

コア温度の変化に適応する OSによる細粒度パワーゲーティング制御方式

木村 一樹^{†1} 近藤 正章^{†5} 天野 英晴^{†4}
宇佐美 公良^{†6} 中村 宏^{†3}
佐藤 未来子^{†2} 並木 美太郎^{†2}

本論文では、省電力プロセッサコア Geysler において、ソフトウェアのサポートにより電力削減効果を高めるための一方式を提案する。提案方式では、コアの温度に適応して電力オーバーヘッドが少なくなるようスリープの時間粒度を OS が判定する。その制御を行うためのタスクごとの閾値温度について、アドミッションテストにより事前に決める方法と、実行時に取得した情報により決める方法の二種類を適用した。実際に FPGA ボード上に構築した評価基盤上で OS に実装し評価を行い、演算ユニット全体の平均リーク電力を最大約 12%削減することを確認した。

Fine Grain Power Gating Control Adapting to a Change in Core Temperature with Operating System

KAZUKI KIMURA,^{†1} MASAOKI KONDO,^{†5}
HIDEHARU AMANO,^{†4} KIMIYOSHI USAMI,^{†6}
HIROSHI NAKAMURA,^{†3} MIKIKO SATO^{†2}
and MITARO NAMIKI^{†2}

This paper describes a method using software that is for achieving further power saving on the processor core 'Geysler'. The proposed method is that controls the temporal granularity of power gating to reduce overhead power consumption, adapting to the CPU core temperature with OS support. To determine temperature thresholds for each tasks, two methods, deciding beforehand by admission test and deciding by runtime information are applied. The result of the evaluation of the method implemented in OS using FPGA-board, the leakage power consumption of computing units has been reduced by up to 12%.

1. はじめに

計算機工学の分野では近年、消費電力削減技術の確立が大きな課題の一つとなっている。特にシステム LSI では、処理性能の著しい向上を実現するためにクロックの高速化や製造プロセスの微細化などの技術革新が行われてきたが、消費電力の増大とそれに伴う発熱の問題から、更なる性能向上が限界に近づいているという状況にある。このような背景から、科学技術振興機構 (JST) の戦略的創造研究推進事業「CREST」におけるプロジェクト「情報システムの超低消費電力化を目指した技術革新と統合化技術」¹⁾ では、省電力プロセッサコア Geysler²⁾、および、電力削減効果を高めるためのシステムソフトウェア^{3),4)} の研究を進めている。Geysler の様な演算ユニットに自律的なパワーゲーティング機能を持つプロセッサは、従来より多く提案されている⁵⁾⁻⁷⁾。このようなプロセッサでは、パワーゲーティングによる頻繁なスリープとウェイクアップが大きな電力オーバーヘッドとなる。そのためパワーゲーティングにおけるスリープ時間の粒度をソフトウェアから制御し、電力オーバーヘッドを抑えて電力削減効果をより向上させるための研究として、コンパイラによりスリープ時間の粒度の最適化を図る提案も行われている⁸⁾⁻¹⁰⁾。

筆者らは、OS における動的なスリープ時間粒度の最適化方式について研究しており、これまでに、プロセッサのパワーゲーティング実施時の電力的特性が演算ユニットの置かれたコア温度により変動する点に着目し、温度に適応する OS による制御手法を提案している⁴⁾。今回、パワーゲーティングを適用する演算ユニットのスリープ頻度がタスクの特性によって異なる点を考慮しながら、スリープの時間粒度を変更するための閾値温度を決める新たな方

†1 東京農工大学/株式会社モルフォ
Tokyo University of Agriculture and Technology/Morpho, Inc.

†2 東京農工大学
Tokyo University of Agriculture and Technology

†3 東京大学
The University of Tokyo

†4 慶應義塾大学
Keio University

†5 電気通信大学
The University of Electro-Communications

†6 芝浦工業大学
Shibaura Institute of Technology

法を検討した。閾値温度を決めるための方法として検討したのは、アドミッションテストにより事前に決定する方法と、実行時の情報により決定する、二種類の方法である。本稿では、それらの制御機構の設計を示し、本機構を実装した OS について述べ、FPGA ボード上に構成した Geysler をコアとする評価基盤上で、MiBench に含まれるいくつかのベンチマークプログラムを用いた評価を行い、二種類の方法を比較した結果を示す。

