# パイプラインステージ統合とDVSの併用による消費電力の削減

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，パイプラインステージ統合（PSU: Pipeline Stage Unification）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．しかし，DVS と PSU は排他的にしか利用できないものではなく，併用することによってさらなる消費電力の削減を達成できると考えられる．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 それぞれ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．この機構を種々の目標のスループットに対して評価した結果，提案するハイブリッド制御機構は DVS 単独に対して最大 14%，PSU 単独に対して最大 28%消費電力を削減できることを示した．

# Power Consumption Reduction through Combining Pipeline Stage Unification and DVS

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 that it can reduce energy consumption than that of dynamic voltage scaling (DVS) which is currently employed. However, DVS and PSU are not exclusive techniques, and so further reduction of power comsumption can be achieved through combining them. This paper proposes a hybrid control mechanism which combines DVS and PSU to reduce power consumption more. 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 more than standalone DVS and standalone PSU. We evaluated our mechanism with various target throughputs. Our evaluation results show that our mechanism reduces power consumption by a maximum of 14% compared to the standalone DVS or by a maximum of 28% compared to the standalone PSU.

## 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 PentiumM シリーズの SpeedStep[2]，AMD Turion 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，プロセス技術の進歩に応じて最大電源電圧は下げられる一方で，サブスレッショルド・リーク電流を増加させる閾値電圧の低下は得策とはいえない点や，ソフト・エラーの増加という面から，最小電源電圧はそれほど下げられないことから，その有効性は減少していく．

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

† 京都大学大学院情報学研究科
Graduate School of Informatics, Kyoto University
†† 名古屋大学大学院工学研究科
Graduate School of Engineering, Nagoya University

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，投機失敗に対するペナルティ削減によるIPC向上により，プロセッサの消費エネルギーが削減される．

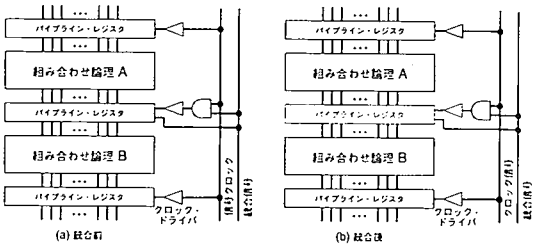
以前の論文では，PSUとDVSそれぞれを単独で用いた場合の評価を行い，ステージ統合の切替え点となるクロック周波数（スイッチング・ポイント）において，PSUはDVSと比べ消費エネルギーをより多く削減できることを示した[4)~6)]．しかしながら，PSUとDVSは排他的にしか利用できないものではなく，併用することが可能である．前述したとおり，DVSは将来のプロセス技術では有効性は減少していくが，その有効性がなくなることはないと考えられる．PSUだけでは，スイッチング・ポイント間ではクロック周波数の変化に比例した量でしか消費電力を削減できないが，このとき電源電圧も同時に下げれば，さらに消費電力を削減できる．そこで我々は，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抑えるハイブリッド制御機構について提案し[7),8)]，それを用いたときの消費電力をDVSと比較する．

本論文の構成は以下のとおりである．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ている状態を示しており，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，信号は統合信号により制御されるマルチプレクサ（図では省略）によりパイプライン・レジスタをバイパスする．この場合，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．




注：灰色の部分は動作していない部分

図1 PSUの実装
Fig. 1 Implemenation of PSU.

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を統合でき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

この章では，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．クロック周波数は商用のプロセッサにおけるDVSと同様，あらかじめ定められた離散値のいずれかをとることとする．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．通常，DVSでは制御はOSより下のソフトウェア・レイアで行われるが，本ハイブリッド制御機構においても同様であり，$TP_{target}$が指示される以外にOSとの間にやりとりはない．

### 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すべてについて，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

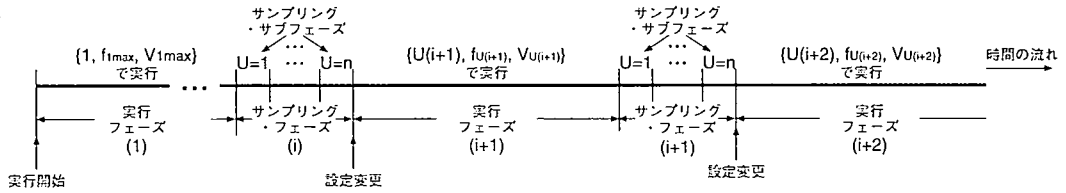



図 2 制御の概略
Fig. 2 Outline of control.

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．サンプリング・フェーズのある統合度 $u$ 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，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 $u$ における最適なパラメータとする．

すべてのサンプリング・サブフェーズが終了したなら，各パラメータにおけるプロセッサの消費電力を推定し，それが最小となるパラメータを次の実行フェーズのパラメータ $\{U, f_U, V_U\}$ として採用し，次の実行フェーズに入る．

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，単位時間あたりの実行命令数で定義され，以下の式で定義される：

$$TP = \frac{total\_insts}{total\_time} \quad (1)$$
$$= IPC \times f \quad (2)$$

ここで，$f$ はクロック周波数，$total\_insts$ はプログラムの実行開始からスループット計算時までの総実行命令数，$total\_time$ はプログラムの実行開始からスループット計算時までの総実行時間である．

以下，アルゴリズムの説明のために，実行フェーズ，および，サンプリング・フェーズに実行順に番号を付ける．この番号は，実行フェーズ，および，サンプリング・フェーズ各々に独立して付けることとする．よって，各フェーズと番号の出現順は，「実行フェーズ 1，サンプリング・フェーズ 1，実行フェーズ 2，サンプリング・フェーズ 2，…」のようになる．

アルゴリズムは以下のとおりである：

(1) 求めるパラメータ { 統合度，クロック周波数，電源電圧 } と $TP_{current}$ を，それぞれ，$\{1, f_{1max}, V_{1max}\}$，0 に初期化する．ここで，$f_{1max}$，$V_{1max}$ はそれぞれ，統合度 1 における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) サンプリング・フェーズ ($i$) 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) 各サンプリング・サブフェーズでは，クロック周波数を各統合度 $u$ における可動範囲の最大値とし，実行命令数 $insts$ およびクロック・サイクル数 $cycles$ を計測する．サンプリング・サブフェーズ終了時には，以下のようにして，各サンプリング・サブフェーズにおける統合度 $u$ におけるパラメータ $\{u, f_u, V_u\}$ を求める．

