

FIFO 置換方式による CAM ベース 世代分割高連想度キャッシュ

岡部 翔^{†1} 阿部 公輝^{†1}

CAM ベース高連想度キャッシュは、高性能な組み込みシステムに広く用いられている。本論文では、CAM ベース高連想度キャッシュにおいて、消費電力を抑えるために、SRAM と CAM を併用し、SRAM を利用頻度に応じてサブバンクごとに 2 つの領域（第 1 世代 SRAM と第 2 世代 SRAM）に分ける。さらに、第 2 世代 SRAM の置換方式を FIFO とし、性能を維持しつつ回路コストや速度コストの面で改良する。シミュレーション実験により提案手法は従来法と比較し、速度性能の低下を抑制しつつ高い電力削減効果が得られることがわかった。

A CAM-based Highly Associative Cache Partitioned into Generations with FIFO Replacement Policy

SHO OKABE^{†1} and KÔKI ABE^{†1}

Content Addressable Memory (CAM)-based Highly-Associative Caches (HAC) are widely used in high performance embedded microprocessors. In this paper, we partition SRAM in a subbank into first- and second-generations according to access frequency. The method is further improved by employing FIFO replacement policy for the second-generation SRAM, with respect to feasibility and area costs without decreasing cache access performance. Simulation experiments revealed that the proposed method can realize significant power reduction without increasing access delays compared with traditional CAM-based HAC.

1. はじめに

Content Addressable Memory (CAM) は、入力に対し高速並列比較を行うことができ、ネットワークルーターやワイヤレスセンサネットワークに用いられる¹⁾²⁾ ほか、Translation lookaside buffer (TLB) や System on a Chip (SoC) などにも用いられる³⁾⁴⁾。CAM ベースの高連想度キャッシュ (Highly Associative Cache, HAC) は、高性能な組み込みシステムに有用である⁵⁾⁶⁾。CAM ベース HAC は、キャッシュメモリを複数のサブバンクに分割し、アクセスあたりの電力を下げる。各サブバンクは高い連想度を持つことによりミス率を下げる。CAM ベース HAC の問題は、CAM タグがキャッシュアクセスあたりの電力消費の大きな割合を占めることである。CAM は同じサイズの SRAM の 5~10 倍の電力を消費する⁷⁾。

本論文では、はじめに、CAM ベース高連想度キャッシュにおいて、消費電力を抑えるために、SRAM と CAM を併用し、SRAM を利用頻度に応じてサブバンクごとに 2 つの領域、第 1 世代 SRAM と第 2 世代 SRAM、に分ける手法（提案手法 (1)）について概略を述べる。この手法では、SRAM において複数ウェイで同時比較を行い、第 1 世代、第 2 世代共に LRU 置換方式とする。次に、この手法を分析し、回路や速度の面で改良する手法（提案手法 (2)）を述べる。提案手法 (2) では、SRAM において逐次比較を行い、置換は、第 1 世代 SRAM は LRU 方式、第 2 世代 SRAM は FIFO 方式とする。

以下、第 2 章で従来の高連想度キャッシュと関連研究を述べた後、第 3 章で提案手法について説明する。第 4 章で実験方法と実験結果、評価結果を述べ、最後に第 5 章でまとめる。

2. 高連想度キャッシュと関連研究

同容量のセット・アソシアティブ方式のキャッシュと比較し、高連想度キャッシュは、キャッシュの配置に柔軟性があるため、ミス率の改善が期待できる。しかし、高連想度キャッシュはタグ比較の際に、各ウェイにおいて、全ての CAM を動作させねばならない。つまり、1 回のキャッシュアクセスにおいて、連想度分の CAM を動作させる必要があり、電力消費が大きいという問題がある。

CAM を複数のサブバンクに分割し、1 回のキャッシュアクセスに対して、1 つのサブバンクだけをアクティブにして、電力削減を図る。図 1 にラインサイズ 32bytes、容量 8KB の 32 ウェイ高連想度キャッシュの構成例を示す。図では 8 つのサブバンク（各 1KB）に分割している。アクティブな 1 つのサブバンクにおいて、全ウェイに対して並列にタグ比較を行い、マッチした SRAM データを出力する。サブバンクに分けることで消費電力は抑えら

