様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　元 年　 ６ 月 １１ 日現在

機関番号：１３９０１
研究種目：基盤研究(C)（一般）
研究期間：2016～2018
課題番号：１６Ｋ００００７０

研究課題名（和文）高性能・低電力コンピュータの方式に関する研究

研究課題名（英文）Study on computer architcture for high performance and low power consumption

研究代表者
安藤　秀樹（Ando, Hideki）

名古屋大学・工学研究科・教授

研究者番号：４０２９３６６７

交付決定額（研究期間全体）：（直接経費）　　3,400,000円

研究成果の概要（和文）：近年、コンピュータの単一スレッド実行時間はほとんど短縮されず、性能はほぼ停滞している。私は、プロセッサの部品の中で性能(IPC: instructions per cycle)に最も大きな影響を与える発行キュー(IQ:issue queue)について研究を行い、再並べ替えランダム・キュー(RRQ: rearranging random queue)と呼ぶ新しいIQの構成法を提案した。評価の結果、RRQは、既存のIQと比べて低電力、低遅延かつ高IPCを達成できることがわかった。

研究成果の学術的意義や社会的意義
コンピュータの性能向上の最大の源泉は、長年LSIにおけるトランジスタの縮小に起因するゲート遅延の短縮法則、すなわち、デナード・スケーリングであった。しかし、このトレンドはLSI製造技術の限界により2005年に終わった。一方、LSI製造の縮小トレンドは続いたが、電力が冷却の限界に達し、トランジスタを有効に利用することが困難となった。これらの理由により、コンピュータの単一スレッド実行性能はほとんど向上しなくなった。これに対して、本研究は、電力を増加させることなく性能を向上させる構成法を提案した。本技術は、実際のプロセッサに即座に適用できるほど実用的であり、学術的のみならず社会的意義が大きい。

研究成果の概要（英文）：The single-thread performance improvement is very sluggish in recent computers. I studied about the organization of the issue queue (IQ), which affects the performance (IPC: instructions per cycle) most significantly in various structures in a processor, and proposed a new IQ organization called rearranging random queue (RRQ). My evaluation results showed that the RRQ achieved high IPC as well as low power consumption and short delay, compared with conventions IQs.

研究分野：コンピュータ・アーキテクチャ

キーワード：コンピュータ・アーキテクチャ　スーパスカラ・プロセッサ　発行キュー

1　研究開始当初の背景

近年のコンピュータは、複数のプロセッサ・コアにより複数のプログラムを同時に処理することにより高いスループットを達成している。しかし、1 つのプログラム (単一スレッド) の実行時間はほとんど短縮されず、性能はほぼ停滞している。この原因は、いわゆる次の 2 つの障害 (壁) による: メモリの壁と電力の壁。これらの壁を同時に打破しない限り単一スレッド実行における性能は向上しない。

2　研究の目的

2 つの壁を同時に打破し単一スレッド実行性能を向上させるために、プロセッサの部品の中で性能に最も大きな影響を与える発行キュー (IQ: issue queue) について研究を行う。具体的には、次のような研究を行う。

近年の IQ は、電力消費を抑えるために空いているエントリに命令を単に書き込む方式を採っている。この方式では命令の並びが、その古さ (年齢) においてランダムとなる。一方で、高い性能を得るには、古い命令ほど優先して実行することが要求されている。この要求を命令がランダムに並んだキューにおいて満たすことは、回路上難しい。本研究は、命令がランダムに並ぶ近年の IQ において、プロセッサの高い性能を実現する IQ の方式を見出すことを目的とする。

3　研究の方法

考案した方式を模擬するプロセッサのシミュレータを作成する。このシミュレータは、SimpleScalar [1] と呼ばれる近年のプロセッサ (スーパスカラ・プロセッサ) をベースに作成する。評価プログラムには、SPEC2006 を用い、実際に市場で使われる状況を想定した評価を行う。

4　研究成果

4.1　はじめに

IQ は、キューの中で実行可能な命令の中から機能ユニットに発行する命令を選択することにより命令の実行をスケジュールする回路である。プロセッサが高い性能を発揮するには、信号遅延は短く高い IPC (instrucitons per cycle) を達成する必要がある。IQ の遅延はプロセッサ内で元も長い遅延の 1 つであり、これはクロック・サイクル時間に悪影響を与えるため、IQ の短い遅延は重要である。

