圧で実行した際の消費電力で正規化した値である．

ほとんどの TP_{target} において，PI-PSU の消費電力が ideal-DVFS の消費電力を下回り，さらに，FB-PSU の消費電力が PI-PSU の消費電力を下回った．比較を行ったところ，PI-PSU の消費電力は ideal-DVFS の消費電力よりも最大 11.5%($TP_{target} = 60\%$) 消費電力を削減できていることがわかった．また，FB-PSU の消費電力は ideal-DVFS の消費電力よりも最大 15.6%($TP_{target} = 60\%$) 消費電力を削減できていることがわかった．

6. 関連研究

PSU と DVFS を併用したハイブリッド制御機構が提案されている[2]．これに対し，本論文では制御機構を用いてパイプラインステージ数の制御のみを行い，任意のスループットを実現した．また，パイプライン状態の変更を 25 サイクル程度で実行でき，スループットの変化に高速に追随できる利点がある．さらに，本制御機構を用いて電源電圧もあわせて制御することが可能である．また，PSU におけるグリッチの問題を扱った VSP と呼ばれる機構が提案されている[6]．

7. まとめ

本論文では，PSU を用いて消費電力を削減するための，2 つのオンチップ制御機構を提案した．提案した 2 つの PSU 制御機構を評価した結果，DVFS に比べてそれぞれ，最大 11.5%，最大 15.6%消費電力を削減できることが示せた．

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．また，本研究は東京大学大規模集積システム設計教育研究センターを通し，シノプシス株式会社，ならびに，日本ケイデンス株式会社の協力で行われたものであり，本チップ試作は東京大学大規模集積システム設計教育研究センターを通し，ローム (株) および凸版印刷 (株) の協力で行われたものである．

参考文献

[1] H. Shimada, et al., "Pipeline Stage Unification: A Low-Energy Consumption Technique for Future Mobile Processors," ISLPED2003, pp. 326-329, 2003.

[2] H. Shimada, et al., "A Hybrid Power Reduction Scheme Using Pipeline Stage Unification and Dynamic Voltage Scaling," Nineth IEEE Symposium on Low-Power and High-Speed Chips, COOL Chips IX, pp.201-214, 2006.

[3] P. N. Glaskowsky, "Pentium 4 (Partially) Previewed," Vol. 14, Archive 8, pp. 1–4, 2000.

[4] F. E. Anderson, et al., "The Core Clock System on the Next-Generation Itanium Microprocessor," 2002 IEEE International Solid-State Circuits Conference Visual Supplement to the Digest of Technical Papers, pp. 110–111, 2002.

[5] J. Hart, et al., "Implementation of a 4th-Generation 1.8 GHz Dual Core Sparc V9 Microprocessor," 2005 IEEE International Solid-State Circuits Conference Visual Supplement to the Digest of Technical Papers, pp. 186–187, 2005.

[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), pp.231-242, 2006.