

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,†2} 石原 亨^{†3}
甲斐 康司^{†2} 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・メモリバンド幅を有効に活用し、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,†2} TOHRU ISHIHARA,^{†3} KOJI KAI^{†2}
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・ポトルネックの顕在化によるメモリバンド幅の限界、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ともない、3) 高性能化と低消費エネルギー化の両立、に対する要求がきわ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・ポ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幅可能となり、メモリバンド幅が劇的に向上する。また、オフチップ・アクセス回数がきわ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

†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 財団法人九州システム情報技術研究所
Institute of Systems & Information Technologies/
KYUSHU

†3 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

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

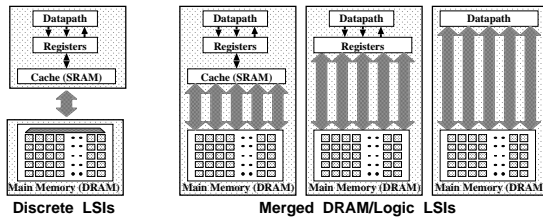


図1 オンチップ/オフチップ・メモリバス・アーキテクチャ
Fig. 1 On-chip/off-chip memory-path architectures.

(System-On-a-Chip)時代において核となる要素技術である。今後、さらなる高性能化/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。

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とりを行う 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% ときわめて低い。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⁾
- 動的変換ラインサイズ・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、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・メモリバンド幅を活用する際の利点と欠点を整理する。そして、さらなる高性能化/低消費エネルギー化を実現するための手段として、3 章ではウェイ予測キャッシュ、4 章では動的

変換ラインサイズ・キャッシュについて述べる。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 6 章でまとめる。

2. 高オンチップ・メモリバンド幅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ー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)、 $LineSize$ はリプレースされるキャッシュ・ラインのサイズ、 $BandWidth$ はキャッシュ-主記憶間バンド幅を表す。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、ラインリプレー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・ポトルネックによりメモリバンド幅が制限される。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)、それにともないミス・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・ポトルネックが解消されるため、高オンチップ・メモリバンド幅を実現できる (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とまなうことなくラインサイズを拡大可能となる

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

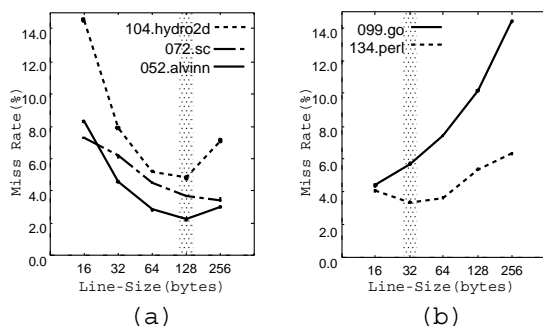


図2 16 KBダイレクトマップ・キャッシュにおけるラインサイズとミス率の関係

Fig. 2 Cache-miss rate versus cache-line size.

($LineSize \leq Bandwidth$ の条件下では, $\frac{LineSize}{BandWidth}$ が一定).

2.2 性能に関する問題点

2.1 節において, 高オンチップ・メモリバンド幅活用の利点は「ミス・ペナルティの増加をとま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, それにともないキャッシュ・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い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・メモリバンド幅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・メモリ面積がきわめて大きくなる. 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・アクセスあ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・アクセスあ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} に与える影響がきわ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とも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・アクセスあたりの消費エネルギーを E_{Cache_cell} で近似する.

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ことで, 高オンチップ・メモリバンド幅を活用可能となる. しかしながら, 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, それにともない拡幅された 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, さら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・アクセスごとに行われる. 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 カ所(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 個のウェイをすべて同時に活性化し、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かかわらず、すべ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、2.2 節および 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ともな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 方式を採用することでコンフリクト・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 方式がつね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ともな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ともない、拡幅された 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、主記憶アクセスあ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ともない、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かかわらず、従来型 SA キャッシュではすべ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、2.4 節で述べた問題 4 を解決する手法として、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、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、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 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・アクセスあた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 個あ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・アクセスあたりに読み出されるタグおよびラインの個数である。

