

演算ユニットのスリープ頻度に応じてパワーゲーティング制御を行う Linux プロセススケジューラ

高橋 昭宏 *1, 茂木 勇 *2, 佐藤 未来子 *3, 並木 美太郎 *4

東京農工大学工学部情報工学科 *1, 東京農工大学工学部情報コミュニケーション工学科 *2,
東京農工大学大学院工学府 *3, 東京農工大学大学院工学研究院 *4

1 緒言

LSI 技術の進歩により年々LSIは高性能化し、それに伴って消費電力は増大している。LSIの消費電力の中でもとくにリーク電力が年々増大している。リーク電力を削減する手法として、パワーゲーティング (Power Gating) などの技術が注目されている。従来のパワーゲーティング技術はハードウェアが自律的に電源のアクティブ・スリープを制御する。しかし、パワーゲーティングによりアクティブ・スリープへ移行する際に余計なオーバヘッド電力が発生する。そのため、全体として省電力化を行うためには、オーバヘッド電力を考慮する必要がある。

そこで我々は、OSによりオーバヘッド電力による損益を考慮してパワーゲーティングを制御する方式を提案し、簡易 OS 上にてリーク電力を削減できることを確認している。本研究では、本制御方式を汎用 OS に適用し、リーク電力削減の効果を確認する。具体的には、汎用 OS に Linux を使い、カーネルのプロセス管理部にプロセスごとに適切なパワーゲーティングの方式を設定する機構を実装して電力削減を目指す。また、パワーゲーティング制御を行った場合と行わなかった場合の消費電力を測定し、電力削減の効果を評価する。

2 Geysler

Geysler は MIPS R3000 アーキテクチャをベースに、パワーゲーティングを搭載したプロセッサである。Geysler は、コア内の演算器 ALU・シフト (SHIFT)・乗算器 (MULT)・除算器 (DIV) に対して自律的に命令サイクルごとにパワーゲーティングを行う機能、2.1 節に示すソフトウェアからパワーゲーティングに関する制御を行う機構を備えている。

Linux Process Scheduler using Sleep Frequency Information of Arithmetic Units for the Power Gating Control

Akihiro TAKAHASHI*1, Isamu MOGI*2, Mikiko SATO*3, Mitaro NAMIKI*4

*1Department of Computer and Information Sciences, Tokyo University of Agriculture and Technology, *2Department of Computer, Information and Communication Sciences, Tokyo University of Agriculture and Technology, *3*4Graduate school of Engineering, Tokyo University of Agriculture and Technology

2.1 細粒度パワーゲーティングの制御

Geysler にはパワーゲーティングを制御するための特権レジスタを備えており、このレジスタを用いて、次の3種類のスリープポリシーを ALU・SHIFT・MULT・DIV の各ユニットに対して個別に指定する。

1. 動的パワーゲーティング: 通常はユニットをスリープ状態にしておき、ユニットを使用するときだけアクティブにする。
2. キャッシュミス時スリープ: 通常はユニットをアクティブ状態にしておき、キャッシュミスが発生したときのみスリープする。
3. 常にアクティブ: ユニットの常にアクティブ状態にし、パワーゲーティングを行わない。

2.2 パワーゲーティング制御アルゴリズム

パワーゲーティング時のオーバヘッド電力のため、頻繁にパワーゲーティングを行うとオーバヘッド電力が削減電力を越え、電力的に損となる。パワーゲーティングによるスリープの長さが電力的に損となるかどうかの分岐点を BEP と呼ぶ。BEP は温度に依存し、温度ごとの BEP は既に明らかになっている [2]。

BEP より短いサイクルのスリープを頻発する場合、OS はパワーゲーティングのスリープポリシーを変更し、頻繁なパワーゲーティングを抑えることで省電力化を図る。このスリープポリシー変更の方法として、先行研究 [1] は OS からパワーゲーティングを制御するアルゴリズムを提案している。本研究でも本アルゴリズムを Linux へ適用し、BEP とスリープサイクルの頻度情報をもとに、スリープポリシーの変更を行う。

3 Linux プロセススケジューラでのパワーゲーティング制御

本研究では、Linux のプロセススケジューラに 2.2 節で述べたアルゴリズムを実装した制御機構を搭載し、各プロセスの命令列の特徴に応じて個別に制御する。

Linux によるパワーゲーティング制御の流れを図 1 に示す。まず、タイマ割り込みなどを契機としてプロセススケジューラが呼ばれる。スケジューラ内でパワー

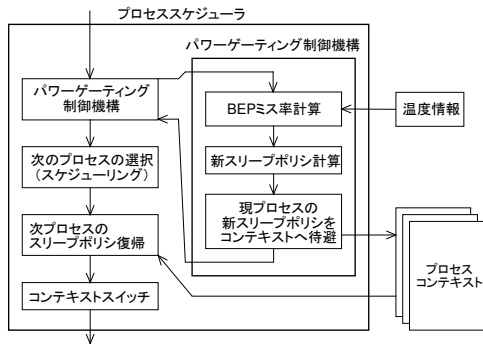


図 1: Linux プロセススケジューラのパワーゲーティング制御の流れ

ゲーティング制御機構を呼び出し、前述のアルゴリズムを用いてスリープポリシーを計算する。スリープポリシーの計算には、スリープサイクルの頻度情報および温度情報を用いる。