(a) $TP_{current}$ を式 (1) を用いて更新する．

(b) 統合度 $u$ における IPC，$IPC_u$ を以下の式で求める．

$$IPC_u = \frac{insts}{cycles} \quad (3)$$

(c) $TP_{target}$ を満たす最小のクロック周波数 $f_u$ を，得られた $IPC_u$ を式 (2) に代入して求める．ただし，$f_u > f_{umax}$ ならば $f_u = f_{umax}$，$f_u < f_{umin}$ ならば $f_u = f_{umin}$ とする．ここで，$f_{umax}$ は統合度 $u$ におけるクロック周波数の可動範囲における最大値，$f_{umin}$ は統合度 $u$ におけるクロック周波数の可動範囲における最小値である．

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

$$error = \frac{TP_{current}}{TP_{target}} - 1 \quad (4)$$

次の実行フェーズで $error$ がより小さくなるように，(c) で得られた $f_u$ を修正する．ここで，$f_u$

は離散値であるため，その離散値のステップに従って修正する．このとき，修正の最大ステップ数は，あらかじめ $k_{max}$ と定めておく．

具体的には，*error* 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，$f_u$ を $k$ ステップだけ増減させる．

$$|error| > k \times E \quad (5)$$

*error* が非負ならば減少させ，負ならば増加させる．ただし，$f_u > f_{umax}$ ならば $f_u$ を $f_{umax}$，$f_u < f_{umin}$ ならば $f_u$ を $f_{umin}$ とする．

(e) $u$, $f_u$ でインデクスされる表を引いて $V_u$ を求め，*CAND* に $\{u, f_u, V_u\}$ を加える．

(f) すべての統合度のサンプリングが終わっていなければ，まだサンプリングが終わっていない統合度を選択し，サンプリング・サブフェーズに入る．この場合，アルゴリズムは(3)の最初に戻る．

(4) *CAND* に含まれる各パラメータ候補の消費電力を求め，最も消費電力が小さい候補の $\{u, f_u, V_u\}$ を次の実行フェーズのパラメータ $\{U(i+1), f_{U(i+1)}, V_{U(i+1)}\}$ とする．

ただし，$TP_{target}$ を達成することを重視し，*error* が負かつ $|error| > k_{max} \times E$ であれば，$TP_{current}$ が $TP_{target}$ を大幅に下回っているとして，スループットが最大になるよう，統合度1，クロック周波数100%で動作させる．

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，可能なすべ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よく，サンプリング・フェーズ時間を短縮できる．

### 3.2 実　　装

3.1節のアルゴリズムはソフトウェアで実装するが，その際に3つの方法が考えられる．1つ目の方法は，そのソフトウェアの実行を別スレッドとし，必要に応じて切り替える方法である．この方法は，OSを介したスレッド切替えを要し，その時間オーバヘッドが問題となる．2つ目の方法は，同じく別スレッドとするが，SMT (Simultaneous Multi-Threading) でプログラムの実行と同時に実行するものである．この方法は，アルゴリズムのスレッドは，つねに「起きている」ので，スレッド切替えのオーバヘッドはないが，プロセッサをSMTに対応させなければならならず，プロセッサが複雑化したり，面積が増大したりする (文献9) によれば5%) という問題がある．3つ目の方法は，アルゴリズムを実行する専用のRISCコアを備えることである．この方法は，先の2つの方法のような欠点はないが，RISCコアの面積コストが問題となる．しかし，このコストは以下に述べるように十分小さく問題ない．

たとえばMIPS 4KEを搭載した場合について検討する．文献10) によれば，MIPS 4KEコアのサイズは130 nmのプロセス技術で1.2〜1.5 mm$^2$ である．上記のコアは，アルゴリズムの実行に必要な浮動小数点ユニットを含まないので，これを追加しなくてはならない．文献11) によれば，この面積は130 nmのプロセス技術で1.8 mm$^2$ である．これらを90 nmのプロセス技術にシュリンクすると，約1.4〜1.6 mm$^2$ となる．これを90 nmのプロセス技術で100 mm$^2$ 前後のPC用モバイル・プロセッサに搭載すると考えると，その面積オーバヘッドは2%程度となるといえる．このオーバヘッドは十分許容できるものと考える．

なお，RISCコアはメインのプロセッサとは独立しており，アルゴリズムのソフトウェアは，RISCコアの命令として専用のROMに埋め込まれる．また，アルゴリズムの実行のために以下のカウンタ/レジスタが追加される．

(1) *total_insts* を記憶するレジスタ
(2) *total_time* を記憶するレジスタ
(3) *insts* を数えるカウンタ
(4) *cycles* を数えるカウンタ

これらのカウンタ/レジスタはRISCコアの近傍に配置され，(3)，(4) のカウンタは毎サイクル，メイン・プロセッサによって更新される．メイン・プロセッサからカウンタまでは距離が遠く，1サイクルでカウンタを更新できない懸念があるが，これはパイプライン化すれば問題ない．(1)，(2) のレジスタについては，(3)，(4) のカウンタの値を用いて，毎サンプリング・サブフェーズの開始時にRISCコアにより計算され更新される．

## 4. 評価環境

### 4.1 シミュレーション環境

SimpleScalar Tool Set [12] 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，SPECint2000の8本を用いた．ベ

