九州大学学術情報リポジトリ
Kyushu University Institutional Repository

動的可変ラインサイズ・キャッシュ・アーキテクチャとその性能およびオンチップDRAMの消費エネルギーに関する評価

井上, 弘士
九州大学大学院システム情報科学研究科情報工学専攻

甲斐, 康司
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情報工学専攻

https://hdl.handle.net/2324/7345

出版情報：電子情報通信学会技術研究報告. ICD, 集積回路. 100 (5), pp.25-30, 2000-04-13. 電子情報通信学会
バージョン：
権利関係：

動的可変ラインサイズ・キャッシュ・アーキテクチャとその性能および オンチップ DRAM の消費エネルギーに関する評価

井上 弘士 †　　甲斐 康司 ‡　　村上 和彰 †

† 九州大学 大学院システム情報科学研究科 情報工学専攻

〒 816-8580 福岡県春日市春日公園 6-1

‡ (財) 九州システム情報技術研究所

〒 814-0001 福岡県早良区百道浜 2 丁目 1 番 22 号

E-mail : †*ppram@c.csce.kyushu-u.ac.jp*,‡*kai@k-isit.or.jp*

あらまし：我々は，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，動的可変ラインサイズ・キャッシュ (D-VLS キャッシュ) を提案している．D-VLS キャッシュは，高オンチップ・メモリバンド巾を活用し，プログラムの特性に応じて動的にラインサイズを変更する．これにより，ヒット率向上に伴うメモリシステムの高性能化，ならびに，ラインサイズに基づいたオンチップ DRAM の選択的活性化による低消費エネルギー化を同時に実現する事ができる．本稿では，D-VLS キャッシュを搭載したオンチップ・メモリパス・アーキテクチャに関して，その性能および消費エネルギーを評価する．14 個のベンチマーク・プログラムを用いて実験を行った結果，128 バイト固定ラインサイズ従来型キャッシュと比較して，32 バイト，64 バイト，および，128 バイトで変更可能なラインサイズを有する D-VLS キャッシュは，約 20%のメモリシステム性能向上を達成した．また，オンチップ DRAM アクセスにおける消費エネルギーを約 68%削減できた．

キーワード　DRAM/ロジック混載 LSI，キャッシュ，可変ラインサイズ，高メモリバンド巾

Performance and Energy Evaluation of a Dynamically Variable Line-Size Cache

Koji Inoue †　　Koji Kai ‡　　Kazuaki Murakami †

† Department of Computer Science and Communication Engineering, Kyushu University

6–1 Kasuga-koen, Kasuga, 816-8580 Japan

‡ Institutes of Systems and Information Technologies/Kyushu

2–1–22 Momochihama, Sawara-ku, Fukuoka, 814-0001 Japan

E-mail : †*ppram@c.csce.kyushu-u.ac.jp*,‡*kai@k-isit.or.jp*

Abstract : We have proposed the "dynamically variable line-size cache (*D-VLS cache*)" as a novel on-chip cache architecture for merged DRAM/logic LSIs. The D-VLS cache can make good use of the high on-chip memory bandwidth, and optimizes its cache-line size at run time according to memory-reference characteristics of programs. For an on-chip memory-path architecture having the D-VLS cache, the memory-system performance will be improved due to high cache-hit rate. In addition, the energy consumption for accessing to the on-chip main memory (DRAM) will also be reduced by activating only the on-chip DRAM subarray corresponding to the cache line to be replaced. This paper evaluates the performance/energy efficiency of an on-chip memory-path architecture with the D-VLS cache. In our simulation, it is observed that the performance improvement achieved by employing a D-VLS cache is about 20% while it reduces the energy consumption for accessing to the on-chip DRAM by 68%, compared to an on-chip memory-path architecture with a conventional cache having fixed line size.

key words　merged DRAM/logic LSI, cache, variable line-size, high memory bandwidth

1　はじめに

プロセッサ–主記憶間の性能差拡大を解決する有効な手段として，DRAM/ロジック混載 LSI の活用がある [5][6]．プロセッサと主記憶 (DRAM) を 1 チップ化し，オンチップ・メモリバスを拡幅する事で，プロセッサ–主記憶間データ転送能力は劇的に向上する．

