

High-performance/Low-power Cache Architectures for Merged DRAM/Logic LSIs

井上, 弘士

九州大学大学院システム情報科学研究科 | 財団法人九州システム情報技術研究所

石原, 亨

東京大学大規模集積システム設計教育研究センター

甲斐, 康司

財団法人九州システム情報技術研究所

村上, 和彰

九州大学大学院システム情報科学研究科

<http://hdl.handle.net/2324/7655>

出版情報 : 情報処理学会論文誌. 42 (3), pp.419-431, 2001-03-15. Information Processing Society
in Japan

バージョン :

権利関係 :



DRAM/ロジック混載 LSI 向け高性能/低消費電力キャッシュ・アーキテクチャ

井上 弘 士^{†1,†3} 石原 亨^{†2}
甲斐 康 司^{†3} 村上 和 彰^{†4}

DRAM とロジックの混載は、21 世紀の SOC(System-On-a-Chip) 時代を支える最も重要な技術の 1 つである。従来は分チップ構成であった CPU と主記憶を 1 チップ化することにより、今までにない新しい計算機構成法の実現が可能となる。本稿では、特にオンチップ・メモリシステムに焦点を当て、著者らがこれまでに提案してきた DRAM/ロジック混載 LSI 向けキャッシュ・アーキテクチャについて議論する。これらのキャッシュは、DRAM/ロジック混載 LSI の特徴である高オンチップ・メモリバンド巾を有効に活用し、その潜在能力を引き出すことで、高性能かつ低消費エネルギーなオンチップ・メモリシステムの実現を可能にする。

High-Performance/Low-Power Cache Architectures for Merged DRAM/Logic LSIs

KOJI INOUE,^{†1,†3} THORU ISHIHARA,^{†2} KOJI KAI^{†3}
and KAZUAKI MURAKAMI^{†4}

Integrating main memory and microprocessors into the same chip is one of the most important technologies for future SOC (System-On-a-Chip). The integration makes it possible to realize novel memory architectures because of eliminating the chip boundary between the main memory and microprocessors. This paper discusses cache architectures for high-performance/low-power memory systems on merged DRAM/logic LSIs. The caches make good use of the attainable high on-chip memory bandwidth, and try to bring out potential advantages of the merged DRAM/logic LSIs.

1. はじめに

集積回路技術の順調な進歩により、ついに「ギガの世界」が現実となった。1GHz を超える超高速マイクロプロセッサ (CPU) や Gbit 級の超大容量 DRAM が次々と報告されている。このように、CPU と主記憶 (DRAM) は、それぞれが個別デバイスとして目覚ましい発展を遂げてきた。しかしながら、これらを組み合わせ合わせたコンピュータ・システムにおいては、1) CPU-主記憶間の性能差拡大、2) I/O ピン・ボトルネックの顕在化によるメモリバンド巾の限界、といった高性能

化阻害要因が依然として存在する。また、PDA やノート型 PC に代表される携帯型電子機器システムの普及に伴い、3) 高性能化と低消費エネルギー化の両立、に対する要求が極めて強くなってきた。

前述した 1) と 2) の技術的課題を解決し、かつ、3) の市場要求を満足できる有力なデバイスとして、DRAM/ロジック混載 LSI がある。従来は分チップ構成であった CPU と主記憶を 1 チップ化することで、チップ間 I/O ピン・ボトルネックが解消される。その結果、オンチップ化されたメモリバスは容易に拡張可能となり、メモリバンド巾が劇的に向上する。また、オフチップ・アクセス回数が極めて少なくなるため、外部入出力ピン駆動による消費エネルギーを大幅に削減できる。DRAM とロジックの混載は、21 世紀の SOC(System-On-a-Chip) 時代において核となる要素技術である。今後、更なる高性能化/低消費エネルギー化を達成するためには、単に CPU と主記憶を 1 チップ化するだけでなく、混載の利点を最大限有効に活用し、かつ、その潜在能力を十分に引き出すことのできる新しいメモリ・アーキテクチャを考案する必要がある。

^{†1} 九州大学大学院システム情報科学研究科
Department of Computer Science and Communication
Engineering, Kyushu University

^{†2} 東京大学大規模集積システム設計教育研究センター
VLSI Design and Education Center, The University of
Tokyo

^{†3} (財)九州システム情報技術研究所
Institute of Systems & Information Technologies/KYUSHU

^{†4} 九州大学大学院システム情報科学研究科
Department of Informatics, Kyushu University

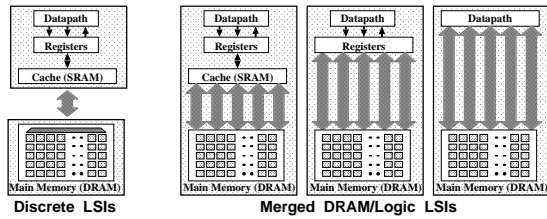


図1 オンチップ/オフチップ・メモリバス・アーキテクチャ

DRAM/ロジック混載 LSI では、図 1 に示すように、階層メモリ構造の自由度が広がる。これらは、1) 演算器と主記憶間で直接データのやり取りを行う DM(Datapath-Memory) 型、2) キャッシュ・メモリは搭載せず、データのロード/ストアは直接レジスタ-主記憶間で行う DRM(Datapath-Register-Memory) 型¹²⁾、3) 現在の多くのコンピュータ・システムが採用している DRCM(Datapath-Register-Cache-Memory) 型¹⁰⁾に分類できる。しかしながら、年率 22%の割合で動作周波数を向上し続ける CPU に対し、DRAM 行アクセス時間の向上率は年 7%と極めて低い。よって、CPU-主記憶 (DRAM) 間の性能差を十分隠蔽するためには、DRAM/ロジック混載 LSI においてもなお、オンチップ・キャッシュの搭載が必要であると考えられる。そこで本稿では、オンチップ・メモリシステムの高性能化/低消費エネルギー化を目的とした DRAM/ロジック混載 LSI 向けキャッシュ・アーキテクチャとして、

- ウェイ予測キャッシュ・アーキテクチャ⁵⁾
- 動的可変ラインサイズ・キャッシュ・アーキテクチャ⁶⁾⁷⁾

について議論する。これらのキャッシュは、メモリ参照の局所性を活用し、メモリ参照履歴に基づいて投機的にキャッシュ内の処理を最適化することで、高性能化と低消費エネルギー化という相反する要求を同時に満足する。

以下、第 2 章では、DRCM 型の DRAM/ロジック混載 LSI において、高オンチップ・メモリバンド巾を活用する際の利点と欠点を整理する。そして、更なる高性能化/低消費エネルギー化を実現するための手段として、第 3 章ではウェイ予測キャッシュ、第 4 章では動的可変ラインサイズ・キャッシュについて述べる。第 5 章ではベンチマーク・プログラムを用いた定量的評価を行い、最後に第 6 章でまとめる。

2. 高オンチップ・メモリバンド巾活用における利点と欠点

2.1 利点

メモリシステム性能の評価尺度として平均メモリアクセス時間 ($AMAT$: *Average Memory Access Time*) があり、以下の式で表される。

$$AMAT = T_{Cache} + CMR \times 2 \times T_{Main} \quad (1)$$

$$T_{Main} = T_{DRAM} + \frac{LineSize}{BandWidth} \quad (2)$$

ここで、 T_{Cache} はキャッシュ・アクセス時間、 CMR はキャッシュ・ミス率、 T_{Main} はキャッシュ-主記憶間でのラインリプレイスにおける主記憶アクセス時間 (ミス・ペナルティ) である。また、 T_{DRAM} はオンチップ DRAM アクセス時間 (DRAM スタートアップ時間)、 $LineSize$ はリプレイスされるキャッシュ・ラインのサイズ、 $BandWidth$ はキャッシュ-主記憶間バンド巾を表す。最悪の場合、キャッシュ・ミスが発生した際には 2 回の主記憶アクセス (ライトバックとリフィル) が生じる。キャッシュは、メモリ参照の時間的/空間的局所性を利用することで、高いヒット率を達成する (CMR を低くする)。特にこの空間的局所性は、ラインリプレイス時に多くのデータを一度に主記憶からキャッシュへリフィルすること (つまり、ラインサイズ $LineSize$ を拡大すること) で活用できる。キャッシュ・ミスを引き起こしたデータの近傍データも一度にキャッシングされ、プリフェッチ効果によるキャッシュ・ヒット率の向上を期待できるためである。