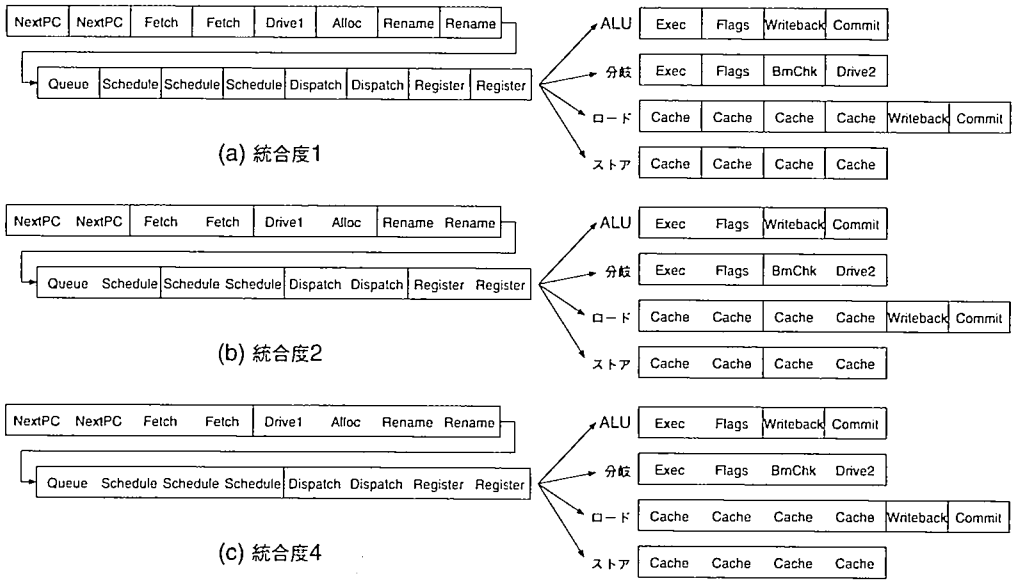



図 3 仮定した PSU のパイプライン
Fig. 3 Assumed PSU pipeline.

表 1 ベンチマーク
Table 1 Benchmark.

| ベンチマーク | 入力 |
|---|---|
| bzip2 | train/input.compressed |
| gcc | train/cp-decl.i |
| gzip | train/input.combined |
| mcf | ref/inp.in |
| parser | train/train.in |
| perlbmk | train/scrabbl.in |
| vortex | train/lendian.raw |
| vpr | train/route.in |

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．入力は train 入力もしくは ref 入力を用い，最初の 1G 命令をスキップした後，1.5G 命令を測定に用いた．

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．パイプラインの段数は 20 段と仮定した．

### 4.2 パイプラインの仮定

本評価では，統合度 1，2，4 の 3 種を仮定する．図 3 に統合度 1，2，4 のパイプラインを示す．図 3 (a) に示すように，評価においては，180 nm の Pentium 4[13] のパイプラインとほぼ等しいパイプラインをベースとした．図中の各ステージの動作については，以下のとおりである．

- NextPC：分岐予測による次の PC の決定
- Fetch：命令キャッシュからの命令フェッチ

表 2 プロセッサの構成
Table 2 Processor configuration.

| | |
|---|---|
| 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 |
| L1 命令キャッシュ | 64 KB/32B ライン/1-way |
| L1 データ・キャッシュ | 64 KB/32B ライン/1-way |
| L2 キャッシュ | 2 MB/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 |

- Drive1：フェッチした命令のデコーダへの転送
- Alloc：RUU (Register Update Unit)，LSQ (Load/Store Queue) の割当て
- Rename：レジスタ・リネーミング
- Queue：RUU への書き込み
- Schedule：命令スケジューリング
- Dispatch：RUU からの発行
- Register：レジスタ読み出し
- Exec：実行
- Flags：フラグの書き込み
- BrnChk：分岐命令の実行結果と分岐予測の比較
- Drive2：分岐予測の結果のフロントエンドへの

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
Table 3 Assumptions of execution latencies, cache hit latencies, and branch misprediction penalty in PSU processor.

| 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 |

転送

- Cache：データ・キャッシュ・アクセス
- Writeback：ライトバック
- Commit：コミット

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

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した．各統合度における電源電圧とクロック周波数の関係は，90 nm のプロセス技術で製造されている Intel Pentium M Model 755[14] 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%～30%の場合の電源電圧は PentiumM Model 755 の値を用いた．同プロセッサの最小クロック周波数である 30%より下のクロック周波数の場合は，同プロセッサの最小電源電圧である 0.988 V より低下させることができないと仮定した．これは，トランジスタの閾値電圧，ソフト・エラー，ノイズ・マージン等から，同プロセッサにおいて，電源電圧を下げることが困難であるため実現されていないと推測したからである．

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

$$V(U,f) = V(1, U \times f) \tag{6}$$

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，

表 4 電源電圧とクロック周波数の関係
Table 4 Relationship between supply voltage and clock frequency.

| クロック周波数 | 統合度 1 | 統合度 2 | 統合度 4 |
|---|---|---|---|
| 100% | 1.340 V | — | — |
| 95% | 1.316 V | — | — |
| 90% | 1.292 V | — | — |
| 85% | 1.268 V | — | — |
| 80% | 1.244 V | — | — |
| 75% | 1.220 V | — | — |
| 70% | 1.196 V | — | — |
| 65% | 1.172 V | — | — |
| 60% | 1.148 V | — | — |
| 55% | 1.124 V | — | — |
| 50% | 1.100 V | 1.340 V | — |
| 45% | 1.076 V | 1.292 V | — |
| 40% | 1.052 V | 1.244 V | — |
| 35% | 1.020 V | 1.196 V | — |
| 30% | 0.988 V | 1.148 V | — |
| 25% | 0.988 V | 1.100 V | 1.340 V |
| 20% | 0.988 V | 1.052 V | 1.244 V |
| 15% | 0.988 V | 0.988 V | 1.148 V |
| 10% | 0.988 V | 0.988 V | 1.052 V |
| 5% | 0.988 V | 0.988 V | 0.988 V |

$$V\left(U, \frac{f}{U}\right) = V(1,f) \tag{7}$$

である．これより，式 (6) 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 0.988 V 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 $V$，リーク電流を $I_{leak}$ とすると，動的消費電力 $P_{DVSdynamimc}$ と静的消費電力 $P_{DVSstatic}$ の和であり，以下の式で表される：

$$P_{DVS} = P_{DVSdynamic} + P_{DVSstatic} \tag{8}$$

$$P_{DVSdynamic} = a \times C \times f \times V^2 \tag{9}$$

$$P_{DVSstatic} = I_{leak} \times V \tag{10}$$

ここで，最大電源電圧 $V_{max}$，最大クロック周波数 $f_{max}$ のときの静的消費電力の最大値 $P_{DVSstatic,max}$ の動的消費電力の最大値 $P_{DVSdynamic,max}$ に対する割合を $s$ とする．すなわち，

$$P_{DVSstatic,max} = P_{DVSdynamic,max} \times s \tag{11}$$

$I_{leak}$ は，5 章の評価における電源電圧の範囲において，電源電圧にかかわらず一定と仮定すると（この仮定の妥当性については付録 A.1 節を参照），式 (10) より，$P_{DVSstatic}$ は電源電圧に比例するので，

$$P_{DVSstatic} = P_{DVSstatic,max} \times \frac{V}{V_{max}} \tag{12}$$

と書くことができる．式 (9)，(11) を式 (12) に代入す

ると，

$$P_{DVSstatic} = a \times C \times f_{max} \times s \times V_{max} \times V \quad (13)$$

