Vol.2015-ARC-217 No.14
2015/10/8

# 演算器におけるオペランド値を考慮したパワーゲーティングに関する初期検討

石川 雄介[1]　小柴 篤史[2]　坂本 龍一[3]　和田 康孝[4]　三輪 忍[1]　近藤 正章[3]　並木 美太郎[2]
本多 弘樹[1]

**概要**：近年のプロセッサではリーク・エネルギーが指数関数的に増加しており，リーク・エネルギーを削減するための手法としてパワー・ゲーティングが広く用いられている．我々は，リーク・エネルギーの大きなユニットの 1 つである演算器に対して，プログラムの実行中にパワー・ゲーティングを行う手法を提案しており，またこの手法を適用可能な実チップを開発してきた．パワー・ゲーティングによるエネルギー削減量を最大化するためには，パワー・ゲーティングによって削減可能なエネルギーとパワー・ゲーティングそのものによって消費されるエネルギーが釣り合う点（損益分岐点）を考慮し，パワー・ゲーティングを行うか否かを判断する必要がある．今回，実チップを使用した実験により，入力オペランドによって演算器のリーク電力が変化し，それによって損益分岐点が変化することを確認したので報告する．

## 1. はじめに

半導体プロセスの微細化により，近年のプロセッサではリーク・エネルギーが消費エネルギー全体の多くを占めるようになっており，これを削減することが重要な課題となっている．リーク・エネルギーの削減に有効な手法の 1 つとして，未使用の回路ブロックの電源供給を遮断するパワー・ゲーティング（Power Gating, **PG**）が知られている．ただし，PG には，パワー・スイッチの ON/OFF などの PG を行うことで消費されるエネルギー・オーバヘッドが存在する（**図 1**）．したがって，PG によって回路全体のエネルギーを削減するためには，リークエネルギー削減量がこのエネルギー・オーバーヘッドを上回らなければならない．そのためにはこの 2 つのエネルギーが釣り合う期間（**損益分岐点**（BreakEven-Point, **BEP**））と未使用期間（=PG を適用可能な期間）を正確に見積もり，未使用期間が BEP よりも長くなるときのみ PG を行う必要がある．

回路のリーク電流は，トランジスタの ON/OFF 状態や温度などの要因によってプログラムの実行中に変化する．そのため，プログラムの実行開始前に BEP を正確に見積もることは困難である．回路の消費エネルギーのさらなる削減のためには，プログラムの実行中に変化する BEP に応じて PG の制御を変更した方がよい．例えば，文献 [1] では，温度ごとに最適化された PG 制御を行うオブジェクト・コードを生成しておき，実行中のプロセッサの温度情報をもとにそれらを切り替える方式が提案されている．しかし，トランジスタの ON/OFF 状態を考慮して PG を行う制御手法は存在しない．

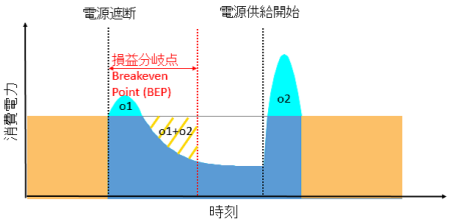


**図 1**　パワーゲーティングにともなうオーバーヘッド

そこで我々は，トランジスタの状態を考慮して PG 制御を行う手法を検討する．このような制御手法を開発するにあたり，まずはトランジスタの状態によって BEP がどの程度異なるかを調査した．以下ではその結果を報告する．

## 2. 演算器におけるパワー・ゲーティング

プロセッサ内の各ユニットはそれぞれ使用頻度が異なるため，より多くのリーク・エネルギーを削減するためにはユニット毎に PG を適用した方がよい．我々は PG 対象のユニットとして演算器に着目する．演算器は，一般に高性能なトランジスタで構成されており，ある程度の回路規模を有していることから，リーク電流が大きい．我々は，この演算器に対してプログラムの実行時に PG を行う手法を提案し，また，この手法を適用可能な実チップである Geyser を開発してきた [2]．Geyser プロセッサではプロセッサ内の 4 つの演算器（ALU，シフト器，乗算器，除算器）に対してそれぞれ独立に PG を行うことが可能である．

対象の演算器を PG するか否かは，命令に付加された制

1　電気通信大学
2　東京農工大学
3　東京大学
4　明星大学

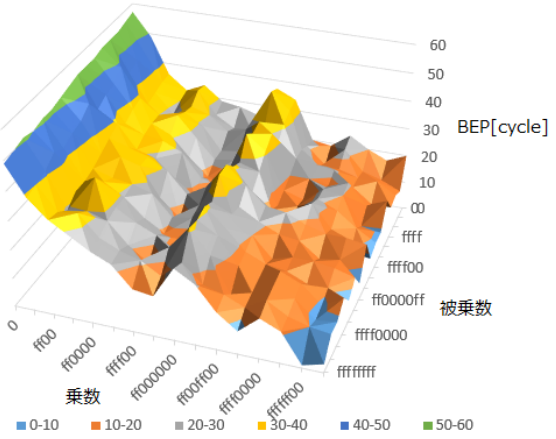


図 2　オペランド値と BEP の対応関係

御コードを用いて制御する．ある命令がある演算器で実行されると，Geyser の電力管理機構はその命令の制御コードを参照する．そして，電力管理機構は，参照した制御コードの値に応じて，使用した演算器のモード（アクティブ・モード =PG しないモード，あるいは，スリープ・モード =PG を適用したモード）を直ちに変更する．この時選択されたモードは，基本的には，次にこの演算器が使用されるまでは変更されることがない．このように，Geyser は命令毎に演算に使用した演算器の電源状態を変更できる．