オンチップ・キャッシュを搭載した DRAM/ロジック混載 LSI では，高オンチップ・メモリバンド巾を利用して，ミス・ペナルティを増加する事無くキャッシュ・ラインサイズを拡大できる [3][7]．実行プログラムのメモリ参照が高い空間的局所性を有する場合，ラインサイズの拡大はプリフェッチ効果によるヒット率の向上をもたらす．しかしながら、メモリ参照が低い空間的局所性を有する場合，頻繁なキャッシュ・コンフリクトが発生しヒット率は低くなる．その結果，キャッシュ・リプレイスに伴う主記憶アクセス回数が増加し，メモリシステム性能は大きく低下する．また，拡幅されたオンチップ DRAM およびオンチップバスが頻繁に活性化されるため，主記憶アクセスにおける消費エネルギーが増大する．

そこで我々は，DRAM/ロジック混載 LSI 向きキャッシュ・アーキテクチャとして，動的可変ラインサイズ・キャッシュ(D-VLS キャッシュ) を提案している [1][4]．D-VLS キャッシュは，実行プログラムが有する空間的局所性の度合に応じて動的にラインサイズを変更する．これにより，1) 大きなキャッシュ・ラインによるプリフェッチ効果の活用，2) 小さなキャッシュ・ラインによるコンフリクトミスの削減，を同時に達成できる．また，リプレイス対象キャッシュ・ラインに対応したオンチップ DRAM およびオンチップバスのみを活性化する事で，3) 主記憶アクセスにおける無駄なエネルギー消費を回避できる．文献 [1][4] では，従来型ダイレクトマップ・キャッシュを比較対象とし，ミス率を用いて D-VLS キャッシュの性能を評価した．これに対し，本稿では，キャッシュ・アクセス時間を反映した平均メモリアクセス時間を用い，従来型セット・アソシアティブ・キャッシュも含めた性能比較を行う．これと同時に，主記憶 (オンチップ DRAM) アクセスにおける消費エネルギーに関して評価し，D-VLS キャッシュの有効性を明らかにする．

以下，第 2 章では，D-VLS キャッシュの基本概念とその構成を簡単に説明する．次に，第 3 章では，D-VLS キャッシュまたは従来型キャッシュを搭載したオンチップ・メモリシステムに関して，その性能および主記憶アクセスでの消費エネルギーを評価する．最後に，第 4 章で簡単にまとめる．

2　動的可変ラインサイズ・キャッシュ

2.1　基本概念

D-VLS キャッシュの基本概念を図 1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 SRAM サブアレイおよび DRAM サブアレイの分割構造となる[1]．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 1に示す構成の場合，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．

[1]ただし，SRAM サブアレイと DRAM サブアレイに関して，互いのサブアレイ数，および，語長は同じでなければならない．

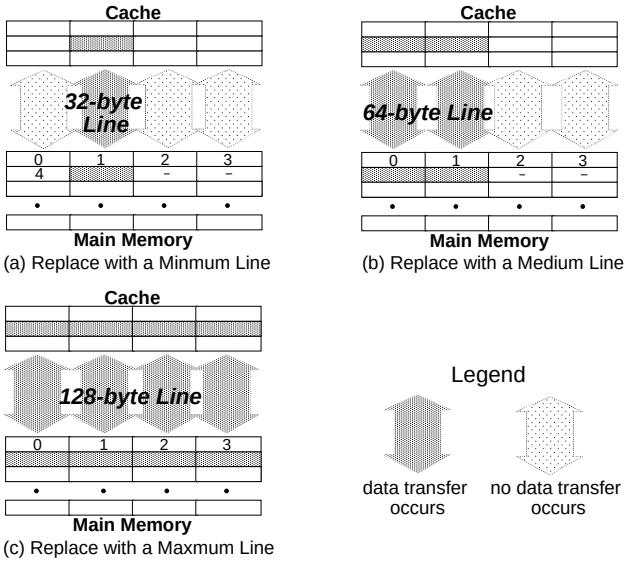

図 1: D-VLS キャッシュの基本概念

- 最小ライン：図 1(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 1(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 1(c) で示すように，全ての SRAM-DRAM サブアレイがリプレイス対象となる．