となる．

評価においては，$s$ を 0.11（静的消費電力がプロセッサの全消費電力の約 10%を占める）と仮定した．これは，最近のプロセス技術によるプロセッサの代表的な値である[15]．静的消費電力削減にあまり注力されていなかった数年前のプロセス技術では，$s$ は約 0.43〜0.67 程度（静的消費電力がプロセッサの全消費電力の約 30〜40%を占める）であったが[16]，最近では改良がなされ[17]，0.11 程度となっている．

PSU の消費電力 $P_{PSU}$ は，動的消費電力 $P_{PSUdynamic}$ と静的消費電力 $P_{PSUstatic}$ の和であり，以下の式で表される：

$$P_{PSU} = P_{PSUdynamic} + P_{PSUstatic} \quad (14)$$

$$P_{PSUdynamic} = P_{DVSdynamic} \times \left(1 - \frac{U-1}{U} \times m \times k\right) \quad (15)$$

$$P_{PSUstatic} = P_{DVSstatic} - P_{PRstatic} \times \frac{U-1}{U} \quad (16)$$

ここで，$m$ はプロセッサの全動的消費電力に対するクロック・ネットワークの動的消費電力の割合，$k$ はパイプライン・レジスタを駆動するクロック・ネットワークの最終段のクロック・ドライバの動的消費電力がクロック・ネットワークの全動的消費電力に占める割合，$P_{PRstatic}$ はパイプライン・レジスタの静的消費電力である．

PSU の動的消費電力は，パイプライン・ステージ統合によるパイプライン・レジスタへのクロック分配が抑制されることにより削減される．統合度 $U$ では，全パイプライン・レジスタの $(U-1)/U$ へのクロック分配が抑制される．全パイプライン・レジスタへのクロック分配に要する最終段クロック・ドライバの動的消費電力は，$P_{DVSdynamic} \times m \times k$ であるから，これに $(U-1)/U$ を乗じた値が削減される動的消費電力である．

PSU の静的消費電力は，パイプライン・ステージの統合時，バイパスされるパイプライン・レジスタをパワー・ゲーティングすることにより削減される．したがって，統合度 $U$ のとき，パイプライン・レジスタが消費する静的消費電力 $P_{PRstatic}$ に $(U-1)/U$ を乗じた値が削減される．

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 18)〜21) から得た商用プロセッサの $m$ の値（18%〜40%）のほぼ中央値である．また，$k$ は文献 19) に示されている 130 nm の Intel Itanium 2 の値より，88%を仮定した．$m$，$k$ の値は用いるプロセッサ・アーキテクチャ，半導体テクノロジ，クロック・ネットワークへの工夫により大きく変動する．また，今回は $P_{PRstatic}$ は保守的に 0 と仮定した．つまり，PSU によるパイプライン・レジスタのパワー・ゲーティングの効果は計算に反映させず，PSU には不利となるよう仮定した．

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．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%で動作するときのスループットを測定し，そのスループットの 10〜100%を 10%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．

後に 5.4 節で述べるが，アルゴリズムの実行によるオーバヘッドは非常に小さいため，この章の評価において，これを含めていない．

### 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．

- 最大クロック周波数：2 GHz
- 実行フェーズ：2 ms
- サンプリング・サブフェーズ：20 $\mu$s
- 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，DVS，PSU の場合の消費電力を示す．図の横軸は $TP_{target}$ を示し，縦軸は 100%のスループットのときの消費電力で正規化し

表 5 統合度 1，クロック周波数 100%での IPC
Table 5 IPC in unification degree 1 and 100% clock frequency.

| ベンチマーク | IPC |
|---|---|
| bzip2 | 2.67 |
| gcc | 1.39 |
| gzip | 1.25 |
| mcf | 0.32 |
| parser | 1.03 |
| perlbmk | 1.06 |
| vortex | 2.29 |
| vpr | 0.97 |

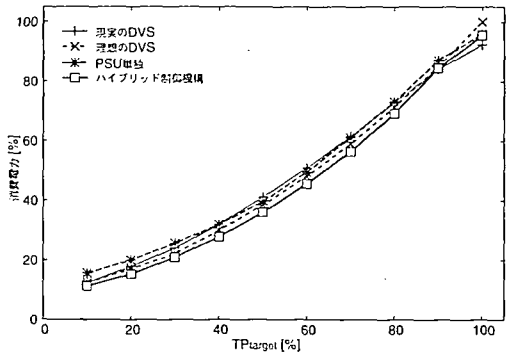



図 4 ハイブリッド制御機構，DVS，PSU の消費電力
Fig. 4 Power consumption of hybrid control mechanism, DVS, and PSU.

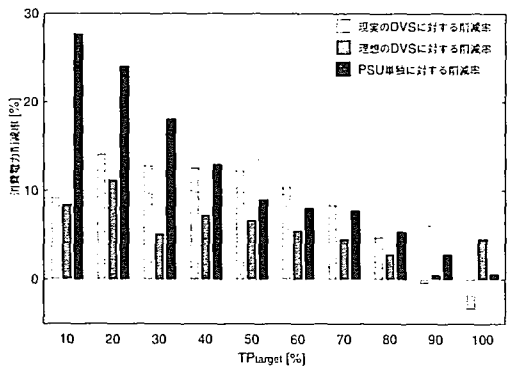



図 5 DVS，PSU に対するハイブリッド制御機構の消費電力削減率
Fig. 5 Power consumption reduction ratio of hybrid control mechanism compared to DVS and PSU.

