名古屋大学大学院工学研究科博士前期課程
修士学位論文

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電力削減

令和 3 年 3 月

情報・通信工学専攻
森　健一郎

概要

発行キューは電力密度の大きいホット・スポットとして知られている．ホット・スポットは，デバイスの摩耗故障を引き起こし，誤動作やタイミング・エラーを引き起こす．発行キューが大きな電力を消費する原因は，ウェイクアップ論理のタグ比較回路である．この回路は CAM で構成されており，全てのデスティネーション・タグと発行キュー内の全てのソース・タグとの多数の比較を一斉に行うため，非常に大きな電力を消費する．そこで本論文では，CAM の分野で提案されている手法を応用し，タグ比較による消費電力を削減する手法を提案する．本手法では，発行キューを複数のセグメントに分割する．命令は，ソース・タグの下位ビットがセグメント番号と一致するセグメントにディスパッチする．そして，ウェイクアップ時には，ディスティネーション・タグの下位ビットが一致するセグメントにあるタグ比較器のみを動作させる．一致しないセグメントの比較器は動作しないため，タグ比較器の動作回数を削減できる．

本手法では，命令がディスパッチされるセグメントに空きがない場合，他のセグメントに空きがあってもディスパッチできないためストールする．この結果，発行キューの容量効率が低下するという問題が生じる．この問題は，発行キューの容量効率が重要なプログラムにおいて性能低下を引き起こす．そこで本論文では，容量効率を重視したディスパッチ・アルゴリズムと，タグ比較の積極的な削減を重視したディスパッチ・アルゴリズムを動的に切り替える手法を提案する．本手法は，発行キューの容量効率が重要な場合は容量効率の低下による性能低下を抑制し，そうでない場合は積極的にタグ比較器の動作回数を削減することを可能とする．提案手法を SPEC CPU 2017 を用いて評価を行った．結果，性能低下を最大で 5% 以下（平均 -1%）に抑えつつ，タグ比較器の動作回数を平均で 85% 削減できることを確認した．

目 次

第 1 章　はじめに

現在のプロセッサは，非常に微細な LSI 技術で製造される．このような LSI の微細化に伴い，デバイスの信頼性低下の問題が深刻になっている [1]．微細化は，経年劣化や摩耗故障を加速し，その結果，タイミング・エラーや誤動作を引き起こし，デバイスの寿命を縮める．経年劣化や摩耗故障は温度に関して指数関数的に加速し [2–4]，温度 10〜15 ℃の上昇でデバイスの寿命は半分以下になる [5]．

プロセッサ・チップ上には，ホット・スポットと呼ばれる単位面積あたりの電力が大きい場所が存在する．ホット・スポットは，そうでない場所と比べて温度上昇が激しいため，上述した故障を引き起こす確率が高くなる．従って，ホット・スポットを生成する回路の消費電力を低下させる必要がある．

ホット・スポットを生成する回路の 1 つに，発行キューがある．発行キューのサイズはプロセッサの世代が進むごとに大きくなっており，より深刻なホット・スポットとなっている．従って，発行キューの電力削減に対する要求は非常に大きい．

発行キューの中で最も電力を消費する回路は，タグ比較の回路である．タグ比較は，発行幅分のディスティネーション・タグとすべてのソース・タグとの間で行われるため，非常に多くの電力を消費する．そこで本論文では，タグ比較器が動作する回数を削減する以下のような手法を提案する．

- 発行キューを複数の**セグメント**に分割する．命令を発行キューにディスパッチする際，第 1 ソース・タグの下位ビットが n である命令は，第 n 番目のセグメントに書き込む．タグ比較時には，ディスティネーション・タグの下位ビットがセグメント番号と一致するセグメントでのみ，第 1 ソース・タグの比較を行う．一致しないセグメントでは比較が行われない．これによりタグ比較回数が削減される．

