Vol.2014-ARC-213 No.5
Vol.2014-HPC-147 No.5
2014/12/9

タグの2段階比較を行う発行キューによる消費エネルギー削減の評価

小林 誠弥[1]　塩谷 亮太[1]　安藤 秀樹[1]

概要：スーパスカラ・プロセッサでは発行キューは非常に多くのエネルギーを消費する．発行キューの構成要素の中ではウェイクアップ論理の消費エネルギーが最も大きい．これはタグ比較に多くのエネルギーを消費するためである．これに対し我々はタグの 2 段階比較という手法を提案してきた．この手法では最初にタグの下位ビットのみを比較し，それが一致した場合に限って上位ビットを比較する．これにより比較するビット数を減らすことで消費エネルギーを削減する．この手法は発行キューの動作に 2 サイクルをかける必要があるため，これによる IPC 低下を抑制する手法として，先頭 1 段化方式と後方 1 段化方式という手法も合わせて提案してきた．これらの手法は，性能に悪影響を与えると推測される少数のエントリのタグ比較を従来通りの 1 段階で行うものである．これまで，提案手法の評価は性能とタグ比較の消費エネルギーに対する削減率のみにとどまっており，発行キュー全体，プロセッサコア全体の消費エネルギーに対する評価や既存手法との比較は行われていなかった．そこで本研究では，提案手法と既存手法における発行キュー全体およびプロセッサコア全体に対する消費エネルギー削減率を評価し，提案手法と既存手法の比較を行った．評価の結果，タグの 2 段階比較に先頭 1 段化方式と後方 1 段化方式を適用した場合，発行キューの消費エネルギーをそれぞれ 64%，67%削減し，プロセッサコア全体の消費エネルギーをそれぞれ 7.4%，8.0%削減できることがわかった．この時の性能低下はそれぞれ 0.79%と 0.47%で，提案手法は性能対消費エネルギーの面で既存手法より優れていることを確認した．

1. はじめに

ここ数年，プロセッサのシングルスレッド性能はあまり改善していない．この理由の一つに，性能を改善しようとすると通常は消費エネルギーが増加する一方で，その消費エネルギーは許容される限界に達しているという事情がある．そのため，プロセッサの消費エネルギー削減は今後の高性能化のために非常に重要な課題となっている．

また近年では，別の理由でも消費エネルギー削減が要求されるようになっている．それは，タブレットやスマートフォンといった携帯機器向けのプロセッサの需要が急速に高まっていることである．そのようなプロセッサは，バッテリ駆動時間を伸ばすために低消費エネルギーが強く求められる一方で，デスクトップ向けプロセッサと同等の高い性能も同時に要求される．そのため，性能を犠牲にせず消費エネルギーを削減できればその影響は極めて大きい．

現在広く用いられているスーパスカラ・プロセッサにおいて，発行キューは消費エネルギーが非常に大きい構成要素の一つである．図 1 にプロセッサ・コアと発行キューの電力内訳を示す（プロセッサの構成は表 1 の通り）．この図に示したように，発行キューの消費電力はコア全体の消費電力の 34%を占めている．発行キューの構成要素の中で，ウェイクアップ論理の消費電力が発行キュー全体に対して占める割合は 37%で最も大きく，これはコア全体に対して 13% [*1] に達する．

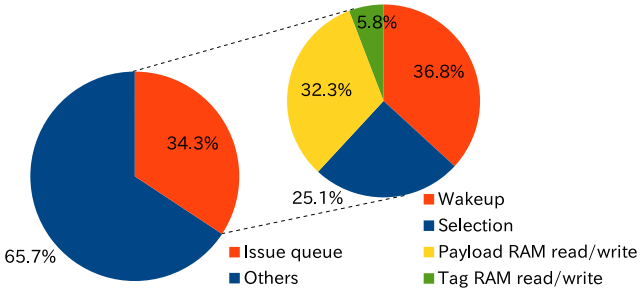


図 1　スーパスカラ・プロセッサのコアと発行キューの電力内訳

ウェイクアップ論理の消費電力が大きい理由は，タグ比較に大きなエネルギーを要するためである．一般に，ウェイクアップ論理はプロセッサのクリティカル・パスにあり，高速化が強く求められる．このため，ウェイクアップ論理

[1] 名古屋大学大学院工学研究科
Graduate School of Engineering, Nagoya University

[*1] 同様の数字が文献 [1] に示されており，こちらではウェイクアップ論理の消費電力はコア全体の 16%を占めるとされている．

の中で最も複雑なタグ比較器はダイナミック論理で構成され [2]，高速化が図られる．しかし，その代償としてエネルギー消費においては非常に不効率になっている．なぜならば，このタグ比較器は比較結果が不一致となった時にエネルギーを消費する構成になっているためである．一般に，ある実行結果を消費する命令はわずかであり，そのわずかな数の命令のタグについて比較結果が一致となる以外は，比較結果のほとんどが不一致となる．これによりタグ比較によって大きなエネルギーが消費される．

これに対し，我々は**タグの 2 段階比較**という手法を提案してきた [3]．この手法では，タグの下位ビットを先に比較し，それが一致した場合のみタグの上位ビットを比較することでタグ比較の消費エネルギーを抑える．一方で，この手法を単純に実装すると，直列な 2 回のタグ比較によって発行キューの遅延時間が増加し，クロック・サイクル時間が伸びる．これを回避するため，発行キューの動作を 2 サイクルにパイプライン化する．しかし，依存する命令を連続するサイクルで発行できなくなり，IPC が低下する．

この IPC 低下を抑えるため，**先頭 1 段化方式** [*2] と**後方 1 段化方式**と呼ぶ 2 つの拡張手法を合わせて提案してきた [3]．これらの手法では，性能に悪影響を与えると推測される少数のエントリのタグ比較を従来通りの 1 段階で行う．先頭 1 段化方式は，発行キュー先頭の一定数のエントリについてのみタグ比較を 1 段階で行う方式である．発行キューの先頭にある古い命令はデータフローのクリティカル・パスにある確率が高いと考えられるため，プログラムの実行時間の増加を抑制できる．後方 1 段化方式は，発行キューにおいて命令が発行されたエントリの後方一定数のエントリについてのみタグ比較を 1 段階で行う方式である．一般に，発行された命令の実行結果を消費する命令は，それらの命令がクリティカル・パスにあるならば，プログラム順で後方の近いところに存在する確率が高いと考えられる．そのため，この手法によってもプログラムの実行時間の増加を抑制できる．

これまで，提案手法の評価は性能とタグ比較の消費エネルギーに対する削減率のみにとどまっており，発行キュー全体，プロセッサコア全体の消費エネルギーに対する評価や既存手法との比較は行われていなかった．そこで本論文では，提案手法と既存手法における発行キュー全体およびプロセッサコア全体に対する消費エネルギー削減率を評価し，提案手法と既存手法の比較を行う．

