

令和 6 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：2020～2023

課題番号：20K11732

研究課題名（和文）ムーアの法則破綻後のマイクロプロセッサの高性能化・低電力化に関する研究

研究課題名（英文）Study on microprocessors for high performance and low power consumption after breakdown of Moore's law

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） 3,300,000円

研究成果の概要（和文）：ムーアの法則によるLSI技術の進歩は近年その速度が低下しており、近い将来進歩しなくなると予想されている。このような状況下においてはマイクロプロセッサの性能を向上させることは難しい。このため商用プロセッサでは、世代が進むごとに命令の実行順序を最適化する回路である発行キュー（IQ: issue queue）のサイズを大きくしている。しかし現在のIQの方式においてはIQを拡大するほど実行順序の十分な最適化を行うことが難しくなり、性能向上は頭打ちになるという問題がある。本研究ではIQ内で最も古い複数の実行可能命令を選択することにより高い性能を達成する方式を提案した。

研究成果の学術的意義や社会的意義

マイクロプロセッサは現在PC、スマートフォンなどに広く使われおり、巨大な市場を形成している。今後、ITがさらに浸透すると、様々な作業がコンピュータにより自動化される。このために、多くの機器に広くコンピュータは利用され、市場規模は飛躍的に拡大していく。一方で、性能の原動力であったムーアの法則が終わることから、性能向上は非常に困難という問題が立ちはだかっている。しかし、本研究により、ムーアの法則の破綻後も性能を向上させることができ、学術上の大きなブレイクスルーとなるだけでなく、産業上もきわめて大きな意義がある。

研究成果の概要（英文）：LSI technology advancement by Moore's law is sluggish recently. In the near future, it is predicted that the advancement is stopped. In this situation, it is difficult to improve the performance of microprocessors. Commercial processors increase the size of the issue queue (IQ) at each of their new generation. However, in the current IQ organization, the optimization of the instruction execution becomes more difficult, as the IQ size is increased and thus performance improvement is suppressed. In this study, I proposed an IQ organization that achieves high performance by selecting multiple ready instructions in the IQ.

研究分野：コンピュータ・アーキテクチャ

キーワード：マイクロプロセッサ 発行キュー

## 1 研究開始当初の背景

ムーアの法則による LSI 技術の進歩 (加工寸法縮小) は、近年その速度が低下しており、近い将来、進歩しなくなると予想されている。このような状況下においては、マイクロプロセッサの性能を向上させることは難しい。すでに 10 年以上前からクロック速度を大きく向上させることはほぼ不可能となっているので、性能向上のためには、クロックの速度向上に期待することなく、アーキテクチャの改善により性能を改善する必要がある。より具体的には、命令の実行順を最適化し、1 サイクル当たり実行される平均命令数 IPC (instructions per clock cycle) を上げる必要がある。このため、商用プロセッサでは、世代が進むごとに、命令の実行順序を最適化する回路である発行キュー (IQ: issue queue) のサイズを大きくしている。これは、一般に、IQ のサイズを大きくするほど命令の実行順序が最適化される機会が増大し、IPC が向上するからである。しかし、現在の IQ の方式においては、IQ を拡大するほど実行順序の十分な最適化を行うことが難しくなり、IPC 向上は頭打ちになるという問題がある。

## 2 研究の目的

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用している。この方式では IQ における命令の並びが、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、実行可能となった命令の中で、古い命令ほど優先して実行することが要求されている。ランダムな並びの命令から古い命令を選択して実行することは難しい。この困難さは、IQ のサイズが大きいほど高まる。IQ を拡大すると命令の実行順が最適化される「機会」は増大するが、実際に十分な最適化を実現するには、現在の IQ の方式の素朴な延長では IQ の遅延が増大し、その結果クロック速度が低下し、逆に性能を低下させてしまうという問題がある。本研究では、IQ 内の回路構成を工夫し、大きな IQ において、IQ の遅延を増大させることなく「実際に」命令の実行順をより最適化し、プロセッサの性能を向上させることを目的としている。

## 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、SPEC2017 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

プロセス技術の進化の速度が低下し、この約 10 年は、クロック周波数をあげることによるプロセッサの性能向上は難しくなった。このため、アーキテクチャの改善が以前にも増して重要となっている。このような改善の中で、IQ サイズの拡大は最も有効な方法の 1 つである。しかし、IQ サイズが増加するに伴って、現在の IQ の構成では、命令の実行順を定めるスケジューリングの最適度が低下する。

