C-022

メモリアクセス命令の特徴を利用したセットアソシアティブキャッシュの低電力アクセス手法

A Low-power Access Method for Set-associative Cache Using Characteristics of Memory Access Instructions

会田康男 † **岡部翔** † **矢崎俊志** † **阿部公輝** †
Yasuo Aita Sho Okabe Syunji Yazaki Kôki Abe

1. はじめに

近年，コンピュータシステム，特に携帯電話やノート PC などにおいて小型化や消費電力の削減が求められている．その中でキャッシュメモリにおける消費電力は，マイクロプロセッサ全体における消費電力の中でも大きな割合を占めている．本研究では，キャッシュにおける消費電力の削減に着目する．スーパースカラープロセッサでは複数の命令を 1 サイクルあたりに発行するが，命令同士の依存関係によっては同時に発行できないことがある．サイクル毎に同時に発行される命令数を IPC(Instructions Per Cycle) という．4 命令同時発行スーパースカラープロセッサでは，IPC は 1 から 4 となる．

本研究では，セットアソシアティブ方式のキャッシュメモリのアクセスにおいて，メモリアクセス命令の特徴を利用する．特徴としてそれらの命令が発行されたときの IPC を用いる．各メモリアクセス命令の特徴に応じ，キャッシュのウェイの適切なサブセットにアクセスする．特徴付けが適切であれば，小さいサブセットへのアクセスのヒット率を上げることができ，その結果として消費電力を削減できる．本稿では，提案手法と従来手法について消費電力とプログラム実行速度の比較評価を行う．

2. 関連研究

IPC に着目したセットアソシアティブキャッシュにおける消費電力の削減手法が S. Nadathur と A.Tyagi により提案されている [1]．彼らは，ロード・ストア命令を IPC の値に着目して分類し，IPC で分類された命令は同じウェイでヒットする傾向にあることを利用し，ロード・ストア命令の IPC に応じて，ウェイ集合のサブセットに順にアクセスする．この手法では，エネルギーの削減は期待できるが，それに対する速度の低下が大きい [1].

A.Veidenbaum と D.Nicolaescu は，次にアクセスすべきウェイを予測し，タグへのアクセスをスキップする手法を提案した [2]．本研究では，IPC という特徴でメモリアクセス命令を分類し，同じ特徴を持つメモリアクセス命令の過去の振る舞いから，次にアクセスすべきウェイを予測する．

3. 提案手法

本研究では，最も最近ヒットした命令の IPC をヒットしたウェイに記録する．次のロード・ストア命令は，IPC が同じウェイに優先的にアクセスする．

†電気通信大学, The University of Electro-Communications

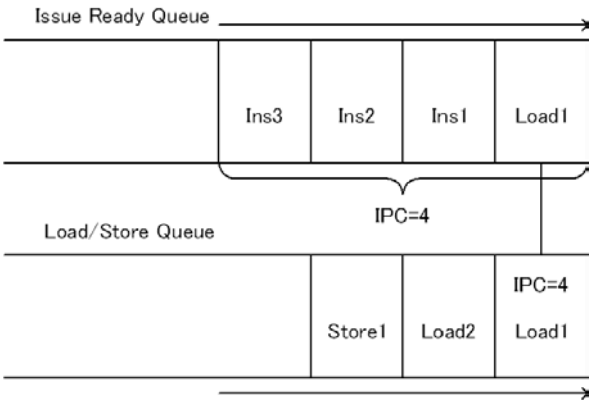

図 1: ロード・ストア命令への IPC 情報の付加

図 1 において，発行可能な命令を置くキューを Issue Ready Queue(IRQ)，デコード済みであるが発行ができないロード・ストア命令を置くキューを Load Store Queue(LSQ) と呼ぶ．本稿では，各ロードストア命令が発行される時点に IRQ にある命令の数をそのロード・ストア命令の IPC とする．IRQ の大きさを N とする．IPC は 1 から N の値をとる．

[1] では特徴に対応する固定したサブセットへアクセスする．適切なサブセットの区切り方は時間と共に変化すると考えられる．ある時間帯では，少ないサブセットで足りていた IPC のクラスの命令が別の時間帯ではより大きいサブセットを必要とするかもしれない．特徴ごとにサブセットを固定しては，キャッシュメモリを有効に利用できないと考えられる．

本提案手法では，図 2 に示すように，最も最近ヒットした命令の IPC をヒットしたウェイに記録する．次のロード・ストア命令は，IPC が同じウェイに優先的にアクセスする．ミスしたときは残りのウェイにアクセスする．このように，本手法では特徴に応じて動的にウェイを選択し，優先的にアクセスする．特徴の選び方が適切であれば，同じ特徴を持つロード・ストア命令は同じ振る舞いをする，すなわち同じウェイでヒットしやすいと考えられる．

4. シミュレーション実験と結果の評価

キャッシュメモリに文献 [1] と提案手法の 2 種類について，プログラムの実行速度と消費電力を計測した．また，対比のために通常のセットアソシアティブ方式での実行速度と消費電力も計測した．