分チップ構成の従来型コンピュータ・システムでは、キャッシュ-主記憶間 I/O ピン・ボトルネックによりメモリバンド巾が制限される。そのため、空間的局所性の活用を目的としてラインサイズを拡大した場合、キャッシュ-主記憶間でのデータ転送時間が増大し (式 2 の $\frac{LineSize}{BandWidth}$ が増大)、それに伴いミス・ペナルティが増加する (式 2 の T_{Main} が増加)。これに対し、DRAM/ロジック混載 LSI では、I/O ピン・ボトルネックが解消されるため、高オンチップ・メモリバンド巾を実現できる (式 2 の $BandWidth$ を大きくできる)。その結果、ミス・ペナルティの増加を伴うことなくラインサイズを拡大可能となる ($LineSize \leq Bandwidth$ の条件下では、 $\frac{LineSize}{BandWidth}$ が一定)。

2.2 性能に関する問題点

第 2.1 節において、高オンチップ・メモリバンド巾

本稿では、プログラム実行に必要なメモリ領域は全てオンチップ DRAM に格納されると仮定する。

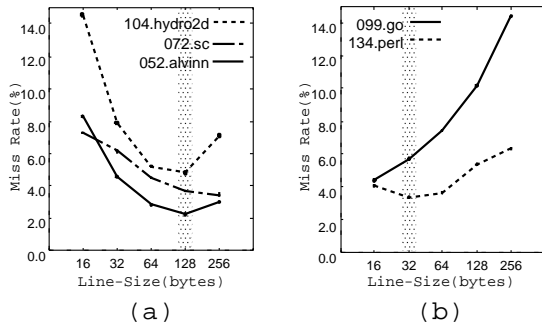


図2 16 KB ダイレクトマップ・キャッシュにおけるラインサイズとミス率の関係

活用の利点は「ミス・ペナルティの増加を伴わずにラインサイズを拡大可能な点」にあると述べた。しかしながら、キャッシュ・サイズが一定の場合、ラインサイズの拡大により、キャッシュ内に格納可能な総ライン数が減少する。そのため、メモリ参照の空間的局所性が低い場合には、頻繁なキャッシュ・コンフリクトが発生し、それに伴いキャッシュ・ヒット率が低下する(式1の CMR が高くなる)。

ほとんどのプログラムにおいて、命令参照は高い空間的局所性を有する。そのため、命令キャッシュでのラインサイズ拡大は非常に有効である¹³⁾。一方、図2に示すように、データ参照における空間的局所性の度合はプログラムにより様々である。たとえば、図2(a)のプログラムは比較的高い空間的局所性を有するため、128 バイトの大きなラインサイズで高いヒット率(低いミス率)を得る。これに対し、図2(b)のプログラムは低い空間的局所性を有するため、ラインサイズの拡大は大幅なヒット率の低下(ミス率の増大)を招く。つまり、高オンチップ・メモリバンド巾の活用を目的としてラインサイズを拡大した場合、プログラムによっては性能が低下する場合がある。

2.3 消費電力に関する問題点

一般的に、DRAM/ロジック混載 LSI では、CPU コア面積に比べ、キャッシュおよび主記憶で構成されるオンチップ・メモリ面積が極めて大きくなる。そのため、オンチップ・メモリシステムにおける消費電力の増大は、チップ全消費電力に大きな悪影響を与える。キャッシュと主記憶でオンチップ・メモリシステムが構成される場合、平均メモリアクセス消費エネルギー ($AMAE$: *Average Memory Access Energy*) は以下の式で近似できる。

$$AMAE = E_{Cache} + CMR \times 2 \times E_{Main} \quad (3)$$

ここで、 E_{Cache} はキャッシュ・アクセス当たりの消費エネルギー、 E_{Main} はキャッシュ・リプレイスのため

の主記憶アクセスで消費されるエネルギー、 CMR はキャッシュ・ミス率である。式1と同様に、キャッシュ・ミスが発生した際、最悪時には2回の主記憶アクセスが生じる。

キャッシュ・アクセス当たりの消費エネルギー (E_{Cache}) は、主に、アドレス・デコードに要するエネルギー (E_{Cache_dec})、および、SRAM セルへのアクセスに要するエネルギー (E_{Cache_cell}) の和で表すことができる。これに加え、分チップ構成である従来のメモリシステムでは、ライン・リプレイスを行う際、外部入出力ピン駆動に要するエネルギー (E_{Cache_io}) が消費される¹⁴⁾。ここで、 E_{Cache_dec} は、 E_{Cache_cell} に比べ、 E_{Cache} に与える影響が極めて小さいことが報告されている¹⁾。また、DRAM/ロジック混載 LSI では、キャッシュと主記憶が共にオンチップ化されているため、ライン・リプレイス時に外部入出力ピンを駆動する必要はない。そこで本稿では、キャッシュ・アクセス当たりの消費エネルギーを E_{Cache_cell} で近似する。

第2.1節で述べたように、DRAM/ロジック混載 LSI では、DRAM アレイおよびキャッシュ-主記憶間バスを拡幅し、かつ、ラインサイズを拡大する事で、高オンチップ・メモリバンド巾を活用可能となる。しかしながら、第2.2節で示したように、メモリ参照の空間的局所性が低い場合、頻繁なコンフリクト・ミスの発生によりヒット率が低下する(式3の CMR が高くなる)。その結果、主記憶アクセス回数が増加し、それに伴い拡幅された DRAM アレイおよびオンチップ・バスが頻繁に活性化されるため、多くのエネルギーを消費する。

また、キャッシュ・アクセスにおける低消費エネルギー化(式3の E_{Cache} の削減)も重要である。キャッシュはオンチップ主記憶と比較して小容量となる。しかしながら、更なる高ヒット率の達成を目的として、キャッシュ・サイズは年々増加傾向にある。また、主記憶アクセスはキャッシュ・ミス時のみ発生するのに対し、キャッシュ・アクセスはメモリ・アクセス毎に行われる。したがって、キャッシュ・アクセスにおける低消費エネルギー化は、オンチップ・メモリシステム全体の低消費エネルギー化に大きく寄与する。

2.4 問題点のまとめと従来技術での解決策

キャッシュ・コンフリクトを回避する従来手法として、セット・アソシアティブ方式の採用がある¹¹⁾¹³⁾。通常のセット・アソシアティブ・キャッシュ(SA キャッシュ)では、あるデータを格納可能なキャッシュ内ロケーションが n ケ所 (n ウェイ SA キャッシュの場合) 存在する。そのため、キャッシュ・コンフリクトの発

生を回避して、高ヒット率を達成できる¹¹⁾。しかしながら、 n 個のウェイを全て同時に活性化し、タグ比較結果に基づき参照データを選択しなければならない。そのため、ダイレクト・マップ(DM)方式と比較してキャッシュ・アクセス時間が長くなる(式1の T_{Cache} が大きくなる)。また、キャッシュ・ヒットの場合、ある唯一のウェイにのみ参照データが存在するにも関わらず、全てのウェイを活性化するため、多くのエネルギーを浪費する(式3の E_{Cache} が大きくなる)。以下、第2.2節および第2.3節に基づき、DRAM/ロジック混載LSIにおける問題点をまとめる。また、これら問題点に対し、従来型SAキャッシュの有効性を考察する。