評価の結果，タグの 2 段階比較に先頭 1 段化方式と後方 1 段化方式を適用した場合，発行キューの消費エネルギーをそれぞれ 64%，67%削減し，プロセッサコア全体の消費エネルギーをそれぞれ 7.4%，8.0%削減できることがわかった．この時の性能低下はそれぞれ 0.79%と 0.47%で，提案手法は性能対消費エネルギーの面で既存手法より優れていることを確認した．

[*2] 文献 [3] の 2 階層発行キュー方式と論理的には同一である．

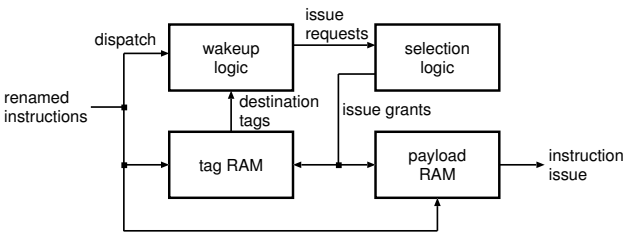


図 2 発行キューの構成

本論文の残りの部分は，以下の構成となっている．2 節では発行キューの構成と，ウェイクアップ論理およびタグ比較器の回路について説明する．3 節では提案手法であるタグの 2 段階比較方式および 2 つの拡張手法について説明する．4 節では評価の方法と結果について述べ，5 節でまとめを述べる．

2. 発行キューの構成とタグ比較器の動作

本節では，最初に発行キューの構成について説明する．その後，発行キューの中のウェイクアップ論理およびタグ比較器の回路と消費エネルギーについて説明する．

2.1 発行キューの構成

発行キューはリネームされて実行を待つ命令を保持し，その中から発行する命令を決定するユニットである．発行キューの構成として 2 つの方式が提案されており，それぞれ CAM 方式 [4] と RAM 方式 [5] である．本研究では CAM 方式を仮定する．RAM 方式については，4.3.2 節で消費エネルギーを比較する．

CAM 方式の発行キューは図 2 に示すように，ウェイクアップ論理，セレクト論理，タグ RAM，ペイロード RAM から構成される．一般に，ウェイクアップ論理は 1 次元のアレイで，各エントリには対応する命令の 2 つのソース・オペランドのタグ（以下，ソース・タグと呼ぶ）と，そのオペランドが利用可能かどうかを示すレディ・ビットを保持する．全てのオペランドがレディとなった場合，セレクト論理に発行要求を出す．セレクト論理では，発行要求の優先度や資源競合の有無を考慮し，発行許可信号を出す．発行許可信号はペイロード RAM に送られ，対応する命令のデータが読み出されて命令が発行される．発行許可信号は，対応する命令のディスティネーション・オペランドのタグ（以下，ディスティネーション・タグと呼ぶ）を保持するタグ RAM にも送られる．タグ RAM はアドレス・デコーダのない SRAM で構成され，発行許可信号はワード線に直接接続されている．発行許可信号によって発行される命令のディスティネーション・タグが読み出され，ウェイクアップ論理の全エントリに放送される．ウェイクアップ論理はエントリ毎に，放送されてきたディスティネーショ

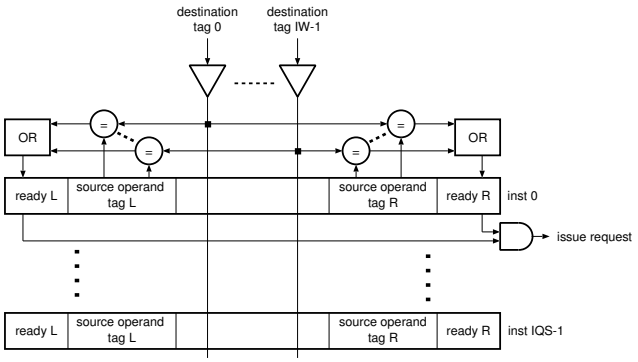


図 3　ウェイクアップ論理

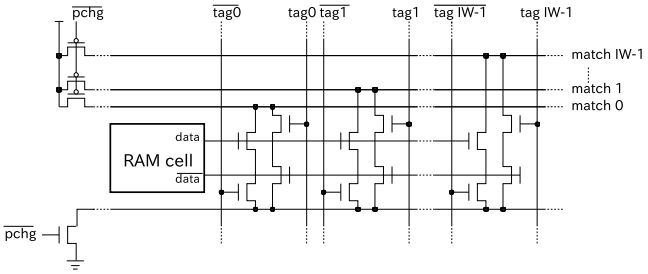


図 4　タグ比較を行う CAM セルの回路図

ン・タグと保持する命令のソース・タグの比較を行い，レディ・ビットを更新する．

発行キューのクリティカル・パスは，ウェイクアップ論理→セレクト論理→タグ RAM →ウェイクアップ論理でループを形成している．依存する命令を連続するサイクルで発行するためには，このループは 1 サイクルで完結する必要がある．そのため，このループはプロセッサのクリティカル・パスの 1 つであり，高速化が強く求められる．

発行キューの実装にはサーキュラ・バッファを仮定する．この実装では．ヘッドとテールの 2 つのポインタによってキューの論理的な先頭と終端を管理する（以降，特に断りがなければ発行キューの先頭と終端は論理的なものを指すとする）．この実装はエントリのシフトがそれらのポインタを動かすだけでよく，エントリの内容を実際に動かす必要がないためエネルギー効率が良い．

2.2　ウェイクアップ論理とタグ比較器

本節では，最初にウェイクアップ論理とその中のタグ比較器の回路について説明する．その後，タグ比較器の消費エネルギーについて議論する．

2.2.1　回路

ウェイクアップ論理の構成を図 3 に示す．同図中の IW は発行幅，IQS は発行キューのサイズを表す．タグ RAM から読み出された IW 個のディスティネーション・タグは，タグ・ドライバによってウェイクアップ論理の全 IQS エントリに放送される．各エントリは 2 つのソース・タグを持ち，それらと放送されてくるディスティネーション・タグとを比較する．比較が一致すればレディ・フラグがセットされる．両方のレディ・フラグがセットされると，発行要求が出力される．

図 4 にタグ比較を行う CAM セルの回路を示す．ウェイクアップ論理の 1 つのエントリはこの CAM セルをタグ・ビット幅だけ横に並べたものである．図の左にある SRAM セルはソース・タグの 1 ビットを保持する．右端に match と書かれた水平の線は，比較結果の論理値を出力するマッチ線である．マッチ線の左端にあるトランジスタは，マッチ線のプリチャージ・トランジスタである．マッチ線の下に直列に接続されたトランジスタは，タグの比較結果に応じてマッチ線をディスチャージするプルダウン・トランジスタである．