IQ には 3 つのタイプがある: シフティング・キュー (SHIFT)、サーキュラ・キュー (CIRC)、ランダム・キュー (RAND)。SHIFT は、完全な年齢順選択を行えるため、これらのキューの中で最も高い IPC を達成できる。しかし、その遅延は最も長く、また、多くの電力を消費する。このため、SHIFT は、小さな IQ を持つかつてのプロセッサ [2] で用いられただけであり、現在では用いられていない。これに対して、CIRC と RAND は、回路が単純なため、その遅延は短い。しかし、年齢を考慮した選択が不完全か、無考慮のため、IPC が低い。さらに、CIRC は容量効率が悪い。このため、CIRC は現在のプロセッサでは用いられていない。RAND あるいは RAND をベース

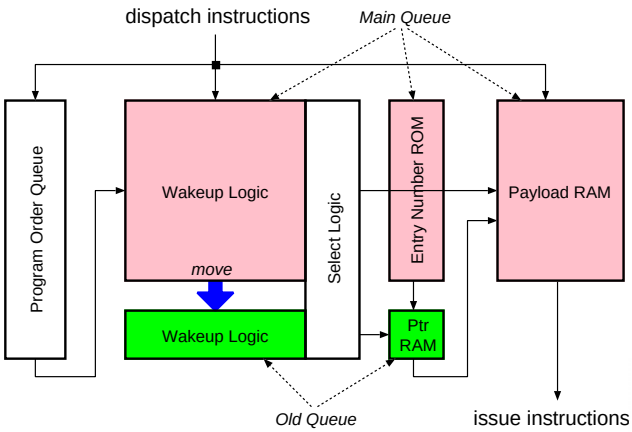

図 1: RRQ の構成

とした IQ は、完全に要求を満たす構成ではないものの、現在の多くのプロセッサで用いられている [3–5]。

これに対して本研究では、RAND をベースとする**再並べ替えランダム・キュー** (RRQ: rearranging random queue) と呼ぶ IQ の構成法を提案した。RRQ は、低電力、低遅延、高 IPC を同時に実現する IQ の構成法である。

4.2 再並べ替えランダム・キュー

4.2.1 概要

概念的には、RRQ はランダム IQ を以下の 2 つの部分に分割したものである: **古いキュー** (OQ: old queue) と呼ぶ小さな部分と**主キュー** (MQ: main queue) と呼ぶ残りの大きな部分。ただし、選択論理は MQ と OQ の間で共有している。MQ も OQ もランダム・キューであり、論理的には 2 つ合わせて 1 つの単一のキューを構成している。命令は MQ の方にディスパッチされる。

毎サイクル、MQ の中の最も古い少数の命令を OQ に移動する。OQ は MQ の下に配置し、命令は従来の位置ベースで優先づけする選択論理によって選択され、発行される。選択論理は、IQ の下のエントリほど高い優先度で命令を選択するので、年齢を意識した選択が実現される。

MQ の中で最も古い命令を見つけるために、**プログラム・オーダ・キュー** (PQ: program order queue) と呼ぶ追加のキューを用意する。PQ はサーキュラ・キューであり、そのエントリはディスパッチされる命令にプログラム順に割り当てられる。命令が MQ の第 n エントリに割り当てられた時、n が PQ の末尾エントリに書き込まれる。つまり、PQ のエントリは、関連する命令を保持する MQ のエントリへのポインタを保持する。毎サイクル、PQ の先頭から複数のポインタが読み出され、その後、これらのポインタが指している MQ の命令が読み出され、OQ の空きエントリに書き込まれる。

4.2.2 構成と実装

図 1 に、RRQ の詳細な構成を示す。同図に示すように、MQ はウェイクアップ論理、ペイロード RAM、エントリ番号 ROM(ENR) を持ち、OQ はウェイクアップ論理とポインタ RAM(ptr RAM)

表 1: ベース・プロセッサの構成

Pipeline width	4-instruction wide for each of fetch, decode, issue, and commit
Reorder buffer	128 entries
IQ	64 entries
Load/store queue	64 entries
Physical registers	128(int) + 128(fp)
Branch prediction	12-bit history 4K-entry PHT gshare, 2K-set 4-way BTB, 10-cycle misprediction penalty
Function unit	2 iALU, 1 iMULT/DIV, 2 Ld/St, 2 FPU
L1 I-cache	32KB, 8-way, 64B line
L1 D-cache	32KB, 8-way, 64B line, 2 ports, 2-cycle hit latency, non-blocking
L2 cache	2MB, 16-way, 64B line, 12-cycle hit latency
Main memory	300-cycle min. latency, 8B/cycle bandwidth
Data prefetch	stream-based: 32-stream tracked, 16-line distance, 2-line degree, prefetch to L2 cache

