

## 負荷変動に瞬時適応可能なマルチパフォーマンスプロセッサの設計と評価

山口誠一郎<sup>†</sup> 大山裕一郎<sup>†</sup> 国武 勇次<sup>††</sup> 松村 忠幸<sup>†</sup> 石飛百合子<sup>†</sup>  
山口 聖貴<sup>†</sup> 李 東勲<sup>†</sup> 金田 裕介<sup>†††</sup> 舟木 敏正<sup>††</sup> 室山 真徳<sup>††††</sup>  
石原 亨<sup>††††</sup> 佐藤 寿倫<sup>††††</sup>

<sup>†</sup>九州大学大学院システム情報科学府 〒819-0395 福岡市西区元岡 744 番地

<sup>††</sup>九州工業大学院情報工学研究科 〒820-8502 福岡県飯塚市川津 680-4

<sup>†††</sup>九州大学工学部 〒819-0395 福岡市西区元岡 744 番地

<sup>††††</sup>九州大学システム LSI 研究センター 〒814-0001 福岡市早良区百道浜 3-8-33

E-mail: †{seiichiro,yuichiro,matsumura,ishitobi,masaki,donghoon}@c.csce.kyushu-u.ac.jp,

††{y-kunitake,t-funaki}@klab.ai.kyutech.ac.jp, †††kaneda@c.csce.kyushu-u.ac.jp ,

††††{muroyama,ishihara,tsato}@slrc.kyushu-u.ac.jp

あらまし 動的可変電圧プロセッサ (以下 DVS プロセッサ) の代用となるマルチパフォーマンスプロセッサについて述べる。マルチパフォーマンスプロセッサはプロセッサコア内に複数の PE (Processing Element) コアと連想度を動的に変更可能なキャッシュメモリを搭載する。PE コアはすべて同じ命令セットアーキテクチャを持つが消費エネルギーと動作性能が異なる。アプリケーションの負荷に応じて適応的に一つの PE コアとキャッシュメモリの連想度を選択することによりピーク性能を保ちつつプログラム実行に必要なエネルギー消費を削減することが出来る。PE コアの切り替えは約 1 $\mu$  秒で行うことが出来る。本稿では、商用の 90nm プロセスを利用して設計したマルチパフォーマンスプロセッサの実現例を紹介しその評価結果を報告する。

キーワード マイクロプロセッサ, 低消費電力設計

### An Adaptive Multi-Performance Processor and its Evaluation

Seiichiro YAMAGUCHI<sup>†</sup>, Yuichiro OYAMA<sup>†</sup>, Yuji KUNITAKE<sup>††</sup>, Tadayuki MATSUMURA<sup>†</sup>,  
Yuriko ISHITOB<sup>†</sup>, Masaki YAMAGUCHI<sup>†</sup>, Donghoon LEE<sup>†</sup>, Yusuke KANEDA<sup>†††</sup>, Toshimasa  
FUNAKI<sup>††</sup>, Masanori MUROYAMA<sup>††††</sup>, Tohru ISHIHARA<sup>††††</sup>, and Toshinori SATO<sup>††††</sup>

<sup>†</sup> Graduate School of IS & EE, Kyushu Univ. 744, Motoooka, Nishi, Fukuoka, 819-0395 Japan

<sup>††</sup> Graduate School of CS & SE, Kyushu Inst. of Tech. 680-4, Kawazu, Iizuka, Fukuoka, 820-8502 Japan

<sup>†††</sup> Department of EE & CS, Kyushu Univ. 744, Motoooka, Nishi, Fukuoka, 819-0395 Japan

<sup>††††</sup> System LSI Research Center, Kyushu Univ. 3-8-33, Momochihama, Sawara, Fukuoka, 814-0001 Japan

E-mail: †{seiichiro,yuichiro,matsumura,ishitobi,masaki,donghoon}@c.csce.kyushu-u.ac.jp,

††{y-kunitake,t-funaki}@klab.ai.kyutech.ac.jp, †††kaneda@c.csce.kyushu-u.ac.jp ,

††††{muroyama,ishihara,tsato}@slrc.kyushu-u.ac.jp

**Abstract** This paper presents an energy efficient processor which can be used as a design alternative for the dynamic voltage scaling (DVS) processors in embedded system design. The processor core consists of multiple PE (processing element) cores and a scalable set-associative cache memory. The major advantage over the DVS processors is a small overhead for changing its operating speeds. Our processor can change its speeds in 1 $\mu$  second while conventional DVS processors need hundreds of microseconds for the performance transition [1], [7].

