

## ヒストリ・ベース・タグ比較キャッシュの設計と評価

田中, 秀和  
福岡大学大学院工学研究科電子工学専攻

井上, 弘士  
福岡大学工学部電子情報工学科

モシニヤガ, ワシリー  
福岡大学工学部電子情報工学科

村上, 和彰  
九州大学大学院システム情報科学研究院

<https://hdl.handle.net/2324/6115>

---

出版情報：第66回情報処理学会全国大会講演論文集（1），pp.83-84，2004-03. 情報処理学会  
バージョン：

権利関係：ここに掲載した著作物の利用に関する注意 本著作物の著作権は（社）情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。

# ヒストリ・ベース・タグ比較キャッシュの設計と評価 Design and Evaluation of A History-Based Tag-Comparison Cache

田中秀和<sup>†</sup> 井上弘士<sup>‡</sup> モシニヤガ・ワシリー<sup>‡</sup> 村上和彰<sup>††</sup>

福岡大学大学院工学研究科電子工学専攻<sup>†</sup> 福岡大学工学部電子情報工学科<sup>‡</sup>  
九州大学大学院システム情報科学研究院<sup>††</sup>

Hidekazu TANAKA<sup>†</sup> Koji INOUE<sup>‡</sup> Vasily MOSHNYAGA<sup>‡</sup> Kazuaki MURAKAMI<sup>††</sup>  
Dept. of Electronics Engineering, Fukuoka University<sup>†</sup>  
Dept. of Electronics Engineering and Computer Science, Fukuoka University<sup>‡</sup>  
Dept. of Informatics, Kyushu University<sup>††</sup>

## 1 はじめに

現在のプロセッサ・チップにはプロセッサ主記憶間の性能差を隠蔽するために、当然のようにキャッシュ・メモリが搭載されている。そして、プロセッサの性能向上を背景に、更なるキャッシュ・ヒット率の向上を目的としてキャッシュ・サイズは年々増加傾向にある。その結果、キャッシュ・アクセスにおける消費エネルギーが増大し、チップ全体の消費エネルギーに大きな影響を与えるようになってきた。特に命令キャッシュはアクセス頻度が高いため、その消費エネルギー増大が与える影響は非常に大きい。例えば、ARM920T プロセッサではチップ全体の消費電力の約 25%が命令キャッシュで消費される[3]。

そこで我々は、命令キャッシュの低消費エネルギー化を目的として、ヒストリ・ベース・タグ比較 (History Based Tag Comparison: HBTC) キャッシュを提案した[1]。従来型セット・アソシアティブ・キャッシュでは、参照データが1つのウェイのみに存在するにも関わらず、アクセス時には全てのウェイを活性化するため多くの無駄なエネルギーを消費する。これに対し、HBTC キャッシュでは、過去のタグ比較結果を再利用して参照データが存在するウェイのみを活性化するため、性能に影響を与えることなく低消費エネルギー化を実現できる。しかしながら、文献[1]では、0.8  $\mu\text{m}$  CMOS テクノロジーを前提とし、文献[2]で提案された消費エネルギー・モデルを用いて評価を行っているため、先端技術を用いた場合の有効性は未だ不明確である。そこで本稿では、HBTC キャッシュの有効性をより詳細に評価するため、0.18  $\mu\text{m}$  CMOS プロセスによる SRAM アレイの実設計に基づく消費エネルギー評価を行う。

## 2 HBTC キャッシュ

### 2.1 従来型キャッシュ

多くの高性能 CPU では高ヒット率を達成するため、セット・アソシアティブ方式 (以下、SA と呼ぶ) を採用している。一般に、連想度が  $N$  の SA キャッシュは  $N$  個のウェイで構成されており、また、各ウェイは1つのタグ・サブアレイとそれに対応するライン・サブアレイから成る。CPU からのメモリ参照が発生した時、従来型キャッシュでは、1つのウェイのみに参照データが存在するにも関わらず、高速アクセスを実現するために全てのウェイが同時に活性化される。具体的には、各ウェイからタグとラインの組が読出され、参照アドレスに基づきタグ比較を行う。もし、一致しているタグが存在すればキャッシュ・ヒットとなり、このタグに対応するライン

が CPU に供給される。ここで、説明を簡単にするため、キャッシュ・アクセス当りの消費エネルギー ( $E_{cache}$ ) を以下の式で近似する。

$$E_{cache} = T_{num} \times E_{tag} + L_{num} \times E_{line} \quad (1)$$

上式において、 $E_{tag}$  および  $E_{line}$  は、タグ 1 個およびライン 1 個の読出しに要する平均消費エネルギーを表す。また、 $T_{num}$  と  $L_{num}$  は、それぞれ、アクセス時に読出されるタグおよびラインの総数である。連想度が 4 の場合、従来型キャッシュでは、 $T_{num}=4$ 、 $L_{num}=4$  となる。