この回路は次のように動作する．まず，マッチ線をプリチャージし，それと同時にタグ・ドライバがディスティネーション・タグを放送する．プリチャージの後に比較が行われる．ディスティネーション・タグとソース・タグに 1 ビットでも不一致があれば，直列接続されたプルダウン・トランジスタがマッチ線をプルダウンして L が出力される．全てのビットが一致すればマッチ線はプルダウンされず，H が出力される．

2.2.2　タグ比較器の消費エネルギー

2.2.1 節で説明したウェイクアップ論理の回路から直感的にわかるように，タグ比較によって大きなエネルギーが消費される．本節では，この消費エネルギーが比較するタグのビット数に比例することを説明する．提案手法のタグの 2 段階比較方式では，ウェイクアップ論理で比較されるタグのビット数を削減することによって消費エネルギーを削減する．

一般に，CMOS デバイスが動作すると以下の式で表される動的エネルギーが消費される [6].

$$E = C_L V_{DD}^2 \tag{1}$$

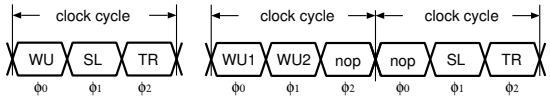
ここで，C_L と V_{DD} はそれぞれ，デバイスの静電容量と供給電圧を表す．

この式から，タグ比較器の動的な消費エネルギーは以下のようになる．

$$E_{cmp}(n_{tagbits}) = \{ (C_{pdown} + C_{mline}) \times n_{tagbits} + C_{pchg} \} V_{DD}^2 \tag{2}$$

ここで，C_{pdown} は 1 ビットあたりのプルダウン・トランジスタの接合容量，C_{mline} は 1 ビットあたりのマッチ線の配線容量，C_{pchg} はプリチャージ・トランジスタの接合容量，$n_{tagbits}$ はタグのビット幅を表す．

これに加えて，プルダウン・トランジスタのリーク電流によってサイクル毎に以下の式で表される静的エネルギーが消費される．



(a) 従来の場合　　(b) タグの 2 段階比較の場合

図 5　発行キューにおける動作のタイミング

$$E_{leak}(n_{tagbits}) = I_{leak} \times V_{DD} \times k \cdot T \times n_{tagbits} \quad (3)$$

ここで，I_{leak} は 1 ビットあたりのプルダウン・トランジスタのリーク電流，T はクロック・サイクル時間，k はマッチ線がチャージされ，かつ 2 つのプルダウン・トランジスタのうち少なくとも 1 つがオフになっている期間のクロック・サイクル時間に対する割合を表す．

注意すべきは，動的エネルギーは比較結果が不一致であった時にだけ，プリチャージされたマッチ線がプルダウンされて消費されることである．比較が一致すれば，マッチ線のプリチャージ状態が維持されて動的エネルギーは消費されない．この性質が発行キューの大きな消費エネルギーをもたらしている．一般に，ある命令の結果を消費する命令の数は少ない（多くの場合 1 つだけ）．そのため，結果が一致となる比較器はわずかで，その他多くの比較結果は不一致となり大きなエネルギーが消費される．

式 (2) から，E_{cmp} はおおよそタグのビット幅に比例することがわかる．なぜなら，$(C_{pdown} + C_{mline}) \times n_{tagbits} \gg C_{pchg}$ だからである．同様に，式 (3) から E_{leak} もタグのビット幅に比例することがわかる．そのため，提案手法のタグの 2 段階比較方式では比較するタグ・ビットの総数を削減することで発行キューの消費エネルギーを削減する．

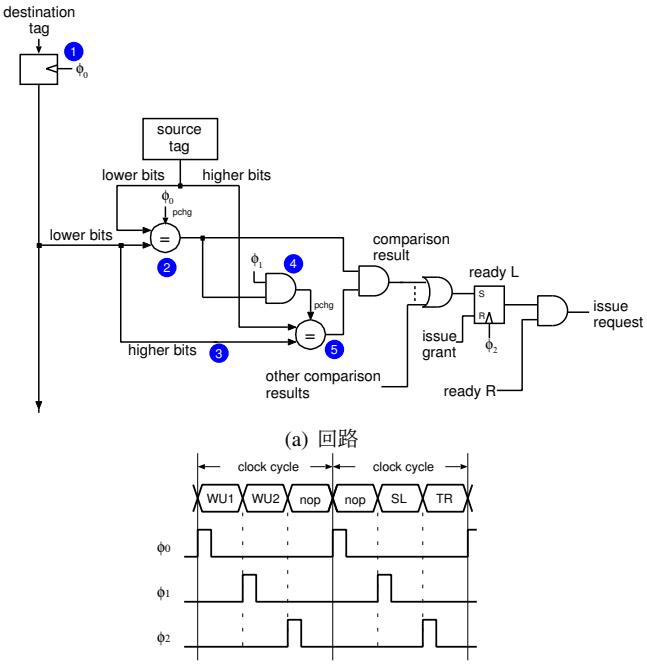
ウェイクアップ論理は多数のタグ比較器を持つ（4 節で評価するプロセッサでは 2048 個）ため，その消費エネルギーは大きい．これはプロセッサ・コア全体の消費エネルギーに対して大きな割合を占める（図 1 で示したように，ウェイクアップ論理はプロセッサ・コア全体の 13%の消費エネルギーである）．

3. タグの 2 段階比較方式

最初にタグの 2 段階比較の基本方式における回路と動作タイミングについて説明する．次に，基本方式で生じる IPC の低下を抑制する 2 つの拡張手法について説明する．

3.1 基本方式の回路と動作タイミング

発行キューの消費エネルギーを削減するため，**タグの 2 段階比較**という手法を提案してきた．この手法では，タグの下位ビットのみを先に比較し，それが一致した場合のみタグの上位ビットの比較を行う．回路面から見ると，下位ビットの比較器は必ずプリチャージされ動作するが，上位ビットの比較器は下位ビットの比較が一致した場合のみプリチャージされる．このため，以下の理由により消費エネルギーが削減される．



(a) 回路

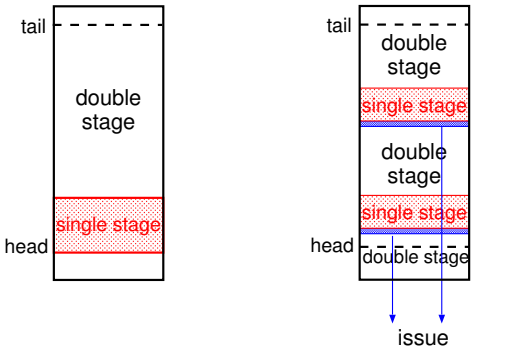
(b) 動作タイミング

図 6　タグの 2 段階比較を行うウェイクアップ論理の回路

- 必ず動作する比較器のビット幅は狭められる．式 (2) と (3) から，比較器の動的および静的な消費エネルギーはどちらもタグのビット幅におおよそ比例する．
- 上位ビット比較器は，下位ビットが一致した時のみ動作する．下位ビットが不一致だった場合，上位ビット比較器はプリチャージされないため，エネルギーは動的にも静的にも消費されない．