**Key words** microprocessor, low power design

## 1. まえがき

携帯型情報端末の市場拡大を背景にコンピュータシステムの省エネルギー化が非常に重要な課題となっている。2000年には新たな省電力プロセッサとして動的可変電圧プロセッサ（以下DVSプロセッサ）が製品化されたが、テストコストの問題やタイミング設計にかかる手間、オンチップDC-DCコンバータのコスト、電圧・周波数切り替えのオーバーヘッドおよびこれに伴うリアルタイム性保証の困難さなどの理由から、DVSプロセッサがリアルタイム組込みシステムで利用された例は少ない。DVS制御を行う組込みOS（またはOSの一部の機能として電力管理をするソフトウェア）の例として、SpeedStep, Enhanced SpeedStep (Intel), PowerNow! (AMD), LongHaul (VIA Technologies), LongRun, LongRun2 (Transmeta), SmartReflex (TI), IEM (ARM), XEC (Freescale Semiconductor) などがある。何れもリアルタイムシステムを主なターゲットとはしていない。リアルタイムシステムとは、決められた制約時間内に与えられたタスクを完了するコンピュータシステムのことを意味する。あるいは、厳密に時間制約を守れなくても、タスクの最悪実行時間が制約時間よりも小さくなるように設計されたシステムのことを指す。例えばTransmeta社のLongRunは、プロセッサのサイクルあたりの実行命令数（IPC）の値を実行時に観測し、IPC値が小さい場合は、プロセッサに高い性能が求められていないと判断し、電圧を下げて省エネルギー化を図る。逆にIPC値が大きい場合は、プロセッサに高い性能が求められていると判断し電圧を上げて高速化する。この方法はリアルタイム性を必ずしも保障しないため、スループット重視のシステムの省エネルギー化には有効であるがリアルタイムシステムには適用が難しい。本稿では、DVSプロセッサに代わるプロセッサとしてマルチパフォーマンスプロセッサ [6] のアーキテクチャを提案し、最小機能を搭載したプロセッサチップの設計例とその評価結果について述べる。マルチパフォーマンスプロセッサは既存のDVSプロセッサに対して性能切り替えのオーバーヘッドを約2桁小さくすることを可能にした。

## 2. マルチパフォーマンスプロセッサ

### 2.1 アーキテクチャ

マルチパフォーマンスプロセッサは同じ命令セットアーキテクチャを持つ複数の演算要素（以下PE）と動的に実効容量を変更できるキャッシュメモリとローカルメモリから構成される（図5参照）[6]。各PEはすべて同一の命令セットアーキテクチャを持つが、異なる消費エネルギー特性と動作性能を持つ。一般には低消費エネルギーと高速動作はトレードオフの関係にあるため、高速で消費エネルギーの大きいPEと低速で消費エネルギーの小さいPEをCPUコア内に複数搭載し、状況に応じて適切なPEを稼働させる。同時には一つのPEのみが稼働する。使用しないPEへは信号供給とクロック供給を遮断する。長時間使用しないことがあらかじめ分かっているPEにはパワーゲーティングによって電流供給を遮断しリーク電流を削減する。キャッシュメモリやローカルメモリはPE間で共有す

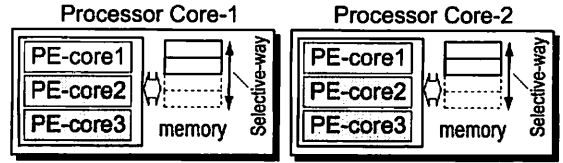


図1 マルチパフォーマンスプロセッサの概略図

る。ただし、同時に一つのPEのみが動作するため、メモリの入出力ポートはそれぞれ1ポートのみである。また、PEはすべて同じ命令セットアーキテクチャを持つため、PEを切り替えることによりソフトウェアからは1つのPEが動的にその性能と消費エネルギーを変更しながら動作しているように見える。従ってマルチパフォーマンスプロセッサは、マルチプロセッサでも命令レベル並列プロセッサでもない。タスクレベルの並列性が要求されるアプリケーションにはマルチパフォーマンスプロセッサコアを並列にバス接続することにより並列化する。

PEを切り替える際には、PEの内部状態（レジスタの値）を現在稼働中のPEから次に稼働するPEへ移動させる必要がある。ローカルメモリを経由して移動させる方法と直接専用バスで移動させる方法が考えられる（図2参照）。ハードウェアコストとPEコアの切り替え時間のトレードオフを考慮してレジスタ状態の移動方法を決定する必要がある。