^{†1} 電気通信大学 情報工学専攻
Department of Computer Science, The University of Electro-Communications

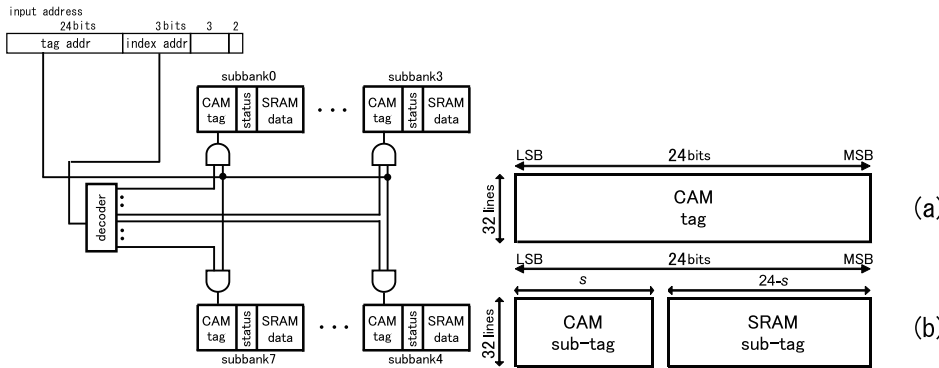


図1 キャッシュ容量 8KB で 8 つのサブバンクを持つ高連想度キャッシュ (各サブバンクはフルアソシアティブ方式のキャッシュ)

Fig.1 Traditional highly-associative cache with eight subbanks each of a fully associative cache.

れるが、サブバンク内での電力削減は行っておらず、十分とは言えない。

性能を低下させずに電力消費を削減する手段として、いくつかの方法が知られている。A.Veidenbaum と D.Nicolaescu は、次にアクセスすべきウェイを予測し、CAM タグへのアクセスをスキップする手法を提案した⁸⁾。最も最近アクセスがあったウェイが次もアクセスされるといふ単純な予測でも 86% 当たる。A.Ma や M.Zhang らは、最も最近アクセスがあったウェイをカウンターに記録し、次に同じウェイへのアクセスがあったとき、タグ検索をしない手法を提案した⁹⁾。この手法により 21% の電力削減ができる。

C.Zhang は、タグを上位ビットと下位ビットの 2 つのサブタグに分ける手法 (Low Power Highly Associative Cache, LPHAC 法) を提案した¹⁰⁾。従来の高連想度キャッシュでは、図 2(a) のように、全て CAM タグで構成されているのに対し、LPHAC 法では、図 2(b) のように、タグの下位 s ビットのサブタグを CAM で、上位ビット (この例では $24-s$ ビット) のサブタグを SRAM で構成する。はじめに CAM のサブタグ比較を行い、ヒットするラインがあれば、SRAM のサブタグ比較とデータ読み出しを行う。

LPHAC 法では、CAM のサブタグ比較では高々 1ヶ所しか一致しないようにしている。これは、SRAM のサブタグ比較とデータ読み出しを 1 ラインに限定するためである。そのため、CAM のサブタグでヒットし SRAM のサブタグでミスした場合、置換ポリシーが無

視されることから、ミス率が増加する。 $s=8$ のとき、このケースでのキャッシュミスが約 15% 発生する¹⁰⁾。そのため、 s を小さく設定することができない。このように LPHAC 法では、CAM 部分の縮小に限界があり、電力削減は十分でない。

3. 提案手法

岡部, 阿部は、LPHAC 法に対し、SRAM のサブタグにおいて複数ラインとの比較を許し、かつ世代分割を用いる手法を提案した¹¹⁾¹²⁾。ここでは、その手法を提案手法 (1) と呼び、その概略を述べる。そして、提案手法 (1) の分析を行い、問題点を明らかにする。さらに、提案手法 (2) で手法 (1) の問題点を解決する。

3.1 提案手法 (1)