### 2.2 タグ比較結果の再利用による低消費電力化

HBTC キャッシュは過去のタグ比較結果を再利用することにより、参照データが存在するウェイのみを活性化し、性能に影響を与えることなく低消費エネルギー化を実現する。ここで、タグ比較結果を記憶しておくためのメモリ領域として BTB (Branch Target Buffer) を利用する。通常、BTB の各エントリには分岐命令アドレスとその分岐先アドレスが格納される。プログラム実行中、条件が成立した分岐命令は BTB に登録されていく。このような従来の BTB エントリに対し、HBTC キャッシュでは、図 1 に示すように以下のハードウェアを分岐成立用/不成立用それぞれに追加する。

- ウェイ・ポインタ (WP) : 分岐後続命令列に関するタグ比較結果であり、分岐後続命令が存在するウェイ番号を保持する。エントリ当りに  $N_{wp} (\geq 2)$  個の WP を実装することにより、連続する  $N_{wp}$  ライン分 (通常、1 個のラインには複数命令が格納される) のタグ比較結果を記録できる。なお、命令キャッシュの連想度を  $W$  とすると、1 個の WP は  $\lceil \log W \rceil$  ビットで実装できる。
- 有効フラグ : 対応する WP が有効であるか否かを示す有効フラグ。

そして、プログラム実行中、HBTC キャッシュは以下のように動作する。

1. ある分岐命令  $B_i$  の実行時に BTB ヒットが発生した際、 $B_i$  のアドレスを PBA レジスタに保存する。その後、分岐後続命令に関するキャッシュ・アクセスを行う際、タグ比較結果を順次 WP 記録レジスタに保存していく。この時、HBTC キャッシュは従来型キャッシュと同様に全てのウェイを活性化する。
2. 次分岐命令  $B_j$  の実行により BTB ヒットが発生した際、PBA レジスタ (分岐命令  $B_i$  のアドレス) を用いて WP 記録レジスタの値を BTB に書込む。また、対応する有効フラグをアクティブにする。
3. 分岐命令  $B_i$  が再度実行される際、BTB アクセスと同

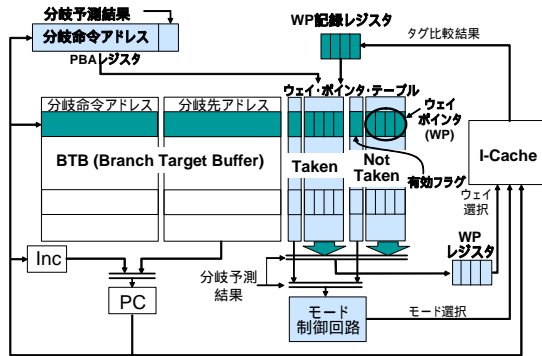


図 1: HBTC キャッシュの内部構成

時に記録した WP ならびに有効フラグを讀出す。もし、有効フラグがアクティブであれば、讀出した WP を WP レジスタに格納する。以降、命令キャッシュは WP レジスタから供給されるタグ比較結果を用いてアクセスを遂行する(参照ウェイのみが活性化される)。一方、有効な WP が存在しない場合には上記 1 の処理を行う。ただし、キャッシュ・ミスが発生した場合には、記録されたタグ比較結果に対応するデータがキャッシュから追いついてしまう可能性がある。そのため、キャッシュ・ミス発生時には BTB に記録した全てのタグ比較結果を無効にする。また、制御を簡単にするため、BTB エントリの追出し発生時にもタグ比較結果を無効にする。

HBTC キャッシュでは、タグ比較結果を再利用する場合、ラインを 1 個讀出すだけでアクセスを完了できるため大幅な消費エネルギーの削減を期待できる。しかしながら、分岐予測のために毎クロック・サイクル BTB アクセスが発生する。よって、タグ比較結果(WP)へのアクセスも毎クロック・サイクル発生し、BTB の消費エネルギーを増大させるといった欠点がある。