2. 課題と目標

本章では、先行研究におけるパワーゲーティング制御方式の課題と、本研究の目標を示す。

2.1 既存研究と課題

従来の電力削減のためのシステムソフトウェアレベルの取り組みとして、DVFS や ACPI を始めとするソフトウェアから制御可能な電源、クロック管理のインタフェースを利用して、省電力化を図る取り組みも行われている^{11),12)}。一方、プロセッサに備えるパワーゲーティングをソフトウェアレベルで制御する取り組みも行われている。Komoda ら⁸⁾ はコンパイラの静的解析により、パワーゲーティングを施した演算ユニットのアイドル期間を予測し、アイドル期間の長さからパワーゲーティングのスリープ・ウェイクアップ時に発生する電力オーバーヘッドの大きさを換算して、パワーゲーティングを行うか否かを制御する手法を提案している。また、You ら⁹⁾ や Roy ら¹⁰⁾ も、コンパイラにおける静的解析手法をそれぞれ提案しており、パワーゲーティングを制御するための sleep 命令や wakeup 命令を挿入する方法により、パワーゲーティングを実施する期間を制御している。しかし、パワーゲーティングのスリープ・ウェイクアップ時に発生する電力オーバーヘッドというものは、プロセッサのコア温度により変動するものであり、従来の静的解析手法による制御では、パワーゲーティング実施時の電力特性に大きな影響を与えるコア温度の変化には対応できていなかった。

そこで、本研究では OS の動的な制御により、コア温度情報に基づくスリープ時間粒度を制御する手法に取り組んでいる。文献 4) では、スリープサイクル数の計測によってあらかじめ得たキャッシュミス時のスリープサイクル長を基準に、一様と仮定したスリープの度数分布のうちすべて、または少なくとも半分が電力ロスとならないように温度の閾値を決め、実行時のスリープ時間粒度の切り替えを行った。この方法では、ハードウェアによる自律的な細粒度パワーゲーティングの実施に比べ、すべての演算ユニットの平均で約 7[%] のリーク電力削減を達成した。

しかし、細粒度パワーゲーティング実施時に演算ユニットがスリープする頻度は命令列に依存するため、タスクにより特性が異なる。したがって、プラットフォーム固有のパラメー

タであるキャッシュミス時のスリープサイクル長をもとに閾値温度を決めるよりも、タスク固有の特性の違いを考慮し、タスクそれぞれに対し別の閾値温度を決めることで、電力削減効果を向上できるという課題が得られた。

2.2 本研究の目標

本稿では、コア温度の変動に適応してスリープの時間粒度を制御するとともに、細粒度パワーゲーティング実施時の演算ユニットのスリープ頻度がタスクにより異なる点も考慮した制御方式の実現を目標とする。

具体的には、細粒度パワーゲーティング実施時に実行時のコア温度において適切なスリープの時間粒度をソフトウェアにより判定し、パワーゲーティングを行うハードウェアに知らせることで、細粒度パワーゲーティングの電力削減効果の向上を実現する。タスクそれぞれの特徴の違いを考慮し、細粒度パワーゲーティング実施時のタスクごとのスリープの度数分布を計測して、電力オーバーヘッドを小さくするようにスリープ時間の粒度を切り替えるための温度の閾値を決定する。

3. 細粒度パワーゲーティング制御方式の設計

本章では、コア温度に適応する細粒度パワーゲーティング制御の方式について設計を述べる。

3.1 コア温度に適応するスリープポリシー制御

パワーゲーティングにおけるスリープ・ウェイクアップ時に発生する電力オーバーヘッドと、パワーゲーティングによる電力削減量の損益分岐点となるサイクル数を **Break-Even Point(BEP)** と呼ぶ。BEP は図 1 に示すように、コア温度が高いほど短く、低いほど長くなるという特性を持つ。演算ユニットのスリープ頻度として同じ度数分布を仮定すると、BEP が長ければそのサイクル数を下回る長さのスリープの割合が増え、電力ロスが大きくなる。一方、BEP が短ければ電力ロスとなる短いサイクル数のスリープが占める割合は少なくなる。

そこで本研究では、閾値温度 θ_{TH} を定め、コア温度が θ_{TH} より高い場合はパワーゲーティングのスリープ時間粒度が細くなるように制御し、コア温度が θ_{TH} より低い場合はスリープ時間粒度を粗くするように制御する、というスリープポリシーの制御を OS において行う。また、温度対 BEP の特性は演算ユニットごとに異なるので、演算ユニットごとの閾値温度設定とスリープポリシー選択制御を前提とする。