提案手法 (1) について説明する。LPHAC 法と同様、図 2(b) のようにタグを上位ビットと下位ビットに分けるが、LPHAC 法とは異なり、CAM のサブタグとの部分マッチングにおいて、ヒットするキャッシュラインを 1 つに絞らない (複数ラインとの一致を許す)。さらに、SRAM 部分をアクセスされる可能性の高いウェイトとそうでないウェイトに分割する。これにより、ヒット率の低下を抑制しつつ、CAM 部分をさらに縮小する。

ウェイ数を w とする。図 2(b) において、SRAM のサブタグのウェイのうち、最近アクセスがあった r 個 ($r \leq w$) のウェイに目印をつける。図 2(b) の例では、 $w = 32$ である。ここで、 r 個のウェイを第 1 世代ウェイ、その他のウェイを第 2 世代ウェイと呼ぶ。

動作の説明をする。まず CAM のサブタグとの部分マッチングを行う。ヒットした (一般には複数の) キャッシュラインについて、SRAM サブタグの第 1 世代ウェイでマッチングとデータ出力を同時に行う。一致するウェイがあればキャッシュヒットとしてデータを出力する。一致するウェイがなければ、出力されたデータは破棄する。第 1 世代ウェイでヒットしなかった場合のみ、第 2 世代ウェイの探索を行う。

$r = 4$ の時の動作例を図 3 に示す。CAM 部分のサブタグを 4 ビットとし、メモリアドレスのタグ部の下位 4 ビット (tag addr の lower) が 1001 であるとする。図 3 では、第 1 世代でヒットした場合を示している。この場合、まず CAM のサブタグ比較において、1001 を持つウェイに一致フラグを立てる。SRAM サブタグ比較は、一致フラグが立っている第 1 世代ウェイで行う。データの出力も同時に行う。図 3 では、第 1 世代 SRAM でヒットしたため、第 2 世代 SRAM で一致フラグが立っているウェイは、動作させずに済み、その分の消費電力を削減できる。もし、第 2 世代 SRAM でミスした場合は、次のサイクルで第 2 世代 SRAM の探索を同様に行う。

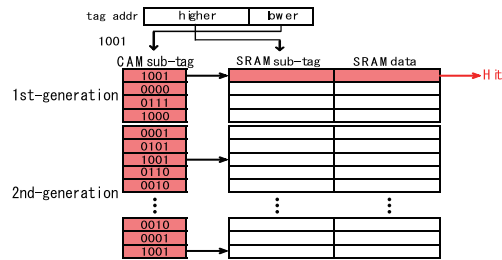


図3 提案手法(1)の動作例:第1世代ウェイトでヒットした場合.濃い色の部分は検索処理(データ出力)を要するサブタグ(データ)を表す.

Fig.3 The case when the first generation SRAM exhibits a hit. The shaded tags or data require energy for matching or output.

3.2 提案手法(1)の問題点

提案手法(1)は, CAMのサブタグ比較において, 複数ラインで一致した際は, それらのラインについて, 第1世代SRAMのサブタグ比較を同時に行う. 第1世代でヒットしなければ第2世代SRAMのサブタグ比較を同時に行う. これは, 実現可能性の面で問題がある. 一般的なSRAMセルは, 読み書きは同時に1つだけしか行えない. 同時比較を可能にするためには, 各メモリセルにワードラインとビットラインを増やすマルチポート化を行う必要がある. 提案手法(1)では, 連想度の数だけ重複する可能性があるため, その数だけポートを用意しなければならない. これは, 回路面積が増大し, 実現性が低い.

また, 提案手法(1)では, 第1世代SRAMも第2世代SRAMも, 置換方式としてLRU方式を用いている. これは, 回路面積や速度のコストの面で問題がある. LRU方式はキャッシュアクセスごとに全てのウェイトの利用順序の履歴を更新する. 連想度 w が32の場合では, 全てのウェイトにカウンタを持たせるといった簡単な手法を用いたとしても, 各ウェイトで5ビット必要で, さらに, アクセス毎に全てのウェイトでアクセス情報を更新する必要がある. これは, 回路面積や速度面でのコストが大きい.