た消費電力を示している．4 本の折れ線グラフは，現実の DVS，理想の DVS，PSU 単独，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する．DVS において，上記のような実行が最も消費電力を小さくすることは，文献 22) に示されている．また，PSU 単独とは，スイッチング・ポイント間のクロック周波数を選択した場合においても，電源電圧を変更しないものである．

図 4 より，100%の $TP_{target}$ の場合を除いて，ハイブリッド制御機構は他の 3 つの手法よりも消費電力を削減できることが分かる．100%の $TP_{target}$ では現実の DVS よりも消費電力が大きい．これは，後に 5.3 節で示すように，100%の $TP_{target}$ において，現実の DVS はハイブリッド制御機構よりも最終的なスループットが低く，その分だけ消費電力が低くなっているからである．また，同じ理由で，理想の DVS が達成した最終的なスループットは $TP_{target}$ を満たしているのに対し，その他の 3 つは満たしていないため，理想の DVS は 90%，100%の $TP_{target}$ において，他の 3 つの手法より多く電力を消費している．

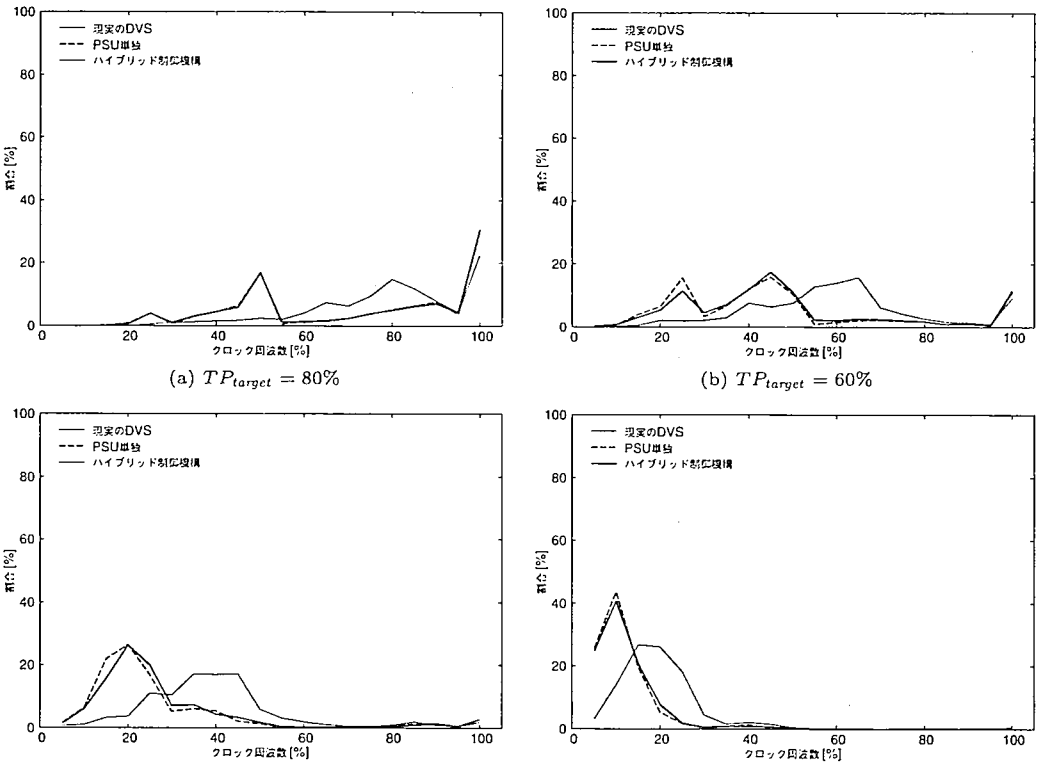
ハイブリッド制御機構の他の 3 手法に対する消費電力削減率を図 **5** に示す．概して，$TP_{target}$ が小さい部分でハイブリッド制御機構が消費電力が大きく削減されている．これは，低い $TP_{target}$ の方がより大きな統合度を利用できる局面が多いからである．また，$TP_{target}$ が大きい部分では，統合度 1 で動作する局面が増えるために DVS との差が少なくなる．また，$TP_{target}$ が小さい部分で PSU 単独に対する削減率が非常に大きくなるのは，図 4 からも見てとれるように，小さい $TP_{target}$ では，PSU の消費電力が大きいからである．この理由は，今回の評価では PSU には静的消費電力削減能力がないとしたため，$TP_{target}$ が小さい部分では静的消費電力によって消費電力の下限が抑えられているためである．

ハイブリッド制御機構の DVS に対する消費電力削減率は，$TP_{target} = 20\%$時において最大を示し，理想の DVS に対して 11%，現実の DVS に対して 14%と大きな削減率を達成した．また，PSU 単独に対する消費電力削減率は，$TP_{target} = 10\%$時において最大を示し，その大きさは 28%であった．

### 5.2 選択されたクロック周波数の内訳

5.1 節の測定において，現実の DVS，PSU 単独，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調べた．図 **6** に，$TP_{target}$ をそれぞれ 80%，60%，40%，20%に指定したときの分布を示す．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．

図より，現実の DVS ではそれぞれの $TP_{target}$ に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，ハイブリッド制御機構では PSU を多用するために，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．特に，図 6 (d) のように $TP_{target}$ が低い場合は，より高い統合度を利用することができるため，この傾向が大きい．逆に，図 6 (a) のように $TP_{target}$ が高い場合は，PSU を適用できない局面が多いため，現実の DVS に近い分布となっている．こ




(a) $TP_{target} = 80\%$ (b) $TP_{target} = 60\%$

(c) $TP_{target} = 40\%$ (d) $TP_{target} = 20\%$

図 6 種々の $TP_{target}$ に対する選択されたクロック周波数の分布
Fig. 6 Distribution of selected clock frequencies for varios $TP_{target}$.