クロック・サイクル時間が増加するのを防ぐため，提案手法では発行キューの動作を 2 サイクルとする．図 5(a) に従来の発行キューの動作タイミングを示し，図 5(b) にタグの 2 段階比較方式での動作タイミングを示す．従来の発行キューでは，図 5(a) に示すように，ウェイクアップ (WU)，セレクト (SL)，タグ RAM 読み出し (TR) が順に 1 サイクルで行われる．それぞれの動作は，ϕ_0 から ϕ_2 の 3 つのフェーズで行われ，それぞれの遅延はおおよそ等しい [7]．一方で，タグの 2 段階比較においては，図 5(b) のようにウェイクアップ（と第 3 フェーズの NOP）に 1 サイクルをかけ，次のサイクルで（第 1 フェーズの NOP の後）セレクトとタグ RAM 読み出しを行う．最初のサイクルでは，ϕ_0(WU1) でディスティネーション・タグを放送し，下位ビットの比較を行う．下位ビットが一致していれば，上位ビット比較器は ϕ_1 の最初にプリチャージされる．この場合のみ，ϕ_1(WU2) の残りの時間で上位ビットの比較が行われる．そして，上位ビットと下位ビットの両方が一致していれば，ϕ_2 の最初でレディ・ビットがセットされる．次のサイクルで，セレクトとタグ RAM 読み出しが従来と同様に行われる．

図 6 にタグの 2 段階比較方式におけるウェイクアップ



(a) 先頭 1 段化方式　　(b) 後方 1 段化方式

図 7　IPC 低下を抑える手法

論理の回路を示す．この回路は次のように動作する．タグ RAM から読み出されたディスティネーション・タグは ϕ_0 でラッチされ（①），ウェイクアップ論理の全エントリに放送される．各エントリでは，ϕ_0 が H になっている間に下位ビット比較器がプリチャージされる．ϕ_0 が L になると，ディスティネーション・タグとソース・タグの下位ビットが比較される（②）．このフェーズでは，ディスティネーションの上位ビットは単に上位ビット比較器に送られるだけである（③）．下位ビットの比較結果は，上位ビット比較器のプリチャージ信号 ϕ_1 をゲーティングする（④）．つまり，下位ビットが一致した時だけ上位ビット比較器がプリチャージされる．ϕ_1 が L になると，上位ビットが比較される（⑤）．そして，下位ビットと上位ビットの比較結果の AND をとり，ϕ_2 でレディ・ビットを更新する．

図 6 の回路の面積オーバヘッドは，2 つの AND ゲート（上位ビットのプリチャージをゲーティングするものと，2 つの比較器の結果を AND するもの）だけである．また，回路の遅延がクロック・サイクル時間に悪影響を与えないことは明らかである．

3.2　IPC 低下を抑制する手法

3.1 節で述べたように，タグの 2 段階比較では発行キューの動作を 2 サイクルかけて行う．そのため，依存する命令を連続するサイクルに発行できなくなり，IPC の低下が生じる．この IPC 低下を抑えるため，基本方式を拡張する 2 つの手法を提案する．これらの手法では，IPC に悪影響を与える可能性の高い少数のエントリでのみ，従来通りに 1 段階のタグ比較を行う（以降，これを 1 段化と呼ぶ）．つまり，1 段化したエントリでは発行キューの動作は 1 サイクルで行われる．1 段化するエントリが少数であっても，4.2 節で評価するように IPC 低下は劇的に改善される．提案する 2 つの手法の概要は以下の通りである．

(1)　**先頭 1 段化方式**：図 7(a) のように，発行キューの先頭の一定数のエントリのみを 1 段化する．

(2)　**後方 1 段化方式**：図 7(b) のように，命令が発行されたエントリの後方一定数のエントリのみを，発行直後だけ 1 段化する．

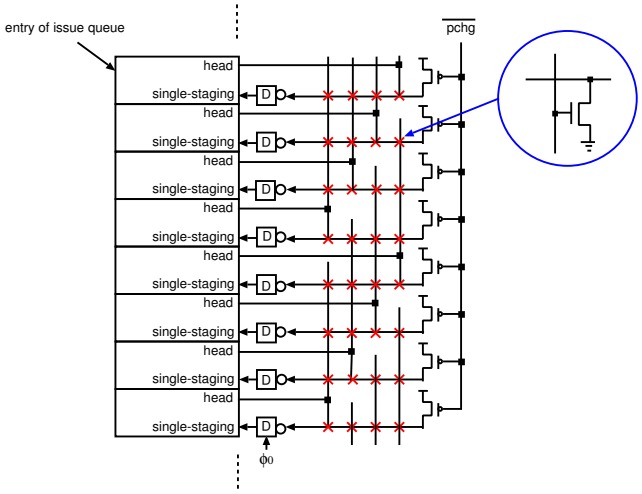


図 8　1 段階比較を指示する信号の生成回路 ($N_{head} = 4$)

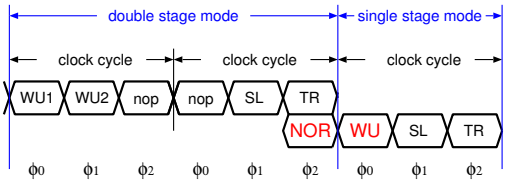


図 9　2 段階比較モードから 1 段階比較モードへの切り替え

以下，この 2 つの手法について詳しく説明する．

3.2.1　先頭 1 段化方式

ある命令がデータフローにおけるクリティカル・パス上にある時，その命令の依存チェーンには先行する多くの命令があるはずである．したがって，そのような命令は発行キューの中に長時間残り，発行キューのヘッド・ポインタはその命令に近づいていく．そのため，その命令が発行される時には先頭エントリの近くにある可能性が高い．このヒューリスティックを IPC 低下を抑えるために利用する．

この方式は図 7(a) に示したように，発行キューの先頭の一定数のエントリを 1 段化する．1 段化したエントリではエネルギー削減が得られないが，他の多くのエントリでタグの 2 段階比較を行うことで大きなエネルギー削減を得る．

この方式では，タグ比較が 2 段階と 1 段階の 2 つのモードに切り替えられるようにタグ比較器の回路を変更する必要がある．これは図 6 の回路にわずかな変更を加え，上位ビット比較器とレディ・ラッチの動作タイミングを切り替えられるようにするだけでよい．具体的には，以下の 2 つの変更を加える．