- 問題1: ラインサイズの拡大に伴い頻繁なコンフリクト・ミスが発生し、メモリシステム性能が低下する(式1における CMR の増加)。SA方式を採用する事でコンフリクト・ミスの発生を回避できる。しかしながら、メモリシステム性能は、式1に示すように、キャッシュ・ミス率(CMR)とキャッシュ・アクセス時間(T_{Cache})の両方に依存する。この点に関して、SA方式が常に有効であるとは限らない。
 - 問題2: ラインサイズの拡大に伴い頻繁なコンフリクト・ミスが発生し、主記憶アクセス回数が増加する。これにより、主記憶アクセスによる消費エネルギーが増大する(式3における CMR の増加)。従来型のキャッシュにおいて、主記憶アクセス回数はヒット率にのみ依存する。そのため、高ヒット率を達成できるSA方式の採用は、この問題に対して有効である。
 - 問題3: ラインサイズの拡大に伴い、拡張されたDRAMアレイおよびオンチップ・バスが活性化される。これにより、主記憶アクセス当たりの消費エネルギーが増大する(式3における E_{Main} の増大)。大きな固定ラインサイズを有する従来型SAキャッシュでは、この問題を解決できない。
 - 問題4: キャッシュの大容量化および高速化に伴い、キャッシュ・アクセスでの消費エネルギーが増大する(式3における E_{Cache} の増大)。キャッシュ・ヒットの場合、唯一のウェイにのみ参照データが存在するにも関わらず、従来型SAキャッシュでは全てのウェイが活性化される。そのため、多くのエネルギーを浪費する。
3. 投機的ウェイ選択によるセット・アソシアティブ・キャッシュの低消費エネルギー化
本章では、第2.4節で述べた問題4を解決する手法

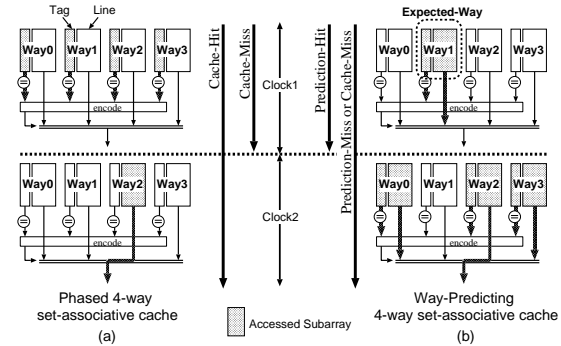


図3 段階型キャッシュとウェイ予測型キャッシュ

として、ウェイ予測セット・アソシアティブ・キャッシュ(Way-Predicting Set-Associative Cache: WPキャッシュ)について議論する⁵⁾。本来、ウェイ予測技術は、キャッシュの高性能化を目的として考案された²⁾。我々が提案したWPキャッシュでは、低消費エネルギー化を目的としてこのウェイ予測技術を活用する。まず、第3.1節では、比較対象として、過去に提案された段階的アクセスによる低消費エネルギー化技術を説明する。次に、第3.2節で我々が提案したWPキャッシュを紹介する。なお、WPキャッシュの有効性に関する評価は第5章で行う。

3.1 段階的アクセスによる低消費エネルギー化
キャッシュ・アクセス当りの消費エネルギー E_{Cache} は、以下の式で表すことができる。

$$E_{Cache} = N_{Tag} \times E_{Tag} + N_{Line} \times E_{Line}$$

ここで、 E_{Tag} および E_{Line} は、タグ1個およびライン1個当たりの読み出しに要するエネルギーを表す。また、 N_{Tag} と N_{Line} は、それぞれ、キャッシュ・アクセス当りに読み出されるタグおよびラインの個数である。

第2.4節で述べたように、従来型4ウェイSAキャッシュでは、ヒット/ミスに関係なく、キャッシュ・アクセス毎に4個のタグと4個のラインが同時に読み出される。これに対し、図3(a)に示すように、ライン読み出しをタグ比較の後に遅らせ、参照ライン(プロセッサが要求するデータを含むライン)だけを読み出すことで消費エネルギーを削減できる⁴⁾。以下、このようなキャッシュを「段階型キャッシュ(Phased Cache:Pキャッシュ)」と呼ぶ。段階型キャッシュでは、タグ読み出しとライン読み出しを逐次的(段階的)に行い、キャッシュ・ヒット時のアクセス時間を犠牲にすることで低消費エネルギー化を実現する。従来型4ウェイSAキャッシュおよび段階型4ウェイSAキャッシュのアクセス時間(T_{Cache} :ここでは所要クロック・サイクル数)と消費エネルギー(E_{Cache})を表1に示す。

表 1 従来型/段階型キャッシュのアクセス時間と消費エネルギー

キャッシュ	E_{Cache}	$T_{Cache}(\text{cycle})$
従来型	$4E_{Tag} + 4E_{Line}$	1
段階型	$4E_{Tag} + CHR \times 1E_{Line}$	$1 + (CHR \times 1)$

CHR:Cache-Hit Rate

なお、表中の CHR はキャッシュ・ヒット率（つまり、 $1 - CMR$ ）である。

3.2 ウェイ予測キャッシュ

3.2.1 基本概念

第 3.1 節で述べたように、段階型キャッシュはキャッシュ・ヒット時間（ヒット時のキャッシュ・アクセス時間）を犠牲にして低消費エネルギー化を実現する。しかしながら、多くのプログラムにおいて、キャッシュ・ヒット率は非常に高い。そのため、キャッシュ・ヒット時の低速アクセスは平均メモリアクセス時間の増大を招き、ひいては、システム性能に大きな悪影響を及ぼす。キャッシュ性能の低下を伴うことなく、低消費エネルギー化を実現するためには、

- タグとラインの並列読み出しによる高速化、
- 参照ラインだけの読み出し（無駄なライン読み出しの回避）による低消費エネルギー化、

を同時に達成する必要がある。しかしながら、従来型キャッシュや段階型キャッシュはいずれか一方の要求は満足するが、これら 2 つの要求を同時に満たすことはできない。

WP キャッシュは、CPU の参照データが存在する唯一のウェイをタグ比較とは独立に予測することで、上記 2 つの相反する要求を同時に満足する。以下、CPU の参照データが存在すると予測されたウェイを「見込みウェイ (Expected-Way)」と呼ぶ。そして、通常のキャッシュ・ヒット/ミス以下のように分類する。

- ウェイ予測ヒット (Way-Prediction Hit:WPH) : 通常のキャッシュ・ヒットであり、かつ、ウェイ予測が正しい場合。
- ウェイ予測ミス (Way-Prediction Miss:WPM) : 通常のキャッシュ・ヒットであり、かつ、ウェイ予測が誤りの場合。
- キャッシュ・ミス (Cache Miss:CM): 通常のキャッシュ・ミスであり、リプレースが発生する場合。

WP キャッシュでは、図 3(b) に示すように、タグ比較を行う前に参照データが存在するウェイを見込みウェイとして予測する。そして、見込みウェイに対応するタグおよびラインだけを並列に読み出し、タグ比較を行う。もし、タグ比較結果が一致（ウェイ予測ヒット）であれば、見込みウェイから読み出したライン内に参

表 2 4ウェイ WP キャッシュの性能と消費エネルギー

ケース	E_{Cache}	$T_{Cache}(\text{cycle})$
WPH	$1E_{Tag} + 1E_{Line}$	1
WPM	$4E_{Tag} + 4E_{Line}$	2
CM	$4E_{Tag} + 4E_{Line}$	2
総合	$WPHR \times (1E_{Tag} + 1E_{Line}) + (1 - WPHR) \times (4E_{Tag} + 4E_{Line})$	$WPHR \times 1 + (1 - WPHR) \times 2$

