パイプラインステージ統合とダイナミック・ボルテージ・スケーリングを併用したハイブリッド消費電力削減機構

嶋田　創[†]　安藤秀樹[†]　島田俊夫[†]

近年のモバイルプロセッサでは，低消費電力と高性能の両方が要求されている．この要求に応える手法として我々は，Pipeline Stage Unification(PSU) を提案し，現在主流の Dynamic Voltage Scaling(DVS) よりも消費エネルギーを削減可能であることを示した．本論文では，DVS と PSU を複合し消費電力を削減するハイブリッド制御機構を提案する．この機構はシステムが要求するスループットに応じて動的に統合するステージ数とクロック周波数と電源電圧を適応させることにより，DVS と PSU を単独で用いるよりも多くの消費電力の削減を達成する．この機構を様々なスループットにおいて評価した結果，提案するハイブリッド制御機構は最大のスループットの 25%〜80%において，理想的な DVS よりも最大で 28%，現実の DVS に対して最大 34%消費電力を削減できることを示した．

A Hybrid Power Reduction Mechanism Using Pipeline Stage Unification and Dynamic Voltage Scaling

Hajime Shimada,[†] Hideki Ando[†] and Toshio Shimada[†]

Recent mobile processors are required to exhibit both low-power consumption and high performance. To satisfy these requirements, we proposed pipeline stage unification (PSU), and showed it can reduce energy consumption than that of dynamic voltage scaling (DVS) which is currently employed. This paper proposes a hybrid control mechanism which combines DVS and PSU to reduce power consumption. This mechanism adapts the number of unifying stages, clock frequency, and supply voltage according to the throughput that the system requires, and consequently it reduces power consumption than DVS and PSU. Our evaluation results in various throughput show that our mechanism reduces power consumption by a maximum of 28% compared to the real DVS or by a maximum of 34% compared to the ideal DVS.

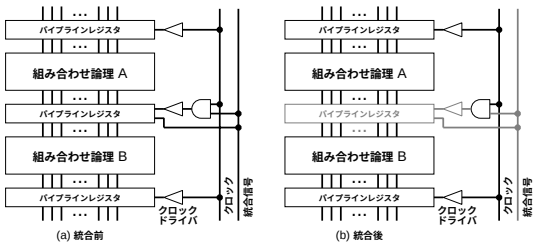
1. はじめに

近年のモバイルプロセッサでは，低消費電力と高性能の両方が要求されている．この要求を満たすために，現在**DVS**(dynamic voltage scaling) と呼ばれる方式が導入されている (たとえば，Transmeta Crusoe の LongRun[1)]，Intel Mobile Pentium シリーズの SpeedStep[2)]，AMD Mobile Athlon 4 の PowerNow![3)])．DVS はバッテリ持続時間要求やプロセッサ負荷に応じて，動的にクロック周波数と電源電圧を変更するものである．バッテリ持続時間要求が強いか，与えられた負荷が低ければ，クロック周波数を低下させ，消費電力を削減する．さらに，延びたクロックサイクル時間に信号の遅延を合わせ，電源電圧を低下させる．これにより，プログラム実行に要する消費電力を削減する．このように，DVS は消費電力を削減する有効な手法であるが，サブスレッショルドリーク電流の増加によって閾値電圧が下げにくくなる点や，過渡故障の増加という面から，将来のプロセス技術ではその有効性は減少する．

これに対して，我々はパイプラインのステージを動的に統合する**パイプラインステージ統合** (PSU: pipeline stage unification) と呼ぶ手法を提案した[4)〜6)]．PSU では DVS と同様に，プロセッサの消費電力を削減するためにクロック周波数を低下させるが，DVS と異なり，電源電圧を低下させるのではなく，パイプラインレジスタをバイパスさせることによって複数のパイプラインステージを統合する．クロックドライバの消費電力削減や，投機によるペナルティ削減のための IPC 向上により，プロセッサの消費電力が削減される．

以前の論文では，PSU と DVS それぞれを単独で用

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



注：灰色の部分は動作していない部分

図 1 PSU 実装

いた場合の評価を行い，ステージ統合の切り替え点となるクロック周波数 (**スイッチングポイント**) において，PSU は DVS と比べ消費エネルギーをより多く削減できることを示した[4)~6)]．スイッチングポイント間では，PSU だけではクロック周波数の変化に比例した量でしか消費電力を削減できないが，このとき電源電圧も同時に下げれば，さらに消費電力を削減できる．本論文では，DVS と PSU を複合し，統合するステージ数とクロック周波数を動的に変更することによって目標とするスループットを満足しつつ，消費電力を可能なかぎり押さえるハイブリッド制御機構について提案を行い，それを用いた時の消費電力を DVS と比較する．