## 3. オペランド値による BEP の違い

演算器は組み合わせ回路であるため，演算器内の各トランジスタの ON/OFF 状態は演算器の入力値によって一意に決まる．すなわち，演算器に対する 2 つの入力オペランド値と入力制御信号の値が同じならば，演算実行後の各トランジスタの ON/OFF 状態も同じとなり，その結果，リーク電力も同じになる．そこで我々は演算に用いる 2 つの入力オペランド値と BEP との相関関係を調査した．

## 4. 実験

実チップを用いて演算器の入力オペランド値と BEP の関係を調査した．チップの温度を一定に保った状態で入力オペランド値を変更した時の演算器の消費エネルギーを計測した．数種類の入力パターンを用いて予備評価を行った結果，乗算器が入力オペランド値の違いによる BEP の差が最も大きかった．そこで今回，乗算器に与える入力パターンをさらに細かくし，入力オペランド値と BEP との関係を詳細に分析した．

今回の実験では，被乗数と乗数それぞれについて 16 パターンの入力オペランド値を生成し，計 256（$= 16 \times 16$）パターンの入力オペランド値の組み合わせに対して乗算器の BEP を算出した．16 個のパターンは，0x00000000, 0x000000ff, 0x0000ff00, ···, 0xffffffff のように，0x00000000 から 0xffffffff まで 0xff 刻みで 16 通り生成した．なお，0xf

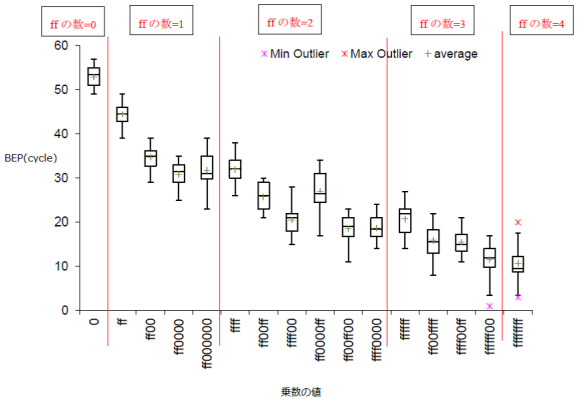


図 3　乗数ごとの BEP の分布

刻みで入力オペランド値を振って BEP の計測を行ってみたが，全体の傾向はほとんど変わらなかったため，今回は 0xff 刻みの結果のみを示す．ちなみに，今回の実験で使用した乗算器の入力オペランド値は 32 ビットである．チップ温度を 25°C に保ち，実験を行った．

実験結果を**図 2** に，また乗数を 0xff の数でソートした上で 2 次元化したグラフを**図 3** に示す．これらのグラフは

(1) BEP は被乗数より乗数に強く依存する
(2) オペランド値に 0xff が多いほど BEP が小さくなる

傾向があることを示している．このとき，入力オペランド値の違いによる BEP の差は最大 56 サイクルであった．またチップ温度 65°C で，同様の実験を行ったところ入力オペランド値による BEP の差は最大 15 サイクルであった．

以上の結果より，入力オペランド値の違いによる BEP の差は大きく，入力オペランド値を考慮して PG を行うべきと考えられる．また，BEP と乗数の 0xff の数は強い相関があることから，乗数の値が 1 のビット数から BEP を推測できる見込みが高いことがわかった．

## 5. おわりに

我々は，入力オペランド値の違いによって乗算器の BEP に大きな差があることを確認した．今後は，さまざまなベンチマーク・プログラムにおける入力オペランド値と演算器の使用間隔を調査し，上記の BEP の差が，実際にプログラムを実行した時の演算器の消費エネルギーに与える影響を調査する．また，入力オペランド値の違いを考慮する新しい PG 手法の開発を進めていく．

**謝辞**　本研究の一部は，JSPS 科研費 25220002 の助成により行われたものである．

### 参考文献

[1] 小林 弘明 他：OS における細粒度パワーゲーティング向けオブジェクトコードの実行時管理機構の研究，情報処理学会研究報告，2011-OS-117(1), pp.1-8, 2011-04-06
[2] Zhao, L. et al.: *Geyser-2: The second prototype CPU with fine-grained run-time power gating*, 2011 16th Asia and South Pacific Design Automation Conference (ASP-DAC), pp.87-88, 25-28 Jan. 2011