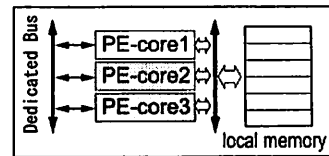


図2 マルチパフォーマンスプロセッサの内部状態切り替え

### 2.2 開発の動機

マルチパフォーマンスプロセッサはリアルタイムシステムを主なターゲットとしている。リアルタイムシステムは与えられたソフトウェアタスクを決められた時間内に完了すれば良く、それ以上の性能は消費エネルギーの観点からは無駄となる。従って、プロセッサは必要最小限の性能とエネルギーでタスクを処理し、決められた時間ちょうどに終了することが望ましい。しかし、タスクの実行時間は表1に示すとおり入力データに依存して大きく変化する[4]。表1のBCET (Best Case Execution Time) は最短実行時間を意味し、WCET (Worst Case Execution Time) は最悪実行時間を意味する。例えばバブルソートのプログラムではBCETとWCETで約3600倍値が異なる。リアルタイムシステムは通常WCETでプログラムが実行された場合でも、プログラムが制限時間内に終了するように設計されるため、多くの場合プロセッサは計算能力をもちあますことになる。

文献[5]では、あるタスクがそのWCETより早く終了した場合に、余ったCPU時間を後続のタスクに割り当てて、電源電圧とクロック周波数を低減することによりプロセッサの消費エネルギーを削減するアルゴリズムが提案されている。後続タスクが常に余ったCPU時間を利用できれば良いが、あるタスク

表 1 プログラムの実行時間

Program	Description	BCET	WCET
DES	Deata Encryption	73,912	672,298
DJPEG	JPEG decoder	12.7M	122.8M
FFT	1024-point FFT	1.59M	3.97M
Sort	Bubble sort of 500 elements	14.0K	50.2M

が WCET より早く終了しても、後続のタスクが常に実行可能状態であるとは限らないため、余った CPU 時間を有効に使えない場合がある。この問題を解決するために文献 [3] では、タスクをいくつかのスライスと呼ばれるサブタスクに分割し、スライス毎にタスクの進行状況をチェックすることにより、きめ細かく CPU 時間を利用する方法が提案されている。タスクが実行可能状態であれば、各スライスはすべての先行スライスが終了した時点で実行可能となるため、先行スライスが WCET より早く終了することによって発生した CPU 時間は後続スライスで有効に使用することが出来る。文献 [3] の方法は結果的に約数ミリ秒～10 ミリ秒毎にプロセッサの電源電圧と動作周波数を切り替えることになる。文献 [3] では切り替え時間を 500 $\mu$ 秒としている。数ミリ秒のスライスに対して 500 $\mu$ 秒のオーバーヘッドは無視できない。電源電圧の切り替え中にプロセッサを動作させることも可能であるが、この場合、電圧切り替え中の正常動作を保障するためのテストが必要となりコストがかかる。

リアルタイムシステムには最悪タスク応答時間が一定値より小さいことも要求される。ここでタスクの応答時間とはタスクが実行可能となった時点からそのジョブが完了する時点までに経過した時間と定義する。例えば自動車のエンジン制御には数十 $\mu$ 秒から数百 $\mu$ 秒の応答時間が要求される。最近では携帯電話にも数ミリ秒程度の最悪タスク応答時間が期待されている。例えばメールの編集中に電話の着信が入った場合に、速やかに電話の着信処理に移行する必要があるためである。携帯電話のように時間厳守が求められない、いわゆるソフトリアルタイムシステムでも数百 $\mu$ 秒のオーバーヘッドは無視できない。結果として CPU 時間に余裕が出来ても簡単には電源電圧とクロック周波数を下げられない。

マルチパフォーマンスプロセッサは性能と電力の切り替えにかかるオーバーヘッドを小さくすることを目標に考案したプロセッサである。3章で述べるマルチパフォーマンスプロセッサのプロトタイプでは、切り替えにかかる時間が約 1 $\mu$ 秒で、エネルギー消費は約 10nJ と非常に小さい。

### 2.3 DVS プロセッサとの比較

DVS プロセッサとは動作電圧と動作周波数を稼働時に変更することが出来るプロセッサである。表 2 に商用の DVS プロセッサとその電圧切り替え時間を示す [7]。