### 3 評価

HBTC キャッシュの有効性をより詳細に評価するため、0.18  $\mu\text{m}$  CMOS テクノロジー(電源電圧 1.8V)を用いて SRAM アレイを設計した。命令キャッシュに関して、16KB の 4 ウェイ SA キャッシュを想定して 1 ウェイ分のデータメモリを設計し、負荷容量の抽出ならびに回路シミュレーションを行った。一方、BTB に関しては、連想度 4 の 512 エントリ構成を前提とし、エントリ当りに分岐成立/不成立用それぞれ 4 個のウェイ・ポイント(WP)が実装されると仮定した(文献[1]の実験結果より)。また、BTB に保存された全ての WP を無効にするため、WP 有効フラグ用強制リセット機能付メモリセルを設計した。設計結果を表 1 に示す。ここで、 $E_{tag}$ 、 $E_{line}$  ならびに  $E_{dec}$  は、それぞれ、タグ 1 個の讀出し、ライン 1 個の讀出し、ならびに、アドレス・デコードにおける消費エネルギーである。また、BTB 拡張部分に関して、 $E_{wprdb}$ 、 $E_{wprwt}$  ならびに  $E_{wpinv}$  は、それぞれ、1 回の WP 讀出し、1 回の WP 書込み、ならびに、1 回の WP 無効化処理に要するエネルギーを表している。なお、本評価では周辺回路による消費エネルギーは考慮していない。また、メモリセルで消費されるエネルギー( $E_{tag}$ 、 $E_{line}$ 、 $E_{wprdb}$ 、 $E_{wprwt}$ )に関しては、プリチャージ動作を含むビット・アクセス当りの平均消費エネルギーを測定し、アクセス・ビット数との積をとって求めた。

表 1 に示した設計結果に基づき、詳細な消費エネルギー

表 1: 設計結果 (各動作当りの消費エネルギー)

命令キャッシュ (pJ)		BTB 拡張部分 (WP) (pJ)	
$E_{dec}$	82.2	$E_{wprwt}$	75.8
$E_{tag}$	63.6	$E_{wprdb}$	229.1
$E_{line}$	814.7	$E_{wpinv}$	160.0

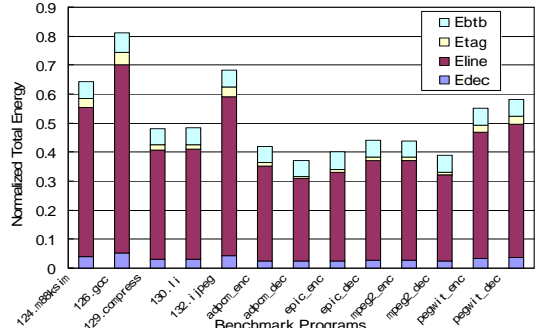


図 2: HBTC キャッシュの総消費エネルギー

一評価を行うために SimpleScalar[4]を用いたサイクル・レベル・シミュレーションを行った。

各ベンチマークにおける HBTC キャッシュの総消費エネルギーを図 2 に示す。全ての結果は、従来型キャッシュでの消費エネルギーで正規化している。ここで、 $E_{btb}$  は BTB に追加した WP を操作するために発生する消費エネルギーであり、 $E_{wprdb}$ 、 $E_{wprwt}$  ならびに  $E_{wpinv}$  の合計である。図 2 より、 $E_{btb}$  のオーバ・ヘッドはキャッシュ全体の消費エネルギーに対し比較的大きな割合を占めており、削減効果を下げていることが分かる。しかしながら、 $E_{btb}$  のオーバ・ヘッドに対し、 $E_{line}$  および  $E_{tag}$  の削減効果が非常に大きいため、従来型キャッシュと比較した場合、HBTC キャッシュは最大で約 63% ( $adpcm_{dec}$ ) の低消費エネルギー化を達成している。

### 4 おわりに

本稿では、実設計に基づく HBTC キャッシュの消費エネルギー評価を行った。その結果、従来方式と比較して、最大で約 63% の低消費エネルギー化を達成した。今後は、本稿で考慮しなかった周辺回路の設計を行い、HBTC キャッシュのより詳細な評価を行う。

### 謝辞

本研究は東京大学大規模集積システム設計教育研究センターを通し、シノプシス株式会社ならびにケイデンス株式会社の協力で行われたものである。なお、本研究は一部、文部省科学研究費補助金(課題番号: 14GS0218, 14702064, 14580399)による。

### 参考文献

- [1] K. Inoue, V.G.Moshnyaga, and K. Murakami, "A Low Energy Set-Associative I-Cache with Extended BTB," *Proc. Of the 2002 IEEE International Conference on Computer Design*, pp.187-192, Sep. 2002.
- [2] M. B. Kamble, and K. Ghose, "Analytical Energy Dissipation Models For Low Power Caches," *Proc. Of the 1997 Int. Symp. On Low Power Electronics and Design*, pp.143-148, Aug. 1997.
- [3] S. Segars, "Low Power Design Techniques for Microprocessors," *ISSCC Tutorial*, Feb. 2001.
- [4] "SimpleScalar Simulation Tools for Microprocessor and System Evaluation," URL: <http://www.simplescalar.org/>.