3.3 提案手法(2)

提案手法(1)の問題点を解決するために, 置換方式については, 第1世代SRAMはLRU方式, 第2世代SRAMはFIFO方式にする. さらに, 各世代SRAM内でのサブタグ比較を逐次的に行う. 比較の順序は, 第1世代SRAMでは, より最近使われたウェイトから, 第2世代SRAMでは, より最近置換されたウェイトからとする.

第1世代のみLRU方式で管理する方法は, 第1世代SRAMのウェイト数 $r = 4$ の場合,

第1世代, 第2世代の識別に加え, LRU管理の状態は5つになる. さらにLRUの更新も第1世代SRAMでのヒットならば4ウェイト分, 第2世代SRAMでヒット, またはキャッシュミスの場合, 5ウェイト分で済む.

図4を例に動作を説明する. 図4では, 第1世代SRAMをウェイト4, 7, 8, 11とし, 赤色で示し, さらに第1世代SRAM内に利用履歴を示す. 利用履歴は小さい数ほど最近利用されたウェイトを表す. 図の例では, ウェイト4が最も最近利用されたウェイトである. また, キャッシュミス時の置換対象となるウェイトを置換対象ウェイトと呼ぶ. 図の例では, 置換対象ウェイトはウェイト24である. このとき, キャッシュミスが発生するとウェイト24のラインが置換され, 置換対象ウェイトはウェイト25に移る. すなわち, 図4の例では, 第2世代ウェイトの中で最も新しくキャッシュに入ったウェイトは, ウェイト23である. なお, 置換対象ウェイトがウェイト31にあり, キャッシュミスが発生した場合は, 次の置換対象ウェイトはウェイト0になる. さらに, 置換対象ウェイトはキャッシュミス時の置換を行うためのものであることから, 第1世代に移動してきた場合, 置換対象ウェイトとせず先へ移動する.

CAMのサブタグでヒットしたウェイトが7, 11, 12, 13であったとする. まず, 第1世代SRAMでタグ比較・データ読み出しを行う. 第1世代SRAMの最近利用されたウェイトを優先して, SRAMサブタグ比較・データ読み出しを逐次的に行う. ウェイト7とウェイト11の利用履歴の番号はそれぞれ1, 2であるため利用履歴の番号が小さいウェイト7のサブタグ比較とデータ読み出しを先に行う. ウェイト7がヒットした場合はデータを出しウェイト11, 12,

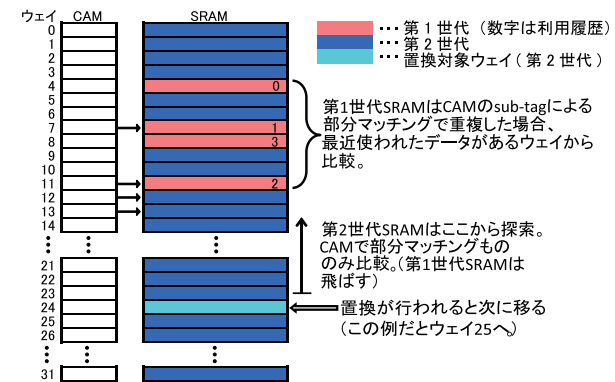


図4 提案手法(2)の動作例

Fig.4 Operation example of the proposed method (2).

13ではサブタグ比較とデータ読み出しを行わない。ウェイ7がミスした場合はウェイ11のサブタグ比較を行う。ウェイ11でもミスした場合は、比較操作を第2世代SRAMで行う。

第2世代SRAMでは、より最近置換されたウェイから逐次的にサブタグ比較・データ読み出しを行う。ウェイ12とウェイ13では、ウェイ13の方がより新しいウェイなので、ウェイ13のサブタグ比較とデータ読み出しを先に行う。ウェイ13もウェイ12もヒットしなかった場合は、キャッシュミスとなる。

CAMのサブタグとの部分マッチングにおいて、ヒットするキャッシュラインを1つに絞らず、かつ第1世代SRAMにLRU方式を採用しているため、第2世代SRAMの置換方式をFIFO方式にしても、ヒット率の低下を招かないと考えられる。