(1)　上位ビット比較器のプリチャージ信号入力にマルチプレクサを挿入し，1 段階モードの時に ϕ_0 でプリチャージできるようにする．

(2)　レディ・ラッチのクロック入力にマルチプレクサを挿

入し，1 段階モードの時に ϕ_1 でレディ・フラグを更新できるようにする．

この 2 つのマルチプレクサによる影響は，2 段階比較回路の遅延をわずかに増加させるだけである．

ウェイクアップ論理の変更に加えて，タグ比較のモード切り替えを示す信号を生成する回路も追加する必要がある．あるエントリに対するモード切り替え信号は，自身のエントリを含めて前方 N_{head} エントリのヘッド・ポインタの信号の OR をとることで生成できる．ここで，N_{head} は 1 段化するエントリの数を表す．その回路を図 8 に示す．この図では例として $N_{head} = 4$ としている．図の右側でトランジスタが行列になっている部分はダイナミック NOR 回路の集合（1 行が 1 つの NOR を表す）である．NOR の入力はヘッド・ポインタがそのエントリを指すかどうかの信号である．この回路は図 9 に示すように，タグ RAM 読み出しフェーズ (TR) と並列して動作する．つまり，モード切り替えはタグ RAM 読み出しによって隠されるため，この NOR 回路はクロック・サイクル時間に影響しない．この NOR 回路は追加のエネルギーを消費するが，これは 4.3.1 節で評価するように無視できる大きさである．

発行キューはサーキュラ・バッファを用いた実装を仮定しているため，ラップ・アラウンドが生じる．これは，キューの物理的な末尾近くのエントリに対応するヘッド・ポインタ信号の配線を，物理的な先頭まで伸ばす必要があることを意味する．しかし，そのような長い配線は NOR 回路の遅延を大きく増加させる．そのため，そのような配線は削除し，ラップ・アラウンドした状況での一部の 1 段化を諦めることにする．この妥協による性能低下は無視できるほど小さい（評価では 0.05%）．

この方式による 1 段化を実装するための面積オーバヘッドは非常に小さい．追加回路（2 つのマルチプレクサ，1 つの NOR 回路，NOR 出力のための 1 つのラッチ）のトランジスタ数は $N_{head} = 16$（4.2 節の評価による最適値）のとき 1 エントリあたり 37 個である．これは 4 節の仮定（発行幅とタグのビット幅はそれぞれ 8 と 9）における従来のウェイクアップ論理の 3.5%の量にすぎない．

3.2.2 後方 1 段化方式

多くの場合，あるデータのプロデューサとそのコンシューマはプログラム順で近い位置にある．それらがクリティカル・パス上にあれば，その間の距離は特に短い．そのため，この方式では図 7(b) に示したように，命令が発行されたエントリの後方一定数のエントリのみ，命令の発行直後だけタグ比較器の回路を 2 段階モードから 1 段階モードに切り替える．あるプロデューサが発行された時，そのコンシューマのタグ比較がうまく 1 段階モードに切り替えられていれば，コンシューマは次のサイクルに発行できる．1 段階モードに切り替えられたエントリは，そのエントリの前方一定数のエントリで命令が発行されなければ，次のサイクルに 2 段階モードに戻る．

この方式は先頭 1 段化方式と同様に，2 つのモードを切り替えるためのウェイクアップ論理の変更と，モード切り替え信号を生成する NOR 回路の追加が必要である．ただしこの方式では，あるエントリの NOR 回路の入力は前方 N_{rear} エントリの選択論理からの発行許可信号である．ここで N_{rear} は 1 段化するエントリ数である．この NOR 回路の動作タイミングは図 9 に示したものと同じである．ゆえに NOR 動作はタグ RAM 読み出しに隠され，クロック・サイクル時間には影響しない．

後方 1 段化方式の先頭 1 段化方式に対する利点は，データフローにおけるクリティカル・パス上にある命令がプログラムの実行状況によって発行キュー内に分散していても，この方式では適応的にそれらの命令をカバーできることである．一方，先頭 1 段化方式で性能低下を効果的に抑えるには，そのような命令は先頭の一定数のエントリに集まっていなければならない．

3.2.1 節で説明したラップ・アラウンドにより生じる問題を考慮して，NOR 回路は先頭 1 段化方式と同様にキューの物理的な末尾から先頭へ伸びる配線を削除する．このため，やはりラップ・アラウンドした状況での一部の 1 段化を諦めることになる．これにより性能低下が生じるが，先頭 1 段化方式の場合と同様に無視できるほど小さい（評価では 0.05%）．

4. 評価

4.1 節では評価環境について説明する．4.2 節では各提案手法の性能を評価し，1 段化エントリ数の最適化を行う．4.3 節では消費エネルギーについて評価し，RAM 方式の発行キューと提案手法を比較する．最後に，4.4 節で提案手法と電圧/周波数スケーリングのエネルギー対性能比を比較する．

4.1 評価環境

性能の評価には，SimpleScalar/MASE [8] をベースにしたシミュレータを用いた．命令セットは Alpha ISA を使用した．ベンチマーク・プログラムとして，SPECint2006 の全 12 本，SPECfp2006 から wrf を除く 16 本を用いた[*3]．プログラムは，gcc ver. 4.5.3，コンパイル・オプション -O3 でコンパイルした．シミュレーションはプログラムの入力に ref データ・セットを用い，SimPoint 3.2 [9] で選んだ 100M 命令区間で行った．

消費エネルギーの評価には，McPAT [10] を修正および拡張したシミュレータを用いた．LSI テクノロジは 22nm を仮定した．提案手法の評価では，電力ではなく消費エネル

[*3] wrf については，現在のところ正しくシミュレータが動作しない．

表 1　プロセッサの構成

Pipeline width	4-instruction wide for each of fetch, decode and commit, 8-instruction wide for issue
Reorder buffer	256 entries
Issue queue	128 entries
Load/store queue	64 entries
Physical registers	Int and fp 168 registers each
Tag bit width	9 bits
Branch prediction	16-bit history 16K-entry PHT gshare, 1K-set, 4-way BTB 10-cycle misprediction penalty
Function unit	6 iALU, 2 iMULT/DIV, 2 Ld/St, 2 fpALU, 2 fpMULT/DIV/SQRT
L1 I-cache	32KB, 2-way, 32B line
L1 D-cache	32KB, 2-way, 32B line, 2 ports, 2-cycle hit latency, non-blocking
L2 cache	Unified, 2MB, 4-way, 64B line, 12-cycle hit latency
Main memory	300-cycle min. latency, 8B/cycle bandwidth
Data prefetcher	Stride-based, 4K-entry, 4-way table, 16-data prefetch to L2 cache on miss

ギーを指標として用いる．提案手法では IPC 低下によって電力が削減されるが，この効果は提案手法の利点から生じるものではない．そのため，評価指標としては消費エネルギーの方が適している．

評価においてベースとなるプロセッサの構成を表 1 に示す．パイプラインの幅および物理レジスタ数は Intel Haswell [11] を基にしている．物理レジスタの総数は 336 であるため，タグのビット幅は 9 ビットとなっている．