本論文の構成は以下の通りである．2 章ではこの研究のベースとなる PSU について述べる．3 章では提案する DVS と PSU を複合するハイブリッド制御機構について述べる．4 章では評価における仮定について説明し，5 章で評価結果を示す．6 章では関連文献について述べ，最後に，7 章でまとめる．

2. PSU の概要

図 1 に PSU に関連する信号線とパイプラインレジスタとの結線関係を示す．説明を簡単にするために，2 ステージの統合を例としている．図 1 に示すように，パイプラインレジスタには，クロックの階層ネットワークの最終段のクロックドライバの出力が入力されている．また，PSU のための信号線として，**統合信号**と呼ぶ，統合を指示する信号線が追加される．

図 1 (a) はステージを統合していない状態を，図 1 (b) は統合した状態を示す．図中の黒い部分は動作部分を示し，灰色の部分は動作していない部分を示す．図 1 (a) は通常のパイプラインとして動作し，統合信号は 1 である．隣接する組合せ論理回路 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，異なったステージとして動作する．一方，図 1 (b) では，統合信号を 0 にすることによって組合せ論理回路 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，信号はバイパスされる．したがって，このパイプラインレジスタは動作せず，2 つの組合せ論理回路は 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のみ述べたが，統合信号を複数用意し，クロックドライバ停止のための信号を適切に制御することにより，さらに多くのステージの統合を行えるように拡張可能である．

3. DVS と PSU を複合するハイブリッド制御機構

この節では，DVS と PSU を複合し消費電力を削減するハイブリッド制御機構を提案する．以下の説明において，**統合度**とは PSU によって統合されているステージ数とする．統合度 1 は統合しないことを意味する．クロック周波数はある定められたステップで変更するものとする．目標とするスループットを TP_{target} と表し，これは OS より指示されるものとする．これは，文献 7) を参考に，OS が各タスクの実行後に余った時間を調べ，各タスクを規定の時間内で終らせるのに必要な，最小のスループットを計算して指示すると仮定したためである．

3.1 アルゴリズム

制御アルゴリズムは，基本的には，最小の電力でスループットが TP_{target} を満たすよう，定期的に，統合度，クロック周波数，電源電圧の 3 つのパラメータを調整させるものである．調整のために，**サンプリング・フェーズ**と呼ぶ区間を設ける．このフェーズでは，設定可能な統合度の全てについて，実際に IPC を測定し，パラメータ決定に用いる．決定したパラメータで，次のサンプリング・フェーズまでプロセッサを動作させる．この期間を，**実行フェーズ**と呼ぶ．**図 2** に 2 つのフェーズが切り替わる様子を示す．

図 2 に示しているように，サンプリング・フェーズは，さらに，設定可能な統合度を変化させ，その各々における IPC を測定する**サンプリング・サブフェーズ**よりなる．ある統合度 U におけるサンプリング・サブフェーズにおいては，まず，測定した IPC より，TP_{target} を満たす最小のクロック周波数 f_U を求める．次に，実行開始から現在までのスループット $TP_{current}$ を計算し，TP_{target} との誤差を計算する．その量に応じて，次の実行フェーズにおいて誤差を縮められるよう，あらかじめ定められた数のステップだけクロック周波数 f_U を調整する．U と f_U の組に対応する最小の電源電圧 V_U を求め，(U, f_U, V_U) を統合度 U にお

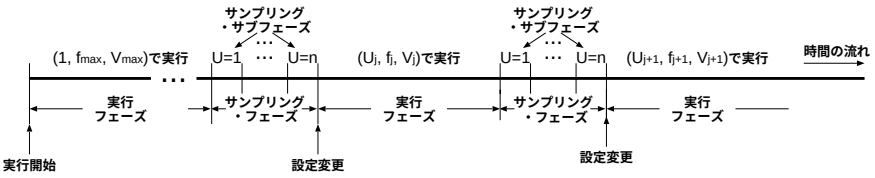


図 2 制御の概略

ける最適なパラメータとする．

全てのサンプリング・サブフェーズが終了したなら，各パラメータにおける消費電力を推定し，それが最小であるパラメータを採用し，実行フェーズに入る．

以上では，説明を容易にするために，サンプリング・フェーズでは，可能な全ての統合度について IPC の測定を行うと述べたが，実際には，実行フェーズでも IPC の測定を行い，その統合度における最適なクロック周波数と電源電圧を，サンプリング・サブフェーズでの方法と同じ方法で求める．これにより，サンプリング・フェーズでは，実行フェーズで設定されていた統合度以外の場合でのサンプリングのみを行えば良く，サンプリング・フェーズ時間を短縮できる．