SRAM内を逐次的に比較・読み出しを行うため、クロックサイクル数が増加する可能性がある。しかし、命令アドレスの局所参照性からヒットしやすい第1世代SRAMの最初の比較・読み出しでヒットする可能性は高いため、クロックサイクル数の増加を抑制できると考えられる。同時に、無駄な比較・読み出しを行わないことから、消費電力も削減できると考えられる。

4. 実験方法と実験結果、評価結果

提案手法(2)を評価するためにシミュレーション実験を行った。本章では、実験方法、ミス率の測定結果、および、クロックサイクルと消費電力量の評価結果を述べる。通常の高連想度キャッシュであるTraditional CAM、および、LPHAC法のCAMのサブタグビット長 $s=8$ の場合と比較する。

4.1 方法

シミュレータSim-Panalyzer¹³⁾を用いて、組込み向けベンチマークプログラムMiBench¹⁴⁾を実行する。今回の実験では、MiBenchのうち、図5に示す12個のベンチマークプログラムをlarge data setで実行した。以下に示す実験結果では、図5のプログラム番号を用いている。

実験条件は以下のとおりである。プロセッサアーキテクチャは、組込み向けARMとし、L1キャッシュのみ、キャッシュ容量8KB、ラインサイズ32Bytes、サブバンクの数を8と

1. bitcount
2. qsort
3. susan_s
4. tiff2bw
5. lame
6. patricia
7. blow_d
8. rijndael_d
9. rijndael_e
10. sha
11. rawcaudio
12. untoast

図5 使用したベンチマークプログラム名及び対応番号

Fig.5 Benchmark programs used for evaluation.

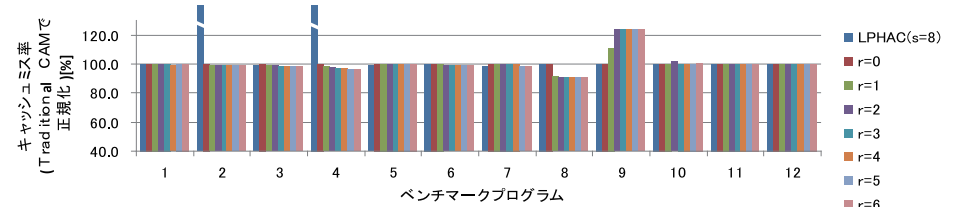


図6 i11におけるTraditional CAM, LPHAC法, 及び提案手法のキャッシュミス率

Fig.6 Instruction cache miss rates of benchmarks for the traditional CAM, LPHAC, and proposed method.

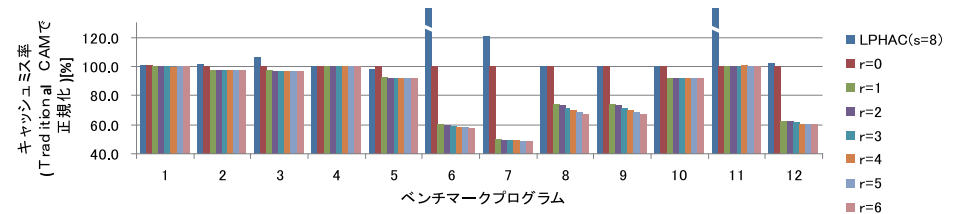


図7 dl1におけるTraditional CAM, LPHAC法, 及び提案手法のキャッシュミス率

Fig.7 Data cache miss rates of benchmarks for the traditional CAM, LPHAC, and proposed method.