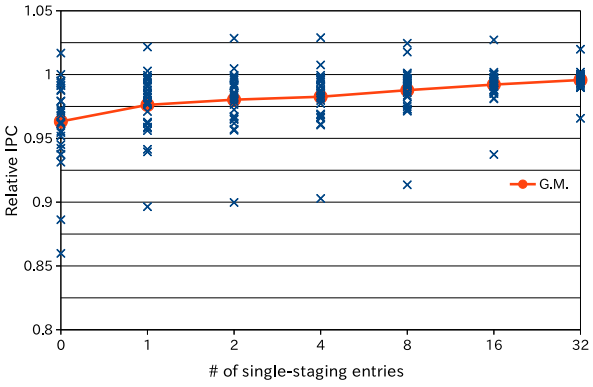
以降の評価では，以下に示すモデル名を使用する．

- **DSC-noext**：単純なタグの 2 段階比較方式のモデル (IPC 低下を抑制する措置は講じない)
- **DSC-head**：タグの 2 段階比較方式に先頭 1 段化方式を加えたモデル
- **DSC-rear**：タグの 2 段階比較方式に後方 1 段化方式を加えたモデル

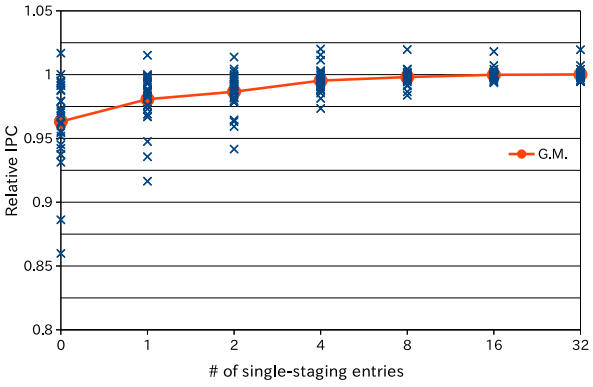
4.2　1 段化エントリ数に対する性能

図 10 に，(a)DSC-head モデル，(b)DSC-rear モデルにおいてタグ比較を 1 段化するエントリ数を変えた時の，ベースのプロセッサに対する相対 IPC を示す．縦軸は相対 IPC で，横軸は 1 段化エントリ数である．×で表された点は個々のベンチマークに対応し，赤い線は全ベンチマークの幾何平均を表す．横軸左端の 0 は DSC-noext モデルを表す．

図 10 に示したように，DSC-noext では大きな IPC 低下（平均 3.7%，最大 14.0%）が生じる．一方で拡張手法と組み合わせると，1 段化エントリ数を増やすにつれて性能は大きく改善される．同じ 1 段化エントリ数（すなわち，1 段化



(a) DSC-head



(b) DSC-rear

図 10　1 段化エントリ数を変化させた時の相対 IPC

回路の規模が同じ）では，DSC-rear の方が DSC-head よりもわずかに優れている．性能とエネルギー消費にはトレードオフがあるが，本論文の評価では性能を優先する．そこで，許容する IPC 低下を全ベンチマーク平均で 1%以下とする．そのため，最適な 1 段化エントリ数は IPC 低下が平均 1%以下となる最小のエントリ数となる．図 10 の結果から，最適な 1 段化エントリ数は DSC-head で 16，DSC-rear で 4 と決定した．この時，DSC-head と DSC-rear の平均の性能低下はそれぞれ 0.8%と 0.5%で，最大の性能低下はそれぞれ 6.3%と 2.7%である．

図 10 において，一部のベンチマーク（特に *lbm*）で相対 IPC がわずかに 1.0 を超えている．これは提案手法による命令の発行タイミングの変化が，分岐予測やキャッシュの挙動にプラスに働いたためである．

4.3　消費エネルギー

4.3.1 節では提案手法におけるウェイクアップ論理の消費エネルギーについて評価し，4.3.2 節ではプロセッサ・コア全体の消費エネルギーについて評価する．どちらの節でも，評価結果を RAM 方式の発行キューと比較する．

4.3.1　ウェイクアップ論理の消費エネルギー

図 11 に，提案手法の 3 つのモデルにおいてタグの下位ビット比較器のビット幅を変えた時の，ベースに対するウェイクアップ論理の平均のエネルギー削減率を示す．縦

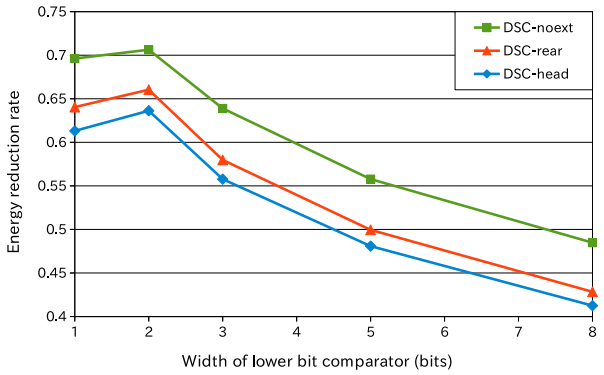


図 11 DSC-noext モデルにおいて比較器の幅を変化させた時のウェイクアップのエネルギー削減率

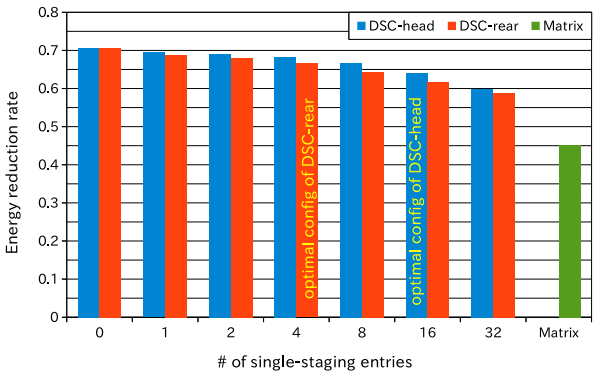


図 12 1 段化エントリ数を変えた時のウェイクアップのエネルギー削減率

軸はエネルギー削減率であり，点が上にあるほど削減率が大きいことを表す．横軸はタグの 2 段階比較における 1 段目の下位ビット比較器のビット幅である．DSC-head と DSC-rear の 1 段化エントリ数はそれぞれ 16 と 4 である（4.2 節で決定した最適値）．ここで評価するウェイクアップ論理のエネルギーには，ウェイクアップ論理全体の動的および静的な消費エネルギーが含まれている（後述の図 12 についても同様）．タグのビット幅は 9 ビットであるため，下位ビット比較器の幅は 1 から 8 まで変化させている．

この図に示したように，常に動作する下位ビット比較器の幅が小さいほど，概してエネルギー削減は大きくなる傾向がある．そして，比較器の下位ビットと上位ビットの幅がそれぞれ 2 ビットと 7 ビットの時に削減率が最大になっている．いずれのモデルにおいても，エネルギー削減は 2 段階比較を行うエントリによってなされるため，最適な比較器のビット幅は 3 つのモデルで同一となる．以降の評価ではこの設定を使用する．