また，与えられたプロセッサに対し，統合度とクロック周波数が定まれば，そのパラメータでプロセッサが正しく動作するための最小の電源電圧は一意に定まる (4.3 節で述べる)．したがって，サンプリング・フェーズで実際に計算により求める必要があるのは，統合度とクロック周波数だけである．

アルゴリズムの詳細を説明する前に，準備として，スループットの計算について説明する．一般にスループット TP は，IPC とクロック周波数 f より，以下の式で求められる：

$$TP = IPC \times f \tag{1}$$

サンプリング・サブフェーズ，および，実行フェーズを，単に，**フェーズ**と呼び，各フェーズに実行順に番号を付ける．式 (1) より，i 番目のフェーズにおける $TP_{current}(i)$ は，次式で求めることができる．

$$TP_{current}(i) = \frac{TP_{current}(i-1) \cdot \sum_{j=1}^{i-1} cycles(j) + IPC(i) \cdot cycles(i)}{\sum_{j=1}^{i} cycles(j)} \tag{2}$$

ここで，$cycles(i)$ は第 i フェーズのクロックサイクル数，$IPC(i)$ は第 i フェーズでの IPC である．

アルゴリズムは以下の通りである:

(1) 求めるパラメータ (統合度，クロック周波数，電源電圧) と現在のスループット $TP_{current}(0)$ を，それぞれ，$(1, f_{max}, V_{max})$，0 に初期化する．ここで，f_{max}, V_{max} はそれぞれ，クロック周波数と電源電圧の可動範囲における最大値である．

(2) 実行フェーズに入ったら，次の実行フェーズにおけるパラメータ候補集合 $CAND$ を空に初期化する．

(3) 実行フェーズからサンプリング・フェーズの最後までの各フェーズ i において，以下のようにして，パラメータ $(U(i), f(i), V(i))$ を求める：

(a) $TP_{current}(i)$ を式 (2) により求める．また，TP_{target} を満たす最小のクロック周波数 $f(i)$ を式 (1) より求める．ただし，$f(i) > f_{max}$ ならば $f(i)$ を f_{max} とする．

(b) $TP_{target}(i)$ と $TP_{current}(i)$ との誤差 $error(i)$ を以下の式で計算する：

$$error(i) = \frac{TP_{current}(i)}{TP_{target}(i)} - 1 \tag{3}$$

次の実行フェーズで $error(i)$ がより速く小さくなるように，(a) で得られた $f(i)$ を修正する．具体的には，$error(i)$ の絶対値が，あらかじめ定めた誤差範囲 E に対し，以下の式を満たす最大の非負の整数 $k(\leq k_{max})$ を見つけ，$f(i)$ を k ステップだけ増減させる．

$$|error(i)| > k \times E \tag{4}$$

具体的には，$error(i)$ が非負ならば減少させ，負ならば増加させる．ただし，$f(i) > f_{max}$ ならば $f(i)$ を f_{max}，$f(i) < f_{min}$ ならば $f(i)$ を f_{min} とする．ここで，f_{min} はクロック周波数の可動範囲における最小値である．

(c) $U(i)$, $f(i)$ から $V(i)$ を求め，$CAND$ に $(U(i), f(i), V(i))$ を加える．

(4) $CAND$ に含まれる各パラメータ候補の消費電力を推定する．各統合度に対し，その統合度でプロセッサが正しく動作するクロック周波数の上限を越える候補を除き，消費電力が最も小さい候補を次の実行フェーズのパラメータとする．

3.2 実　　装

3.1 節のアルゴリズムは，プロセッサ上で動作するソフトウェアとして実装する．

コードや変数を通常のプログラムと同様にメモリに格納すると，キャッシュ・ミスの発生により，アルゴリ

表 1 ベンチマーク

ベンチマーク	入力	実行命令数
compress95	ref/bigtest.in (300000 e 2231 に変更)	953M
gcc	train/amptjp.i	1257M
go	train/2stone9.in	547M
ijpeg	ref/specmun.ppm (1800M 命令で測定終了)	1800M
li	test/test.lsp	956M
m88ksim	test/ctl.in	422M
perl	train/jumble.in	2272M
vortex	train/vortex.in	2506M

表 2 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/1-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

表 3 統合度と最大クロック周波数，実行レイテンシ，キャッシュヒットレイテンシ，分岐予測ミスペナルティの関係

統合度		1	2	4
最大クロック周波数		100%	50%	25%
実行レイテンシ	int mult	3	2	1
	fp ALU	2	1	1
	fp mult	4	2	1
キャッシュヒットレイテンシ	L1	4	2	1
	L2	16	8	4
分岐予測ミスペナルティ		20	10	5