なお、本研究は、パワーゲーティング機能を備えた演算ユニットに対して、スリープの時

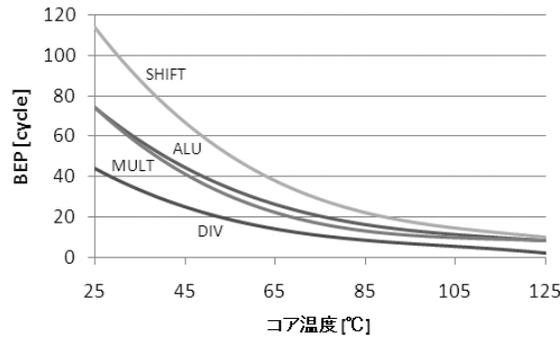


図1 演算ユニットごとの BEP とコア温度との関係 (Geysler²⁾ の例

間粒度を選択的に実施できる機能と、このスリープ時間粒度をシステムソフトウェアが制御できる機能をプロセッサが備えていることが要件となる。例えば、Geysler²⁾には、次の3種類のスリープポリシーが実装されている。

- (1) ハードウェアによる自立的な動的パワーゲーティング (細粒度スリープモード)
- (2) キャッシュミス時のみスリープするパワーゲーティング
- (3) パワーゲーティングしない (常にアクティブ)

スリープポリシー (2) は、スリープポリシー (1) に比べて一段階時間粒度の粗い設定であり、キャッシュミスが生じてメモリアクセスが発生した場合に、まとまったサイクル数で演算ユニットをスリープするというポリシーである。Geysler では、これらのポリシーを、Geysler の特権レジスタ “PGStatus” を介して演算ユニットごとに設定することができる。

3.2 タスク固有の特性を考慮した閾値温度の決定方法

閾値温度を演算ユニットごとに分ける必要があることに加え、スリープの頻度がタスクの性質によっても変化するため、タスク固有の特性の違いを考慮してそれぞれに閾値温度を決めることで、スリープの時間粒度制御による電力削減効果のさらなる向上を図る。閾値温度 θ_{TH} の設定には、スリープ頻度情報を用いる。スリープ頻度情報とは、長さ n サイクルのスリープに対するその回数 m_n の度数分布である。スリープ頻度はタスクごとの命令列に依存して性質が変わるため、タスク固有の特性を持つ情報であるといえる。図2にスリープ頻度のイメージを図示する。このスリープ頻度情報を用いて閾値温度 θ_{TH} を決める方法を次に説明する。

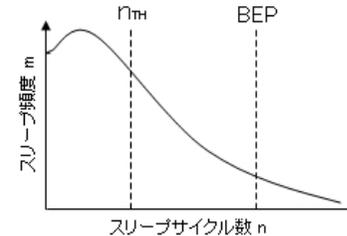


図2 スリープ頻度のイメージ図

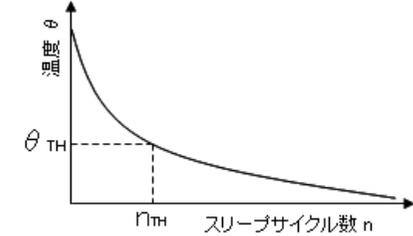


図3 コア温度と BEP サイクル数との関係

まず、スリープ頻度情報 $T_{sleep} = \{t_{n,m} | n, m \in \mathcal{N}\}$ における全スリープサイクル数

$$t_{sleepAll} = \sum_{t \in T_{sleep}} t \quad (1)$$

に対し、

$$\sum_{n=1}^{n_{TH}} (m_n \times n) = \frac{1}{2} t_{sleepAll} \quad (2)$$

なる閾値サイクル数 n_{TH} を求める。 n_{TH} はスリープ頻度分布の中央値といえる。 n_{TH} に対して、ユニットごとに温度と BEP サイクル数の関係から図3に示すように θ_{TH} が得られる。この θ_{TH} を閾値温度とする。このようにスリープ頻度分布の中央値に相当する BEP サイクル数から閾値温度を決めることで、BEP が長くなったときには、全スリープサイクル数のうち少なくとも半分のスリープが BEP ミスとならないうちにスリープポリシーの変更を行うことができる。

ところで、閾値サイクル数 n_{TH} を求める方法には、次の二種類が考えられる。

- (1) アドミッションテストにより事前に得た情報を用いる決定方法
- (2) 実行時の情報を用いる動的な決定方法

方法 (1) のアドミッションテストにより事前に得た情報を用いる決定方法では、事前にスリープ頻度の計測を行って各演算ユニットの閾値温度を決定する。そのため実行時にはスリープ頻度を計測するためのモジュールや閾値温度決定のための処理を必要としないが、スリープ頻度分布の特性がタスクの性質によって異なるため、その変化に適応することが難しいというデメリットがある。