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．なお，PSU 単独はハイブリッド制御機構とほぼ同じ形の分布となっている．

### 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 7 に $TP_{target}$ に対する達成率のベンチマーク平均を示す．$TP_{target}$ 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 7 より，ハイブリッド制御機構は $TP_{target} = 90\%$ 時に約 1% $TP_{target}$ を下回り，$TP_{target} = 100\%$時には 3%ほど下回る結果となった以外は，$TP_{target}$ を達成できている．なお，高い $TP_{target}$ において，$TP_{target}$ を達成できないことが問題な状況においては，$TP_{target}$ をやや高めに設定することにより，達成率を 100%以上にすることができる．

### 5.4 アルゴリズムの実行時間の評価

3.1 節で示したアルゴリズムを C 言語で記述し，4.1 節のベンチマーク・プログラムのコンパイル条件と

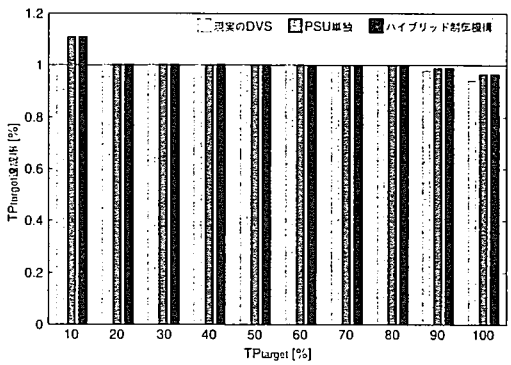



図 7 $TP_{target}$ 達成率
Fig. 7 Achievement rate of $TP_{target}$.

同じ条件でコンパイルし，得られたアセンブリ・コードより，アルゴリズムの実行サイクル数を評価した．また，得られた実行サイクル数より，アルゴリズムの実行時間を評価した．

アルゴリズムを実行する RISC コアを MIPS 4KE とし，以下の条件で評価した．

- 1 命令イン・オーダ発行の 5 段パイプラインで実行される．

- 分岐予測はなく，条件分岐命令がフェッチされた場合，実行完了までパイプラインはストールする．
- クロック周波数は400 MHzとする．これは，参考文献10)に示されている，130 nmにおけるMIPS 4KEコアの動作周波数340 MHzをベースとし，90 nmにシュリンクして約1.2倍になったものとした．

RISCコアのクロック周波数は400 MHzと，メイン・プロセッサの2 GHzのちょうど1/5を仮定している．よって，互いに同期してカウンタを更新/参照できる．

得られた実行サイクル数は以下のとおりである．なお，3.1節の箇条書きのアルゴリズムの(1)，(2)には計算が存在しないため，評価からは省いてある．

(1) 各サンプリング・サブフェーズ終了後の$(u, f_u, V_u)$の計算（アルゴリズムの(3)）：144サイクル

(2) 次の実行フェーズのパラメータの決定（アルゴリズムの(4)）：66サイクル

4章の評価条件と同様に3種類の統合度を利用することを仮定した場合，1組のサンプリング・フェーズと実行フェーズの実行においては，アルゴリズムの(3)が3回，(4)が1回実行されることになる．よって，この1組の実行において，アルゴリズムの実行サイクル数は498サイクルとなる．これを実行時間に換算すると1245 nsとなり，これは，実行フェーズの時間の約0.06%でしかない．アルゴリズム計算用コアは，アルゴリズムの実行を行うとき以外は停止するため，上記の実行時間しか電力を消費しない．よって，アルゴリズムの実行による消費電力の増加は，問題にならないほど小さいといえる．

### 5.5 アルゴリズムの実行時のオーバヘッド

本ハイブリッド制御機構では，DVSとPSUのモード切替えのオーバヘッドを被る．本評価では，これを考慮に入れなかったが，それはこのオーバヘッドが評価結果にほとんど影響を与えないためである．以下これを説明する．

一般に，DVSにおけるモード切替え（クロック周波数と電源電圧の変更）にはいくらかのオーバヘッドを被る．たとえば，Enhanced Intel Speed Step[23]では，10 $\mu$sの間プロセッサを停止させなければならない．

PSUのモード切替えのオーバヘッドは，DVSと同じくクロック周波数変更のための時間のほかに，統合信号の変更がプロセッサ全体に行きわたるまでのサイクル数である．後者は，多くても1000サイクルは超えないと容易に推測できるが，5.1節の評価パラメータでは，この場合，0.5 $\mu$sとなる．これは，クロック周波数変更のオーバヘッドに比べてきわめて小さいので，PSUのモード切替えのオーバヘッドとしては，クロック周波数変更についてのみ考えればよいことが分かる．つまり，たかだかDVSのモード切替えと同じだけのオーバヘッドがかかるといえる．

本ハイブリッド制御機構では，サンプリング・フェーズ開始時と終了時，および，各サンプリング・サブフェーズ間に，PSU，DVSのモード切替えのオーバヘッドを被る．今回の評価では3種類の統合度を仮定したので，1回のサンプリング・フェーズで3回のモード切替えが生じる．1回のモード切替えのオーバヘッドを10 $\mu$sとすると，1回のサンプリング・フェーズにおけるオーバヘッドは，合計で30 $\mu$sとなる．一方，本ハイブリッド制御機構の1フェーズ周期は，サンプリング・フェーズと実行フェーズからなり，オーバヘッドの重みは，この1フェーズ周期に対して判断されなければならない．今回の評価では，実行フェーズを2 ms，サンプリング・フェーズを40 $\mu$s（20 $\mu$sのサンプリング・サブフェーズが2回☆）としたので，1フェーズ周期に対するオーバヘッドの割合は，わずか1.5%でしかない．よって，オーバヘッドが本研究で行った評価に与える影響はきわめて小さい．

## 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，Koppanalilらは，我々に遅れてではあるが，Dynamic Pipeline Scaling（DPS）と呼ぶ方式を提案した[24]．このほか，Efthymiouらは，非同期プロセッサにおける方式を提案した[25]．

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[26]．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[27]．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

☆ 3.1節の最後の段落で述べたように，前の実行フェーズでもサンプリングを行うために，サンプリング・サブフェーズの回数は，用いる統合度の数より1回分少なくなる．

電力を削減することを提案した[28]．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する命令数を削減する．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[29]．

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[30]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っているもの[31]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[32),33)]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，および，以前に提案した PSU 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，次の期間の統合度とクロック周波数と電源電圧を決定し，目標とするスループットに適応しつつ，電力を削減するものである．