- 上記の方法では，第 2 ソース・タグの比較回数は削減されない．そこで提案手法では**スワップ**と**サブ・セグメント**と呼ぶ 2 つの方法を導入し，第 2 ソース・タグの比較回数も削減する．スワップは，ディスパッチ時に第 1 ソース・オペランドがレディで，第 2 ソース・オペランドがレディでない命令において，第 1 ソース・タグと第 2 ソース・タグを格納するフィールドを交換し，第 2 ソース・タグの下位ビットを用いてディスパッチするセグメントを決定する手法である．サブ・セグメントは，各セグメントを第 2 ソース・タグにもとづきさらに分割する手法である．

- セグメント化によりディスパッチできるエントリが制限されるため，発行キューの容量効率が低下し，容量に敏感なプログラムにおいて性能が低下するという問題が存在する．この問題に対応するため，本論文では **SWITCH** という手法を提案する．SWITCH では，容量効率を重視したディスパッチ・アルゴリズムと，タグ比較回数の削減を重視したディスパッチ・アルゴリズムを，容量効率の重要性に応じて切り替えて使用することにより，性能低下を抑制する．

提案手法を SPEC CPU 2017 ベンチマークを用いて評価し，性能低下を 最大でも 5% 以下（平均 -1%）に抑えつつ，タグ比較の回数を平均で 85% 削減できることを確認した．

本論文の残りの構成は次の通りである．まず，**??**節で発行キューの基本的な事項を説明する．そして，**??**節で提案手法の基本となるアイデアに関して説明した後，**??**節で提案手法における第 2 ソース・タグのタグ比較回数削減方法に関して述べる．その後，**??**節で提案手法の問題点である発行キューの容量効率の低下に関して説明した後，**??**節で容量効率の低下に対する対策方法を説明する．**??**節で評価を行い，3 節でまとめる．

第 2章　発行キュー (IQ：Issue Queue)

本章では，本研究の研究対象である、IQ に関して説明する．まず，IQ の概要と動作を 2.1 節で説明したあと，IQ の回路構成を 2.2 節で述べる．その後，2.3 節で IQ の方式に関して説明し，最後に研究の動機である IQ の問題点に関して 2.4 節で述べる．

2.1　概要と動作

IQ はアウト・オブ・オーダ実行を行うプロセッサにおいて，リネームされた命令を保持し，実行順序をスケジューリングして，機能ユニットへ発行する回路である．IQ は，ディスパッチ，発行，ウェイクアップと呼ばれる 3 種類の動作を行う．以下でそれぞれの動作に関して説明する．

- ディスパッチ：リネームされた命令は，IQ にエントリが割り当てられ，命令の情報が格納される．この動作をディスパッチと呼ぶ．ディスパッチの動作は，IQ の方式により異なる．IQ の方式に関しては，2.3 節で詳しく説明する．

- 発行：IQ 内の命令のうち，ソース・オペランドが両方共レディとなった命令は，依存関係が解消し，実行が可能となる．このような命令を実行ユニットに送出する動作を発行と呼ぶ．なお，発行可能な命令が機能ユニットの数を超える場合 (このような場合を発行コンフリクトと呼ぶ) は，各命令の発行優先度に基づき命令を選択して発行する．発行された命令のエントリは IQ より削除される．

- ウェイクアップ : 命令が発行されると，その命令のディスティネーション・オペランドのタグと発行キュー内にある全命令のソース・オペランドのタグの比較が行われる．比較が一致した場合には, 対応するソース・オペランドのレディ・ビットをセットす