表 2: RRQ 固有のパラメータ

OQ	4 entries
MQ	60 entries
PQ	96 entries, 6 bits per entry, 4 ports

を持つ。選択論理とペイロード RAM は 2 つのキューで共有されている。ただし、ペイロード RAM は MQ にしか含まれない。ここで、ENR の各エントリは自身のエントリ番号を保持し、ptr RAM は、ペイロード RAM の対応するエントリへのポインタを保持する回路である。MQ 内の命令を OQ に移動するとは、具体的には、MQ のウェイクアップ論理と ENR が保持するデータを OQ のウェイクアップ論理と ptr RAM にそれぞれ移動することである。

4.3 評価結果

4.3.1 評価方法

SimpleScalar ツール・セット (ver.3.0a) [1] をベースとしたシミュレータを作成し評価した。使用した命令セットは Alpha ISA である。評価プログラムに SPEC2006 を用いた。プログラムは gcc ver.4.5.3 を用い、最適化オプション`-O3` でコンパイルした。ベース・プロセッサの構成を表 1 に示す。

ref 入力を用い、16B 命令スキップした後の 100M 命令をシミュレートした。RRQ に関するパラメータを表 2 に示す。

4.3.2 IPC 評価結果

図 2 に、各ベンチマーク・プログラムについて、種々の IQ の SHIFT に対する IPC の低下率を示す。低下率なので、高い棒グラフほど IPC は悪いことを示しており、低い棒グラフほどよい結果である。X 軸において、プログラムは “comp”(計算インテンシブ) と “mem”(メモリ・インテンシブ) に分類されている。また、“GM comp” と “GM mem” は、それぞれ、計算インテンシブ、メモリ・インテンシブ・プログラムの性能低下の幾何平均を示している。“comp” と “mem” に分類

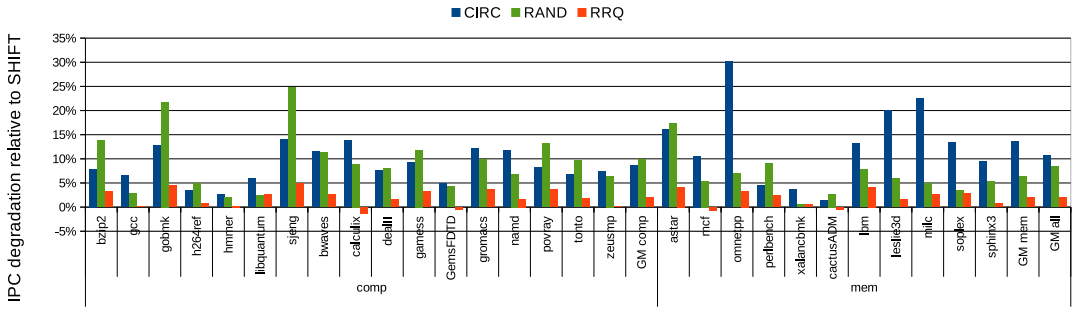

図 2: 種々の IQ の SHIFT に対する性能低下率

したのは、以下の理由による。計算インテンシブ・プログラムでは性能が命令の年齢に敏感であるのに対し、メモリ・インテンシブ・プログラムではそうではない。メモリ・インテンシブ・プログラムでは、最終レベル・キャッシュ・ミスが頻繁に起こり、ロード・レイテンシが非常に長い。このため、命令の年齢は性能に対して重要でない。

図 2 に示すように、計算インテンシブ・プログラムにおいては、CIRC と RAND の性能低下は著しい。平均では、CIRC と RAND の SHIFT に対する性能低下は、それぞれ、8.7%、9.8%であった。CIRC では、容量使用における不効率性が、RAND では、年齢を考慮しない命令スケジューリングが、大きな IPC 低下を引き起こしている。これに対して RRQ は、年齢を意識した命令スケジューリングが実現されているため、IPC 低下を非常に小さく (1.9%) に抑えることができている。