スケジューラは、プロセスごとにスリープポリシーの計算を行い、プロセスコンテキストへ計算したスリープポリシーを待避する。その後スケジューラは、スケジューリングを行った後、次プロセスのスリープポリシーの復帰、コンテキストスイッチを行う。

4 実装と評価

評価環境に Geysler プロセッサを FPGA 上に実装した Xilinx ML501 ボードを用いた。これには Geysler プロセッサ本体に加え、スリープサイクル数の頻度情報を計測するハードウェアを有している。なお温度については、実験環境に温度デバイスがないため、固定値とした。スリープサイクル数に応じたスリープ時平均リーク電力はすでに求められており、このデータを用いて頻度情報からリーク電力を算出した。

評価対象としてクイックソート、行列計算、Dhrystone を選び、これらを Linux 上で実行させて電力評価を行った。本パワーゲーティング制御あり・なしの場合のスリープ時リーク電力を図 2、図 3 に示す。

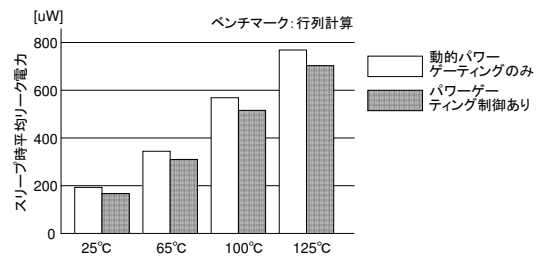


図 3: 温度別リーク電力

パワーゲーティング制御を行った場合、行わなかった場合に比べてリーク電力を 2.3~12.9%削減することができた。温度別で見ると、25°C の場合はリーク電力を 12.9%削減できたものの、125°C の場合は 8.6%にとどまった。これは温度が高くなるにつれ BEP が小さくなり、損となる連続スリープ期間が減少するため、OS の制御の余地が少なくなったためだと考える。この結果は先行研究 [1] の結果のとおりであり、汎用 OS においても先行研究と同様のリーク電力の削減を実現できた。

プロセススケジューラに本制御機構を実装するための主な変更点は、制御機構の呼び出し手続きを追加した点であり、およそ 5 行の追加であった。また、制御機構の手続きは約 400 行程度であった。実行時オーバーヘッドは実行時間のうちの 0.75%程度であった。以上より、既存のカーネルの変更箇所を最小にしつつ、パワーゲーティング制御を実現することができた。

5 結言

本研究では、Linux において Geysler プロセッサのパワーゲーティングを制御する方法について述べた。プロセススケジューラの変更箇所を最小限に抑えることができ、先行研究と同様のリーク電力の削減が確かめられた。しかし一方で、本制御アルゴリズムでは条件によって効果の出にくい部分も見られ、今後の課題としてアルゴリズムを改良していくことが必要である。

謝辞 本研究は、科学技術振興機構「JST」戦略的創造研究推進事業「CRSET」における研究領域「情報システムの超低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」によるものである。

参考文献

[1] 砂田徹也著『省電力を目的とした OS による細粒度パワーゲーティング制御方式の研究』東京農工大学大学院 / 2009 年度修士論文
 [2] 白井利明・香嶋俊裕・武田清大・中田光貴・宇佐美公良・長谷川揚平・関直臣・天野英晴著『ランタイムパワーゲーティングを適用した MIPS R3000 プロセッサの実装設計と評価 (低消費電力化技術)』芝浦工業大学・慶応義塾大学 / 情報処理学会研究報告. SLDM, [システム LSI 設計技術] 2008(2), 43-48, 2008-01-16

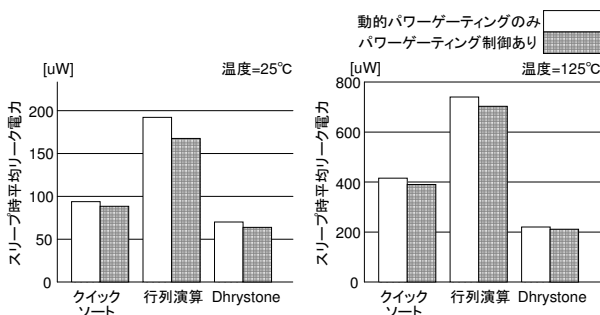


図 2: ベンチマーク別リーク電力