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・アクセスごとに 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

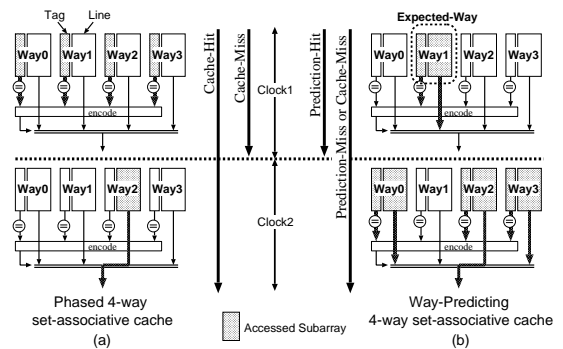


図 3 段階型キャッシュとウェイ予測型キャッシュ

Fig. 3 Operations of the phased cache and the way-predicting cache.

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

Table 1 Cache-access time and energy of the conventional cache and the phased cache.

| キャッシュ | E_{Cache} | T_{Cache} (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

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。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

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とまなうことなく、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。

表 2 WPキャッシュのアクセス時間と消費エネルギー

Table 2 Cache-access time and energy of the way-predicting cache.

| ケース | E_{Cache} | T_{Cache} (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

- 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、リプレイスが発生する場合。

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すべ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・リプレイスを実行する。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、そのときのアクセス時間は、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

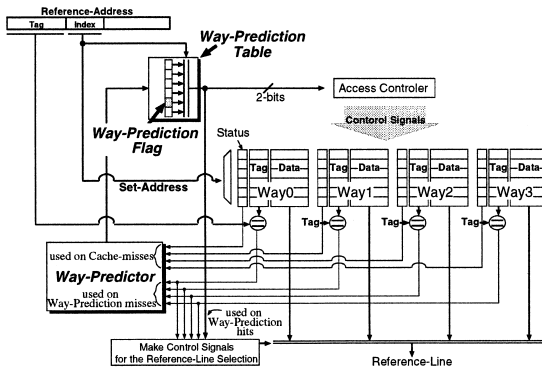


図4 WP キャッシュの内部構成

Fig. 4 Block diagram of a way-predicting cache.

(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): 各セットごとに設ける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・リプレイ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などに関しては、文献5)を参照されたい。

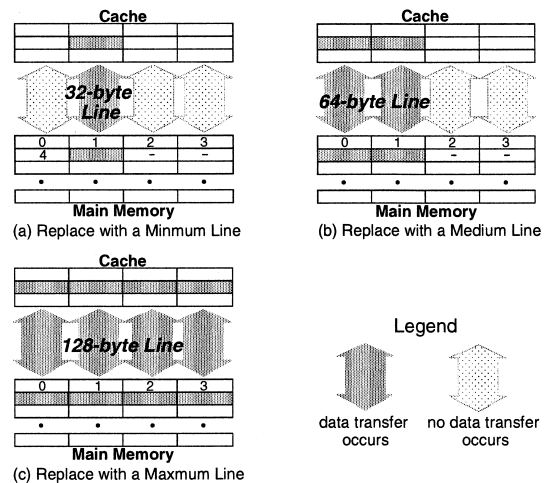


図5 D-VLS キャッシュの基本概念

Fig. 5 Concept of the dynamically variable line-size cache.

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、2.4節で示した問題1~3をすべて解決する手法として、動的に変化するラインサイズ・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5章で行う。

4.1 基本概念

2.1節で述べたように、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・メモリバンド幅を活用できる。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、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。これらの問題をすべて解決し、さらなる高性能化/低消費エネル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ともしないキャッシュ・コンフリクトの回避、
- ラインサイズ拡大にともな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、これらすべ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イス対象となる。
- 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イス対象となる。
- 最大ライン: 図 5 (c) で示すように、すべての SRAM-DRAM サブアレイがリプレイス対象となる。

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・マップ (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 キャッシュとよく似た構成である。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うるキャッシュ内ロケーションが 4 カ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とまなうことなく、コンフリクト・ミス削減できる(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とき、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(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