100 $\mu$ F のキャパシタ容量を持つ DC-DC コンバータが例えば 0.6V から 1.0V に電圧を昇圧するのにかかる時間は約 80 $\mu$ 秒で、エネルギー消費は 6.4 $\mu$ J になる [2]。これは、DC-DC コンバータが電圧切り替え時のノイズを低減するために、大容量 (典型的には 100 $\mu$ F) のキャパシタを内蔵しており、電圧の昇降圧時にはこのキャパシタを充放電する必要があるためである [2]。実際にはチップ自体が大きな容量を持つため、オーバー

表 2 商用の DVS プロセッサとその電圧切り替え時間

Processor	Voltage (V)	Transition Time
Transmeta Crusoe	1.1-1.65	300 $\mu$ s
AMD Mobile K6	0.9-2.0	200 $\mu$ s
Intel PXA250	0.85-1.3	500 $\mu$ s
Compaq Itsy	1.0-1.55	189 $\mu$ s
TI TMS320C55x	1.1-1.6	3.2ms (1.6 $\rightarrow$ 1.1V) 300 $\mu$ s (1.1 $\rightarrow$ 1.6V)
UCB [2]	1.2-3.8	520 $\mu$ s

ヘッドはさらに大きくなる。特に消費電力の小さい組み込みプロセッサに DVS 制御を適用する場合、電圧切り替えにかかる 6.4 $\mu$ J のエネルギー消費は無視できない。例えば 90nm プロセスで設計した組み込みプロセッサは周波数範囲が $\sim$ 400MHz 程度であり、この時の消費電力は $\sim$ 100mW 程度である。従ってクロックあたりの消費エネルギーは約 250pJ となる。電圧切り替えの際に DC-DC コンバータで消費される 6.4 $\mu$ J のエネルギーはプロセッサの 2 万 5 千クロックサイクル分に相当するエネルギーである。このオーバーヘッドはパーソナルコンピュータやサーバなどで使用されるハイエンドのプロセッサにとっては比較的小さいが、組み込みプロセッサにとっては無視できない。

### 2.4 ヘテロジニアスマルチプロセッサとの比較

図 3 に示すように、命令セットアーキテクチャが同じで性能と消費エネルギーの異なるプロセッサコアを複数搭載した単一命令セットヘテロジニアスマルチプロセッサを考える。MPU1 は MPU2 に比べて性能、消費エネルギー共に大きいものとする。

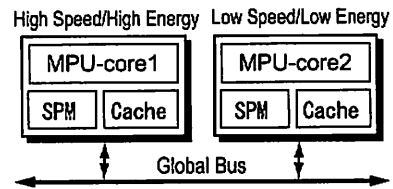


図 3 ヘテロジニアスマルチプロセッサの例

図 4 で示すように MPU1 と MPU2 がそれぞれタスク 1 とタスク 2 を実行中に、タスク 2 がそのデッドラインより大幅に早く終了した場合を想定する。タスク 1 の残り分を、性能の低い MPU2 で実行してもデッドラインまでに間に合う場合は、MPU2 にタスク 1 を移動させて実行することにより消費エネルギーを削減できる可能性がある。ただし、タスク 1 がスクラッチパッドメモリやキャッシュメモリなどの MPU1 と密に接続されたメモリを使用しているとすると、タスク 1 の移動先での性能劣化を防ぐためには、タスクの移動に合わせてこれらのメモリの内容も移動させることが望ましい。スタック領域をスクラッチパッドメモリに配置している場合は MPU1 の内部状態 (レジスタの内容) と共にスタックの内容も移動させる必要がある。3章で述べるプロトタイププロセッサを使用した実験では、256 バイトのデータを 67MHz の AMBA AHB 仕様に準拠するバスを使って MPU1 のスクラッチパッドメモリから MPU2 のスクラッチパッドメモリへバースト転送した場合、約 3.5 $\mu$ 秒の時間と約 300nJ のエネルギーを要する。例えば 4

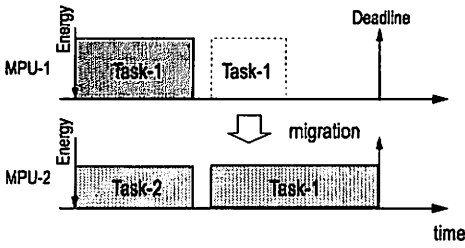


図4 動的タスクマイグレーションの例

章で述べる実験で使用した高速フーリエ変換 (FFT) のプログラムは 2,392 バイトのスタックを使用するため、単純計算ではスタックの移動に約 35 $\mu$  秒の時間と約 3 $\mu$ J のエネルギーを要することになる。これは DVS プロセッサが電源電圧を切り替えるオーバーヘッドとそれほど違わない大きさである。命令キャッシュやデータキャッシュの内容を移動させる場合はさらに大きなオーバーヘッドが必要となるため現実的ではない。結果的にタスク 1 を MPU1 から MPU2 に移動させて実行すると MPU2 のキャッシュヒット率は低下し、性能、消費エネルギー共に劣化する場合がある。