WPHR:Way-Prediction-Hit Rate

照データが存在する。よって、CPU にこの参照データを供給してキャッシュ・アクセスを終了する。一方、見込みウェイでのタグ比較結果が不一致であれば、見込みウェイを除く全てのウェイにおいて、タグとラインの並列読み出しを行う。その後、通常のキャッシュと同様にタグ比較（ウェイ予測ミスもしくはキャッシュ・ミスの判定）を行い、キャッシュ・ミスであれば置換えアルゴリズムに従ってライン・リプレースを実行する。このように、WP キャッシュは、ウェイ予測ヒットの場合には上記 2 つの相反する要求を同時に満足する。しかしながら、ウェイ予測ミスもしくはキャッシュ・ミスの場合には、上記 2 つの要求のいずれも満足することができない。

3.2.2 アクセス時間と消費エネルギー

4ウェイ WP キャッシュのアクセス時間（所要クロック・サイクル数）と消費エネルギーを表 2 に示す。ここで、WPHRとは、ウェイ予測ヒット率（ウェイ予測ヒット回数/メモリ参照回数）である。ウェイ予測ヒットの場合、WP キャッシュは見込みウェイ 1 つのみ活性化する。また、その時のアクセス時間は、従来型キャッシュのそれと同じである。一方、ウェイ予測がはずれた場合（ウェイ予測ミス、もしくは、キャッシュ・ミスの場合）には、従来型キャッシュと同程度のエネルギーを消費する。また、アクセス時間は、段階型キャッシュにおけるキャッシュ・ヒット時間と同じ（つまり、従来型キャッシュ・アクセス時間の 2 倍）になる。したがって、WP キャッシュの有効性は、ウェイ予測ヒット率に大きく依存する。メモリ参照の時間的/空間的局所性を有効に活用するため、ウェイ予測には MRU(Most Recently Used) アルゴリズムを採用した。つまり、各セットにおいて、最も近い過去に参照されたウェイが見込みウェイとなる。

3.2.3 内部構成

WP キャッシュの内部構成を図 4 に示す。ウェイ数は 4 を仮定する。WP キャッシュは、従来型 SA キャッシュに対し、主に以下のハードウェア機構を追加することで実装できる。

- ウェイ予測フラグ (Way-Prediction Flag) : 各セッ

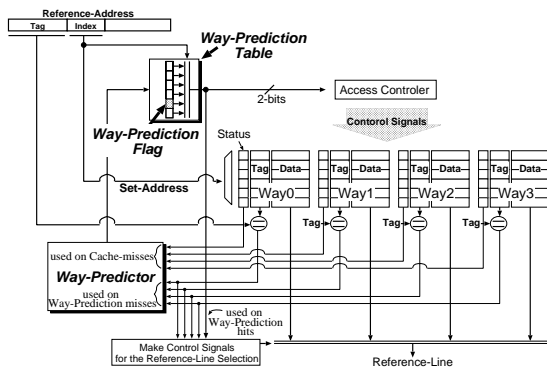


図4 WP キャッシュの内部構成

ト毎に設ける2ビットで、4ウェイの中から1つの見込みウェイを指定する。各ウェイ予測フラグは、ウェイ予測テーブル(Way-Prediction Table)に格納される。

- ウェイ予測回路(Way-Predictor)：MRU ウェイ予測アルゴリズムに従って、参照セットに対応するウェイ予測フラグを更新する。

ウェイ予測ミスの場合、ウェイ予測回路は、タグ比較結果に基づいてウェイ予測フラグの値を更新する。一方、キャッシュ・ミスの場合には、ライン・リプレースの対象となるウェイを見込みウェイとし、ウェイ予測フラグの値を更新する。なお、本稿では、キャッシュ・アクセス開始前に、検索対象となるセットに対応したウェイ予測フラグを読み出し可能とする。WP キャッシュの動作の詳細や、ウェイ予測テーブル・アクセスによって生じる性能ペナルティの評価等に関しては、文献⁵⁾を参照されたい。

4. ラインサイズの可変化による高性能/低消費エネルギー化

本章では、第2.4節で示した問題1~3を全て解決する手法として、動的可変ラインサイズ・キャッシュ(Dynamically Variable Line-Size Cache: D-VLS キャッシュ)について議論する⁶⁾⁷⁾。なお、D-VLS キャッシュの有効性に関する評価は第5章で行う。

4.1 基本概念

第2.1で述べたように、ラインサイズを拡大することで、DRAM/ロジック混載LSIの特徴である高オンチップ・メモリバンド巾を活用できる。しかしながら、従来型キャッシュは固定のラインサイズを有するため、SA方式を採用してキャッシュ・コンフリクトの発生を回避する必要がある。この場合、第2.4節で示した問題1~3を必ずしも解決できるとは限らない。これらの問題を全て解決し、更なる高性能化/低消費エネルギー

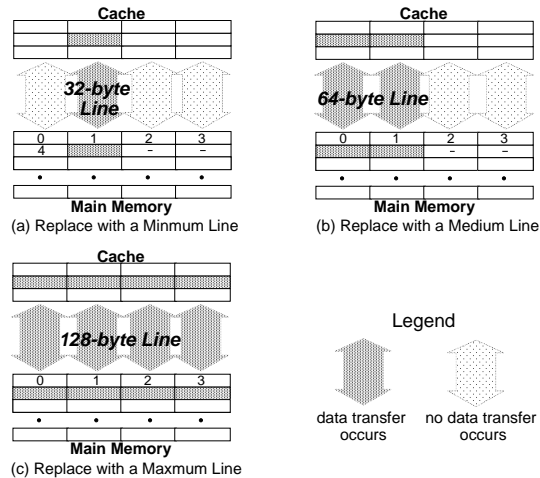


図5 D-VLS キャッシュの基本概念

ギー化を実現するためには、

- ラインサイズの拡大によるプリフェッチ効果の積極活用、
- アクセス時間オーバーヘッドを伴わないキャッシュ・コンフリクトの回避、
- ラインサイズ拡大に伴う無駄な主記憶アクセス消費エネルギーの削減、

を同時に達成する必要がある。D-VLS キャッシュは、空間的局所性の度合いに応じて動的にラインサイズを変更することで、これら全ての要求を同時に満足する。D-VLS キャッシュの基本概念を図5に示す。D-VLS キャッシュにおいて、オンチップ化されたSRAMアレイ(キャッシュ)およびDRAMアレイ(主記憶)は、複数のSRAMサブアレイおよびDRAMサブアレイにそれぞれ分割される。また、キャッシュ-主記憶間のデータ転送は、対応するSRAMサブアレイ-DRAMサブアレイ間でのみ行われる。図5に示す構成では、SRAMサブアレイ-DRAMサブアレイの組が4組あり、D-VLS キャッシュは以下に示す3種類のラインサイズを選択可能である。

- 最小ライン:図5(a)で示すように、1組のSRAM-DRAMサブアレイがリプレース対象となる。
- 中間ライン:図5(b)で示すように、2組の連続したSRAM-DRAMサブアレイがリプレース対象となる。
- 最大ライン:図5(c)で示すように、全てのSRAM-DRAMサブアレイがリプレース対象となる。