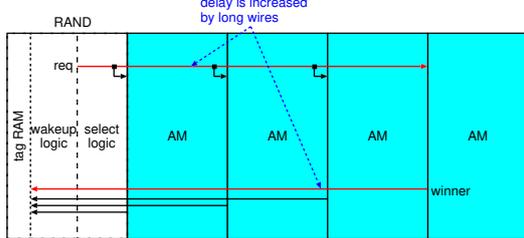


図 1: 4 つの AM を配置した IQ のレイアウト

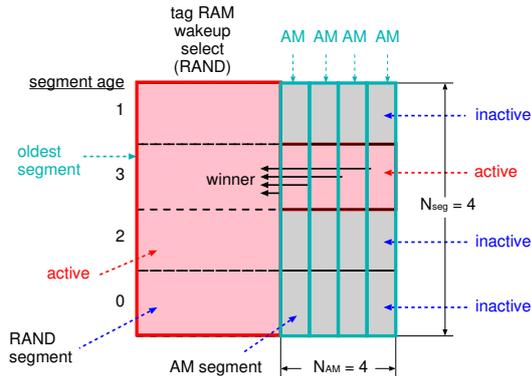


図 2: SegAM の構成例 ( $N_{AM} = 4, N_{seg} = 4$ )

命令スケジューリングの最適化手法として、過去に様々なものが検討されたが [2]、現在では年齢ベースのスケジューリングが単純で最も有効として採用されている。年齢ベースのスケジューリングを実現するために現在のプロセッサは、ランダム・キュー (RAND) にエイジ論理 (AM: age matrix) [3,4] を加えた構成を採用している。この方式は、命令を発行によって空いたエントリに新たな命令は書き込み、AM は、実行可能な命令の中から最も古い 1 つの命令を同定し、最も高い優先度で発行するものである。

この方式は、単純、低電力という利点があるが、IQ が大きくなるに伴って最適度が低下し、IPC の向上が抑制されるという問題がある。なぜなら、AM は最も古い「1 命令」だけを選択するものであり、他の発行命令はランダムに選ばれるからである。より高い最適度を得るためには、最も古い「複数」の命令を選択する必要がある。

現在の IQ の構成を単純に拡張し、最も古い複数の命令を選択できるようにするには、複数の AM を用意すればよい。しかし、この方法は IQ の遅延を著しく増加させ、クロック速度を低下させてしまうという問題がある。

図 1 に、例として 4 つの AM を備えた IQ のレイアウトを示す。各回路の大きさは実際の大きさに比例している (実際の大きさは、MOSIS 設計ルール [5] に基づきトランジスタ・レベルでの設計により求めた)。AM を横断する発行要求信号 (req) と最も古い命令を指示する信号 (winner) が複数の AM 配置により著しく伸びていることがわかる。これによる配線遅延の増加は非常に大きく、IQ の遅延を著しく増加させる。

## 4.2 セグメント化エイジ論理

本研究では、IQ の遅延を増加させることなく最も古い複数の命令を選択する SegAM(segmented age matrices) と呼ぶ IQ の構成と方式を提案する。図 2 に、例として、AM の数 ( $N_{AM}$ ) とセグメント数 ( $N_{seg}$ ) がともに 4 である SegAM 構成のレイアウトを示す。図の各回路の大きさは実際の大きさに比例している。

SegAM では、従来の 1 つの AM を、セグメントと呼ぶ複数の部分に分割する。この分割は物理的なものであり、各 AM セグメントは小さな独立した AM である。AM は、入力 (req) 数の 2 乗の次元の行列回路なので、1 つの AM を  $N_{AM}$  に分割すると、1 つの AM セグメントは  $IQS/N_{seg}$  の次元の行列回路となる ( $IQS$  は IQ のサイズ)。したがって、各 AM セグメントは元の AM の  $1/N_{seg}^2$  に非常に小さくなる。これにより、AM セグメント自身の遅延と共に、AM の上を横断する配線遅

表 1: ベース・プロセッサの構成

6-instruction wide pipeline, 352-entry reorder buffer, 176-entry w/ 1-AM IQ, 3 iALU, 1 iMULT/DIV, 2 Ld/St, 2 FPU 48KB L1 I-cache, 48KB L1 D-cache, 2MB L2 cache
---

延も減少する。

ただし、セグメント化すると IQ 全体から最も古い命令を選択することができなくなってしまう。そこで、この能力を回復するために新たな手法を導入する。まず、IQ の AM を除いた部分 (RAND) を論理的に分割する。各 RAND セグメントは、図 2 に示すように、AM セグメントと行で対応している。