例えば，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．

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．

2.2　内部構成

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ダイレクト・マップ D-VLS キャッシュの構成を図 2に示す．図 2の D-VLS キャッシュは，動的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搭載する．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 [1] を参照されたい．

参照フラグ (reference-flag)　：各最小ライン (32 バイトライン) 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)：キャッシュ・リプレイス時のラインサイズを指定するフラグであり，各キャッシュ・セクタ (同一インデックスを有する 4 個の 32 バイト最小ライン) 毎に搭載される．

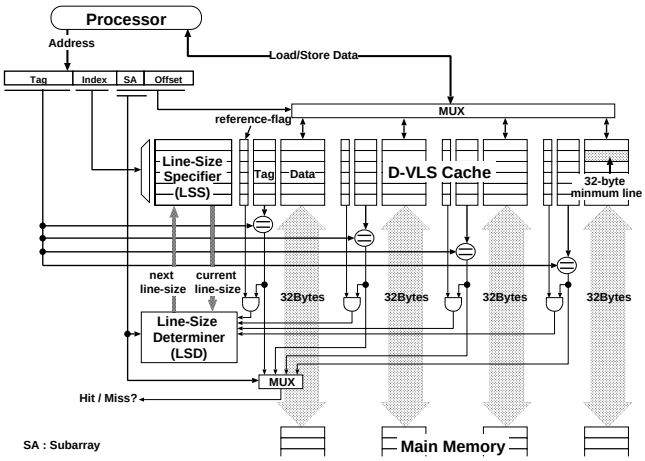

図 2: ダイレクト・マップ D-VLS キャッシュの内部構成

ラインサイズ決定機構 (Line Size Determiner:LSD) : 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．

3 評価

3.1 実験概要

D-VLS キャッシュの有効性を明らかにするため，オンチップ化されたメモリパス (キャッシュと主記憶) に関して，平均メモリアクセス時間ならびに主記憶アクセスでの消費エネルギーを評価した．

大きなラインサイズによるキャッシュ・コンフリクトの発生は，高いキャッシュ連想度 (セット・アソシアティブ方式) の採用，または，キャッシュ・サイズの拡大により回避する事ができる．そこで，本評価においては，以下に示す従来型キャッシュまたは D-VLS キャッシュ搭載のオンチップ・メモリパスを比較対照とした[2]．

- Fix128 : 16K バイトの従来型ダイレクトマップ・データキャッシュ．ラインサイズは 128 バイトで固定．
- Fix128W2 : 16K バイトの従来型 2 ウェイ・セットアソシアティブ・データキャッシュ．ラインサイズは 128 バイトで固定．
- Fix128W4 : 16K バイトの従来型 4 ウェイ・セットアソシアティブ・データキャッシュ．ラインサイズは 128 バイトで固定．
- Fix128double : 32K バイトの従来型ダイレクトマップ・データキャッシュ．ラインサイズは 128 バイトで固定．
- D-VLS128-32 : 16K バイトのダイレクトマップ動的可変ラインサイズ・データキャッシュ．ラインサイズは，32 バイト，64 バイト，および，128 バイトで可変．

[2]実行プログラムの全ワーキング・セットは，オンチップ DRAM に格納されていると仮定する．また，セット・アソシアティブ・キャッシュについては，キャッシュ・リプレイスメントにライトバック・ポリシおよび LRU アルゴリズムを用いる．

表 1: キャッシュ・ミス率

Benchmark	Fix128	Fix128 W2	Fix128 W4	Fix128 double	DVLS 128-32
026.compress	0.1871	0.1755	0.1711	0.1634	0.1724
072.sc	0.0371	0.0285	0.0263	0.0276	0.0465
052.alvinn	0.0224	0.0087	0.0080	0.0175	0.0166
099.go	0.1024	0.0695	0.0302	0.0541	0.0638
124.m88ksim	0.0202	0.0045	0.0028	0.0068	0.0153
126.gcc	0.0611	0.0344	0.0254	0.0349	0.0526
130.li	0.0341	0.0203	0.0182	0.0226	0.0358
132.ijpeg	0.0244	0.0048	0.0036	0.0068	0.0175
134.perl	0.0542	0.0230	0.0105	0.0295	0.0286
147.vortex	0.0505	0.0292	0.0195	0.0307	0.0374
101.tomcatv	0.0633	0.0182	0.0062	0.0546	0.0578
102.swim	0.2612	0.3007	0.3137	0.1016	0.1419
103.su2cor	0.2600	0.0840	0.0242	0.2396	0.0758
104.hydro2d	0.0481	0.0217	0.0179	0.0259	0.0295
Average of normalized miss rate to Fix128	**1.000**	**0.540**	**0.418**	**0.617**	**0.754**