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，最大スループットの 10%から 100%の目標スループットにおいて，DVS のみ，および，PSU 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 11%，現実の DVS に対して 14%，PSU 単独に対して 28%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 on 90nm Process with 2MB L2 Cache Datasheet* (2006).
3) Advanced Micro Devices, Inc.: *AMD Athlon64 Processor Power and Thermal Data Sheet* (2006).
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, *Proc. Int. Symp. on Low Power Electronics and Design 2003*, pp.326–329 (2003).
7) 嶋田 創，安藤秀樹，島田俊夫：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，先進的計算基盤システムシンポジウム SACSIS2004, pp.11–18 (2004).
8) Shimada, H., Ando, H. and Shimada, T.: Hybrid Power Reduction Scheme Using Pipeline Stage Unification and Dynamic Voltage Scaling, *9th IEEE Symposium on Low-Power and High-Speed Chips (COOL Chips IX)*, pp.201–214 (2006).
9) Marr, D.T., Binns, F., Hill, D.L., Hinton, G., Koufaty, D.A., Miller, J.A. and Upton, M.: Hyper-Threading Technology Architecture and Microarchitecture, *Intel Technology Journal*, Vol.6, pp.1–12 (2002).
10) Snyder, C.D.: MIPS SoCs it to EPF 2001, *Microprocessor Report 2001/8/20*, Vol.34, Archive 1, pp.1–3 (2001).
11) Halfhill, T.R.: MIPS 24KE: Better Late Than Never, *Microprocessor Report 2005/5/31*, Vol.34, Archive 1, pp.7–9 (2005).
12) 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).
13) Glaskowsky, P.N.: Pentium 4 (Partially) Previewed, *Microprocessor Report*, Vol.14, Archive 8, pp.1–4 (2000).
14) Intel Corporation: *Intel Pentium M Processor Datasheet* (2004).
15) Hart, J., Choe, S.Y., Cheng, L., Chou, C., Dixit, A., Ho, K., Hsu, J., Lee, K. and Wu, J.: Implementation of a 4th-Generation 1.8 GHz Dual-Core SPARC V9 Microprocessor, *2005 IEEE Int. Solid-State Circuits Conf. Digest of Technical Papers*, pp.186–187 (2005).
16) 大石基之，進藤智則，堀切近史：リーク電流と闘う，日経エレクトロニクス 2004/4/26, No.872, pp.99–127 (2004).
17) Bai, P., Auth, C., Balakrishnan, S., Bost, M., Brain, R., Chikarmane, V., Heussner, R., Hussein, M., Hwang, J., Ingerly, D., James, R., Jeong, J., Kenyon, C., Lee, E., Lee, S.H., Lindert, N., Liu, M., Ma, Z., Marieb, T., Murthy, A., Nagisetty, R., Natarajan, S., Neirynck, J., Ott, A., Parker, C., Sebastian,

J., Shaheed, R., Sivakumar, S., Steigerwald, J., Tyagi, S., Weber, C., Woolery, B., Yeoh, A., Zhang, K. and Bohr, M.: A 65 nm Logic Technology Featuring 35 nm Gate Length, Enhanced Channel Strain, 8 Cu Interconnect Layers, Low-k ILD and 0.57 $\mu m^2$ SRAM Cell, *2004 International Electron Device Meeting Technical Digest*, pp.657–660 (2004).

18) Gowan, M.K., Biro, L.L. and Jackson, D.B.: Power Considerations in the Design of the Alpha 21264 Microprocessor, *the 35th Design Automation Conf.*, pp.726–731 (1998).

19) Anderson, F.E., Wells, J.S. and Berta, E.Z.: The Core Clock System on the Next-Generation Itanium Microprocessor, *2002 IEEE Int. Solid-State Circuits Conf. Visual Supplement to the Digest of Technical Papers*, pp.110–111 (2002).

20) Clark, L.T., Hoffman, E.J., Miller, J., Biyani, M., Liao, Y., Strazdus, S., Morrow, M., Velarde, K.E. and Yarch, M.A.: An Embedded 32-b Microprocessor Core for Low-Power and High-Performance Applications, *IEEE Journal of Solid-State Circuits*, Vol.36, No.11, pp.1599–1608 (2001).

21) Gronowski, P.E., Bowhill, W.J., Preston, R.P., Gowan, M.K. and Allmon, R.L.: High-Performance Microprocessor Design, *IEEE Journal of Solid-State Circuits*, Vol.33, No.5, pp.677–686 (1998).

22) Ishihara, T. and Yasuura, H.: Voltage Scheduling Problem for Dynamically Variable Voltage Processors, *Proc. Int. Symp. on Low Power Electronics and Design 1998*, pp.197–202 (1998).

23) McGregor, J.: x86 Power and Thermal Management, *Microprocessor Report 2004/12/6*, Vol.18, Archive 12, pp.1–6 (2004).

24) Koppanalil, J., Ramrakhyani, P., Desai, S., Vaidyanathan, A. and Rotenberg, E.: A Case for Dynamic Pipeline Scaling, *Proc. Int. Conf. on Compilers, Architecture, and Synthesis for Embedded Systems 2002*, pp.1–8 (2002).

25) Efthymiou, A. and Garside, J.D.: Adaptive Pipeline Depth Control for Processor Power-Management, *Proc. Int. Conf. on Computer Design 2002*, pp.454–457 (2002).

26) Albonesi, D.H.: Dynamic IPC/Clock Rate Optimization, *Proc. 25th Annual Int. Symp. on Computer Architecture*, pp.282–292 (1998).

27) Bahar, R.I. and Manne, S.: Power and Energy Reduction Via Pipeline Balancing, *Proc. 28th Annual Int. Symp. on Computer Architecture*, pp.218–229 (2001).

28) Manne, S., Klauser, A. and Grunwald, D.: Pipeline Gating: Speculation Control For Energy Reduction, *Proc. 25th Annual Int. Symp. on Computer Architecture*, pp.132–141 (1998).

29) Canal, R., Gonzalez, A. and Smith, J.E.: Very Low Power Pipelines using Significance Compression, *Proc. 33rd Annual Int. Symp. on Microarchitecture*, pp.181–190 (2000).

30) Pering, T., Burd, T. and Brodersen, R.: The Simulation and Evaluation of Dynamic Voltage Scaling Algorithms, *Proc. Int. Symp. on Low Power Electronics and Design 1998*, pp.76–81 (1998).