たとえば、各サブアレイが32バイト語長の場合、選択可能なラインサイズは、32バイト、64バイト、および、128バイトとなる。図5に示したダイレクト・マツ

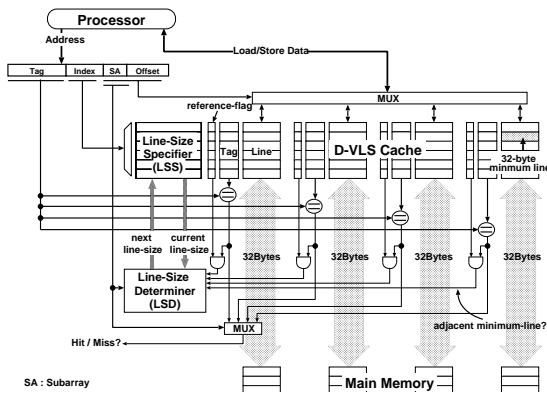


図6 ダイレクト・マップD-VLS キャッシュの内部構成

ブ (DM) 方式の D-VLS キャッシュは、固定 32 バイト・ラインを有する従来型の 4 ウェイ SA キャッシュと良く似た構成である。しかしながら、従来型 4 ウェイ SA キャッシュでは、あるラインが格納され得るキャッシュ内ロケーションが 4ヶ所存在するのに対し、DM 方式の D-VLS キャッシュでは唯一に決定される。つまり、図 5 の D-VLS キャッシュでは、タグ比較結果とは独立に、メモリ参照アドレスを用いて直接サブアレイ選択を行えるため、DM 方式の高速アクセスを維持できる。

D-VLS キャッシュにおけるラインサイズは、過去のメモリ参照パターンに基づき動的に変更される。高い空間的局所性が観測された場合には、積極的にラインサイズを拡大し、プリフェッチ効果によるヒット率の向上を達成する。一方、低い空間的局所性が観測された場合には、ラインサイズを縮小してキャッシュ・コンフリクトの発生を回避する。これにより、キャッシュ・アクセス時間オーバーヘッドを伴うことなく、コンフリクト・ミス削減できる(第 2.4 節における問題点 1 と 2 を解決)。また、最小ラインや中間ラインでのリプレイスが発生した時、リプレイス対象ラインに対応した DRAM サブアレイのみを選択的に活性化させることで、主記憶アクセスにおける消費エネルギーを削減可能となる(第 2.4 節における問題点 3 を解決)。

4.2 内部構成と動作

32 バイト、64 バイト、および、128 バイトのラインサイズを有する DM 方式 D-VLS キャッシュの構成を図 6 に示す。D-VLS キャッシュは、動的に変換ラインサイズを実現するために以下のハードウェア機構を要する。

- 参照フラグ (reference-flag) : 各 32 バイト最小ラインが有する 1 ビットのフラグであり、当該最小ラインがキャッシュにリフィルされた後、少く

とも 1 度はプロセッサにより参照されたか否かを示す。

- ラインサイズ指定フラグ (Line Size Specifier:LSS) : リプレイス時のラインサイズを指定するフラグであり、各セット (同一インデックスを有する 4 個の 32 バイト最小ライン) 毎に設ける。
- ラインサイズ決定機構 (Line Size Determiner:LSD) : キャッシュ・ヒット/ミスに関わらず、プロセッサからのメモリ参照が発生した時、ラインサイズ決定アルゴリズムに従って LSS の値を更新する。

キャッシュ・アクセス時、検索対象セットに対応する LSS を読み出す。同時に、全ての SRAM サブアレイから参照フラグ、タグ、および、データ (最小ライン) を読み出す。参照データの選択は、アドレス中のサブアレイ・フィールドおよびオフセット・フィールドを用いて、タグ比較結果とは独立に行う。また、サブアレイ・フィールドによりタグ比較結果を選択し、キャッシュ・ヒットもしくはミスプロセッサに報告する。キャッシュ・ミスが発生した際には、読み出された LSS が指定するラインサイズに従ってライン・リプレイスを行う。

ラインサイズ決定機構 (LSD) は、各サブアレイから読み出された参照フラグ、および、全てのタグ比較結果に基づき、「隣接最小ライン」を検出する。ここで、隣接最小ラインとは、検索対象セットにおいて、主記憶上のアドレス・ロケーションが参照データと同一のメモリ・セクタ (主記憶上の連続した領域) 上にあり、かつ、リフィル後少なくとも 1 度はプロセッサによって参照された最小ラインである。あるサブアレイにおいて、タグ比較結果が一致し、かつ、参照フラグが 1 である場合、当該最小ラインは隣接最小ラインとなる。連続した隣接最小ラインが多く検出された場合、LSD はメモリ参照の空間的局所性が高いと判断し、LSS の値を最小ラインから中間ラインへ、もしくは、中間ラインから最大ラインへと拡大方向へ遷移させる。一方、隣接最小ラインが少ない場合には、空間的局所性が低いと判断し、LSS の値を最大ラインから中間ラインへ、もしくは、中間ラインから最小ラインへと縮小方向に遷移させる。なお、D-VLS キャッシュの詳細な内部構成とその動作、ならびに、ラインサイズ決定アルゴリズムに関しては、文献⁶⁾を参照されたい。

図 6 の場合、検出される隣接最小ラインは最大で 3 個、最小で 0 個 (参照データを含む最小ラインは除く) である。

表3 データ・キャッシュの評価モデル

	従来型 キャッシュ				段階型 キャッシュ	ウェイ予測型 キャッシュ	可変ラインサイズ キャッシュ
	DM	2SA	4SA	DM32K	P	WP	DVLS
キャッシュ・サイズ	16KB	16KB	16KB	32KB	16KB	16KB	16KB
ウェイ数	1	2	4	1	4	4	1
ラインサイズ	128B 固定	128B 固定	128B 固定	128B 固定	128B 固定	128B 固定	32B, 64B, 128B 可変

5. 総合評価

本章では、第3章で示したウェイ予測キャッシュ(WP キャッシュ) および第4章で示した動的可変ラインサイズ・キャッシュ(D-VLS キャッシュ)に関する定量的評価を行い、それらの有効性について議論する。

5.1 実験環境

C言語を用いてキャッシュ・シミュレータを作成し、プログラム実行時のキャッシュ・ミス率、WP キャッシュにおけるウェイ予測ヒット率、ならびに、D-VLS キャッシュにおけるライン・リプレイス時の平均ラインサイズを測定した。本シミュレータは、QPT¹⁸⁾によって採取したアドレス・トレースを入力とする。また、ベンチマーク・プログラムとしては、SPEC CPU92から3個のプログラムを、SPEC CPU95から10個のプログラムを用いた¹⁷⁾。表3に評価対象となるデータ・キャッシュの評価モデルを示す。

5.2 実験結果

各ベンチマーク・プログラムにおけるシミュレーション結果を表4に示す。CMRはキャッシュ・ミス率を、WPHRはWPキャッシュにおけるウェイ予測ヒット率を、また、Ave.LSはD-VLSキャッシュにおけるライン・リプレイス当りの平均ラインサイズを表す。

WPキャッシュでは、多くのプログラムにおいて80%以上のウェイ予測ヒット率を達成しており、大きな低消費エネルギー効果を期待できる。一方、D-VLSキャッシュでは、従来型2ウェイまたは4ウェイSAキャッシュ(2SAまたは4SA)ほどの高ヒット率は達成できなかった。しかしながら、052.alvinnや134.perlなど複数のプログラムにおいては、2倍のキャッシュ・サイズを有する従来型DMキャッシュ(DM32K)と同程度もしくはそれ以上のヒット率向上を実現した。また、リプレイス当りの平均ラインサイズは、最小で約35バイト(026.compress)、最大で約90バイト

(052.alvinn)と様々であった。

5.3 キャッシュのアクセス時間と消費エネルギー

各キャッシュのアクセス時間(T_{Cache})を求めるため、CACTIモデルを使用した¹⁵⁾¹⁶⁾。また、文献⁸⁾を参考にして、キャッシュ・アクセス当りの消費エネルギー(E_{Cache})を求めた。DMのアクセス時間およびアクセス消費エネルギーを、それぞれ、 T_{unit} および E_{unit} とした際の結果を表5に示す。