プログラム実行時のミス率，および，各ラインサイズでの総リプレイス回数を測定するため，C 言語を用いてキャッシュ・シミュレータを作成した．本シミュレータは，QPT[2] によって採集されたアドレス・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，SPEC92 ベンチマーク・サイトから 3 つのプログラム (入力は reference input)，ならびに，SPEC95 ベンチマーク・サイトから 7 つの整数プログラム (入力は train input) と 4 つの浮動小数点プログラム (入力は test input) を用いた [8]．なお，各プログラムは，Ultra SPARC プロセッサでの実行を想定し，GNU CC を用いてコンパイルした．

3.2 実験結果

実験結果 (ミス率) を表 1に示す．表 1の最終行は，各プログラムにおけるミス率を Fix128 のミス率で正規化し，全プログラムに関して平均をとった値である．D-VLS キャッシュは，同一キャッシュ・サイズである Fix128 のミス率を平均約 25%改善している．特に，幾つかのプログラム (*052.alvinn* や *134.perl* など) においては，2 倍のキャッシュ・サイズを有する Fix128double と同程度もしくはそれ以下のミス率であった．一方，従来型セット・アソシアティブ・キャッシュ(FixW2 および FixW4) は，Fix128 のミス率を平均約 46%および約 52%改善している．これは，D-VLS128-32 のそれと比較して非常に高い改善率である．以上より，D-VLS キャッシュは，ラインサイズをプログラムの特性に応じて変更する事で，キャッシュ・サイズを 2 倍にした場合と同程度のキャッシュ・コンフリクト削減効果を実現可能である．しかしながら，セット・アソシアティブ方式と同程度のキャッシュ・コンフリクト削減効果を達成する事は難しいと考える．

D-VLS キャッシュにおける各ラインサイズでの総リプレイス回数を表 2に示す．表 2の最終行は，キャッシュ・リプレイス 1 回当たりの平均ラインサイズである．*026.compress* を実行する場合，頻繁なキャッシュ・コンフリクトの発生を回避するため，D-VLS キャッシュはラインサイズ縮小方向に動作する．その結果，平均ラインサイズは約 35 バイトと小さくなる．一方，*052.alvinn* のように空間的局所性の高いプログラムを実行する場合，D-VLS キャッシュは積極的にラインサイズを拡大する．そのため，平均ラインサイズは約 90 バイトと大きくなる．全プログラムの平均ラインサイズは，約 54.5 バイトであった．なお，128 バイトの固定ラインサイズを有する従来型キャッシュでは，キャッシュ・ミスが発生した時，常に 128 バイト・ラインサイズでキャッシュ・リプレイスが行われる．そのため，平均ラインサイズは 128 バイトとなる．

表 2: 各ラインサイズにおける総リプレイス回数 (D-VLS128-32)

Line Size	*026.compress*	*072.sc*	*052.alvinn*	*099.go*	*124.m88ksim*	*126.gcc*	*130.li*
32 bytes	3,164,502	1,038,520	11,546,415	6,445,160	317,746	10,092,540	1,190,072
64 bytes	243,979	492,007	1,465,880	1,724,674	53,858	3,463,487	426,488
128 bytes	14,498	352,181	18,806,730	389,746	68,353	1,468,861	189,570
Ave. of Line Size	**34.69 B**	**58.32 B**	**90.22 B**	**42.82 B**	**50.83 B**	**48.76 B**	**49.63 B**

Line Size	*132.ijpeg*	*134.perl*	*147.vortex*	*101.tomcatv*	*102.swim*	*103.su2cor*	*104.hydro2d*
32 bytes	3,530,649	7,987,886	19,805,372	23,539,313	32,465,163	15,340,954	3,784,227
64 bytes	1,179,064	5,250,134	3,593,130	2,608,352	4,163,613	6,701,837	860,802
128 bytes	1,246,695	3,849,457	1,416,595	2,650,269	884,142	3,315,895	6,175,600
Ave. of Line Size	**58.43 B**	**63.46 B**	**42.11 B**	**43.73 B**	**37.81 B**	**53.01 B**	**89.34 B**

