

# A-31 演算の対象となるデータに注目したアルゴリズム実行中の消費電力削減手法 A Reduction Approach to Power Consumption due to Operational Data during Algorithm Running

暮町 尚俊<sup>†</sup> 上土井 陽子<sup>‡</sup> 吉田 典可<sup>†</sup>  
Naotoshi KUREMACHI Yoko KAMIDOI Noriyoshi YOSHIDA

## 1 はじめに

近年、省エネルギー化、計算機の高速度化阻害要因除去、バッテリー駆動機器のバッテリー寿命延長など様々な観点からプロセッサにおける消費電力の削減が大きな課題となっている。一般に、設計作業での低消費電力化はアーキテクチャレベル以下で行われている。しかし、アルゴリズムレベルでの最適化もシステムの消費電力に影響を与え得る [1]。具体的には、アルゴリズム実行中にハードウェアを効率的に使用することで、同じハードウェアを使用しているにも拘らず、消費電力を更に削減できる可能性がある。本研究では、プロセッサ上で実行されるアルゴリズム中の演算に注目し、消費電力の増加を招く一因として知られているデータパス上での計算と無関係なスイッチングを削減することで、アルゴリズム実行中の消費電力の削減を目指す。

本稿では、アルゴリズムレベルにおける低消費電力化を模索するため、比較演算について計算機上での消費電力の観点から実験し、実験結果を基に考察する。

## 2 CMOS LSIにおける消費電力

プロセッサのような複雑なLSIでは様々な回路が混在するため、消費電力を厳密に一つの式で表現することは難しいが、一般的にCMOS LSIの消費電力  $P$  は次の式で表される [2]。

$$P = [\text{充放電電力}] + [\text{貫通電力}] + [\text{直流電力}] \\ = [p_t \cdot f_{clk} \cdot C_l \cdot V_s \cdot V_{dd}] + [p_t \cdot f_{clk} \cdot I_{sc} \cdot \Delta t_{sc} \cdot V_{dd}] \\ + [(I_{dc} + I_{leak}) \cdot V_{dd}]$$

ここに、 $p_t$  はスイッチング確率、 $C_l$  は負荷容量、 $V_s$  は信号振幅、 $V_{dd}$  は電源電圧、 $I_{sc}$  は貫通電流の平均値、 $\Delta t_{sc}$  は貫通電流が流れる時間、 $f_{clk}$  はクロック周波数、 $I_{dc}$  は直流電流、 $I_{leak}$  はリーク電流である。通常、 $V_s$  と  $V_{dd}$  は等しい。 $p_t \cdot f_{clk}$  は、スイッチング確率とクロック周波数を掛け合わせた値で、1秒間のスイッチング回数と等しい。

CMOSの消費電力は、1秒間のスイッチング回数と負荷容量に比例し、電源電圧の2乗に比例する充放電電力が支配的になり、充放電電力の削減が消費電力の削減に効果的である。本稿ではスイッチング回数に注目し、アルゴリズム実行時の低消費電力化を模索する。

### 2.1 スwitchングに伴う消費電力

CMOSの消費電力において充放電電力はスイッチング回数に依存するため、スイッチング回数の削減によって消費電力を削減できる。このスイッチングに伴う消費電力に関して、プロセッサ上で実行されるアルゴリズム中の演算などにおけるデータパス上での計算と無関係なスイッチングが消費電力の増加を招いていることが知られている。この課題に対して、実行するアプリケーションに応じて計算機のデータパス幅を最適化する手法 [3] や、ビット幅を考慮することでキューおよびスタックにおける消費電力を削減する手法 [4] 等、ハードウェアを効率的に使用するためのアプローチが様々な設計レベルで研究されている。

### 2.2 データパス上での計算と無関係なスイッチング

一般に、計算機において演算の対象となるデータは、定義された表現形式を基準にして演算される。データの規定サイズ化は、演算対象となるデータを表現するために最低限必要なビット数に冗長ビットを加えることを意味するため、データビット数の増加につながる。データビット数の増加に伴うスイッチングによって消費電力が増加すると予測される。

### 3 消費電力削減手法

本稿では、アルゴリズム中での特定の処理段階で扱うデータを細分化し、演算上の規定サイズのデータ形式で表現されたデータから、演算結果を導く上で不要なビットを削減する消費電力削減手法を提案する。基本的な演算である減算を例として用い、演算結果を導く上で不要なビットにおけるスイッチングを考慮しない通常の手法による減算を図1、考慮する提案手法を適用した場合の減算を図2にそれぞれ示す。図1、図2は共に  $10 - 8 = 2$  を2の補数表現と加算によって導出する例である。図2において4Bで表現された入力データから演算結果の導出に不要な上位3Bを削減して演算することにより、図1と比べてスイッチング回数が削減される可能性があることが分かる。

$$\begin{array}{r} 0000 \dots \dots 0000 \dots 00001010 \\ - 0000 \dots \dots 0000 \dots 00001000 \\ \hline \end{array}$$