具体的には、0.18umプロセスを想定し、CACTI 2.0により従来型キャッシュのアクセス時間を求めた。また、段階型キャッシュおよびWPキャッシュのアクセス時間に関して、クロック・サイクル時間は従来型4ウェイSAキャッシュ(4SA)のアクセス時間に等しいと仮定した。さらに、D-VLSキャッシュに関して、そのアクセス時間は、16Kバイト従来型DMキャッシュ(DM)のアクセス時間に等しいと仮定した。D-VLSキャッシュにおいて、可変ラインサイズを実現するハードウェア機構は、キャッシュ・クリティカル・パス上に存在しないためである。なお、主記憶アクセス時間(T_{Main})は、16Kバイト従来型DMキャッシュ(DM)におけるアクセス時間の10倍と仮定する。

一方、キャッシュ・アクセス当りの消費エネルギー(E_{Cache})に関して、各トランジスタにおけるソース/ドレイン容量、配線容量といった各種パラメータは文献⁹⁾を参考にした。ここで、 n ウェイSAキャッシュは、 n 個のSRAMサブアレイで構成されると仮定する(つまり、従来型DM方式のSRAMアレイを n 分割)。この場合、DMキャッシュと比較して、ビット線の長さは $1/n$ 倍、ワード線の長さは n 倍になる(つまり、総ビット線数が n 倍)。そのため、ビット線当りの負荷容量は削減されるが、ビット線プリチャージ回路やセンスアンプ回路の増加等に伴う消費エネルギー・オーバヘッドが生じる。これに対し、段階型キャッシュにおけるキャッシュ・ヒットの場合や、WPキャッシュにおけるウェイ予測ヒットの場合には、1個のウェイに対してのみデータ・アクセスが発生する。そのため、活性化されるビット線の長さは $1/n$ 倍、ワード線の長さはほぼ1倍(つまり、活性化される総ビット線数はDM方式の場合とほぼ同じ)になる。その結果、従

各プログラムは、Ultra SPARC プロセッサでの実行を想定し、GNU CC(-O2 オプションを指定)を用いてコンパイルした。また、SPEC CPU92のプログラムに関してはref入力を、SPEC CPU95の整数プログラムおよび浮動小数点プログラムに関しては、それぞれ、train入力およびtest入力を使用した。

表4 各ベンチマーク・プログラムにおけるシミュレーション結果

Benchmarks	DM	2SA	4SA	DM32K	P	WP		DVLS	
	CMR	CMR	CMR	CMR	CMR	WPHR	CMR	CMR	Ave.LS [B]
026.compress	0.1871	0.1755	0.1732	0.1634	0.1732	0.7619	0.1732	0.1724	34.69
052.alvinn	0.0224	0.0087	0.0080	0.0175	0.0080	0.9311	0.0080	0.0166	90.22
072.sc	0.0371	0.0285	0.0263	0.0276	0.0263	0.8623	0.0263	0.0465	58.32
099.go	0.1024	0.0695	0.0302	0.0541	0.0302	0.6986	0.0302	0.0638	42.82
124.m88ksim	0.0202	0.0045	0.0028	0.0068	0.0028	0.9250	0.0028	0.0153	50.83
126.gcc	0.0611	0.0344	0.0254	0.0349	0.0254	0.8442	0.0254	0.0526	48.76
130.li	0.0341	0.0203	0.0182	0.0226	0.0182	0.9181	0.0182	0.0358	49.63
132.jpeg	0.0244	0.0048	0.0036	0.0068	0.0036	0.8793	0.0036	0.0175	58.43
134.perl	0.0542	0.0230	0.0105	0.0295	0.0105	0.8809	0.0105	0.0286	63.46
147.vortex	0.0505	0.0292	0.0195	0.0307	0.0195	0.8275	0.0195	0.0374	42.11
101.tomcatv	0.0633	0.0182	0.0062	0.0546	0.0062	0.8063	0.0062	0.0578	43.73
103.su2cor	0.2600	0.0840	0.0242	0.2396	0.0242	0.6571	0.0242	0.0758	53.01
104.hydro2d	0.0481	0.0217	0.0179	0.0259	0.0179	0.8682	0.0179	0.0295	89.34

表5 各キャッシュにおけるアクセス時間と消費エネルギー

Parameters	DM	2SA	4SA	DM32K	P		WP		DVLS
					CH	CM	WPH	WPM or CM	
$T_{Cache}[Tunit]$	1.000	1.470	1.883	1.195	3.766	1.883	1.883	3.766	1.000
$T_{Main}[Tunit]$	10.000								
$E_{Cache}[Eunit]$	1.00	1.160	1.480	1.838	0.392	0.029	0.370	1.480	1.090
$E_{Main}[Eunit]$	$10.000 \times (AverageLineSize/128bytes)$								

CH:Cache Hit, CM:Cache Miss, WPH:Way-Prediction Hit, WPM:Way-Prediction Miss

来型 DM キャッシュ(DM) よりも低いアクセス消費エネルギーを実現できる。これに対し、D-VLS キャッシュでは、活性化されるビット線の長さ、ならびに、ワード線の長さは、従来型 DM キャッシュの場合と同じである。ただし、第 4.2 節で述べたように、D-VLS キャッシュでは、DM 方式であるにも関わらず、ラインサイズ決定のために全サブアレイからタグが読み出される。その結果、従来型 DM キャッシュと比較して若干の消費エネルギー・オーバーヘッドが生じる。なお、文献³⁾を参考にして、主記憶アクセス当りの消費エネルギーは、16K バイト従来型 DM キャッシュ(DM) におけるアクセス消費エネルギーの 10 倍と仮定した。

5.4 メモリシステムの性能と消費エネルギー

第 5.2 節および第 5.3 節の実験結果に基づき、各プログラムにおける平均メモリアクセス時間 ($AMAT$)、ならびに、平均メモリアクセス消費エネルギー ($AMAE$) を求めた。その結果を図 7 に示す。また、性能と消費エネルギーを同時に評価するため、図 8 に示すように、各プログラムにおける ED 積 ($AMAT \times AMAE$) を計算した。なお、図 8 の各プログラムにおいて、全ての結果は従来型 DM キャッシュ(DM) の結果に正規化している。

まず、従来型 4 ウェイ SA キャッシュ(4SA) を基準にし、WP キャッシュに関する考察を行う。多くのプログラムにおいて、段階型キャッシュでは約 75%、WP

キャッシュでは約 65~70%のキャッシュ・アクセス消費エネルギー (E_{Cache}) を削減している。しかしながら、キャッシュ・ヒット率は 97%以上と高いため、キャッシュ・ヒット時間を犠牲にする段階型キャッシュでは、そのアクセス時間がほぼ 2 倍となる。これに対し、WP キャッシュでは、ウェイ予測が正しい場合にはアクセス時間オーバーヘッドを伴わない。その結果、103.su2cor を除く全てのプログラムにおいて、段階型キャッシュより高い ED 積削減率であった。

次に、従来型 DM キャッシュ(DM) を基準にし、D-VLS キャッシュに関する考察を行う。従来型キャッシュは 128 バイトの固定ラインサイズを有するため、主記憶アクセスでの消費エネルギーはキャッシュ・ミス率にのみ依存する。従来型 4 ウェイ SA キャッシュ(4SA) は、高いヒット率を達成する事で主記憶アクセス消費エネルギーを平均約 60%削減した。しかしながら、キャッシュ・アクセス時間オーバーヘッドが大きいため、性能の向上は実現できていない。これに対し、D-VLS キャッシュにおける主記憶アクセス消費エネルギーは、キャッシュ・ミス率とラインサイズの両方に依存する。従来型キャッシュでは平均ラインサイズが 128 バイトであるのに対し、D-VLS キャッシュのそれは約 54.5 バイトであった。これにより、主記憶アクセス当たりの消費エネルギーを平均約 68%削減した。また、D-VLS キャッシュは、従来型 SA キャッシュとは異なり、DM