3.3 性能に関する評価

3.3.1 性能モデル

本評価では，平均メモリアクセス時間を用いて，D-VLS キャッシュ搭載のオンチップ・メモリシステム性能を評価する．平均メモリアクセス時間とは，プロセッサからのメモリ参照が発生した時，メモリシステムが当該メモリ参照をサービスするのに要する平均遅延時間である．平均メモリアクセス時間 ($AMAT$) は，以下の式で表す事ができる．

$$AMAT \quad = \quad HT + MR \times 2 \times (DRAMstup + \frac{LS}{BandWidth})$$

ここで，HTおよび MRは，キャッシュ・ヒット時間およびミス率である．また，$DRAMstup$, LSおよび $BandWidth$ は，それぞれ，DRAM スタートアップ時間，リプレイス対象となるキャッシュ・ラインサイズ，および，キャッシュ – 主記憶間のメモリバンド巾を表す．DRAM/ロジック混載 LSI では非常に高い $BandWidth$ を活用できるため，ラインサイズに関係なくミス・ペナルティを一定に保つ事ができる [3]．そこで，本評価では，$DRAMstup$ =50ns および $\frac{LS}{BandWidth}$ =10ns で一定とし，全てのキャッシュ・リプレイスにおいて常に 2 回の主記憶アクセス (ライトバックに 1 回とキャッシュ・フィルに 1 回) が生じると仮定する．

3.3.2 平均メモリアクセス時間 ($AMAT$)

各キャッシュ・モデルのヒット時間を求めるため，本評価では CACTI[9] を使用した[3]．その結果，従来型キャッシュである Fix128，Fix128W2，Fix128W4 および Fix128double のヒット時間は，それぞれ，3.510 ns，6.695 ns，8.421 ns および 4.115 ns であった．一方，D-VLS キャッシュに関して，可変ラインサイズを実現するハードウェア機構の追加はキャッシュ・クリティカル・パスに影響しない．そのため，ラインサイズ変更に伴うヒット時間オーバヘッドは無視できる程度に小さい [1]．そこで，ダイレクトマップ方式である D-VLS128-32 のヒット時間は，同一キャッシュ・サイズを有する従来型ダイレクトマップ方式 (Fix128) のそれと等しいと仮定する．

各プログラムにおける平均メモリアクセス時間 ($AMAT$) を図 3に示す．これらの結果は，第 3.2節で示したミス率，および，本節で求めたヒット時間に基づいて算出した．図 3の各プログラムにおいて，全ての結果は Fix128 の平均メモリアクセス時間で正規化している．第 3.2節で示したように，従来型セット・アソシアティブ・キャッシュ (Fix128W2，Fix128W4) は，従来型ダイレクトマップ・キャッシュ(Fix128) と比較して，非常に低いミス率を達成する．しかしながら，高連想度の採用に伴うヒット時間の増大により，平均メモリアクセス時間を向上する事はできなかった．これに対し，D-VLS キャッシュは，ヒット時間オーバヘッドを伴う事無くヒット率を向上する．全プログラムで平均すると，D-VLS128-32 は，Fix128 の平均メモリアクセス時間を約 20%短縮した．これは，2 倍のキャッシュ・サイズを有する従来型ダイレクトマップ・キャッシュ(Fix128double) の採用により達成される性能向上と同程度である．