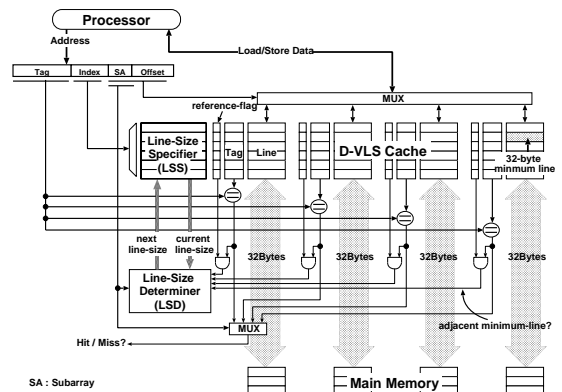


図 6 ダイレクト・マップ D-VLS キャッシュの内部構成
Fig. 6 Block diagram of a direct-mapped D-VLS cache.

図 6 に示す。D-VLS キャッシュは、動的変可変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) ごとに設ける。
- ラインサイズ決定機構 (Line Size Determiner: LSD): キャッシュ・ヒット/ミスにかかわらず、プロセッサからのメモリ参照が発生したとき、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、すべての 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、すべ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、

表3 データ・キャッシュの評価モデル
Table 3 Characteristics of cache models.

| | 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 | 16 KB | 16 KB | 16 KB | 32 KB | 16 KB | 16 KB | 16 KB |
| ウェイ数 | 1 | 2 | 4 | 1 | 4 | 4 | 1 |
| ラインサイズ | 128 B 固定 | 128 B 固定 | 128 B 固定 | 128 B 固定 | 128 B 固定 | 128 B 固定 | 32 B, 64 B, 128 B 可変 |

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
Table 4 Simulation results for benchmark programs.

| 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 |

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。

5. 総合評価

本章では、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4章で示した動的変換ラインサイ

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・リプレイスあたりの平均ラインサイズを表す。

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。

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 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
Table 5 Cache-access time and energy.

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

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

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、リプレイスあた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・アクセスあた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、0.18 μ m 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、16 K 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、16 K 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・アクセスあた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、ビット線あた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などにもなう消費エネルギー・オーバーヘッドが生じる。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、4.2 節で述べたように、D-VLS キャッシュでは、DM 方式であるにもかかわらず、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、主記憶アクセスあたりの消費エネルギーは、16 K 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

5.2 節および 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、すべての結果は従来型 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%、