る．この動作をウェイクアップと呼ぶ．両方のオペランドがレディとなった命令は，依存が解消したため発行可能となる．

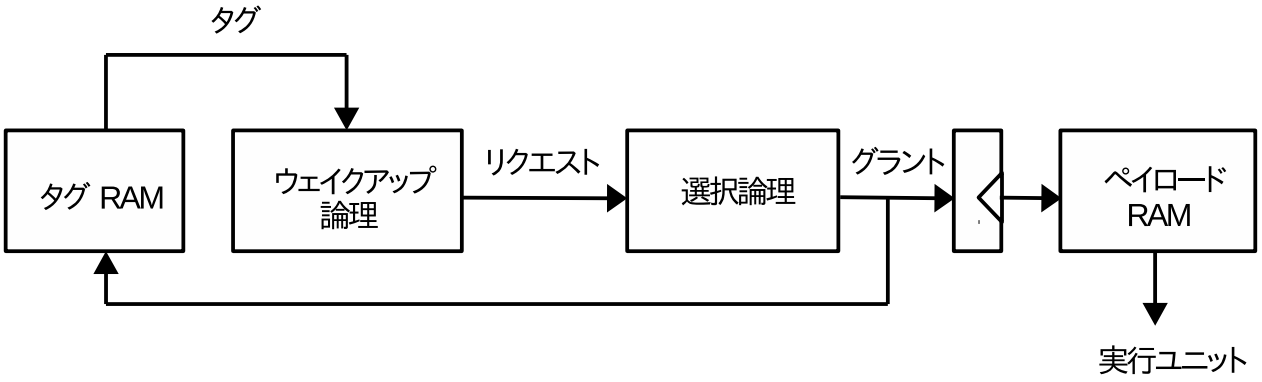

図 2.1: IQ の回路構成

2.2 回路構成

図 2.1 に IQ の回路構成を示す．IQ はウェイクアップ論理，選択論理，タグ RAM，ペイロード RAM と呼ばれる 4 つの回路より構成される．以下で各回路に関して説明する．また，IQ の回路のうちウェイクアップ論理は提案手法に関わる重要な回路であるため，2.2.1 節にて詳細に説明する．

- ウェイクアップ論理 : 命令感の依存関係を管理し，他の命令との依存関係が解消された命令に対して発行要求 (リクエスト信号) を出す．
- 選択論理 : 資源制約を考慮して，発行を要求された命令の中からそれを許可する命令を選択肢，発行許可信号 (グラント信号) を出力する．この選択においては，回路構成の単純化のために IQ の先頭のエントリの命令をより優先する．

- タグ RAM : 発行待機中の命令のディスティネーション・タグを保持する回路で，選択論理から発行許可信号が送られると，対応する命令のタグを読み出し，それをウェイクアップ論理へ送る．