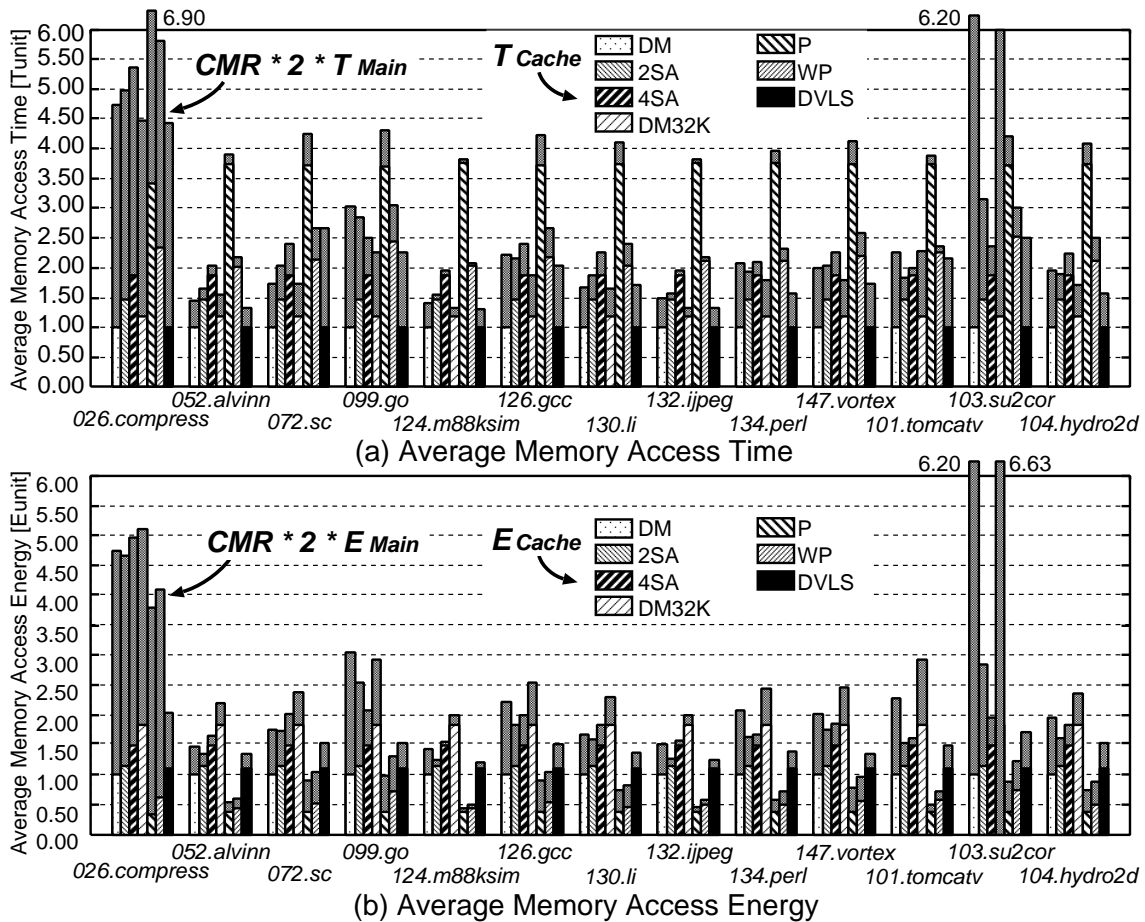


図7 平均メモリアクセス時間と平均メモリアクセス消費エネルギー

方式の高速アクセスを維持でき、かつ、キャッシュ・アクセス当りの消費エネルギー・オーバーヘッドも極めて小さい。その結果、多くのプログラムにおいて高いED積削減率を達成できた。

次に、WP キャッシュとD-VLS キャッシュを比較する。WP キャッシュは、D-VLS キャッシュと比較して、キャッシュ・アクセス当りの消費エネルギーが極めて小さい。また、従来型4ウェイSAキャッシュ(4SA)と同様に高いヒット率を達成できるため、主記憶アクセスにおける消費エネルギーを削減できる。その結果、平均メモリアクセス消費エネルギー(AMAE)に関しては、026.compressを除く全てのプログラムにおいて、D-VLS キャッシュより良い結果となった。一方、前述したように、D-VLS キャッシュは従来型DMキャッシュの高速アクセスを維持できる。これに対し、WPキャッシュのアクセス時間はウェイ予測が正しい場合で T_{4SA} (従来型4ウェイSAキャッシュのアクセス時間)、予測が誤りであった場合には $2 \times T_{4SA}$ となる。そ

の結果、平均メモリアクセス時間(AMAT)に関しては、072.scを除く全てのプログラムにおいてD-VLSキャッシュの方が高い向上率を達成した。これらの結果は、平均メモリアクセス時間/消費エネルギーにおいて、キャッシュ・アクセス時間/消費エネルギーの占める割合が大きいためである。本評価では、主記憶アクセス時間/消費エネルギーは、それぞれ、DMのアクセス時間ならびに消費エネルギーの10倍と仮定した。実際、これらの比はキャッシュ・サイズ、オンチップ主記憶サイズ、プロセス・テクノロジー等に大きく依存する。性能に関しては、主記憶アクセス速度が更に遅い場合、高いキャッシュ・ヒット率を達成できるWPキャッシュの方が有効であると考えられる。一方、消費エネルギーに関しては、主記憶アクセス消費エネルギーが更に大きい場合、オンチップDRAMのサブバンク効果を活用できるD-VLSキャッシュの方が大きな削減率を達成できると考察する。

最後に、WPキャッシュとD-VLSキャッシュの組

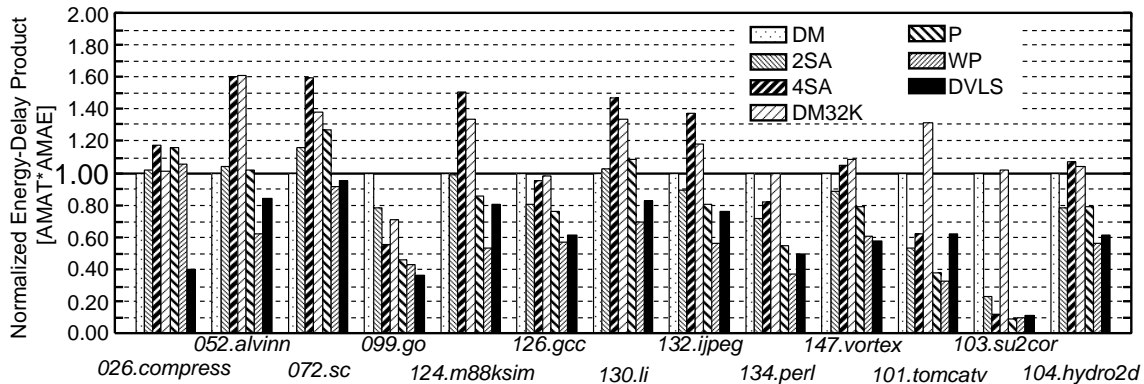


図8 ED積 (AMAT × AMAE)

み合わせについて議論する。本稿では、DM方式のD-VLSキャッシュと、SA方式のWPキャッシュを比較した。D-VLSキャッシュに関しては、SA方式を採用する事も可能である。連想度が n の場合、D-VLSキャッシュは、 n ウェイSAキャッシュのアクセス時間を維持しつつ、より高いヒット率を達成する。このような場合、D-VLSキャッシュとWPキャッシュを組み合わせる事で、より高性能/低消費エネルギーなオンチップ・メモリシステムを構築できる。