一方、方法 (2) の実行時の情報を用いる動的な決定方法では実行時にスリープ頻度の計測を行うためのモジュールと閾値温度決定の処理が必要だが、タスクの性質の変化に応じた閾値温度の決定を行うことで電力削減効果を効率よく向上させる閾値温度の設定をすることが期待できる。

本稿では、これら二種類の閾値温度の設定方法を OS に実装して比較評価を行う。以下、方法 (1) をアドミッションテスト方式、方法 (2) を動的決定方式と呼ぶ。

3.3 システム構成

本研究で提案する機能を実現するために、次に挙げる二つのブロックを OS へ実装する。

- 閾値温度設定部
- スリープポリシー制御部

図 4 に、提案する機能を実現するソフトウェア構成を示す。閾値温度設定部は、スリープ頻度を計測するハードウェアモジュール（図中のスリープ頻度計測モジュール）からスリープ頻度情報を得て、各ユニットの閾値温度を決める。また、スリープポリシー制御部は、閾値温度設定部で決定した閾値温度を得て、実行時のコア温度と比較して各ユニットのスリープポリシーを決定し、そのポリシーを制御対象コアへ設定することでパワーゲーティングのスリープ間隔を制御する。

閾値温度情報に関して、アドミッションテスト方式の場合は、アドミッションテスト時にあらかじめ閾値温度設定部で最適な閾値温度を算出して外部の記憶装置へ格納しておき、実際のプログラム実行時にその記憶装置から閾値温度を読み出してスリープポリシー制御部が稼動するという構成をとる。一方、動的決定方式の場合は、プログラム実行時に閾値温度設定部とスリープポリシー制御部の両方を稼動させ、プログラム実行中に閾値温度設定部で決定した閾値温度を用いながらスリープポリシー制御部で最適なスリープポリシーを定めるという構成をとる。

閾値温度設定部とスリープポリシー制御部は OS の他の部分とは独立したモジュールとして設計しており、任意のタイミングで呼び出すことができる。したがって、OS が管理するタイマ割り込み、システムコール、あるいはスケジューラの起動タイミングなど各種タイミングにより起動することを想定している。

4. 細粒度パワーゲーティング制御方式の実装

本章では、評価基盤上での提案方式の OS への実装について説明する。

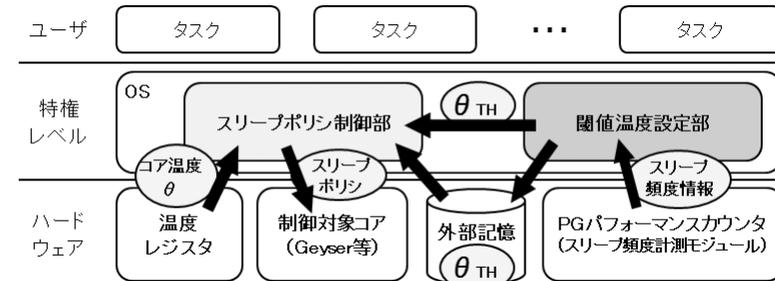


図 4 提案する機能を実現するソフトウェア構成

4.1 FPGA ボードによる評価基盤

本研究では、FPGA ボード上に構築した評価基盤「Geysers on FPGA」を用いて提案手法の実装と評価を行った。Geysers on FPGA は、RTL/ゲートレベルシミュレーションに比べメモリアクセスコストやハードウェア割込みなどの点で実際の計算機に近い評価環境である。

4.1.1 Geysers on FPGA

Geysers on FPGA は主記憶として SDRAM などを持つほか、タイマや UART、経過サイクル数を計測するためのサイクルカウンタ（アップカウンタ）などの各種入出力がある。FlashROM に書き込まれたブートローダにより OS を含むプログラムのバイナリをロードして実行する。Geysers on FPGA の主な仕様を表 1 に示す。

表 1 Geysers on FPGA 仕様

FPGA ボード	Xilinx ML501 (VIRTEX5)
Geysers コア周波数	40MHz
キャッシュ	命令 8KB, データ 8KB
主記憶	DDR 256MB, SRAM 1MB, FlashROM 32MB

4.1.2 スリープ頻度計測モジュール

前述したスリープ頻度情報を計測するためのハードウェアモジュールとして、PG パフォーマンスカウンタが実装されている。このモジュールはソフトウェアから任意のタイミングで計測の開始と終了を指示することができ、その区間のユニットごとのスリープ頻度情報を計測、保持し読み出すことができる。