### 3. プロトタイププロセッサの設計

#### 3.1 回路構成

東芝社製の 32/16 ビット可変命令語長 RISC プロセッサ MeP をベースにマルチパフォーマンスプロセッサのプロトタイプを設計した。プロセステクノロジーは商用の 90nm CMOS プロセスを使用した。CPU コア部分は同一の回路を複製し、性能と消費エネルギーの異なる PE コアとして実現する。各 PE コアは RT レベルでは同一の回路記述であるが別々の電源電圧を使用してキャラクタライズした標準ライブラリを用いてそれぞれ設計する。それぞれ 1.0V、0.68V、0.52V を使って 3 種類の PE コアを設計した。MPU コア 0 には 3 種類の PE コアを、MPU コア 1 には 1.0V と 0.68V を使用する PE コアを、MPU コア 2 には 1.0V と 0.52V を使用する PE コアをそれぞれ搭載した。PE コアの切り替えはスクラッチパッドメモリの特定のアドレスに特定の値を書き込むことにより行う。メモリは 1.0V を使用し、すべての PE コアで共有する。ただし、PE コアは同時に一つしか稼働しないためメモリはすべてシングルポートで実現する。今日のマイクロプロセッサチップは面積の大部分をメモリが占有するため、MPU コア自体を複数搭載するよりは面

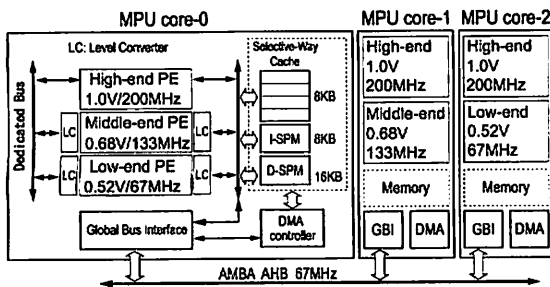


図5 マルチパフォーマンスプロセッサのプロトタイプ

積効率が良い。マルチパフォーマンスプロセッサのプロトタイプでは、オンチップメモリの面積がおおよそ 952,000 $\mu$ m<sup>2</sup> であるのに対し、PE コア一つの面積がおおよそ 207,000 $\mu$ m<sup>2</sup> であるため、PE コアの面積はメモリ部の面積のおおよそ 22% である。

オンチップメモリは 8K バイト 4 ウェイセットアソシアティブの命令キャッシュと 8K バイトの命令コード用スクラッチパッドメモリおよび 16K バイトのデータ用スクラッチパッドメモリを搭載する。命令キャッシュは 4 ウェイのうち、稼働させるキャッシュウェイを選択することが出来る。キャッシュウェイの選択はデータ用スクラッチパッドメモリの特定のアドレスに特定の値を書き込むことにより行う。つまり、キャッシュの選取度を動的に変更できるだけでなく、プログラムの動作状況に応じてどのキャッシュウェイを使うかをプログラマが指定出来る。

#### 3.2 設計フロー

多電源電圧設計を可能とするために、数種類の異なる電圧に対して標準セルのキャラクタライズを行った。キャラクタライズには Cadence 社の SignalStorm を使用した。具体的には、0.5V、0.52V、0.55V、0.68V、0.7V、0.72V、0.75V、1.0V でキャラクタライズを行った。次の手順で設計を行った。

(1) 1.0V の標準セルライブラリを使用して PE コアの論理合成を行い、PE コアのターゲット周波数を決定する。論理合成には SYNOPSIS 社の DesignCompiler を使用した。プロトタイプ設計では、選延制約が 5ns のときに PE コアの選延制約がすべて満たされたため、200MHz を最大動作周波数とした。

(2) 200MHz の 3 分の 1 (67MHz) をバスクロック周波数とし、最も低い電圧を使用する PE コアの動作周波数も 67MHz とした。

(3) 0.5V、0.52V、0.55V の 3 種類のライブラリを使用して PE コアをそれぞれ設計し、PE コアの消費エネルギーを最小化するライブラリを最終的に選別した。プロトタイプでは 0.52V を使用することとした。