ズムの実行時間が非常に長くなる可能性がある．これを避けるため，コードは専用の ROM(以下，PCROM と呼ぶ) に，変数は汎用レジスタに格納し，定数はコード中に埋めこむ．また，実行命令数や実行サイクル数のカウンタを追加する．これにより，アルゴリズムの実行はプロセッサ・コア内で完結し，キャッシュ以下のメモリ階層にアクセスが行くことはない．

プロセッサは SMT(simultaneous multithreading) 動作を可能とし，通常のプログラムの実行と並行してアルゴリズムを実行する．PCROM からの命令フェッチの開始は，フェーズを計測するカウンタの指示によって行う．

4. 評価環境

4.1 シミュレーション環境

SimpleScalar Tool Set[8] 中の out-of-order 実行シミュレータをベースに提案するハイブリッド制御機構を組み込み，測定を行なった．命令セットは SimpleScalar PISA である．**表 1** に示すように，ベンチマークプログラムとして，SPECint95 の 8 本を用いた．ベンチマークプログラムのバイナリは gcc ver.2.7.2.3 を用い，-O6 -funroll-loops のオプションでコンパイルし作成した．

表 2 に，シミュレーションにおいて仮定したプロセッサの構成を示す．プロセッサは近年のプロセッサを考慮して，20 段のパイプラインを持つと仮定した．メモリアクセス時間は，プロセッサのクロック周波数の低下に比例して遅くなると仮定した．そのため，プロセッサのクロック周波数によらず，メモリアクセスのサイクル数は一定である．

4.2 パイプラインの仮定

本研究では，統合度 1，統合度 2，統合度 4 の 3 種類の統合度を仮定する．**図 3** に統合度 1，統合度 2，統合度 4 のパイプラインを示す．このパイプラインの各ステージの説明は，紙面の制限から省略する．参考文献 4)〜6) で仮定したものと同じものであり，これらを参照されたい．また，クロック周波数は 5% きざみで全 20 段階の中から選択されるとする．

表 3 に，これらのパイプラインにおける最大のクロック周波数，命令の実行レイテンシ，分岐予測ミスペナルティ，キャッシュヒットレイテンシを示す．なお，int/fp div と sqrt については，同一資源を繰り返し使用し完全なパイプライン化はされておらず，ステージの統合はできないと仮定した．レイテンシはそれぞれ，20，12，24 サイクルとした．

4.3 電源電圧とクロック周波数の関係

各統合度における電源電圧とクロック周波数の関係は，Crusoe TM5400[1] を基に定めた．まず，統合度 U が 1 の場合について説明する．クロック周波数が 100%〜25%の場合は Crusoe TM5400 の値を用いた．Crusoe TM5400 の最小クロック周波数である 25%より下のクロック周波数の場合は，Crusoe TM5400 の最小電源電圧である 1.10V より低下させることができないと仮定した．

U が 2 以上の場合については，次の式で統合度が 1 の時の電圧より求めた．

$$V(U,f) = V(1, U \times f) \qquad (5)$$

ここで，$V(U,f)$ は統合度 U，クロック周波数 f のときの電源電圧である．これは次のようにして求めることができる．統合度 1，電源電圧 V での最大のクロック周波数を f とすると，統合度を $U(>1)$ とした場合，同じ電源電圧 V での最大クロック周波数は f/U に落ちる．つまり，