6. おわりに

本稿では、DRAM/ロジック混載LSIの潜在能力を引き出すため、著者らがこれまで続けてきた研究の成果として、DRAM/ロジック混載LSI向けキャッシュ・アーキテクチャを紹介した。DRAM/ロジック混載LSIは、21世紀のコンピュータ/電子機器システムにおいて核となるデバイスである。CPU-主記憶間チップ境界線の削除は、分チップ構成を基本とする従来システムの単純な1チップ化だけではなく、今までにない新たなCPUアーキテクチャやメモリ・アーキテクチャの実現を可能にする。今後、「CPUと主記憶の混載」による利点を最大限活用し、より高性能かつ低消費電力な計算機システムを実現するためには、メモリ・アーキテクチャ技術やCPUアーキテクチャ技術だけでなく、最適化コンパイラに代表されるシステム・ソフトウェア技術、プロセス技術、回路技術、さらには、設計最適化技術など、様々な技術の融合が必要であると考える。

謝辞

日頃から御討論頂く、九州大学 大学院システム情報科学研究院 安浦寛人教授、岩井原瑞穂 助教授、PPRAMグループ関係者各位、ならびに、研究室の諸氏に感謝します。なお、本研究は一部、文部省科学研

究費補助金基盤研究(A)(2)展開研究「システムLSI向きカスタム化可能IPコアのアーキテクチャおよび設計支援技術の開発」(課題番号:12358002)、展開研究「メモリ/ロジック混載技術に基づく大規模集積回路システム・アーキテクチャの研究開発」(課題番号:09358005)、ならびに、一般研究「スケーラブル・システムLSIアーキテクチャの設計手法に関する研究」(課題番号:11308011)による。

参考文献

- 1) Bahar, R. I., Albera, G., and Manne, S., "Power and Performance Tradeoffs using Various Caching Strategies," *Proc. of the 1998 International Symposium on Low Power Electronics and Design*, pp.64-69, Aug. 1998.
- 2) Zhang, C., Zhand, X., and Yan, Y., "Two Fast and High-Associativity Caches Schemes," *IEEE Micro*, vol.17, num.5, pp40-49, Sep./Oct. 1997.
- 3) Fromm, R., et al., "The Energy Efficiency of IRAM Architectures," *Proc. of the 24rd Annual International Symposium on Computer Architecture*, pp.327-337, May 1997.
- 4) Hasegawa, A., Kawasaki, I., Yamada, K., Yoshioka, S., Kawasaki, S., and Biswas, P., "SH3: High Code Density, Low Power," *IEEE Micro*, pp.11-19, Dec. 1995.
- 5) Inoue, K., Ishihara, T., and Murakami, K., "A High-Performance and Low-Power Cache Architecture with Speculative Way-Selection," *IEICE Trans. on Electronics*, vol. E83-C, no. 2, pp.186-194, Feb. 2000.
- 6) Inoue, K., Kai, K., and Murakami, K., "Dynamically Variable Line-Size Cache Architecture for Merged DRAM/Logic LSIs," *IEICE Trans. on Electronics*, vol. E83-D, no. 5, pp.1048-1057, May 2000.

- 7) Inoue, K., Kai, K., and Murakami, K., "A High-Performance/Low-Power On-chip Memory-Path Architecture with Variable Cache-Line Size," *IEICE Trans. on Electronics*, vol. ???-C, no. ?, pp.??-??, Nov. 2000. (to be published)
- 8) Kamble, M. B., and Ghose, K., "Analytical Energy Dissipation Models For Low Power Caches," *Proc. of the 1997 International Symposium on Low Power Electronics and Design*, pp.143-148, Aug. 1997.
- 9) Kamble, M. B., and Ghose, K., "Energy-Efficiency of VLSI Caches: A Comparative Study," *Proc. of the 10th International Conference on VLSI Design*, pp.261-267, 1997.
- 10) Murakami, K., Shirakawa, S., and Miyajima, H., "Parallel Processing RAM Chip with 256Mb DRAM and Quad Processors," *1997 ISSCC Digest of Technical Papers*, pp.228-229, Feb. 1997.
- 11) Hennessy, J. L., and Patterson, D. A., "Computer Architecture : A Quantitative Approach," Morgan Kaufmann Publishers, Inc., 1990.
- 12) Patterson, D., Anderson, T., Cardwell, N., Fromm, R., Keeton, K., Kozyrakis, C., Thomas, R., and Yelick, K., "Intelligent RAM (IRAM): Chips that remember and compute," *1997 ISSCC Digest of Technical Papers*, pp.224-225, Feb. 1997.
- 13) Saulsbury, A., Pong, F., and Nowatzky, A., "Missing the Memory Wall: The Case for Processor/Memory Integration," *Proc. of the 23rd Annual International Symposium on Computer Architecture*, pp.90-101, May 1996.
- 14) Su, C. L., and Despain, A. M., "Cache Design Trade-offs for Power and Performance Optimization: A Case Study," *Proc. of the 1995 International Symposium on Low Power Design*, pp.69-74, Apr. 1995.
- 15) Wilton, S. J. E., and Jouppi, N. P., "CACTI: An Enhanced Cache Access and Cycle Time Model," *IEEE Journal of Solid-State Circuits*, vol.31, no.5, pp.677-688, May. 1996.
- 16) "CACTI," <http://www.research.compaq.com/wrl/people/jouppi/CACTI.html>
- 17) "SPEC (Standard Performance Evaluation Corporation)," <http://www.specbench.org/>.
- 18) "WARTS: Wisconsin Architectural Research Tool Set," <http://www.cs.wisc.edu/larus/warts.html>.

(平成 ? 年 ? 月 ? 日受付)

(平成 ? 年 ? 月 ? 日採録)



井上 弘士 (学生会員)

昭和 46 年生 . 平成 8 年九州工業
大学大学院情報工学研究科修士課程
修了 . 同年横河電機 (株) 入社 . 平成
9 年より (財) 九州システム情報技術
研究所研究助手 . 平成 11 年の 1 年

間 Halo LSI Design & Device Technology, Inc. に
て訪問研究員としてフラッシュ・メモリの開発に従事 .
現在九州大学大学院システム情報科学研究科情報工学
専攻博士後期課程 3 年 . 高性能/低消費電力メモリ・
アーキテクチャに関する研究に従事 .



石原 亨 (正会員)

昭和 48 年生 . 平成 12 年九州大学
大学院システム情報科学研究科情報
工学専攻博士課程修了 . 同年東京大
学大規模集積システム設計教育研究
センター 助手に任官 . 平成 9 年か
ら 12 年まで日本学術振興会特別研究員現在東京大学
大規模集積システム設計教育研究センターにて低電力
システム LSI の研究に従事 . 工学博士 . 電子情報通信
学会 , 情報処理学会 , IEEE-CS 各会員 .



甲斐 康司 (正会員)

昭和 41 年生 . 平成 3 年九州大学
大学院総合理工学研究科情報システ
ム学専攻修士課程修了 . 同年松下電
器産業 (株) 入社 . 平成 8 年から 12
年にかけて (財) 九州システム情報技
術研究所に研究員として出向 . 現在松下電器産業 (株)
半導体開発本部にて携帯端末向け LSI の開発に従事 .
情報処理学会 , IEEE-CS 各会員 .



村上和彰 (正会員)

昭和 35 年生 . 昭和 59 年京都大学
大学院工学研究科情報工学専攻修士
課程修了 . 同年富士通 (株) 入社 . 汎
用大型計算機の研究開発に従事 . 昭
和 62 年九州大学助手 . 平成 6 年九
州大学助教授 . 現在九州大学大学院システム情報科学
研究院情報理学部門教授 . 計算機アーキテクチャ , 並
列処理 , システム LSI 設計技術 , 計算科学専用計算機
アーキテクチャに関する研究に従事 . 工学博士 . 平成
3 年情報処理学会研究賞 , 平成 4 年情報処理学会論文
賞 , 平成 9 年坂井記念特別賞受賞 .