31) Krishna, C.M. and Lee, Y.-H.: Voltage-Clock-Scaling Adaptive Scheduling Techniques for Low Power in Hard Real-Time Systems, *IEEE Trans. Comput.*, Vol.52, No.12, pp.1586–1593 (2003).

32) 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, *Proc. 30th Annual Int. Symp. on Computer Architecture*, pp.14–25 (2003).

33) 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, *Proc. 35th Annual Int. Symp. on Microarchitecture*, pp.356–367 (2002).

34) Agarwal, A., Mukhopadhyay, S., Raychowdhury, A., Roy, K. and Kim, C.H.: Leakage Power Analysis and Reduction for Nanoscale Circuits, *IEEE Micro*, Vol.26, No.2, pp.68–80 (2006).

35) 桜井貴康：リーク電流はこう抑える (1)，日経エレクトロニクス 2004/9/13, No.882, pp.154–161 (2004).

36) Taur, Y. and Ning, T.: *Fundamental of Modern VLSI Devices*, Cambridge University Press (1998).

## 付　　録

### A.1 リーク電流 $I_{leak}$ の仮定

4.4 節において，$I_{leak}$ の値は，評価に用いた電源電圧の変動範囲では，電源電圧にかかわらず一定の値をとるものとした．この妥当性を，この節で説明する．

今回の評価で仮定した 90 nm のプロセス技術において，$I_{leak}$ の大部分はサブスレッショルド・リーク電流 $I_{sub}$ とゲート・リーク電流 $I_{gate}$ からなる．このうち，$I_{gate}$ については，90 nm のプロセス技術においては $I_{sub}$ の 1/10 程度であることが文献 34) で示されてお

り，十分に小さく，本研究では消費電力の計算から除外した．なお，現状のプロセスを単純に微細化していくとゲート・リーク電流は増えていくが，high-k 材料によるゲート絶縁膜の実用化によって $10^{-3}$〜$10^{-4}$ 倍になると見込まれており，将来のプロセス技術においても，大きな問題とならない可能性は高い[35]．

以下，$I_{sub}$ の大きさが電源電圧にかかわらず一定の値をとることを説明する．

CMOS のドレイン–ソース間電流 $I_{ds}$ は以下の式で表される[36]．

$$I_{ds} = \mu_{eff} \cdot C_{ox} \cdot \frac{W}{L}(m-1)\left(\frac{kt}{q}\right)^2 \times e^{q\cdot\frac{V_g - V_t}{mkT}}\left(1 - e^{-\frac{q}{kT}\cdot V_{ds}}\right) \quad (17)$$

上記の式中の定数と変数は，以下のとおりである．

| | | |
|---|---|---|
| $\mu_{eff}$ | : | 実効移動度 ($cm^2/V \cdot s$) |
| $C_{ox}$ | : | 単位面積あたりの<br>ゲート絶縁膜容量 ($F/cm^2$) |
| $W$ | : | チャネル幅 (cm) |
| $L$ | : | チャネル長 (cm) |
| $k$ | : | ボルツマン定数<br>($= 1.38 \times 10^{-23}$ J/K) |
| $T$ | : | 絶対温度 (K) |
| $q$ | : | 電子の電荷 ($= 1.6 \times 10^{-19}$ C) |
| $V_g$ | : | ゲート電圧 (V) |
| $V_t$ | : | 閾値電圧 (V) |
| $V_{ds}$ | : | ドレイン電圧 (V) |
| $m$ | : | ボディ効果定数 |

式 (17) の中で，電源電圧に応じて値が変化する変数は，$V_g$ と $V_{ds}$ のみである．このうち，$V_g$ は，$I_{sub}$ の計算時には 0 となるため，実質，変化する変数は $V_{ds}$ のみである．そのため，式 (17) 中の最も右の括弧内のみが電源電圧の変動とともに変化し，それ以外の部分は，まとめて 1 つの定数として見ることができる．さらに，$kt/q$ は室温 (25°C) で 26 mV，プロセッサ動作時に上昇した接合温度，たとえば，100°C では 32 mV であるため，式 (17) の最も右の括弧の値はほぼ 1 となる．よって，$I_{sub}$ は電源電圧にかかわらず一定ということがいえる．

(平成 18 年 7 月 21 日受付)
(平成 18 年 11 月 20 日採録)

嶋田 創（正会員）

1976 年生．1998 年名古屋大学工学部情報工学科卒業．2000 年名古屋大学大学院工学研究科情報工学専攻博士課程前期課程修了．2004 年名古屋大学工学博士．2004 年名古屋大学工学部電気系 COE 研究員．2005 年京都大学大学院情報学研究科特任助手．2006 年京都大学大学院情報学研究科助手．電子情報通信学会/情報処理学会 2003 年先進的計算基盤システムシンポジウム優秀学生論文賞受賞．計算機アーキテクチャの研究に従事．

安藤 秀樹（正会員）

1959 年生．1981 年大阪大学工学部電子工学科卒業．1983 年大阪大学大学院修士課程修了．京都大学工学博士．1983 年三菱電機（株）LSI 研究所．ISDN 用ディジタル信号処理 LSI，第 5 世代コンピュータ・プロジェクトの推論マシン用プロセッサの設計に従事．1991 年 Stanford 大学客員研究員．1997 年名古屋大学大学院工学研究科電子情報学専攻講師．1998 年名古屋大学助教授．1998〜2001 年東京大学大学院理学系研究科助教授併任．2004 年名古屋大学大学院工学研究科計算理工学専攻教授．1998 年，2002 年情報処理学会論文賞受賞．計算機アーキテクチャ，コンパイラの研究に従事．ACM，IEEE，電子情報通信学会各会員．

島田 俊夫（正会員）

1968 年東京大学工学部計数工学科卒業．1970 年東京大学大学院修士課程修了．同年電子技術総合研究所入所．1993 年より名古屋大学大学院工学研究科電子情報学専攻教授．2004 年より名古屋大学大学院工学研究科電子情報システム専攻教授．人工知能向き言語，LISP マシン，データフロー計算機，プロセッサ・アーキテクチャ，チップ内並列処理の研究に従事．最近はシステム LSI の研究を行っている．1988 年度市村賞，1994 年度情報処理学会論文賞，1995 年度注目発明賞，2001 年度情報処理学会論文賞，工学博士．