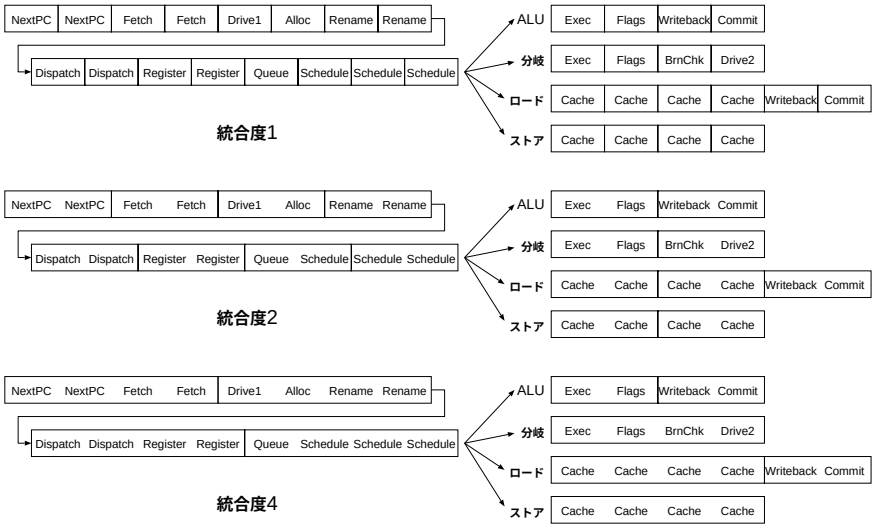


図 3 仮定した PSU のパイプライン

$$V\left(U, \frac{f}{U}\right) = V(1, f) \tag{6}$$

である．これより，式 (5) が求まる．

以上のようにして求めた電源電圧とクロック周波数の関係を**表 4** に示す．なお，どの統合度においても電源電圧は 1.10V を下限とした．

4.4 消費電力の計算方法

DVS の消費電力はアクティビティファクタを a，スイッチするノードの全容量を C，クロック周波数を f，電源電圧を V とすると，以下の式で表される：

$$P_{DVS} = a \times C \times f \times V^2 \tag{7}$$

PSU の消費電力の計算方法は以下の通りである[5),6)]．図 1 に示したようにパイプラインステージを統合すると，パイプラインレジスタへのクロック分配が抑制され，クロック・ネットワークの消費電力が減少する．統合度 U において停止するパイプラインレジスタの割合は $(U-1)/U$ である．m を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とすると，統合度 U で動作する PSU の消費電力は以下の式で表される：

$$\begin{aligned} P_{PSU} &= P_{DVS} \times \left(1 - \frac{U-1}{U} \times m\right) \\ &= a \times C \times f \times V^2 \times \left(1 - \frac{U-1}{U} \times m\right) \end{aligned} \tag{8}$$

評価においては，m は文献 5)，6) と同様に 30%と仮定した．この値は，文献 5)，6) で引用されている商用プロセッサの m の値のほぼ中央値である．

なお，消費電力の評価では，クロック周波数 100%，最大電源電圧での消費電力で正規化した値を用いた．

表 4 電源電圧とクロック周波数の関係

クロック周波数	統合度 1	統合度 2	統合度 4
100%	1.65V	—	—
95%	1.65V	—	—
90%	1.60V	—	—
85%	1.60V	—	—
80%	1.55V	—	—
75%	1.55V	—	—
70%	1.50V	—	—
65%	1.50V	—	—
60%	1.45V	—	—
55%	1.40V	—	—
50%	1.35V	1.65V	—
45%	1.30V	1.60V	—
40%	1.25V	1.55V	—
35%	1.20V	1.50V	—
30%	1.15V	1.45V	—
25%	1.10V	1.35V	1.65V
20%	1.10V	1.25V	1.55V
15%	1.10V	1.15V	1.45V
10%	1.10V	1.10V	1.25V
5%	1.10V	1.10V	1.10V

このため，a と C は正規化のための除算によって消去され，これらの値は評価に関係ない．

5. 評価結果

評価は，統合度 1 のプロセッサがクロック周波数 100%で動作する時のスループットを測定し，そのスループットの 25%〜80%を 5%刻みで TP_{target} として指定して行った．**表 5** に TP_{target} の計算に用いた IPC を示す．

今回の評価では，アルゴリズムの実行によるオーバーヘッドは含めていない．その理由として，まず第

表 5 統合度 1，クロック周波数 100%での IPC

ベンチマーク	IPC
compress95	1.18
gcc	1.03
go	0.87
ijpeg	2.54
li	1.22
m88ksim	1.62
perl	1.43
vortex	2.19

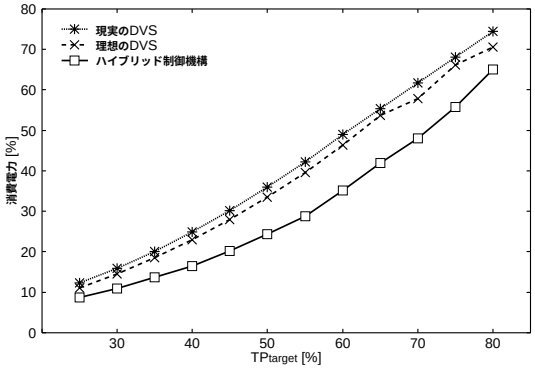


図 4 ハイブリッド制御機構と DVS の消費電力

1 に，アルゴリズムの実行サイクル数は実行フェーズのサイクル数と比較すると十分に短いと考えられる．第 2 に，アルゴリズムは SMT のスレッドとして実行するため，オーバヘッドはアルゴリズムのみを単一スレッドで実行したときよりも小さくなる．