する。置換方法は、提案手法の第1世代SRAMでは、第3章で述べた通りLRU方式とし、Traditional CAM, LPHAC法, 提案手法の第2世代SRAMではFIFO方式とする。また、ウェイ数 $w=32$ とし、CAMのサブタグビット長 s は、 $0 \leq s \leq 8$ 、第1世代SRAMウェイ数 r は、 $0 \leq r \leq 6$ の範囲で実験を行う。

4.2 キャッシュミス率

ベンチマークプログラムごとのミス率を測定した。命令キャッシュi11とデータキャッシュdl1についてTraditional CAMのキャッシュミス率を100%とした場合のミス率を、図6と図7に示す。各ベンチマークプログラムにつき、左から順に、LPHAC法でCAMのサブタグビット長 $s=8$ の場合、提案手法において、 $r=0, 1, 2, 3, 4, 5, 6$ の場合となっている。なお、提案手法の s は、ミス率に影響を与えない。また、 $r=0$ は、世代分割を行わない場合で、ミス率はTraditional CAMと同等である。

図6から、i11のミス率は、LPHAC法では、多くがTraditional CAMとほぼ同じだが、qsortとtiff2bwにおいて大きく悪化する。提案手法では、rijndael_eだけ悪化するが、

23.7%程度の増加に収まる．tiff2bw や rijndael_d では低くなる．

図 7 から，dl1 のミス率は，LPHAC 法では，rawcaudio や patricia, blow_d, susan_s において大きく悪化する．提案手法では，全体として良く，patricia, blow_d, rijndael_d, rijndael_e などでは低くなる．dl1 の方が， r を変化させた場合のミス率低下が顕著に出ている．

ベンチマークプログラムの平均ミス率を il1 と dl1 について図 8 に示す．図 8 より，il1, dl1 のどちらでも，LPHAC 法が，一番ミス率が高いのに対し，提案手法では， $r = 0$ で Traditional CAM と同等で， r を大きくするとミス率が小さくなる．

4.3 消費電力量とクロックサイクル

提案手法 (2) の総消費電力量 E を評価する．これは，CAM のサブタグ部分，SRAM のサブタグ部分，SRAM のデータ部分の消費電力を合わせたものである．SRAM タグ 1 ビットの検出に要する電力量を単位とする．SRAM の電力消費においては，ビットキャパシタンスの充放電に要する電力が支配的である¹⁵⁾ ので，SRAM セル 1 ビットあたりの消費電力

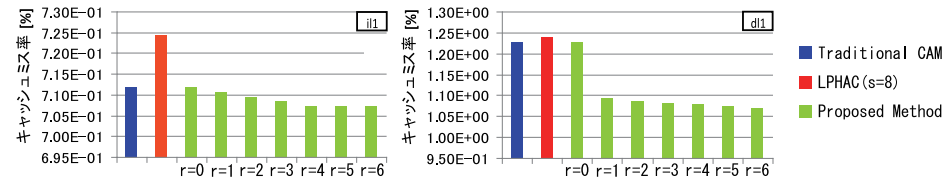


図 8 Traditional CAM, LPHAC 法，及び提案手法の平均キャッシュミス率

Fig. 8 Average miss rates of benchmarks for the traditional CAM, LPHAC, and proposed method.

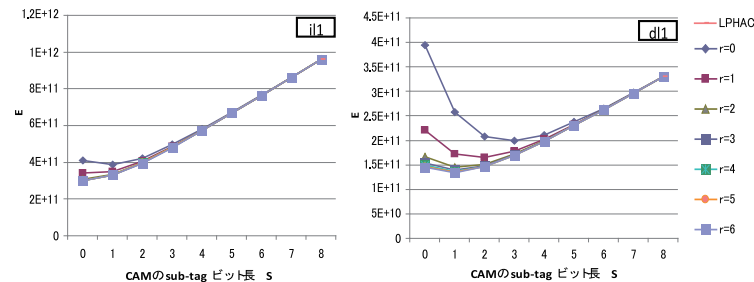


図 9 r と s を変えた場合の平均消費電力量

Fig. 9 Average energy consumption depending on r and s .