4.2 Geysler OS へのスリープポリシー制御機構の実装

本研究では、3章に示したパワーゲーティング制御方式を Geysler OS に実装した。Geysler OS は、東京農工大学並木研究室で開発している組込み OS であり、Geysler アーキテクチャ上で動作するようにポーティングが行われている。Geysler OS はタスク管理、仮想メモリ管理、例外・割込み管理などの基本的な機能から構成され、ユーザプログラム側に対してはシステムコールを提供している。

以下、スリープポリシー制御機構の実装について、二つの閾値温度の決定方法それぞれについて示す。

(1) アドミッションテスト方式の実装

アドミッションテストにより閾値温度 θ_{TH} を決める方法における実装を、図 5 に示す。この方式では、実行時に取得したコア温度に適応するスリープポリシーを判定するための閾値温度 θ_{TH} は、あらかじめアドミッションテストにより得た値を用いる。したがって、演算ユニットごとの閾値温度 θ_{TH} を固定値として記憶装置に保持しておき、実行時にスリープポリシー制御機構がそれを読み出す。まず、タイマ割込みによりスリープポリシー制御部が起動する。スリープポリシー設定部は、保持された閾値温度 θ_{TH} を読み出し、コア温度との比較を行って各演算ユニットに設定するスリープポリシーを決定する。このとき、コア温度としてエミュレーションした値を取得する。最後に、選択された各演算ユニットのスリープポリシーを Geysler コアの PGStatus レジスタに書き込む。

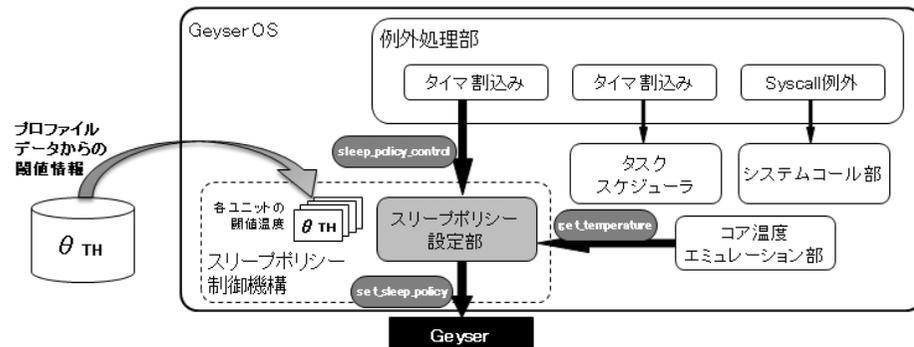


図 5 アドミッションテスト方式で閾値温度を決定する場合の OS の実装

(2) 動的決定方式の実装

実行時のスリープ頻度情報から閾値温度を決める方法の実装を図 6 に示す。この方法では閾値温度を実行時に決めるため記憶装置は必要でない。まずタイマ割込みにより、閾値温度設定部を起動する。閾値温度設定部が PG パフォーマンスカウンタからスリープ頻度情報を取得し、閾値温度を決定する。このとき、スリープ頻度情報からスリープ率も計算し、実行時のスリープ率が高いユニットについてはスリープポリシー設定対象から除外し動的 PG ポリシーに固定する。スリープポリシー設定対象のユニットは、スリープポリシー制御部によってエミュレーションされたコア温度と閾値温度の比較が行われ、その結果選択されたスリープポリシーが Geysler コアの PGStatus レジスタへ書き込まれる。