5.1 消費電力の削減

提案するハイブリッド制御機構を用いた場合と DVS の消費電力を比較する．測定したハイブリッド制御機構のパラメータを以下のようにした．

- 実行フェーズ： 1M サイクル
- サンプリング・サブフェーズ： 10K サイクル
- 誤差範囲 E： 0.005
- k_{max}： 2

これらの値は，予備評価によって決定した．

図 4 にハイブリッド制御機構と DVS の消費電力を示す．図の横軸は TP_{target} を示し，縦軸は 100%のスループットの時の消費電力で正規化した消費電力を示している．3 本の折れ線グラフは，それぞれ上から現実の DVS，理想の DVS，ハイブリッド制御機構の，各 TP_{target} における消費電力のベンチマーク平均である．現実の DVS の消費電力とは，統合度 1 において，3.1 節で述べたアルゴリズムで動的にクロック周波数と電源電圧を変更した場合の消費電力である．理想の DVS の消費電力とは，プログラムの全実行を通しての IPC があらかじめ分かっており，その情報を用いた場合の消費電力である．この場合，全実行を通しての IPC と TP_{target} を式 (1) に代入して，TP_{target} を満たす最小のクロック周波数を求め，そのクロック周波数のみを用いて実行を行う．図 4 より，全ての TP_{target} において，ハイブリッド制御機構は現実の DVS のみならず，理想の DVS よりも消費電力を削減できることがわかる．

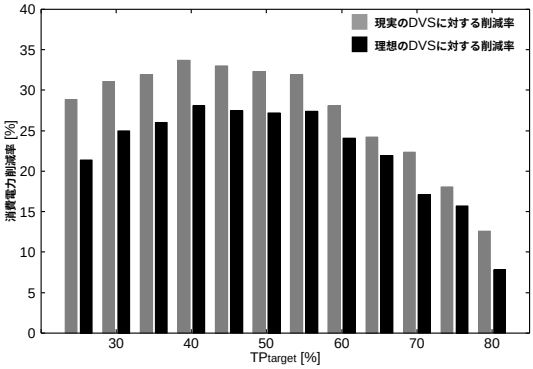


図 5 DVS に対するハイブリッド制御機構の消費電力削減率

また，ハイブリッド制御機構の DVS に対する消費電力削減率を**図 5** に示す．各 TP_{target} における 2 本の棒グラフは，それぞれ現実の DVS に対する消費電力削減率と理想の DVS に対する消費電力削減率である．

図 4 では TP_{target} が小さくなるほど DVS とハイブリッド制御機構の消費電力の差が小さくなることが分かるが，消費電力削減率で比較すると，TP_{target} が小さくても，中程度の TP_{target} と同様に消費電力を削減できていることが分かる．また，TP_{target} が大きくなるほど，統合度 1 で動作する局面が増えるために DVS との差が少なくなる．最大で，TP_{target}=40%時において理想の DVS に対して 28%，現実の DVS に対して 34%と非常に大きな削減率を達成した．

5.2 選択されたクロック周波数の内訳

5.1 節の測定時において，現実の DVS とハイブリッド制御機構の動作時に選択されたクロック周波数の分布を**図 6** と**図 7** に示す．図の横軸はクロック周波数であり，縦軸は選択されたクロック周波数の割合である．図が繁雑になるのを避けるため，測定した TP_{target} のうち，半分のみをプロットした．

図 6 より，DVS ではクロック周波数はそれぞれの TP_{target} と等しいクロック周波数を中心に選択されていることが分かる．一方，図 7 より分かるように，ハイブリッド制御機構では PSU を多用するために，図 6 中の同一の TP_{target} のグラフより低いクロック周波数が選択されていることが分かる．これより，ハイブリッド制御機構が DVS に比べて大きく消費電力を削

