

# 細粒度なヘテロジニアスタコアによる消費電力削減の提案

金子郁未<sup>†</sup>, 小林良太郎<sup>†</sup>, 嶋田創<sup>‡</sup>

豊橋技術科学大学<sup>†</sup> 名古屋大学<sup>‡</sup>

## 1. 研究目的

近年、タブレットやスマートフォンは広く使用されており、これらモバイル端末の性能向上のため、搭載されるプロセッサの処理能力向上に対するニーズは大きく高まっている。一方、プロセッサの高機能化による回路面積の増加や消費電力の増加による消費電力と発熱の問題はモバイル機器向けのプロセッサの課題である。

プロセッサアーキテクチャの1つとして、クラスタ化プロセッサが提案されている[1]。クラスタ化プロセッサは小さな実行コアであるクラスタによって構成されたプロセッサであり、クラスタ1つあたりの回路面積や発行幅が小さく、配線遅延を抑えることが可能である。また、使用していないクラスタの電源を落とすことで消費電力の削減を行うクラスタゲーティングと呼ばれる手法が提案されている[2]。しかし、この手法では命令の実行に使用できるクラスタが減ってしまうため、プロセッサの処理能力の低下を招いてしまう。

本研究ではプロセッサの処理能力を維持しつつ消費電力の削減を行うため、データの規則性を利用したヘテロジニアスタコア化プロセッサを提案する。

## 2. 提案

### 2.1. データの規則性

プロセッサ内で処理されるデータの中には規則性があり、命令の上位ビットに特定の値が頻出することが分かっている[3]。例えば、図1に示すような即値演算を考える。この演算では、

$$\begin{array}{r}
 7FFE\ 6CA3 \\
 +\ 0000\ 0028 \\
 \hline
 7FFE\ 6CCB
 \end{array}$$

計算結果の  
必要ない部分

実際に計算  
されてる部分

図1. 冗長なデータ

実際に数値が計算され結果に反映されているのは下位 8 bit のみであり、上位ビットの結果は計算前のソース 1 のオペランドから変化していない。このような演算の場合は下位 8 bit のみ演算を行い、上位ビットは後から連結させることで、より小さな演算器で演算可能になる。

このデータの規則性を利用し、クラスタ化プロセッサにおけるクラスタコアとして、通常の 64 bit データを扱うことの出来るクラスタに加えて、下位 32 bit の計算を行う 32 bit クラスタ、下位 16 bit の計算を行う 16 bit クラスタ、下位 8 bit の計算を行う 8 bit クラスタを用意する。これらのクラスタは通常のクラスタに比べ、搭載する演算器のサイズが小さなものとなるため、消費電力と回路面積の点で有利である。

### 2.2. ステアリング機構

下位ビットクラスタで正しく命令を実行するためには、命令が実行可能なビット幅を判定し、命令を適切なビット幅のクラスタへステアリングする必要がある。これを実現する機構として、命令ごとに状態を登録したハッシュテーブルを利用して、投機的なステアリングを行う手法が提案されている[4]。

この手法では、投機的なステアリングを行っているため、下位ビットクラスタでは実行することの出来ない命令を下位ビットクラスタにステアリングしてしまう場合がある。このような場合はステアリング失敗を検出し、正しいクラスタで命令を実行し直す必要がある。ステアリングミスを検出する方法としては、演算結果のキャリーフラグを確認する手法が挙げられている。しかし、キャリーフラグにて演算失敗を検出するためには、演算失敗時に必ずキャリーフラグが立つことを保証する必要がある。これを解決するため、本研究ではレジスタファイルにショートフラグ (SF) を付与する。SF はレジスタ値が特定のビット幅以下であることを保証するフラグであり、レジスタファイルにデータが書き込まれる際に、レジスタ値を判定して付与する。Ready ビットが立っているレジスタファイルにおいて、ショートフラグを判定することで、レジスタ値のビット幅を知ることができるほか、レジスタリネーミングに使用する MAP テーブル

Proposal of Power Consumption Reduction by Fine-Grained Heterogeneous Clustered Core

<sup>†</sup>Iikumi KANEKO, Ryotaro KOBAYASHI,  
Toyohashi University of Technology

<sup>‡</sup>Hajime SHIMADA,  
Nagoya University

にも、ショートフラグを付与することで、命令をステリングする際に、命令を演算するために必要なビット幅を予測することも可能となる。

### 3. 予備評価

SF を利用した場合にどの程度のデータを下位ビットクラスタで実行することが出来るかを確認するため、付与した SF からレジスタ値の各ビット幅分布を測定した。測定にはソフトウェアシミュレータ SimCore を SuperScalar 化したものを使用し、ベンチマークとしては SPEC2006 から 9 本を使用した。測定結果を図 2 に示す。縦軸はレジスタファイルに書き込まれたデータにおける、データビット幅の割合を表している。横軸はベンチマーク名である。この図からベンチマークごとに分布にばらつきがあるが、64 bit のデータと 8 bit のデータが多く存在することが分かる。このため、非常に多くの命令が 64 bit または 8 bit クラスタで実行されることが予想される。