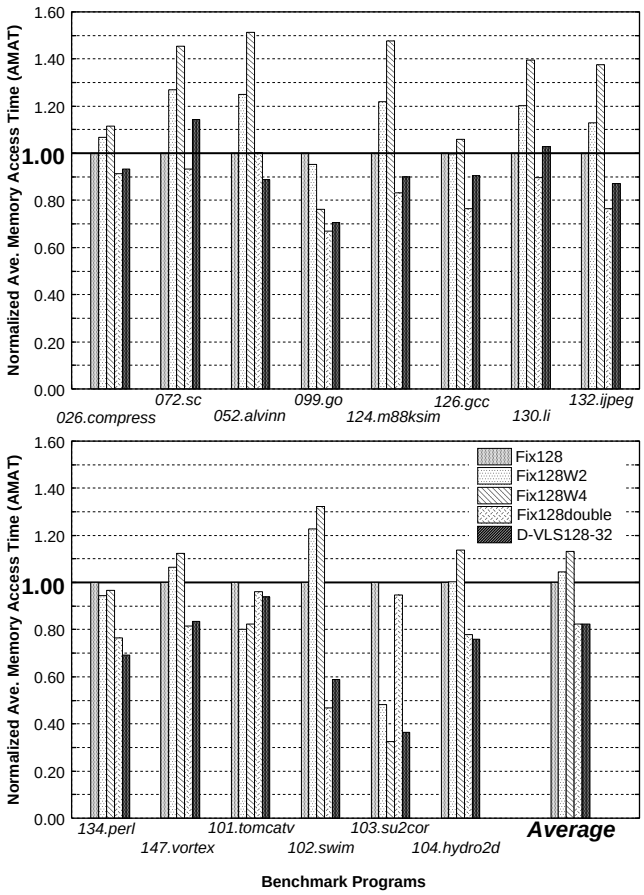

図 3: 平均メモリアクセス時間

[3]本シミュレーションでは，プロセス・ルール 0.5um，キャッシュの入力アドレス巾 30 ビット，ならびに，キャッシュ出力データ巾 32 ビットを仮定した．

3.4 主記憶アクセスにおける消費エネルギー

3.4.1 消費エネルギー・モデル

オンチップ化された主記憶へのアクセスに要するエネルギー (E) は，オンチップ・バスおよびオンチップ DRAM の活性化に伴う消費エネルギーの和で近似できる．ここで，キャッシュ・ミス発生時，リプレイス対象となるキャッシュ・ラインに対応したオンチップ・バス，および，DRAM サブアレイのみが活性化されると仮定する．この場合，固定ラインサイズの従来型キャッシュでは，常に当該ラインサイズでの主記憶アクセスとなる．そのため，オンチップ主記憶アクセスでの消費エネルギー Eは，総リプレイス回数 (つまり，キャッシュ・ミス率) にのみ依存する．これに対し，D-VLS キャッシュは，実行プログラムの特性に応じてラインサイズを変更する．よって，オンチップ主記憶アクセスでの消費エネルギー Eは，総リプレイス回数だけでなく，選択されたラインサイズにも依存する．以上より，オンチップ主記憶アクセスに要するエネルギー (E) は，以下の式で表す事ができる．

$$E = \sum_{N} RepCount(n) \times E_{subarray} \times n/32bytes,$$
$$N = \{n | n = 32bytes, 64bytes, 128bytes\}.$$

ここで，n はキャッシュ・リプレイス時のラインサイズ，$E_{subarray}$は 1 個の 32 バイト DRAM サブアレイに対する 2 回のアクセス (ライトバックに 1 回とキャッシュ・フィルに 1 回) で消費されるエネルギーである．また，$RepCount(n)$ は，ラインサイズ n における総リプレイス回数を表す．なお，128 バイトの固定ライン・サイズを有する従来型キャッシュ(Fix128，Fix128W2，Fix128W4，および，Fix128double)では，nが 32bytes もしくは 64bytes の時，オンチップ主記憶アクセスにおける消費エネルギーは 0 となる (つまり，$RepCount(32bytes) = 0$，かつ，$RepCount(64bytes) = 0$).

3.4.2 消費エネルギー

オンチップ主記憶アクセスでの消費エネルギー (E) を図 4 に示す．各プログラムにおいて，全ての結果は Fix128 の Eで正規化している．第 3.4.1節で述べたように，固定ラインサイズを有する従来型キャッシュでの Eは，キャッシュ・ミス率にのみ依存する．第 3.2 節表 1で示したように，最も低い平均ミス率を達成したのは，16K バイト従来型 4 ウェイ・セットアソシアティブ・キャッシュ(Fix128W4) であった．従って，従来型キャッシュ同士を比較した場合，最も多くの消費エネルギーを削減できるのは Fix128W4 である．実際，図 4に示すように，Fix128W4 は，Fix128 での消費エネルギーを約 60%改善した．