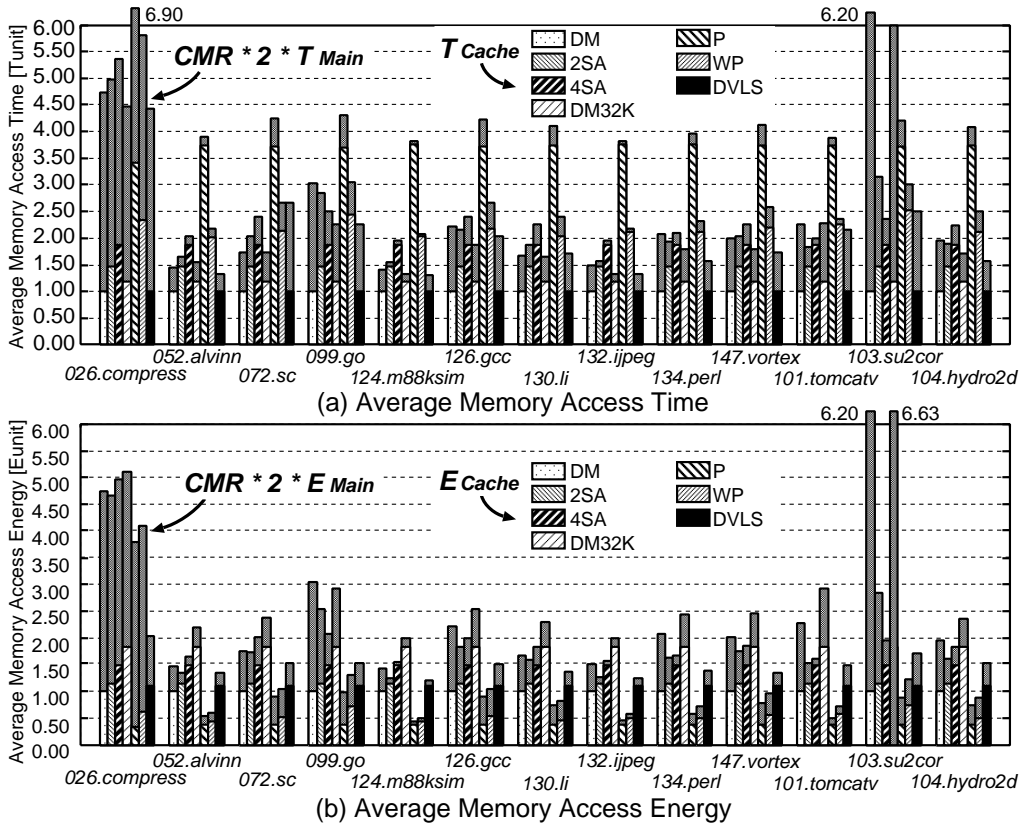


図7 平均メモリアクセス時間と平均メモリアクセス消費エネルギー
Fig. 7 Average memory-access time and energy.

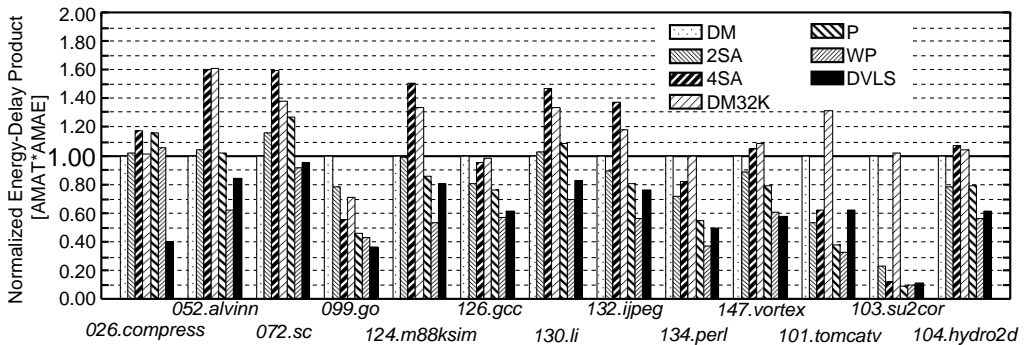


図8 ED積 ($AMAT \times AMAE$)
Fig. 8 Energy-delay product.

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とみなわない。その結

果、103.su2cor を除くすべ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こと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、主記憶アクセスあ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 方式の高速アクセスを維持でき、かつ、キャッシュ・アクセスあたりの消費エネルギー・オーバーヘッドもきわめて小さい。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・アクセスあたりの消費エネルギーがきわめて小さい。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 を除くすべ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 を除くすべ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・テクノロジーなどに大きく依存する。性能に関しては、主記憶アクセス速度がさらに遅い場合、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さらに大きい場合、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 キャッシュの組合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 方式を採用することも可能である。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ことで、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 日頃からご討論いただく、九州大学大学院システム情報科学研究所安浦寛人教授、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. 1998 Interna-*

- tional 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, No.5, pp.40–49 (1997).
 - 3) Fromm, R., et al.: The Energy Efficiency of IRAM Architectures, *Proc. 24th 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. 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. Electronics*, Vol.E83-D, No.5, pp.1048–1057 (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. Electronics*, Vol.E83-C, No.11, pp.1716–1723 (2000).
 - 8) Kamble, M.B. and Ghose, K.: Analytical Energy Dissipation Models For Low Power Caches, *Proc. 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. 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 (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. 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. 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 (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 12 年 7 月 4 日受付)

(平成 12 年 9 月 27 日採録)



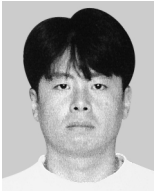
井上 弘士 (学生会員)

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, 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。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科情報理学部門教授。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。