量は，サブタグ部分もデータ部分も変わらないと近似する．CAM セル 1 ビットの消費電力量は SRAM セル 1 ビットの 5 倍とする⁷⁾．

クロックサイクル数の評価では，キャッシュメモリへのアクセスを 1 サイクル，メインメモリへのアクセスを 40 サイクルとし，Traditional CAM のクロックサイクル数を 100% と正規化した．

CAM のサブタグビット長 s と第 1 世代 SRAM ウェイ数 r を変えた場合の消費電力量と平均クロックサイクル数を図 9, 10 に示す．図 9 より， s の値を小さくすると消費電力量は下がるが， r の値が小さいときは s を小さくすると電力は上昇する． r の値が大きいと，消費電力量は下がるが， r が大きいほど変化量が小さくなる．図 10 より， s の値を小さくするとクロックサイクル数は上がる． s の値を大きくするとクロックサイクル数は 100% に漸近する． r の値が大きいと，クロックサイクル数は下がるが， r が大きいほど変化量が小さくなる．このように，実現可能性を考慮に入れなければ， r の値は大きい方が良いが， r が大きいほど消費電力量とクロックサイクル数の変化量は小さくなる． s には最適値があるが，最適値は r の値によって変わる．

Traditional CAM とのクロックサイクル差を 10% 以内に抑えるには，il1 では $s \geq 3$ かつ $r \geq 2$ ，dl1 では $s = 3$ かつ $r \geq 5$ ， $s \geq 4$ かつ $r \geq 2$ とする必要がある．5% 以内に抑えるには，il1 では $s = 4$ かつ $r \geq 3$ ， $s \geq 5$ かつ $r \geq 2$ とする必要がある，dl1 では $s = 4$ かつ $r \geq 4$ ， $s = 5$ かつ $r \geq 3$ ， $s \geq 6$ かつ $r \geq 2$ とする必要がある．

r が大きいと，LRU 管理のためのコストが増えるため r は小さいほうが良い．これと，図 9 の消費電力量を考慮すると，クロックサイクル 5% 以内の上昇で，消費電力量を低く抑えるには，il1 では $s = 4, r = 3$ ，dl1 では $s = 4, r = 4$ が最も良い．

提案手法 (2) で $s = 4, r = 3$ とした時の il1 と dl1 の消費電力量とクロックサイクル数を

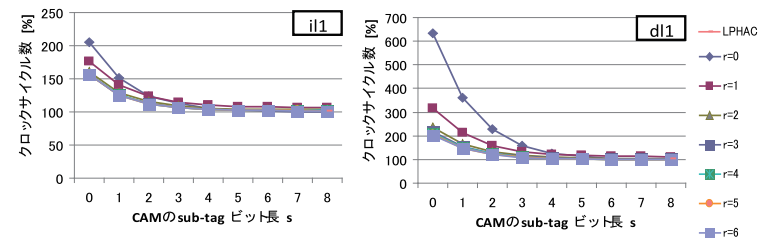


図 10 r と s を変えた場合の平均クロックサイクル

Fig. 10 Average clock cycle depending on r and s .

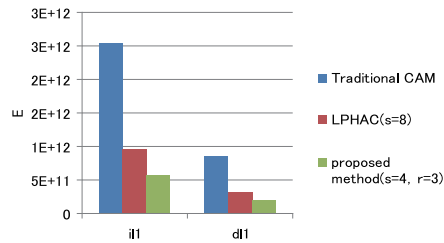


図 11 平均消費電力量の比較

Fig. 11 Average energy consumptions for the traditional CAM, LPHAC($s = 8$), and proposed method($s = 4, r = 3$).

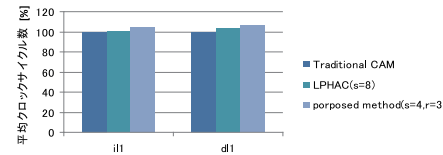


図 12 平均クロックサイクルの比較

Fig. 12 Average clock cycles for the traditional CAM, LPHAC($s = 8$), and proposed method($s = 4, r = 3$).