2's complement ↓

$$\begin{array}{r} 0000 \dots \dots 0000 \dots 00001010 \\ + 1111 \dots \dots 1111 \dots 11111000 \\ \hline 0000 \dots \dots 0000 \dots 00000010 \end{array}$$

図1: 通常の手法による減算

$$\begin{array}{r} 0000 \dots \dots 0000 \dots 00001010 \\ - 0000 \dots \dots 0000 \dots 00001000 \\ \hline \end{array}$$

reduction of bits & 2's complement ↓

$$\begin{array}{r} \dots \dots 00001010 \\ + \dots \dots 11111000 \\ \hline \dots \dots 00000010 \end{array}$$

図2: 提案手法を適用した場合の減算

### 3.1 電力消費実験

演算の対象となるデータに対する所望の演算結果を導く上で不要なビットの削減が、計算機上でのアルゴリズム実行中の消費電力削減につながるかを検討するため、様々なアルゴリズムの基本的演算である比較演算について計算機上での電力消費の観点から実験した。比較演算の対象としてint型、short int型、char型の3種類で表現されたデータを用意し、データの規定サイズ化による冗長ビット数の変化に伴う電力消費の差に注目した。実験プログラム(1)は、配列を用いて入力データを保持し、隣り合う要素間の比較を行うプログラムである。図3に実験プログラム(1)の比較パターンの概略図を示す。

$$\boxed{2} \quad \boxed{8} \quad \boxed{5} \quad \boxed{6} \quad \boxed{1} \quad \boxed{11} \quad \dots$$

図3: 実験プログラム(1)の比較パターンの概略図

#### 3.1.1 入力データ

実験プログラム(1)の入力データは1万要素を持ち、乱数発生プログラムを用いて生成した。使用した計算機のchar型が1B(8bit)であるため、各要素の値は0から255までとした。入力データに冗長ビットを加えることにより、3つの規定サイズint型、short int型、char型での異なる入力データの表現を用意した。入力データの概略図を図4に示す。このため、各要素のint

<sup>†</sup>広島市立大学大学院 情報科学研究科 (Graduate School of Information Sciences, Hiroshima City University)

<sup>‡</sup>広島市立大学 情報科学部 (Faculty of Information Sciences, Hiroshima City University)

型表現は4Bのうち上位3Bが冗長ビット“0”を保持し、short int型表現は2Bのうち上位1Bが冗長ビットを保持する。

upper bit 0000 ..extra bits... 0000 random

図4: 入力データ表現の概略図

### 3.1.2 実験方法

バッテリー駆動型計算機において、バッテリーを使いきるまで実験プログラムを実行し、比較演算の実行回数に差を基に冗長ビット数が異なる入力データの表現型による違いを消費電力の観点から比較した。バッテリーを使いきった状態から十分充電されるであろう時間が3.5時間であることを経験的に得たため、全実験においてバッテリー充電時間を約3.5時間に統一した。本実験ではSONY VAIO ノート PCG-XR1, celeron 500MHz, メモリ 64MB 上で、OS として redhat Linux7.2, コンパイラとして GNU gcc2.95.3 を用い、実験プログラムをC言語で実装した。

### 3.1.3 実験結果

図5に1回の比較演算における消費電力量を示す。

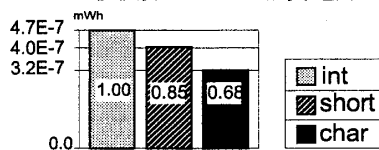


図5: 実験プログラム(1)の1回の比較演算の消費電力量

実験結果から比較演算の対象となる要素から冗長ビットを削減することで、実行中の消費電力が削減されることが分かる。冗長ビットを24ビット削減することで約32%(1.3%/bit)改善され、16ビット削減することで約15%(0.9%/bit)改善されている。

比較演算は2項演算であり、比較の対象となる2個の要素に対して減算を適用し、要素間の大小関係を得ることで実現される。減算は計算機上で2の補数の加算で実現されるため、2個の対象要素のうち一方の要素においてビットの反転が生じる。このときスイッチングが生じて電力が消費される。本提案手法は、演算の対象となる要素のビット数を削減することで、2の補数で表現される要素のみに生じる演算結果の導出に無関係なスイッチングの削減にのみ寄与したと考えられる。これによって演算実行中の消費電力が削減されたと考えられる。1ビット当り改善率に関して、24ビットの削減と16ビットの削減とで比較すると24ビット削減した場合の改善率が高いことから、削減するビット数が多いほど消費電力の削減が大きくなると考えられる。