図 12 に，DSC-head，DSC-rear モデルにおいて 1 段化エントリ数を変えた時のウェイクアップ論理の平均のエネルギー削減率を示す．縦軸はエネルギー削減率で，横軸の数字は 1 段化エントリ数である．横軸左端の 0 は DSC-noext モデルを表す．DSC-head，DSC-rear の評価では，追加の

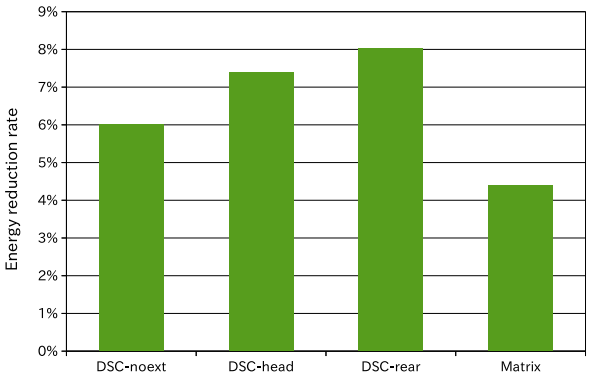


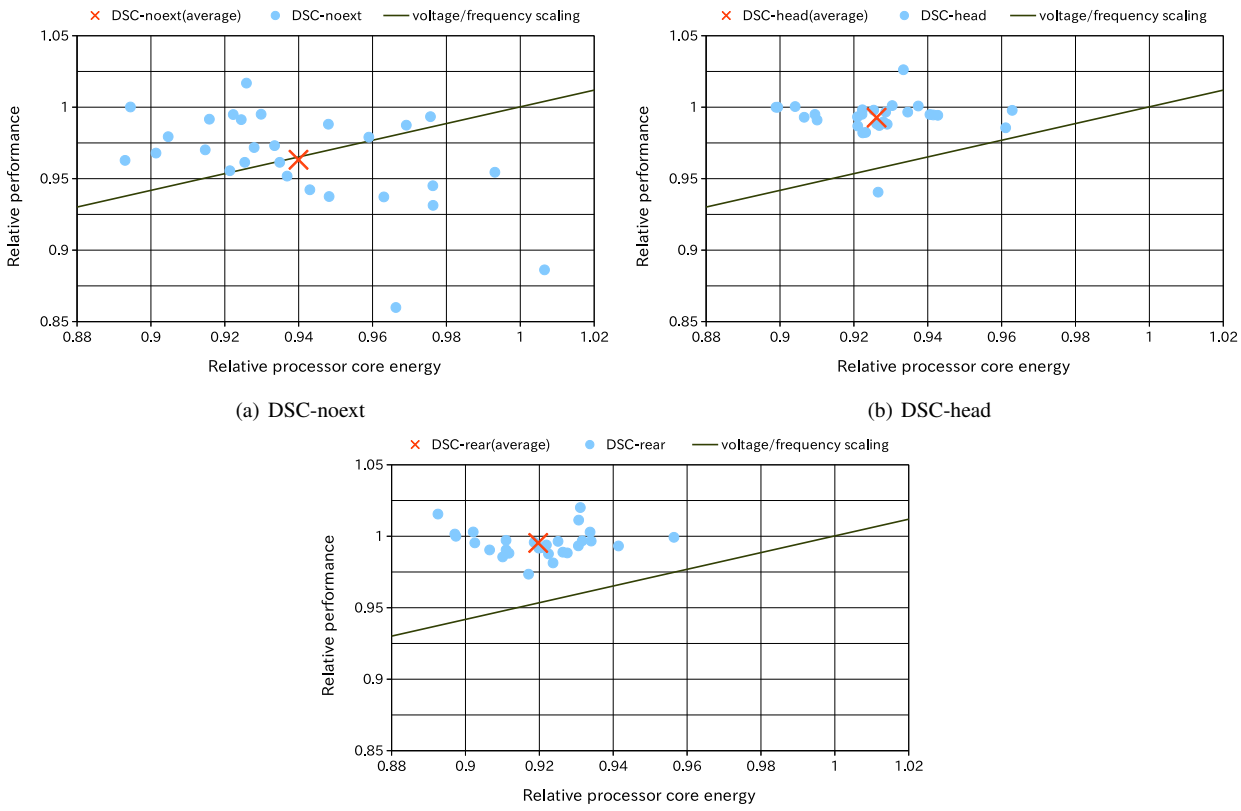
図 13 各手法におけるプロセッサ・コア全体のエネルギー削減率

NOR 回路の消費エネルギーも考慮している．

図からわかるように，1 段化エントリ数を増やすとエネルギー削減率はやはり低下する．しかし，4.2 節で決定した最適なパラメータにおいては，どちらのモデルでも削減率の低下はあまり大きくない．最適なパラメータにおける DSC-head と DSC-rear モデルのエネルギー削減率はそれぞれ 64%と 67%で，DSC-noext に対してそれぞれ 91%と 94%を達成している．このような大きな削減は，不要なタグビットの比較を DSC-head と DSC-rear のそれぞれで 68%と 72%削減したことによる．追加の NOR 回路の消費エネルギーは無視できるほど小さく，ウェイクアップ論理の消費エネルギーに対して DSC-head で 0.19%，DSC-rear で 0.05%である．

図 12 には，マトリクス・スケジューラ [5]（横軸右端の「Matrix」ラベル）の平均のエネルギー削減率も示してある．マトリクス・スケジューラはウェイクアップ論理を CAM ではなく RAM で構成する手法で，最もエネルギー効率の良いウェイクアップ論理を構成する手法の 1 つと考えられている．マトリクス・スケジューラにおいて RAM はビット行列を表し，各ビットは発行キューのエントリ間のデータ依存を表している．ウェイクアップ動作はこの RAM を読み出すだけでよく，タグ比較器が不要になるため，ウェイクアップ論理の消費エネルギーを大きく削減することができる．表 1 の構成をマトリクス・スケジューラで実現する場合，ウェイクアップ行列は 1 つの読み出しポートと 4 つの書き込みポートを持つ 2KB の RAM が 2 つで構成される（各 RAM は各ソース・オペランドに対応する）．

図に示したように，マトリクス・スケジューラも大きなエネルギー削減を達成しているが，提案手法には及ばないレベルである．マトリクス・スケジューラによる削減率は 45%で，DSC-head と DSC-rear のそれぞれの削減率はその 1.42 倍と 1.49 倍である．また，提案手法はこのような大きな削減率をわずかな性能低下 (それぞれ 0.8%と 0.5%) で実現している．提案手法では比較したタグのビット数を約 70%削減しており，これは動作する比較器の数を 30%にまで削減することに等しい．そのため，提案手法を用いた