(4) バスクロックの 2 倍の周波数 (133MHz) で動作する PE コアを 0.68V、0.7V、0.72V、0.75V の 4 種類のライブラリを使用して合成し、PE コアの消費エネルギーを最小化するライブラリを選別した。プロトタイプでは 0.68V を使用することとした。

PE コアの周波数がバスクロックの整数倍になるようにしたのは、バスクロック周波数の整数倍のクロックで動作する MPU のみを接続できる仕様でバスを設計したためである。キャラクタライズの際に、出力端子の信号遅移時間が 10ns を超える論理素子はライブラリから除外した。結果として、低電圧のライブラリを使用すると回路面積が大きくなる傾向にある。

#### 3.3 レベルコンバータ回路

低電圧 PE コアとオンチップメモリ、および PE コア間はレベルコンバータを使用して信号振幅を補正する。レベルコンバータは文献 [8] でチャンらによって提案された回路を参考に自作した (図 6 参照)。この回路を使用することにより 0.52V から 1.0V までの昇圧が最小インバータ回路約 5 段分の遅延で実現できた。スイッチング電力は 1.0V を使用した時の最小インバータ回路の約 2.5 倍、リーク電流は約 1.5 倍であった。レベ

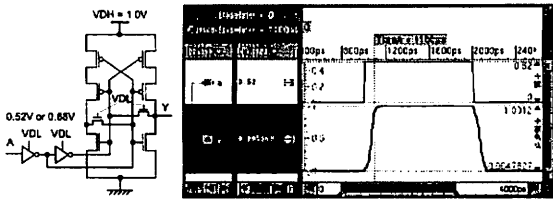


図 6 レベルコンバータ

ルコンバータは標準セルと同様に Cadence 社の SignalStorm を使用してキャラクタライズを行い、プロセッサの論理合成時にパス遅延解析と電力解析が出来るようにした。

### 3.4 DVS プロセッサ設計との比較

各 PE コアは使用するそれぞれの電源電圧に最適化されるため、従来型の DVS プロセッサよりも低電圧動作時のパフォーマンスが高い。図 7 は、1.0V で論理合成した PE コアの最長パスを抽出し、そのパスに対する電圧・遅延特性を SYNOPSIS 社の HSPICE で計測した結果である。最長パスは SYNOPSIS 社の静的遅延解析ツール (STA) で抽出した。1.0V 動作時には 5ns のパス遅延は、0.68V の電源電圧では 12.5ns まで増大する。この場合、プロセッサの動作周波数は 80MHz になる。筆者らの設計したマルチパフォーマンスプロセッサは 0.68V で 133MHz を実現する。さらに 0.52V まで電圧を下げるとパス遅延は 37ns になり、プロセッサは 27MHz でしか動作しない。多入力論理セルと高い閾値電圧を使用する論理セルは低電圧動作時に遅延が急増するためだと考えられる。3.2 節で述べたとおりマルチパフォーマンスプロセッサはそれぞれの電圧に合わせたセルライブラリを使用して設計しているため、低電圧動作時でも回路のクリティカルパス遅延はそれほど大きくならない。

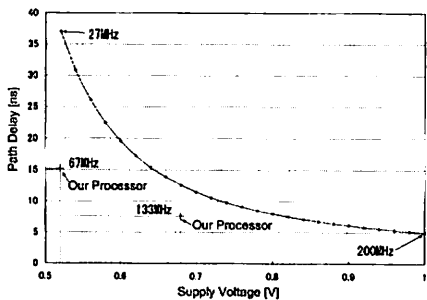


図 7 電圧・遅延特性

## 4. 実験と評価

本章で述べる消費電力と消費エネルギーの結果は、レイアウト前のネットリストに対してゲートレベルシミュレーションを行うことによって得た。ゲートレベルシミュレーションには Cadence 社の Verilog-XL を用いた。オンチップメモリの消費電力および回路遅延は SYNOPSIS 社の NanoSim を使用して計測した。その他の回路の消費電力は、ゲートレベルシミュレーションで生成したスイッチング情報 (SAIF: Switching Activity Interchange Format) と標準セルの電力・遅延情報を入力として、SYNOPSIS 社の PowerCompiler で見積もった。プロセ

表 3 回路規模 (mm<sup>2</sup>)

MPU Core-0			MPU Core-1			MPU Core-2		
PEs	RAM	other	PEs	RAM	other	PEs	RAM	other
0.71	0.95	0.58	0.44	0.95	0.34	0.48	0.95	0.33
2.24			1.74			1.77		
Total Area of Prototype Processor = 5.88								