## 4 ソートアルゴリズムへの拡張

3節から、ソートアルゴリズム中の比較演算において、データの規定サイズ化によって生じる冗長ビットを比較の対象となる要素から削減して扱うことが可能であれば、ソートアルゴリズム実行中の消費電力を削減できると考えられる。提案するアイデアを適用するアルゴリズムとして、基本的なソート手法の一つであるセレクションソートを用いた。セレクションソートは、データの比較コスト( $O(n^2)$ )が高く、交換コスト( $O(n)$ )が低いため、3節で得た比較演算に対する実験結果が反映されやすいと考えられる。

### 4.1 電力消費実験

提案する消費電力削減手法が、ソートアルゴリズムに対して有用であるか否かを検討するため実験を行った。実験プログラムとして、ソートアルゴリズム中の比較演算の対象となる要素のビット数が4B(int型)と1B(char型)の2種類のプログラムを用い、ソート実行回数を基にプログラム間での消費電力を比較した。以下、全ての実験において実験方法、実験環境および入力データは3節の実験と同一である。

## 4.2 実験結果

図6(a)にセレクションソートにおける1回の実行に伴う消費電力量、(b)にセレクションソートアルゴリズム中の要素の交換を削除したプログラムにおける1回の実行に伴う消費電力量をそれぞれ示す。図6(a)から、比較演算の対象となるデータのビット数の削減による効果が、アルゴリズム実行中の消費電力の削減において7%に留まっている。この要因について考察する。

図6(b)から、アルゴリズム中の要素の交換を削除した場合においても削減率に変化がみられず、要素の交換が要因ではないと考えられる。図7に、セレクションソートの比較パターンと酷似した比較パターンを持つ実験プログラム(2)の1回の比較演算における消費電力量を、比較パターンの概略図と共に示す。図6(a)と図7から、セレクションソートの実験結果とほぼ同様の削減率であることが分かり、セレクションソートの比較パターンが、3.1.3節で得られた約32%に対する削減率低下の主な要因であると考えられる。具体的には、セレクションソートの実験プログラムの比較パターンは一方の要素(2の補数表現のためにビットの反転が生じる要素)が毎回に変化しないため、比較演算における全スイッチング回数に対する比較結果の導出に無関係なスイッチング回数の割合が、実験プログラム(1)の割合と比べて低下する。本提案手法の効果が、2の補数で表現される要素に生じる演算結果の導出に無関係なスイッチングの削減にのみ寄与したと考えられ、提案手法によって削減できるスイッチングの割合の減少に伴い、セレクションソートの消費電力の削減率が7%に留まったと考えられる。また、セレクションソートの比較パターンは、実験プログラム(1)と比べて平均スイッチング回数が少なく、1回の比較演算における消費電力量もまた減少している。従って、比較演算における比較パターンがスイッチングでの消費電力に影響を与えると考えられる。

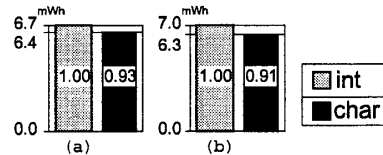


図6: セレクションソートの1回の実行に伴う消費電力量

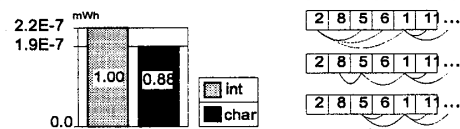


図7: 実験プログラム(2)の1回の比較演算の消費電力量と比較パターン概略図

## 5 おわりに

本稿では、比較演算において、演算上の規定サイズのデータ形式で表現された要素から、演算結果を導くのに不要なビットを削減することが消費電力の削減につながることを示した。また、比較演算において比較パターンが消費電力に影響を与え得るため、今後提案手法と共に考慮していく必要がある。

## 参考文献

- [1] A. Bellaouar and M. I. Elmasry, "Low-power digital VLSI design circuits and systems," Kluwer Academic Publishers, pp. 1-7, 1995.
- [2] 桜井 貴康, "低消費電力, 高速 LSI 技術", 株式会社リアライズ社, 1998.
- [3] Y. Cao and H. Yasuura, "A system-level energy minimization approach using datapath width optimization," Proc. International Symposium on Low Power Electronics and Design, pp.231-236, August 2001.
- [4] V. G. Moshnyaga, "Energy reduction in queues and stacks by adaptive bitwidth compression," Proc. International Symposium on Low Power Electronics and Design, pp.22-27, August 2001.