- ペイロード RAM : 発行待機中の命令の命令のコードを保持する．選択論理から発行許可信号が送られると，対応する命令のコードを実行ユニットに送出する．

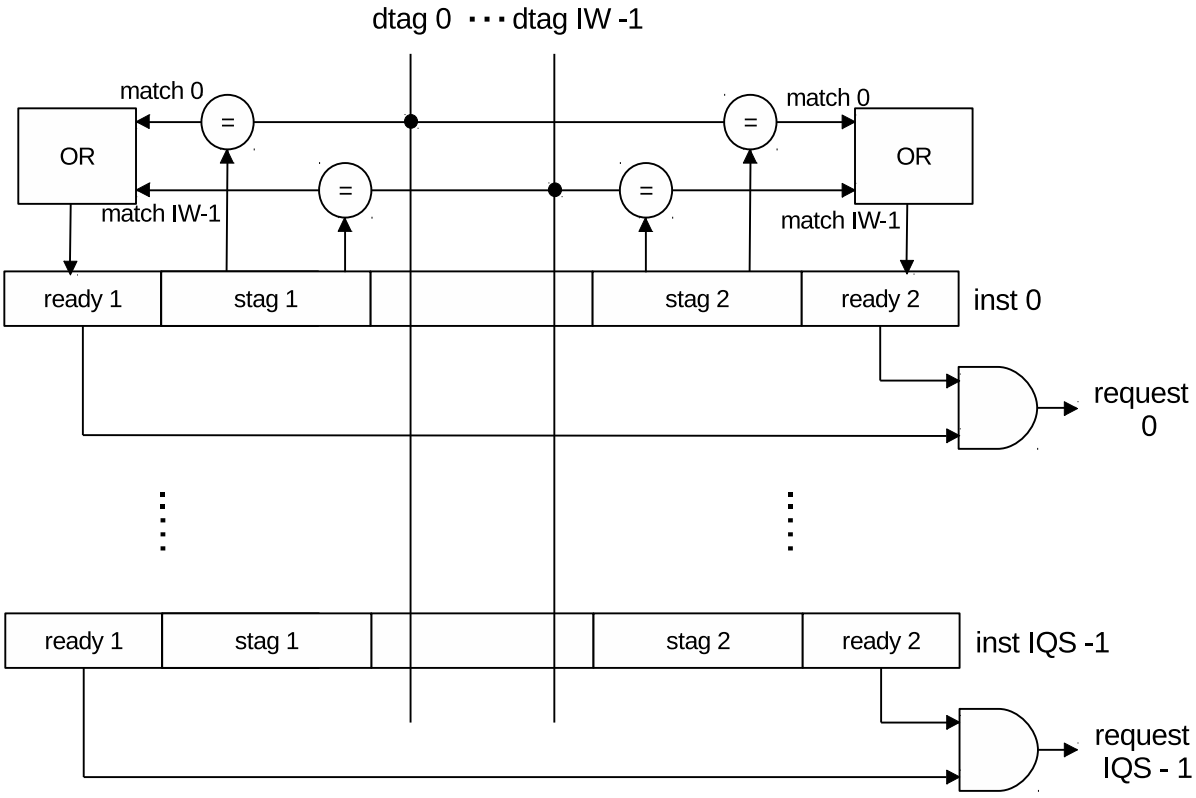

図 2.2: ウェイクアップ論理

2.2.1 ウェイクアップ論理

図 2.2 に，ウェイクアップの回路を示す．図中の IW は発行幅を，IQS は発行キューのエントリ数を表す．ウェイクアップでは，IW 個のディスティネーション・タグ（dtag）が発行キュー内の全命令に放送される．各命令は 2 つのソース・タグ (stag）を保持してお