次に、命令のディスパッチ (IQ への書き込み) をセグメントごとに行う。つまり、命令は 1 つのセグメントに対し、それが一杯になるまでディスパッチを続ける。一杯になったら新しく空いているセグメントを選びそこにディスパッチを続ける。このセグメント単位でのディスパッチにより、セグメントを年齢で順序づけることができる。

最後に、図 2 に示すように、最も古い AM セグメントのみ活性化し、他の AM セグメントは非活性化させる。活性化させた AM セグメントは、最も古いセグメントの中の最も古い実行可能命令を選択するので、選択された命令は IQ 全体の中で最も古い実行可能命令となる。図 2 の例では、1 セグメントに 4 つの AM セグメントが備えられているので、最も古い 4 つの実行可能命令を選択することができる。

## 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。このシミュレータは、近年のアーキテクチャに適合させるよう修正している。使用した命令セットは Alpha ISA である。評価プログラムに SPEC2017 を用いた。ベース・プロセッサの構成を表 1 に示す。IQ を含む命令ウィンドウのサイズは、Intel Sunny Cove [6] を基本とした。refspeed 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。

### 4.3.2 性能評価結果

従来の IQ の遅延と面積を超えないという制約の下で、 $(N_{AM}, N_{seg})$  の種々の組み合わせで性能 (IPC) を測定した。その結果、最も高い IPC を達成したパラメータは  $(N_{AM}, N_{seg}) = (3, 4)$  であった。この時、基準プロセッサ (1AM の IQ を持つプロセッサ) に対する性能向上率は、INT (整数プログラム)、FP (浮動小数点プログラム) における平均でそれぞれ、6.4%、1.2% (最大でそれぞれ、17.7%、11.4%) であった。以下、パラメータ  $N_{AM}, N_{seg}$  を振ったときの性能 (IPC) 評価結果について述べる。

図 3 に、 $N_{seg} = 4$  に固定し、 $N_{AM}$  を振ったときの基準プロセッサに対する性能向上を示す。遅延制約については、測定した全ての場合で満たしているが、面積制約については、 $N_{AM} = 1, 3$  の場合のみが満たしている ( $N_{AM} = 5$  のときは満たしていない)。同図に示すように、 $N_{AM} = 3$  まで AM の数を増加させることにより性能が向上することがわかる。

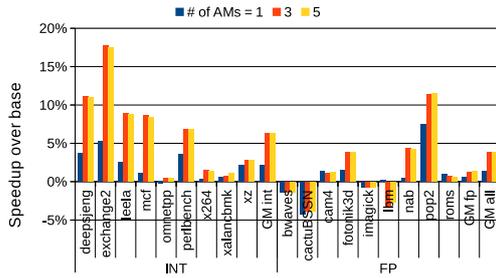


図 3:  $N_{seg} = 4$  に固定し、 $N_{AM}$  を振ったときの基準プロセッサに対する SegAM の性能向上率

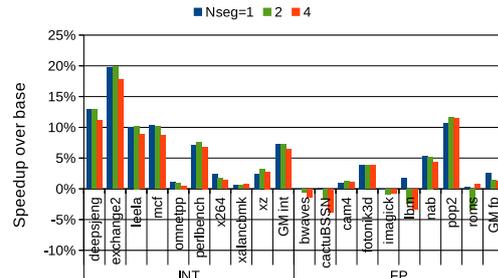


図 4:  $N_{AM} = 3$  で固定し、 $N_{seg}$  を振ったときの基準プロセッサに対する SegAM の性能向上率

図 4 に、 $N_{AM} = 3$  に固定し、 $N_{seg}$  を振ったときの基準プロセッサに対するベンチマーク平均性能向上を示す。遅延制約については、 $N_{seg} = 2, 4$  の場合で満たすが、 $N_{seg} = 1$  では満たさない。また、面積制約については、 $N_{seg} = 4$  では満たすが、 $N_{seg} = 1, 2$  では満たさない。

同図に示すように、 $N_{seg}$  を増加させると、性能はわずかに低下する。これは、IQ をセグメント化しているために、IQ の容量効率が低下するためである。しかし、性能低下はわずかであり、INT、FP でそれぞれ 0.8%、1.2% である。