表 4 閾値電圧別の面積 (mm<sup>2</sup>)

MPU Core-0		MPU Core-1		MPU Core-2	
High V <sub>th</sub>	Low V <sub>th</sub>	High V <sub>th</sub>	Low V <sub>th</sub>	High V <sub>th</sub>	Low V <sub>th</sub>
0.72	1.32	0.53	1.21	0.49	1.28
High V <sub>th</sub>				Low V <sub>th</sub>	
1.83				4.05	

表 5 性能切り替えのオーバーヘッド

切り替え方向	切り替え時間 [ns]	エネルギー [nJ]
High → Middle	1,113	11.84
High → Low	1,290	11.25
Middle → High	968	13.04
Middle → Low	1,443	9.27
Low → High	1,205	13.39
Low → Middle	1,286	8.93
Cache Way	690	10.35

ス条件は Typical, 温度は 75 °C を想定した。PE コアの切り替え、キャッシュウェイ選択および MPU コア間データ転送を含む全ての機能が正常に動作することをゲートレベルシミュレーションにより確認した。

### 4.1 回路規模

設計したプロセッサの回路規模を表 3 に示す。表の値は配線面積を含んでおらず、単純に論理セルと SRAM モジュールの面積を足し合わせたものである。3MPU 構成のマルチプロセッサ全体を見ると RAM の面積がおおよそ 50% を占める。表 4 には閾値電圧別の回路面積を示した。設計には 2 種類の閾値電圧を使用する標準セルライブラリを使用した。低い閾値電圧を使用する論理セルは高い電圧を使う同じ論理セルに比べて約 6 倍のリーク電流を消費する。従って省エネルギー化のためには高い閾値電圧を使用するセルの割合が大きいために望ましい。なお、オンチップメモリはすべて高い閾値電圧を使用する。表 4 に示す結果ではプロセッサ全体の約 70% が高い閾値電圧を使用している。単純計算では、すべて低い閾値電圧を使用した場合と比較してリーク電流を 60% 削減していることになる。

### 4.2 性能切り替えオーバーヘッド

表 5 に PE コア切り替え時とキャッシュウェイ切り替え時のオーバーヘッドを示した。DVS プロセッサと比較すると切り替え時間、消費エネルギー共に約 2 桁小さくなることを確認した。命令キャッシュのウェイ選択も同程度のオーバーヘッドで実現可能であることを確認した。

### 4.3 消費電力

図 8, 9, 10 にそれぞれ、ADPCM デコーダ、FFT、DCT を MPU コア 0 上で実行した場合の消費電力を示す。それぞれのベンチマークプログラムのメインループが存在する関数は命令用のスクラッチパッドに配置し、スタック領域と入力用データ

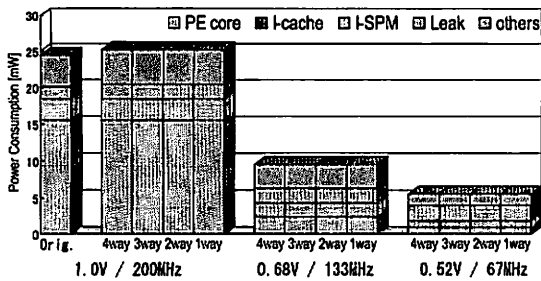


図 8 ADPCM デコーダ実行時の消費電力

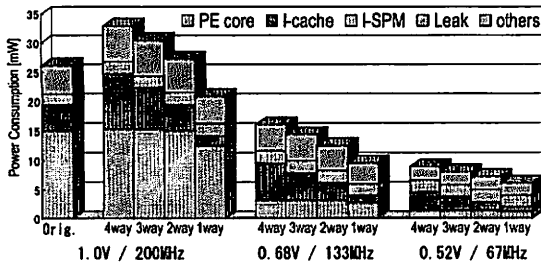


図 9 高速フーリエ変換 (FFT) 実行時の消費電力

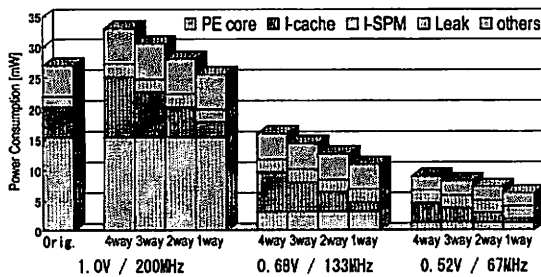


図 10 離散コサイン変換 (DCT) 実行時の消費電力

はデータ用スクラッチパッド上に配置した。図の結果は PE コアの種類と稼働させるキャッシュウェイの数を変化させた際の値を示している。各図の最も左の棒グラフは変更を加える前の MeP プロセッサの消費電力結果である。PE コアの消費電力が電源電圧の 2 乗と周波数にほぼ比例していることが確認できる。ADPCM は使用する命令コードとデータのほとんどがスクラッチパッドメモリ上に載っているためキャッシュアクセスがほとんど発生していない。このためプロセッサの消費電力はキャッシュのウェイ数にほとんど依存しない。