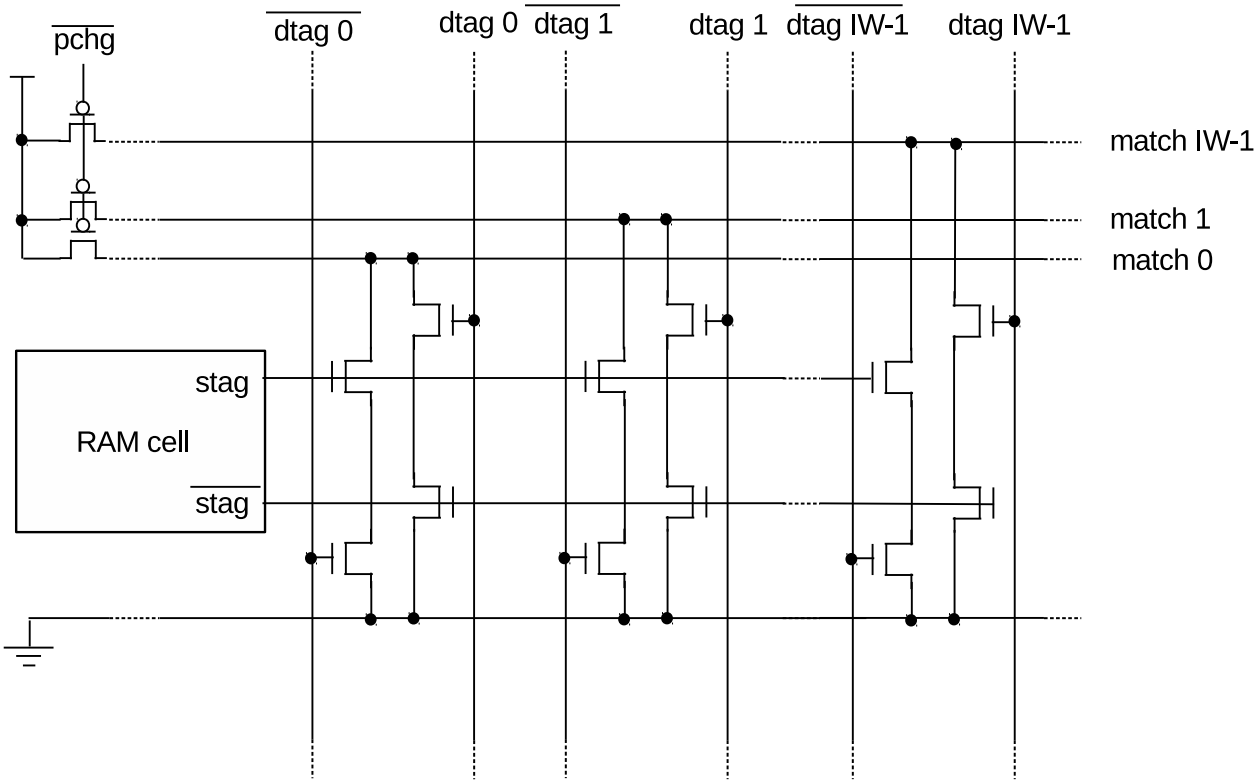

図 2.3: タグ比較器の CAM 回路

り，放送されたディスティネーション・タグと比較が行われる．いずれかのディスティネーション・タグとソース・タグが一致した場合，そのソース・オペランドのレディ・ビットがセットされる．2 つのレディ・ビットがセットされた命令は発行が可能となるため発行要求が出力される．

図 2.3 に発行キューに使用されるタグ比較器の CAM 回路を示す．同図は，ソース・タグ 1 ビット分の比較回路を表す．同図に示すように，高速化のため通常ダイナミック論理によって構成される．比較の動作は，次のように行われる．まず，マッチ線がプリチャージされる．次にデスティネーション・タグが放送され，比較が行われる．タグが不一致であれば，直列に接続された 2 つのプルダウン・トランジスタが両方とも ON となり，マッチ線がディスチャージされる．タグが一致する場合，マッチ線は H の状態が維持される．

比較器はマッチ線のディスチャージ時に電力を消費する．

2.3 IQ の方式

これまで，IQ の方式としてシフト・キュー，サーキュラ・キュー，ランダム・キューの 3 つの方式が提案されている．各方式に関して説明したのち，現在主流な方式であるエイジ論理付きのランダム・キューに関して説明する．

2.3.1 シフト・キュー

シフト・キューは，最も古くに提案され，商用プロセッサに使用された IQ の方式である [6]．シフト・キューでは，IQ の先頭のエントリより順に命令をディスパッチする．これにより，古い命令に高い発行優先度を与えることができる．[1]

また，シフト・キューでは命令を発行したエントリの空きを詰めるコンパクションを行うことにより，高い容量効率も達成することができる．正しい発行優先度と，高い容量効率を同時に達成するため，シフト・キューは IQ の方式の中で最も高い性能を得ることができる．

一方でシフト・キューには，コンパクションの回路が非常に複雑で，また消費電力が非常に大きいという欠点がある．そのため，シフト・キューはスケーリングが困難となっており，現在のプロセッサには使用されていない．

2.3.2 サーキュラー・キュー

サーキュラ・キューは，シフト・キューにおいて問題であったコンパクションを行わない方式である [7]．IQ は，ヘッド・ポインタとテール・ポインタを用いてサーキュラー・バッファとして管理される．