#### 4.4 結論

本研究では、複数の AM を使って高い IPC を達成する SegAM と呼ぶ IQ の構成と方式を提案した。複数の AM により IQ 内で最も古い複数の実行可能命令を選択することができるため、高い IPC を達成することができる。AM をセグメントに物理的に分割し、単純に複数の AM を配置する方法と比べて、非常に小さな面積と小さな遅延で実装することができる。セグメントごとに命令にディスパッチすることにより、AM をセグメント化しても最も古い複数の実行可能命令を IQ 全体から選ぶ方法と変わらない IPC を達成することができた。評価の結果、IQ の遅延、面積を増加させることなく INT、FP についてベンチマーク平均で 6.4%、1.2% (最大、17.7%、11.4%) の性能向上率を得ることができることを確認した。

#### 参考文献

- [1] <http://www.simplescalar.com/>.
- [2] M. Butler *et al.*, “An investigation of the performance of various dynamic scheduling techniques,” in *MICRO-25*, Dec. 1992, pp. 1–9.
- [3] R. P. Preston *et al.*, “Design of an 8-wide superscalar RISC microprocessor with simultaneous multithreading,” in *ISSCC-2002*, Feb. 2002, pp. 334–472.
- [4] P. G. Sassone *et al.*, “Matrix scheduler reloaded,” in *ISCA-34*, Jun. 2007, pp. 335–346.
- [5] <https://www.mosis.com/>.
- [6] D. Kanter, “Intel’s Sunny Cove sits on an Icy Lake,” *MPR*, Feb. 2019.

5. 主な発表論文等

〔雑誌論文〕 計1件（うち査読付論文 1件/うち国際共著 0件/うちオープンアクセス 1件）

1. 著者名 Y. Matsuda, R. Shioya, and H. Ando	4. 巻 E105-D
2. 論文標題 Reducing Energy Consumption of Wakeup Logic through Double-stage Tag Comparison	5. 発行年 2022年
3. 雑誌名 IEICE Transactions on Information and Systems	6. 最初と最後の頁 320-332
掲載論文のDOI（デジタルオブジェクト識別子） 10.1587/transinf.2021EDP7174	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 -

〔学会発表〕 計8件（うち招待講演 1件/うち国際学会 1件）

1. 発表者名 H. Ando
2. 発表標題 Segmenting Age Matrices to Improve Instruction Scheduling without Increasing Delay and Area
3. 学会等名 The 40th IEEE International Conference on Computer Design（国際学会）
4. 発表年 2022年

1. 発表者名 黒川陸，安藤秀樹
2. 発表標題 整数命令のレイテンシ耐性を利用したALUの消費電力削減手法，
3. 学会等名 情報処理学会研究報告，vol.2023-ARC-251，No.7
4. 発表年 2023年

1. 発表者名 彭南翔，安藤秀樹
2. 発表標題 フィルタ・キャッシュにおける低電力化置換ポリシー
3. 学会等名 情報処理学会研究報告，Vol.2022-ARC-248，No.18
4. 発表年 2022年

1. 発表者名 森健一郎, 安藤秀樹
2. 発表標題 容量効率を意識したソース・タグ値に基づくセグメント化による発行キューのエネルギー削減
3. 学会等名 情報処理学会研究報告, Vol.2020-ARC-241, No.3
4. 発表年 2020年

1. 発表者名 後岡瑞希, 安藤秀樹
2. 発表標題 発行キューの電力削減のための発行幅制御方式
3. 学会等名 情報処理学会研究報告, Vol.2021-ARC-244, No.35
4. 発表年 2021年

1. 発表者名 安藤秀樹
2. 発表標題 SWQUE: 優先度を修正するサーキュラー・キューを持ったモード切り替え発行キュー
3. 学会等名 IEICE/IPSJ 第19回情報科学技術フォーラム, トップカンファレンス・セッション(招待講演)
4. 発表年 2020年

1. 発表者名 近藤祐生, 安藤秀樹
2. 発表標題 TAGE分岐予測器がプロセッサ性能に与える影響の評価
3. 学会等名 情報処理学会研究報告, vol.2023-ARC-255, No.26
4. 発表年 2023年

1. 発表者名 松浦淳, 安藤秀樹
2. 発表標題 選択的レベル2命令キャッシュの追加によるキャッシュの消費電力の削減
3. 学会等名 情報処理学会研究報告, vol.2023-ARC-255, No.26
4. 発表年 2023年

〔図書〕 計0件

〔産業財産権〕

〔その他〕

-

6. 研究組織

氏名 (ローマ字氏名) (研究者番号)	所属研究機関・部局・職 (機関番号)	備考

7. 科研費を使用して開催した国際研究集会

〔国際研究集会〕 計0件

8. 本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関