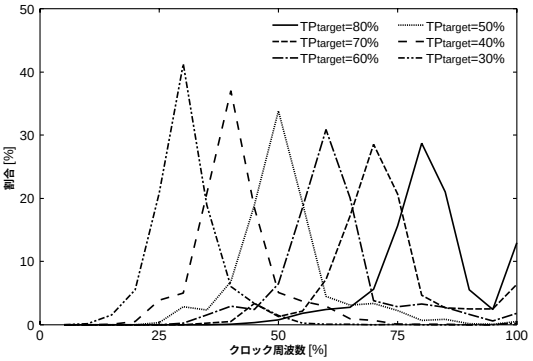


図 6 現実の DVS 使用時に選択されたクロック周波数

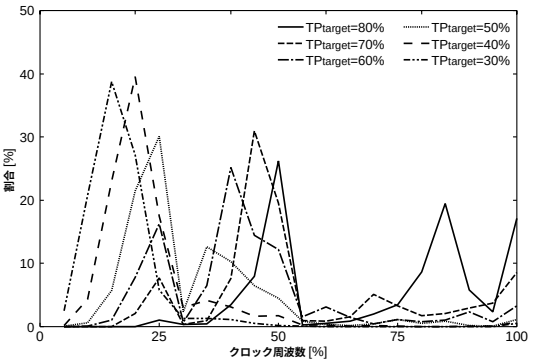


図 7 ハイブリッド制御機構使用時に選択されたクロック周波数

減できる理由は，積極的に PSU を用いてるためであることが分かる．

また，図 7 よりスイッチングポイントであるクロック周波数 50%と 25%，および，それぞれの少し下のクロック周波数が多く選択されていることが分かる．一方，それぞれの少し上のクロック周波数はあまり選択されていない．この理由は，スイッチングポイント少し上のクロック周波数を用いるよりも，統合度を上げてスイッチングポイントかその少し下の周波数を用いた方が，消費電力が削減されることが多いためである．これにより，スイッチングポイントの少し上のクロック周波数の山が小さくなり，代わりに選択されたスイッチングポイント少し下のクロック周波数の山が大きくなる．

5.3 TP_{target} の達成率

ハイブリッド制御機構のアルゴリズムは，定期的にスループットを調整するものなので，最終的なスループットが TP_{target} を下まわる可能性がある．**図 8** に TP_{target} に対する達成率を示す．TP_{target} 達成率は，実行終了時のスループットを TP_{target} で割ったものである．図の縦軸は TP_{target} の達成率であり，横軸は TP_{target} である．各 TP_{target} に対する棒グラフは，

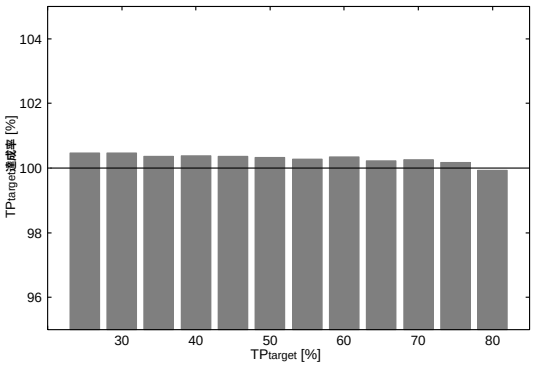


図 8 TP_{target} 達成率

その TP_{target} における，TP_{target} の達成率のベンチマーク平均である．

図 8 より，TP_{target}=80%の時のみ TP_{target} を 0.1%下まわる結果となった以外は，TP_{target} を達成できるという結果になった．紙面の制限で掲載することはできないが，個々のベンチマークごとに見ても，最も大きく TP_{target} を下まわったものでも不足は 0.6%であり，この程度のスループットの不足は問題ないと考える．

6. 関 連 研 究

パイプラインの長さを動的に変更する他の研究として，以下の 2 つがある．Koppanalil らは，Dynamic Pipeline Scaling（DPS）という手法を提案した[9]．彼らは，電源電圧には下限があるため，DVS では消費電力の削減がそれにより制限されることに着目した．そこで，彼らはプロセッサが非常に低いクロック周波数で動作しているときのみパイプラインを短縮し，IPC の向上でより消費エネルギーの削減を行なった．我々の研究は，彼らよりも高いクロック周波数から PSU を用いることを提案している点で，彼らの研究とは異なる．Efthymiou らは，非同期プロセッサにおいて，動的にパイプラインを短縮することを提案した[10]．我々の研究は，同期プロセッサにおいてパイプラインの統合を行う点で彼らの研究とは異なる．

動的にパイプラインのリソースを変更して消費電力を削減する研究には以下のものがある．Albonesi は，動的に命令ウィンドウやキャッシュのサイズとクロック周波数を変更し，実行時間を削減する方法を提案した[11]．一般に，資源サイズと実現可能なクロック周波数は，トレードオフの関係にある．このトレードオフの最適点を求め，プログラムの実行時間を削減する．Bahar らは，プログラムの実行中に同時発行命令数が大きく変化することに着目し，必要に応じて動的に命

令発行論理と機能ユニットの一部を停止させ，消費電力を削減することを提案した[12]．Manne らは分岐予測ミスしたパスからフェッチした命令を減らして消費電力を削減することを提案した[13]．具体的には，分岐予測の信頼性が低い分岐命令が連続してフェッチされた場合，その後の命令のフェッチを停止することにより，分岐予測ミスしたパスからフェッチした命令数を削減する．Canal らは，データ，アドレス，命令をある特別のエンコーディングによって圧縮し，パイプラインの活動量を減少させ，消費電力を削減する方法を提案した[14]．