図 11 に消費エネルギーと実行時間の結果を示した。棒グラフが消費エネルギーを示し、折れ線グラフが実行時間を示している。エネルギー消費の観点では、0.68V/133MHz の PE コアを使用した場合が常に最も良い結果となっている。キャッシュのウェイ数に関しては FFT では 2 ウェイ、DCT ではダイレクトマップの時に最良の結果を得た。

## 5. あとがき

Intel 社で 80 コアを搭載するプロセッサが試作され、ハードウェアの実装技術の観点ではメニーコアの時代に突入した。しかし、80 の並列性が存在するアプリケーションは現状では少

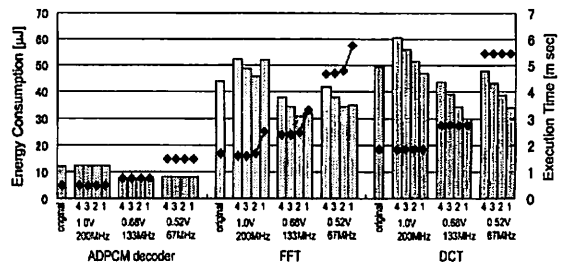


図 11 消費エネルギーと実行時間

ない。一方で、コンピュータシステムに必要とされる機能は多様化し、同じアプリケーションでも、高い速度性能を要求する場面と低速動作が可能な場面が混在する。ピーク性能を維持したまま消費エネルギーを低減するためには、性能と消費エネルギーの観点からヘテロジニアスな PE を状況に応じて瞬時に切替えて使用するアーキテクチャが有効である。

今後はチップの配置配線を行い実配線遅延に基づく評価を行うと共に、名古屋大学高田広章教授のグループが提唱するリアルタイム OS による電力管理技術 DEPS (Dynamic Energy Performance Scaling) の評価を行う予定である。

謝辞 本研究は株式会社東芝および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 (株) 半導体理工学研究センター、富士通 (株)、NEC エレクトロニクス (株)、(株) 東芝、(株) ルネサステクノロジ、ケイデンス (株)、シノプシス (株) の協力で行われたものである。貴重なご意見を頂いた CREST ULP 高田グループの諸氏に感謝する。本研究の一部は JST CREST-ULP によるものである。

## 文 献

- [1] N. Allah, Y. Wang, J. Xing, W. Nisar and A. Kazmi, "Towards Dynamic Voltage Scaling in Real-Time Systems - A Survey," *Int'l Journal of Computer Sciences and Engineering Systems*, Vol.1, No.2, pp.93-104, Apr. 2007.
- [2] T. Burd, and R. W. Brodersen, "Design Issues for Dynamic Voltage Scaling," in *Proc. of Int'l Symposium on Low Power Electronics and Design*, pp.9-14, Jul. 2000.
- [3] S. Lee, and T. Sakurai, "Run-time Voltage Hopping for Low-power Real-time Systems," in *Proc. of Design Automation Conference*, pp.806-809, Jun. 2000.
- [4] Y.-T.S. Li and S. Malik, "Performance analysis of embedded software using implicit pathenumeration," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol.16, Issue 12, pp.1477-1487, Dec. 1997.
- [5] T. Okuma, T. Ishihara, and H. Yasuura, "Real-Time Task Scheduling for a Variable Voltage Processor," in *Proc. of Int'l Symposium on System Synthesis*, pp.24-29, Nov. 1999.
- [6] Y. Oyama, T. Ishihara, T. Sato and H. Yasuura, "A Multi-Performance Processor for Low Power Embedded Applications," in *Proc. of CoolChips 2007*, pp.138, Apr. 2007.
- [7] D. Shin and J. Kim, "Intra-task voltage scheduling on DVS-enabled hard real-time systems," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol.24, Issue 10, pp.1530-1549, Oct. 2005.
- [8] チャン・クワン・カイン, 桜井貴康, "低電圧対応のレベルコンバータ," 電子情報通信学会総合大会予稿集, pp.8, 2004 年 3 月.