(a) DSC-noext

(b) DSC-head

(c) DSC-rear

図 14 提案手法と電圧/周波数スケーリングにおける性能と消費エネルギーの関係

CAM 方式はエネルギー削減の面で RAM 方式を上回る結果となった．

4.3.2 プロセッサ・コア全体の消費エネルギー

本節では，提案手法を用いたプロセッサ・コア全体の消費エネルギー削減について評価する．図 13 は，ベースのプロセッサを基準とした，プロセッサ・コア全体の消費エネルギー削減率を示している．図のように，DSC-head と DSC-rear はコアの消費エネルギーをそれぞれ 7.4%と 8.0%削減している．この結果は，DSC-noext の削減率と比べて約 30%大きく，一見して図 12 の結果と矛盾している．このようになる理由は，DSC-head と DSC-rear は DSC-noext に比べて性能が向上しており，それにより静的な消費エネルギーが減少したためである．

一方，マトリクス・スケジューラの削減率は 4.4%にとどまり，提案手法よりも小さい．この結果は，図 12 に示したようにウェイクアップ論理のエネルギー削減率が提案手法よりも小さいことによる．

4.4 電圧/周波数スケーリングとの比較

電圧/周波数スケーリングは，あるプロセッサの動作電圧とクロック周波数を，対象とする市場における電力および性能の要求に合わせて出荷時に変更する手法である．この手法は，性能を犠牲にしてプロセッサの消費エネルギーを削減する最も単純な手法である．そのため，power-aware な手法は電圧/周波数スケーリングよりも優れたエネルギー対性能を達成することが望ましい [12]．

図 14 に (a)DSC-noext，(b)DSC-head，(c)DSC-rear モデルの，全ベンチマークにおける性能とプロセッサ・コア全体の消費エネルギーの関係を示す．丸い点が各ベンチマークに対応し，X の点が全ベンチマークの平均である．縦軸が性能，横軸が消費エネルギーで，どちらの軸の値もベースのプロセッサの値で正規化してある．この図では，点が左上にあるほどエネルギー対性能が良いことを表す．図中の黒線は，電圧/周波数スケーリングの平均のエネルギー対性能を性能および電力のシミュレーションによって得たものである．評価結果によれば，電圧/周波数スケーリングは 1%の性能低下あたり 1.7%のエネルギー削減を達成した．

図 14 の (a) からわかるように，DSC-noext モデルでは多くの点および平均が電圧/周波数スケーリングの線よりも下にプロットされている．これは DSC-noext が power-aware な手法としては劣っていることを表す．一方で DSC-head と DSC-rear においては，平均を含めてほぼ全ての点が電圧/周波数スケーリングの線よりも上にプロットされている．これらのモデルでは，平均において 1%以下の性能低下で約 8%のエネルギー削減を達成し，電圧/周波数スケーリングの結果を大きく上回っている．ここから，タグの 2 段階比較に拡張手法を組み合わせたものは power-aware な手法として優れていることがわかる．

5. まとめ

発行キューはプロセッサ・コアの消費エネルギーの多くを占める．発行キューの中でも，ウェイクアップ論理は最も大きなエネルギーを消費する．ウェイクアップ論理の中で，タグ比較器はエネルギー的に非常に不効率な回路である．これに対し，タグの 2 段階比較という発行キューの消費エネルギーを削減する手法と，それによる IPC 低下を抑えるために一部の命令のタグ比較を 1 段階で行うようにする，先頭 1 段化方式および後方 1 段化方式を提案してきた．本論文では，提案手法と既存手法における発行キュー全体およびプロセッサコア全体に対する消費エネルギー削減率を評価し，提案手法と既存手法の比較を行った．評価の結果，タグの 2 段階比較に 2 階層発行キュー方式と後方 1 段化方式を適用した場合，発行キューの消費エネルギーをそれぞれ 64%，67%削減し，プロセッサコア全体の消費エネルギーをそれぞれ 7.4%，8.0%削減できることがわかった．この時の性能低下はそれぞれ 0.79%と 0.47%で，提案手法は性能対消費エネルギーの面で既存手法より優れていることを確認した．

謝辞 本研究の一部は，日本学術振興会 科学研究費補助金 基盤研究 (C)（課題番号 25330057），および日本学術振興会 科学研究費補助金 若手研究 (A)（課題番号 24680005）による補助のもとで行われた．

参考文献

[1] Folegnani, D. and González, A.: Energy-effective issue logic, *Proceedings of the 28th Annual International Symposium on Computer Architecture*, pp. 230–239 (2001).
[2] Palacharla, S., Jouppi, N. P. and Smith, J. E.: Quantifying the complexity of superscalar processors, Technical Report CS-TR-1996-1328, University Wisconsin (1996).
[3] 小林誠弥，塩谷亮太，安藤秀樹：タグの 2 段階比較による発行キューの消費エネルギー削減，2013 年先進的計算基盤システムシンポジウム，pp. 2–9 (2014 年 5 月).
[4] Palacharla, S., Jouppi, N. P. and Smith, J. E.: Complexity-effective superscalar processors, *Proceedings of the 24th Annual International Symposium on Computer Architecture*, pp. 206–218 (1997).
[5] Goshima, M., Nishino, K., Kitamura, T., Nakashima, Y., Tomita, S. and Mori, S.: A high-speed dynamic instruction scheduling scheme for superscalar processors, *Proceedings of the 34th Annual International Symposium on Microarchitecture*, pp. 225–236 (2001).
[6] Weste, N. H. E. and Harris, D. M.: *CMOS VLSI Design: A Circuits and Systems Perspective*, Addison-Wesley Publishing Company, USA, 4th edition (2011).
[7] Yamaguchi, K., Kora, Y. and Ando, H.: Evaluation of issue queue delay: Banking tag RAM and identifying correct critical path, *Proceedings of the 29th International Conference on Computer Design*, pp. 313–319 (2011).
[8] Larson, E., Chatterjee, S. and Austin, T.: MASE: A novel infrastructure for detailed microarchitectural modeling, *Proceedings of the 2001 International Symposium on Performance Analysis of Systems and Software*, pp. 1–9 (2001).
[9] Hamerly, G., Perelman, E., Lau, J. and Calder, B.: SimPoint 3.0: Faster and more flexible program analysis, *Journal of Instruction-Level Parallelism*, Vol. 7, pp. 1–28 (2005).
[10] Li, S., Ahn, J. H., Strong, R. D., Brockman, J. B., Tullsen, D. M. and Jouppi, N. P.: McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures, *Proceedings of the 42nd Annual International Symposium on Microarchitecture*, pp. 469–480 (2009).
[11] Krewell, K.: Intel's Haswell cuts core power, *Microprocessor Report* (2012).
[12] Gochman, S., Ronen, R., Anati, I., Berkovits, A., Kurts, T., Naveh, A., Saeed, A., Sperber, Z. and Valentine, R. C.: The Intel® Pentium® M processor: Microarchitecture and performance, *Intel Technology Journal*, Vol. 7, No. 2, pp. 21–59 (2003).