メモリ・インテンシブ・プログラムでは、CIRC の IPC 低下は、容量不効率性によりやはりほとんどのプログラムで大きい。これに対して、RAND はより良い IPC を示している。これは、容量の高効率性によりメモリ・レベル並列が効果的に利用されているためである。一方、RRQ では、さらに良い IPC を達成している。SHIFT に対する性能低下はわずかに 1.9%である。

4.4 結論

本研究では、RRQ と呼ぶ新しい IQ の構成法を提案した。RRQ は、プロセッサの高性能化に対する高い IPC と短い遅延を同時に満たし、低電力なランダム発行キューにおいて年齢を十分に意識した命令選択を世界で初めて可能にした IQ 構成である。評価の結果、RRQ は、既存の IQ の中で最も高い IPC を達成するシフティング・キューと同程度 (わずか 1.9%の性能低下) の高い IPC を達成することがわかった。

参考文献

[1] http://www.simplescalar.com/.

[2] J. A. Farrell *et al.*, “Issue logic for a 600-MHz out-of-order execution microprocessor,” *JSSC*, vol. 33, issue 5, pp. 707–712, May 1998.

[3] R. P. Preston *et al.*, “Design of an 8-wide superscalar RISC microprocessor with simultaneous multithreading,” in *ISSCC*, February 2002, pp. 334–472.

[4] M. Golden *et al.*, “40-entry unified out-of-order scheduler and integer execution unit for the AMD Bulldozer x86-64 core,” in *ISSCC*, February 2011, pp. 80–82.

[5] B. Sinharoy *et al.*, “IBM POWER8 processor core microarchitecture,” *IBM JRD*, vol. 59, issue 1, pp. 2:1 – 2:21, January - February 2015.

5 主な発表論文等

[雑誌論文] (計 2 件)

1. K. Doi, R. Shioya, and H. Ando, “Performance Improvement Techniques in Tightly Coupled Multicore Architectures for Single-Thread Applications,” IPSJ Journal of Information Processing, Vol.26, pp.445-460, June 2018. (査読あり、DOI: https://doi.org/10.2197/ipsjjip.26.445)
2. R. Shioya, R. Takami, M. Goshima, and H. Ando, “FXA: Executing Instructions in Front-End for Energy Efficiency,” IEICE Transactions on Information and Systems, Vol.E99-D, No.4, pp.1092-1107, April 2016. (査読あり、DOI: https://doi.org/10.1587/transinf.2015EDP7316)

[学会発表] (計 7 件)

1. 劉兆良, 塩谷亮太, 安藤秀樹, “グループ化したストリームからのフィードバックを用いたストリーム毎に最適化するストリーム・プリフェチャの高効率化,” 情報処理学会研究報告, Vol.2019-ARC-235, No.21, pp.1-17, 2019 年 3 月.
2. H. Ando, “Performance Improvement by Prioritizing the Issue of the Instructions in Unconfident Branch Slices,” In Proceedings of the 51st Annual International Symposium on Microarchitecture, pp.82-94, October 2018.
3. S. Sakai, T. Suenaga, R. Shioya, and H. Ando, “Rearranging Random Issue Queue with High IPC and Short Delay,” In Proceesings of the 36th IEEE International Conference on Computer Design, pp.123-131, October 2018.
4. 李虹希, 塩谷亮太, 安藤秀樹, “SRAM の電力/遅延シミュレータ CACTI へのシングルエンド方式の対応,” 情報処理学会研究報告, Vol.2018-ARC-232, No.15, 2018 年 7 月.
5. 松尾玲央馬, 塩谷亮太, 安藤秀樹, “パイプライン構造の動的制御による命令フェッチ・スループットの向上,” 情報処理学会研究報告, Vol.2018-ARC-232, No.3, 2018 年 7 月.
6. Y. Chidai, K. Izuoka, R. Shioya, M. Goshima, and H. Ando, “A Tightly Coupled Heterogeneous Core with Highly Efficient Low-Power Mode,” In Proceedings of the 31st International Conference on Architecture of Computing Systems, pp.211-224, April 2018.

6 研究組織

研究分担者、研究協力者はいない。

※科研費による研究は、研究者の自覚と責任において実施するものです。そのため、研究の実施や研究成果の公表等については、国の要請等に基づくものではなく、その研究成果に関する見解や責任は、研究者個人に帰属されます。