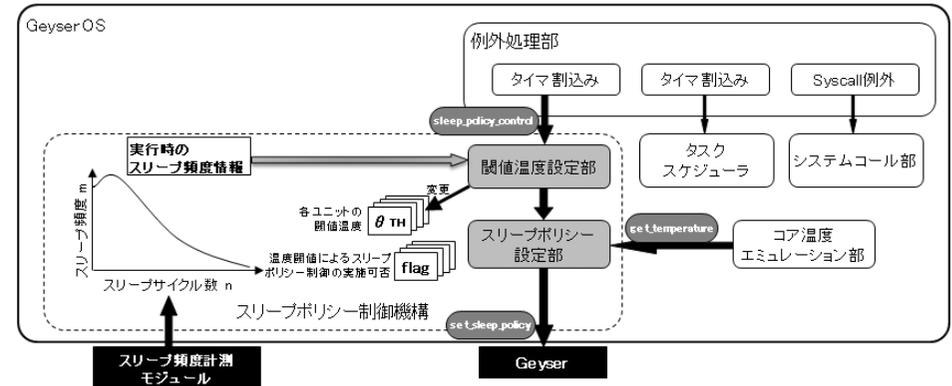


図 6 動的決定方式で閾値温度を決定する場合の OS の実装

4.3 コア温度エミュレーション部の実装

FPGA ボードでは実際の LSI に即した温度を実測することはできない。そのため温度が変化する環境に対して提案手法の評価を行うためにコア温度のエミュレーションを行う必要がある。そこで本稿の評価では、サイクル数をパラメータとして一定の値域において線形に変化するコア温度のテストケースをエミュレーションするプログラムを用いる。このエミュレーションプログラムはスリープポリシー制御プログラムからシステムの経過サイクル数を受け取って時間経過に換算し、あらかじめ設定したテストケースに則った「現在のコア温度」をエミュレートする。

温度は 25℃から 125℃までの範囲をエミュレーションでき、単精度浮動小数点数として読み出すことができる。評価プログラムでは、この温度情報をパラメータとして、BEP や平均リーク電力を実験値から多項式補間により内挿して用いる。

5. 評価と考察

本章では提案した二つの閾値温度決定方法によるスリープポリシー制御方式を評価基盤上の Geysler OS に実装し、MiBench に含まれるいくつかのプログラムをタスクとして実行した場合に関して、各提案方式の電力削減効果の評価した。評価については、アクティブ時とスリープ時の平均リーク電力、BEP ミス率、及びパフォーマンスオーバヘッドを指標とした。BEP ミス率とは、演算ユニットがスリープした時間のうち、BEP サイクル数を下回ったスリープの時間の割合である。BEP ミス率が小さい方が電力オーバヘッドを生じるような期間の短いスリープが少なく抑えられ、電力削減効果の向上に寄与したといえる。ただしこの指標はスリープを行わない場合（細粒度パワーゲーティングを適用しない場合）には定義されない。

評価条件としてコア温度が 25[°C] から 125[°C] の範囲で線形に上昇する場合をエミュレーションした。実行時情報を用いる場合にスリープポリシー制御を適用するスリープ率は 0.9 未満とし、提案方式による制御機構を起動するタイマのインターバルは 500[ms] とした。

5.1 評価結果

以下に、提案手法による細粒度パワーゲーティング制御を行った場合と、比較対象として細粒度パワーゲーティングを行わなかった場合、細粒度パワーゲーティングを行うが OS による制御を行わなかった場合を含め、評価結果を示す。図 7 に平均リーク電力を、図 8 に演算ユニットごとの BEP ミス率を示す。これらのグラフでは各ケースを次の語で表している。

- No PG
細粒度パワーゲーティングを行わなかった場合
- 制御なし
提案方式による制御を行わずに細粒度パワーゲーティングを行った場合
- 制御 (1)
アドミッションテスト方式によりタスクごとの閾値温度を決定してスリープポリシーを制御した場合
- 制御 (2)
動的決定方式によりタスクごとの閾値温度を決定してスリープポリシーを制御した場合

5.1.1 平均リーク電力

図 7 に、制御 (1) に示すアドミッションテスト方式により閾値温度を決める方法を用いた場合、制御を行わず細粒度パワーゲーティングを実施した場合よりも最大で 38[%](Matrix) の削減が見られたが、2.5[%] の増加 (Whetstone) となるケースもあった。すべてのベンチマークプログラムの平均ではおよそ 12[%] の削減を達成した。一方で制御 (2) に示す動的決定方式により閾値温度を決める方法を用いた場合、制御を行わない場合に比べ最小でも 4.0[%](Blowfish)、最大で 16[%](Dijkstra) の削減を達成した。平均ではおよそ 9[%] の削減となった。