サーキュラー・キューでは，既に空いているが，命令をディスパッチできないエントリ

[1] 一般に，古い命令から優先的に発行すると，性能がより高くなることが知られている．

が発生し，IQ の容量効率がシフト・キューと比較して低下してしまう．また，ヘッド・ポインタとテール・ポインタの位置が逆転するラップ・アラウンドが生じた際には，新しい命令に高い優先度が与えられる優先度逆転が起き，選択論理が正しい優先度で命令を選択できない．これらの理由から，サーキュラー・キューはシフト・キューと比較して性能が低下する．

特に，容量効率が低下する影響は大きく，現在のプロセッサには使用されていない．

2.3.3 ランダム・キュー

2.4 IQ の問題点

第 3章　まとめ

本論文では，パイプライン構造の工夫によって命令キャッシュ・ミスによる性能低下を抑制する手法を提案した．本論文ではまず，MAP と呼ばれる独自のパイプライン構造を提案した．MAP は命令キャッシュ・ミスが発生しても性能が低下しない代わりに，分岐予測ミス・ペナルティが増加するという特徴を持つ．そして，本論文では，MAP と従来の構成を組み合わせ，それらを動的に使い分けることで性能向上を図るアーキテクチャと，このアーキテクチャの性能を最大化することができるパイプラインの切り替えアルゴリズムを提案した．提案手法の利点は，命令プリフェッチャと異なり，非常に小さなコストで構成できる上，無駄なメモリ・アクセスを全く行わないことである．

サーバー向けベンチマークで提案手法の評価を行ったところ，提案手法はプリフェッチなしのモデルと比較して最大 25.8%，平均で 13.0%の性能向上を達成し，最先端の命令プリフェッチャと比較して平均 4.8%の性能向上が得られることを確認した．また，提案手法を適用することで，性能へ大きく影響を与えることなく，命令キャッシュのサイズを小さくできることを確認した．

発表実績

- 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
- R. Matsuo, R. Shioya and H. Ando: ”Improving Instruction Fetch Throughput with Dynamic Control of Pipeline Structure”, MICRO-51 ACM Student Research Competition, Poster no.10, Fukuoka, Japan, Oct. 2018

受賞歴

- 情報処理学会システム・アーキテクチャ研究会 若手奨励賞, 2018 年 8 月

謝辞

本研究を進めるにあたり，多大なる御指導と御鞭撻を賜わりました名古屋大学大学院工学研究科 情報・通信工学専攻 安藤秀樹教授に心より感謝いたします．また，本研究の遂行を支えてくださいました，名古屋大学大学院工学研究科情報・通信工学専攻安藤研究室の諸氏に深く感謝します．

参 考 文 献

[1] N. H. E. Weste and D. M. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective, 4th edition.* Addition Wesley, 2010.

[2] F. Monsieur, E. Vincent, D. Roy, S. Bruyre, G. Pananakakis, and G. Ghibaudo, "Time to breakdown and voltage to breakdown modeling for ultra-thin oxides (Tox<32Å)," in *Proceedings of the 2001 IEEE International Integrated Reliability Workshop*, October 2001, pp. 20–25.

[3] S. Khan and S. Hamdioui, "Temperature dependence of NBTI induced delay," in *Proceedings of the 2010 IEEE 16th International On-Line Testing Symposium*, July 2010, pp. 15–20.

[4] J. Black, "Electromigration–a brief survey and some recent results," *IEEE Transactions on Electron Devices*, vol. ED-16, no. 4, pp. 338–347., April 1969.

[5] R. Viswanath, V. Wakharkar, A. Watwe, and V. Lebonheur, "Thermal performance challenges from silicon to systems," *Intel Technology Journal*, vol. 4, no. 3, pp. 1–16, August 2000.

[6] J. A. Farrell and T. C. Fischer, "Issue logic for a 600-mhz out-of-order execution microprocessor," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 5, pp. 707–712, 1998.

[7] J. Abella, R. Canal, and A. Gonzalez, "Power- and complexity-aware issue queue designs," *IEEE Micro*, vol. 23, Issue 5, no. 5, September-October 2003.