本研究では，スーパースカラープロセッサのシミュレーターとして Simplescalar3.0[3] を用いた．シミュ

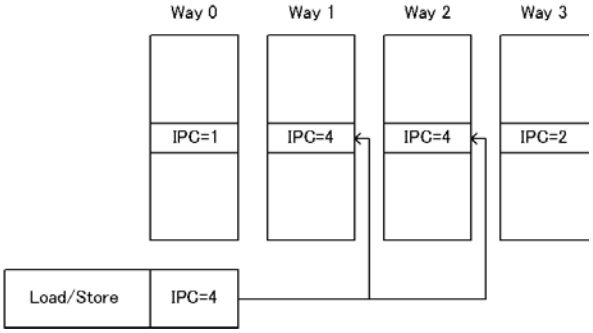

図 2: 最も最近ヒットした命令の IPC をヒットしたウェイに記録する．次のロード・ストア命令は，IPC が同じウェイに優先的にアクセスする．

レーション環境は，次の通りである．L1 データキャッシュはメモリ容量 32KB，4 ウェイ，アクセスレイテンシ 3 サイクル，L1 命令キャッシュはメモリ容量 64KB，1 ウェイ，アクセスレイテンシ 3 サイクル，L2 キャッシュはメモリ容量 256KB，4 ウェイ，アクセスレイテンシ 6 サイクル，メモリアクセスレイテンシは 60 サイクル，フェッチ・デコード・実行命令数を 4，整数・浮動小数点 ALU を 2 個，整数・浮動小数点乗除算器を 4 個として実験を行った．この際，spec2000 ベンチマークの 164.gzip，176.mcf，186.crafty，197.parser，254.gap，256.bzip2 の 6 つのベンチマークプログラムにおいて 5 億回命令を実行した．IRQ の大きさを $N = 4$ とした．

消費電力の計測は，キャッシュメモリの消費電力のシミュレータ Cacti3.2[4] を用いて行った．シミュレーション環境はプロセスルール 100nm，メモリ容量 32KB，ブロックサイズ 32bit，4 ウェイとした．

計測結果を図 3 と図 4 に示す．図 3 の縦軸はプログラム実行速度を実行命令総数 / サイクル数 =IPC で表し，図 4 の縦軸は消費エネルギーを mJ で表す．両図の横軸は各ベンチマークプログラムとそれらの平均を表す．

図 3 から，本手法のプログラム実行速度は通常のセットアソシアティブ方式と殆ど変わらないことが分かる．図 4 から，本手法の消費エネルギーは通常のセットアソシアティブ方式と比べ，約 6.4 %低い．これに対し，文献 [1] の手法では，消費エネルギーの削減率は提案手法より大きいものの，プログラム実行速度の低下は著しい．

5. まとめと今後の課題

セットアソシアティブキャッシュにおいて，最も最近ヒットした命令の IPC をヒットしたブロックに記録しておき，次のロード・ストア命令は，IPC が同じウェイに優先的にアクセスする手法を提案した．シミュレーション実験により，本手法は実行速度をほとんど変えず，消費エネルギーを削減できることが分かった．ウェイに IPC を記録することによるオーバーヘッドの検討については今後の課題である．

参考文献

[1] S. Nadathur and A. Tyagi, “IPC Driven Dynamic Associative Cache Architecture for Low energy”, International Conference on Computer Design (ICCD ’04), pp.472-479, 2004.

[2] A.Veidenbaum and D.Nicolaescu, “Low Energy, Highly-Associative Cache Designfor Embedded Processors”, Proc. IEEE ICCD, pp. 332-335, 2004.

[3] Simplescalar3.0
http://www.simplescalar.com/

[4] CACTI
http://www.hpl.hp.com/research/cacti/

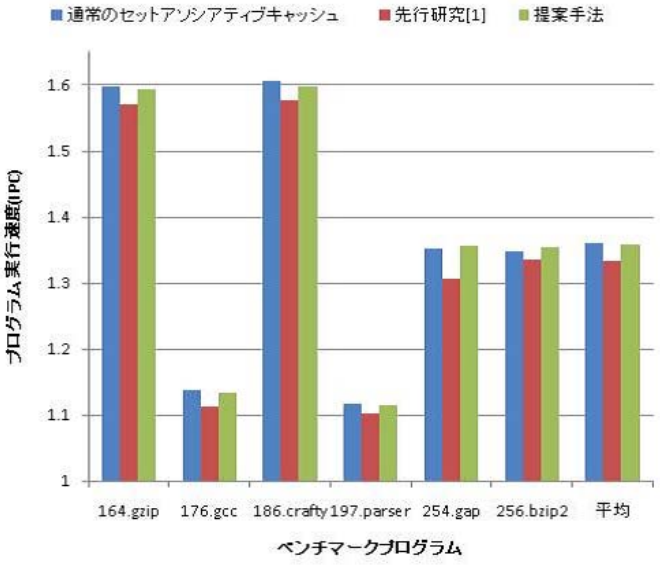

図 3: プログラム実行速度

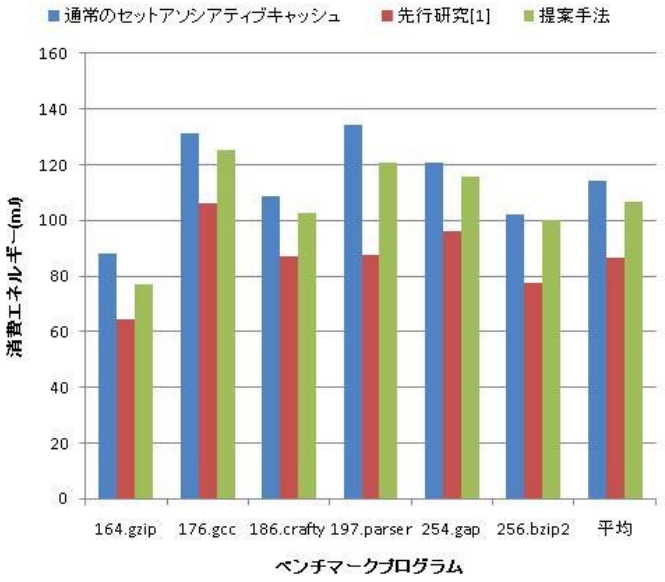

図 4: プログラム実行時の消費電力