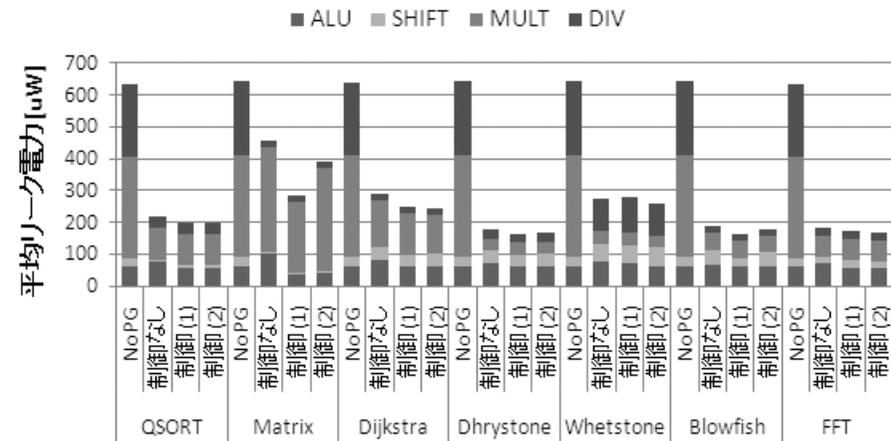


図 7 平均リーク電力

5.1.2 BEP ミス率

図 8 において、制御 (1)、制御 (2) の二種類の方法いずれについても制御なしの場合に比べて ALU で特に大きな改善が見られた。特に Dijkstra では、いずれの方法を用いた場合でも 97[%] からおよそ 3.0[%] まで BEP ミス率を抑制した。一方で MULT ユニットや DIV ユニットでは、ほとんどのベンチマークにおいてもともと BEP ミス率が小さく、ALU のように大きく改善するという事は見られなかった。SHIFT については BEP ミス率や改善の効果はベンチマークによって異なった。

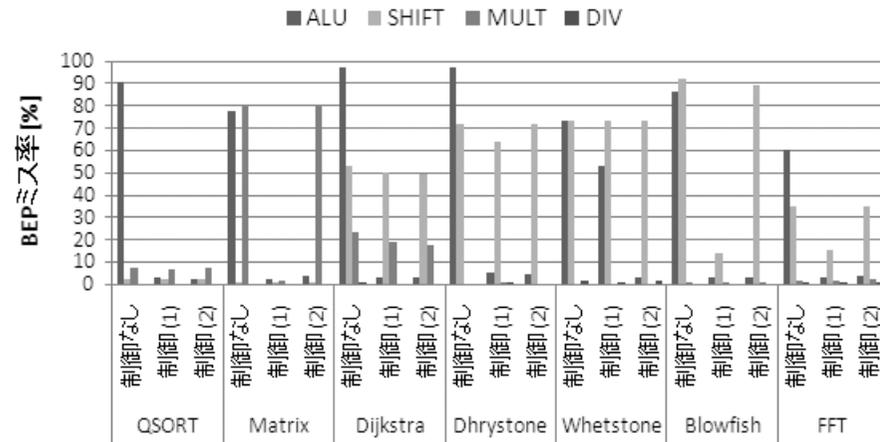


図 8 BEP ミス率

5.1.3 パフォーマンスオーバーヘッド

OS およびタスクを含めたソフトウェア全体の処理に対する、スリープポリシー制御機構のパフォーマンスオーバーヘッドを測定した結果、アドミッションテスト方式の場合およそ 0.25[%] であり、動的決定方式の場合およそ 4.8[%] であった。

5.2 考察

BEP ミス率をみると ALU で大きく改善している。一方で BEP ミス率でそれほど大きな改善が見られていない MULT や DIV といったユニットは、細粒度パワーゲーティング自体の効果により平均リーク電力を大きく抑えている。これらのことから提案方式による制御は ALU に対して特に有効に働いていると考えられ、細粒度パワーゲーティング自体の効果に加えて図 7 の QSORT や Dijkstra に見られるようにさらなるリーク電力削減に寄与したといえる。

制御 (1) のアドミッションテスト方式により閾値温度を決める方法では、実行時に閾値温度決定の処理を行う必要がないため、パフォーマンスオーバーヘッドが小さく抑えられたと考えられる。一方、制御 (2) の動的決定方式では、閾値温度決定のための処理が必要であるためパフォーマンスオーバーヘッドが大きくなった。

スリープ頻度情報から事前に求めた閾値温度によるスリープポリシー制御でも、効果的に電力削減効果を向上できることがわかった。ただしこの方法では、アドミッションテストを

行っていない未知のタスクについてはスリープポリシー制御を適用することができない。

動的決定方式も、パフォーマンスオーバーヘッドはあるものの、電力削減効果を高めることができた。ただしこの方法ではスリープ頻度を取得する際に時間粒度の細かい動的パワーゲーティングのポリシーを適用するため電力ロスが伴うため、電力削減効果がアドミッションテスト方式より平均で小さくなったと考えられる。

二つの方法を総合的に考えると、既知のタスクについてはアドミッションテストにより事前に決定した閾値温度を用いてスリープポリシー制御を行い、未知のタスクについては実行時にスリープ頻度情報を計測し閾値温度を決める、という形で統合すると効果的であるといえる。