DVS に関する研究としては，様々な DVS のアルゴリズムを評価したもの[15] や，リアルタイム・システムにおける DVS のスケジューリングについて提案を行なっているもの[16] がある．また，DVS を発展させたものとして，プロセッサをいくつかの領域に区切り，領域ごとに異なる電源電圧とクロック周波数で動作させる Multiple Clock Domain アーキテクチャ[17),18] がある．

7. ま と め

本論文では，消費電力削減手法である PSU と DVS を複合するハイブリッド制御機構を提案し，その機構によって削減される消費電力を，現在主流の消費電力削減手法である DVS と比較した．提案するハイブリッド制御機構は，定期的に IPC をサンプリングし，次の統合度とクロック周波数と電源電圧を決定するものである．

評価の結果，提案するハイブリッド制御機構を用いることにより，25%〜80%のスループットにおいて，DVS のみの場合よりも消費電力を大きく削減できることを示した．消費電力削減率の最大は，理想な DVS に対して 28%，現実の DVS に対して 34%であった．また，提案する機構を用いたときの TP_{target} 達成率は，もっとも TP_{target} を下まわったものでも 99.4%であり，問題のないレベルであった．

参 考 文 献

1) Laird, D.: *Crusoe Processor Products and Technology*, Transmeta Corporation (2000).
2) Intel Corporation: *Intel Pentium M Processor Datasheet* (2003).
3) Advanced Micro Devices, Inc.: *Mobile AMD Athlon 4 Processor Model 6 CPGA Data Sheet* (2001).
4) 嶋田創, 安藤秀樹, 島田俊夫: 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, pp. 57–62 (2001).
5) 嶋田創, 安藤秀樹, 島田俊夫: パイプラインステージ統合：将来のモバイルプロセッサのための消費エネルギー削減技術, 先進的計算基盤システムシンポジウム SACSIS2003, pp. 283–290 (2003).
6) Shimada, H., Ando, H. and Shimada, T.: Pipeline Stage Unification: A Low-Energy Consumption Technique for Future Mobile Processors, *ISLPED2003*, pp. 326–329 (2003).
7) Pering, T., Burd, T. and Brodersen, R.: The Simulation and Evaluation of Dynamic Voltage Scaling Algorithms, *ISLPED-1998*, pp. 76–81 (1998).
8) Burger, D. and Austin, T. M.: The SimpleScalar Tool Set, Version 2.0, Technical Report CS-TR-97-1342, University of Wisconsin-Madison Computer Sciences Dept. (1997).
9) Koppanalil, J., Ramrakhyani, P., Desai, S., Vaidyanathan, A. and Rotenberg, E.: A Case for Dynamic Pipeline Scaling, *CASES2002*, pp. 1–8 (2002).
10) Efthymiou, A. and Garside, J. D.: Adaptive Pipeline Depth Control for Processor Power-Management, *ICCD2002*, pp. 454–457 (2002).
11) Albonesi, D. H.: Dynamic IPC/Clock Rate Optimization, *ISCA-25*, pp. 282–292 (1998).
12) Bahar, R. I. and Manne, S.: Power and Energy Reduction Via Pipeline Balancing, *ISCA-28*, pp. 218–229 (2001).
13) Manne, S., Klauser, A. and Grunwald, D.: Pipeline Gating: Speculation Control For Energy Reduction, *ISCA-25*, pp. 132–141 (1998).
14) Canal, R., Gonzalez, A. and Smith, J.E.: Very Low Power Pipelines using Significance Compression, *MICRO-33*, pp. 181–190 (2000).
15) Pering, T., Burd, T. and Brodersen, R.: The Simulation and Evaluation of Dynamic Voltage Scaling Algorithms, *ISLPED1998*, pp. 76–81 (1998).
16) Krishna, C.M. and Lee, Y.-H.: Voltage-Clock-Scaling Adaptive Scheduling Techniques for Low Power in Hard Real-Time Systems, *IEEE Transactions on Computers*, Vol. 52, No. 12, pp. 1586–1593 (2003).
17) Magklis, G., Scott, M. L., Semeraro, G., Albonesi, D. H. and Dropsho, S.: Profile-based Dynamic Voltage and Frequency Scaling for a Multiple Clock Domain Microprocessor, *ISCA-30*, pp. 14–25 (2003).
18) Semeraro, G., Albonesi, D.H., Dropsho, S.G., Magklis, G., Dwarkadas, S. and Scott, M. L.: Dynamic Frequency and Voltage Control for a Multiple Clock Domain Microarchitecture, *MICRO-35*, pp. 356–367 (2002).