従来法と比較して、図 11 と図 12 に示す。LPHAC 法は、文献¹⁰⁾より $s = 8$ とした。図 11 より、提案手法は Traditional CAM と比較して、il1 で 77.4%、dl1 で 77.2% 消費電力量を削減できていることが分かった。提案手法は LPHAC 法と比較して、il1 で 40.4%、dl1 で 39.9% 消費電力量を削減できていることが分かった。この時、図 12 より、Traditional CAM と比較して il1 で 4.4%、dl1 で 6.1%の遅延増加で済む。

5. おわりに

タグを CAM と SRAM とで実現する高連想度キャッシュにおいて、実現可能性とコストを考慮した上で、ヒット率の低下を抑制しつつ、CAM の使用量を減らす手法を提案した。提案手法 (1) では、CAM のサブタグとの部分マッチングにおいて、ヒットするキャッシュラインを 1 つに絞らない。また、SRAM 部分をアクセス頻度により 2 世代に分割する。提案手法 (2) では、提案手法 (1) を改良し、第 1 世代 SRAM は LRU 方式、第 2 世代 SRAM は FIFO 方式にする。さらに、各世代 SRAM 内でのサブタグ比較では、第 1 世代 SRAM では最近使われたウェイから、第 2 世代 SRAM では置換対象ウェイのひとつ前のウェイから逆順に、逐次的に比較する。シミュレーション実験により提案手法は従来法と比較し、ヒット率の低下を抑制しつつ高い電力削減効果が得られることがわかった。

参 考 文 献

1) H.J. Chao. Next generation routers. *Proceedings of the IEEE*, Vol.90, No.9, pp. 1518–1558, 2002.

2) S.R. Singh and K.Moez. Ultra low leakage 90nm content addressable memory design for wireless sensor network applications. In *Circuits and Systems, 2009. MWSCAS'09. 52nd IEEE International Midwest Symposium on*, pp. 1191–1194. IEEE, 2009.

3) M.Lin, C.Yunji, S.Menghao, Q.Zichu, Z.Heng, and H.Weiwu. Testing content addressable memories using instructions and march-like algorithms. In *Electronics, Circuits and Systems, 2008. ICECS 2008. 15th IEEE International Conference on*, pp. 774–777. IEEE, 2008.

4) A.R. Patwary, B.M. Geuskens, and S.L.L. Lu. Content Addressable Memory for Low-Power and High-Performance Applications. In *Computer Science and Information Engineering, 2009 WRI World Congress on*, Vol.3, pp. 423–427. IEEE, 2009.

5) Intel. *Intel XScale Microarchitecture*, 2001.

6) S.Santhanam, A.JBaum, D.Bertucci, M.Braganza, K.Broch, T.Broch, J.Burnette, E.Chang, K.T. Chui, D.Dobberpuhl, et al. A low-cost, 300-MHz, RISC CPU with attached media processor. *IEEE Journal of Solid-State Circuits*, Vol.33, No.11, pp. 1829–1839, 1998.

7) A.Efthymiou and J.D. Garside. An adaptive serial-parallel CAM architecture for low-power cache blocks. In *Proc. ISLPED*, pp. 136–141, 2002.

8) A.Veidenbaum and D.Nicolaescu. Low Energy, Highly-Associative Cache Design for Embedded Processors. *Proc. IEEE ICCD*, pp. 332–335, 2004.

9) A.Ma, M.Zhang, and K.Asanovic. Way memoization to reduce fetch energy in instruction caches. In *ISCA Workshop on Complexity Effective Design*, Vol.20, p.31, 2001.

10) Chuanjun Zhang. A Low Power Highly Associative Cache for Embedded Systems. *Proc. IEEE ICCD*, pp. 31–36, 2006.

11) 岡部翔, 阿部公輝. SRAM の世代分割により CAM を有効利用する低電力高連想度キャッシュ. 組込みシステムシンポジウム 2009(ESS2009) 論文集, pp. 117–126, 2009.

12) S.Okabe and K.Abe. A CAM-based Low-power Highly Associative Cache for High-performance Embedded Processors. In *Proc. IWMST2010*, pp. 88–93, 2010.

13) Sim-Panalyzer. <http://www.eecs.umich.edu/panalyzer/>.

14) M.R Guthaus and et al. MiBench: A free, commercially representative embedded benchmark suite. *IEEE WWC*, pp. 83–94, 2001.

15) K.Itoh, K.Sasaki, and Y.Nakagome. Trends in low-power RAM circuit technologies. *Proceedings of the IEEE*, Vol.83, No.4, pp. 524–543, 1995.