6. おわりに

本稿では、コア温度の変動に適応し、かつタスクごとのスリープ頻度特性の違いを考慮した細粒度パワーゲーティングの時間粒度の制御方式を提案した。提案方式では、スリープの時間粒度を切り替えるための温度の閾値をタスク固有の特性であるスリープ頻度情報を用いて決定する設計とした。この決定方法として、アドミッションテストにより事前に決める方法と、実行時に情報を取得して決める方法の二種類を提案した。提案した方法について実際に OS に実装して評価を行った結果、細粒度パワーゲーティング制御を行うことで電力削減効果を高められことを確認した。

細粒度パワーゲーティング制御のためのソフトウェアによる手法では、他にもタスクの性質に適応する方式やコンパイラ情報と連携する方式が考えられており、これらとの統合を検討することや、Linux のような仮想記憶 OS 上での資源管理方式について検討することが今後の課題である。

謝辞 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 (VDEC) を通し、株式会社半導体理工学研究センター、富士通株式会社、松下電器産業株式会社、NEC エレクトロニクス株式会社、株式会社ルネサステクノロジ、株式会社東芝の協力で行われたものである。

本研究は、科学技術振興機構「JST」の戦略的創造研究推進事業「CREST」における研究領域「情報システムの超低消費電力化を目指した技術革新と統合化技術」の研究課題「革新的電源制御による次世代超低電力高性能システム LSI の研究」による。

参 考 文 献

- 1) 中村宏, 天野英晴, 宇佐美公良, 並木美太郎, 今井雅, 近藤正章: 革新的電源制御による超低消費電力高性能システム LSI の構想, 情報処理学会研究報告, 2007-ARC-173, pp. 79-84 (2007).
- 2) Seki, N., Zhao, L., Kei, J., Ikebuchi, D., Kojima, Y., Hasegawa, Y., Amano, H., Kashima, T., Takeda, S., Shirai, T., Nakata, M., Usami, K., Sunata, T., Kanai, J., Namiki, M., Kondo, M., and Nakamura, H.: A Fine-grain Dynamic Sleep Control Scheme in MIPS R3000, *Proc. of the 26th IEEE International Conference on Computer Design*, pp.612-617 (2008).
- 3) 砂田徹也, 木村一樹, 近藤正章, 天野英晴, 宇佐美公良, 中村宏, 並木美太郎: 細粒度パワーゲーティングを制御する OS の資源管理方式, 情報処理学会研究報告, Vol.2010-OS-114, pp.1-8 (2010).
- 4) 木村一樹, 近藤正章, 天野英晴, 宇佐美公良, 中村宏, 佐藤未来子, 並木美太郎: コア温度情報を用いた OS による細粒度パワーゲーティング制御方式の設計, 情報処理学会研究報告, Vol.2011-OS-116, pp.1-8 (2011).
- 5) Hu, Z., Buyuktosunoglu, A., Srinivasan, V., Zyuban, V., Jacobson, H. and Bose, P.: Microarchitectural techniques for power gating of execution units, *Proc. of the 2004 international symposium on Low power electronics and design, ISLPED '04*, ACM, pp.32-37 (2004).
- 6) Shin, Y., Seomun, J., Choi, K.-M. and Sakurai, T.: Power gating: Circuits, design methodologies, and best practice for standard-cell VLSI designs, *ACM Trans. Des. Autom. Electron. Syst.*, Vol.15, pp.28:1-28:37 (2010).
- 7) Liao, W., J.M., B. and He, L.: Microarchitecture-level leakage reduction with data retention, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol.13, No.11, pp.1324-1328 (2005).
- 8) Komoda, T., Sasaki, H., Kondo, M. and Nakamura, H.: Compiler-Directed Fine Grain Power Gating for Leakage Power Reduction in Microprocessor Functional Units, *7th Workshop on Optimizations for DSP and Embedded Systems* (2009).
- 9) You, Y.-P., Lee, C. and Lee, J.K.: Compilers for leakage power reduction, *ACM Trans. Des. Autom. Electron. Syst.*, Vol.11, pp.147-164 (2006).
- 10) Roy, S., Ranganathan, N. and Katkoori, S.: A Framework for Power-Gating Functional Units in Embedded Microprocessors, *IEEE transactions on VLSI Systems*, Vol.17, pp.1640-1649 (2009).
- 11) Donald, J. and Martonosi, M.: Power efficiency for variation-tolerant multicore processors, *Proceedings of the 2006 international symposium on Low power electronics and design, ISLPED '06*, ACM, pp.304-309 (2006).
- 12) Ramamurthy, P. and Palaniappan, R.: Performance-directed energy management

using BOS, *SIGOPS Oper. Syst. Rev.*, Vol.41, pp.66-77 (2007).