クラスタ化プロセッサにおいて、ワークロードバランス (WB) の不均衡は性能低下を招く原因となる。WB を改善し、性能低下を防ぐ方法として、オペランド値を上位と下位に分割して実行することで、数の多い 64 bit と 8 bit の命令を命令数の少ない 32 bit と 16 bit のクラスタで実行する方法がある。しかし、命令を複数回に分けて実行するため、実行サイクルの増加が発生し性能低下が起こるといった問題がある。

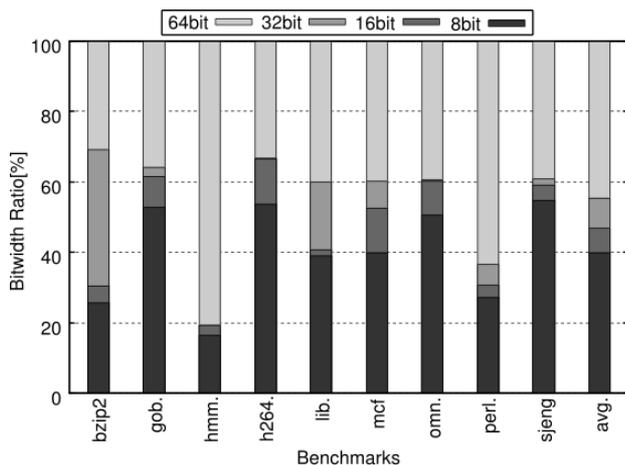


図 2. レジスタ値のビット幅分布

### 4. 今後の予定

前節で述べた通り、WB を改善する手法として命令を複数回に分けて実行する方法がある。この時に追加が必要となる 1 サイクルをスラック命令と呼ばれるクリティカル・パスにない命令

を利用することで隠蔽することが出来ると考えられる。スラック命令は数サイクル実行が遅れてしまっても、プロセッサの処理性能が低下することはない。1 サイクル以上のスラック命令は 3~5 割程度あることが発表されており [5]、これを利用することで少なくとも 3 割ほどの演算を性能低下なく下位ビットクラスタで実行することが可能となると考えられる。

現在、命令の実行可能ビット幅を予測するため、オートマトンと SF を利用している。しかし、予測のための手がかりには、他にもいくつかの手法が考えられる。即値演算命令では、命令フォーマットによって、即値指定に使用できるビット幅が決まっている。また、命令によっては使用するソースオペランドのビット幅が決まっているものもある。こういった手がかりを利用して、命令の投機的ステアリングを改善し、予測ミスによる性能低下を抑えることが出来ると考えられる。

今後の展開としてはまず、上記の手がかりを考慮した命令のステアリング機構とスラック命令によるワークロードバランスを考慮したステアリング機構の 2 つを提案機構に組み込む。その上で、プロセッサの消費電力の削減率の評価、従来機構と比較した IPC などの性能低下率の評価を行っていく。

**謝辞** 本研究の一部は、JSPS 科研費 25330060 及び 26330063 の支援により行った。

### 文 献

- [1] S. Palacharla, et al., "Complexity-effective superscalar processors," In Proc. 24th Annual International Symposium on Computer Architecture, pp. 206-218, 1997.
- [2] 佐藤寿倫ほか, "性能・消費電力・信頼性の間のトレードオフを考慮出来るマルチ・クラスタ型コア・プロセッサ," 電子情報通信学会技術研究報告コンピュータシステム(CPSY), Vol. 107, No. 276, pp. 39-44, 2007.
- [3] G. H. Loh, "Exploiting data-width locality to increase superscalar execution bandwidth," In Proc. 35th Annual ACM/IEEE International Symposium on Computer Microarchitecture, pp. 395-405, 2002.
- [4] 川合翔麻ほか, "レジスタ値の部分更新による低消費エネルギー指向ヘテロジニアス・クラスタ型プロセッサ," 電子情報通信学会技術研究報告コンピュータシステム(CPSY), Vol. 113, No. 282, pp. 1-6, 2013.
- [5] 小林良太郎ほか, "発見的手法に基づくローカル・スラック予測機構," 電子情報通信学会技術研究報告コンピュータシステム(CPSY), Vol. 107, No. 276, pp. 39-44, 2007.