一方，D-VLS キャッシュは，Fix128 を除く全てのキャッシュにおいて，最も高いミス率であった (Fix128 からのキャッシュ・ミス改善率は，Fix128W4 が約 58%であるのに対し，D-VLS128-32 では約 25%)．しかしながら，D-VLS キャッシュの場合，オンチップ DRAM アクセスにおける消費エネルギーは，キャッシュ・ミス率だけでなくキャッシュ・リプレイス時のラインサイズにも依存する．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．従って，D-VLS キャッシュは，オンチップ DRAM アクセス回数の削減 (キャッシュ・ミス率改善) による低消費エネルギー化，ならびに，ラインサイズ縮小 (活性化されるオンチップ DRAM サイズの縮小) による低消費エネルギー化，の両方を同時に達成できる．Fix128 と比較して，D-VLS128-32 は，オンチップ DRAM アクセス消費エネルギーを約 68% (ミス率の改善により 25%，ラインサイズ縮小により 43%) 削減した．

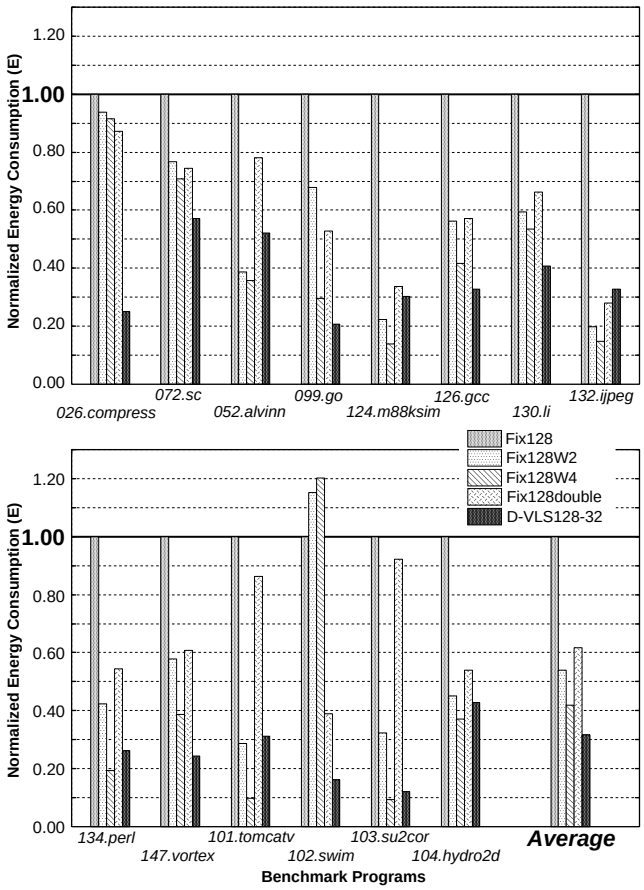

図 4: オンチップ DRAM アクセスにおける消費エネルギー

3.5 ED(Energy-Delay) 積

D-VLS キャッシュにより達成される高性能化/低消費エネルギー化を同時に評価するため，第 3.3.2節と第 3.4.2 節の実験結果に基づいて ED 積 ($E \times AMAT$) を求めた．各キャッシュ・モデルでの ED 積を図 5に示す．各プログラムにおいて，全ての結果は Fix128 の ED 積で正規化している．

Fix128W2，Fix128W4，および，Fix128double の ED 積にはさほど大きな差が見られず，16K バイト従来型ダイレクトマップ方式 (Fix128) と比較した場合，ED 積削減率は約 40%〜50%であった．これに対し，ダイレクトマップ方式の D-VLS キャッシュ(D-VLS128-32) は，70%以上の ED 積削減を達成している．これは，以下の理由に起因すると考える．

32KB ダイレクトマップ方式 (Fix128double)： キャッシュ・サイズ拡大による若干のヒット時間オーバヘッドを伴うだけで，大きなラインサイズによるコンフリクト・ミスを削減できる．そのため，平均メモリアクセス時間は大幅に改善される．しかしながら，セット・アソシアティブ方式と比較してミス率は高いため，大きなラインサイズによるオンチップ DRAM アクセスが比較的頻繁

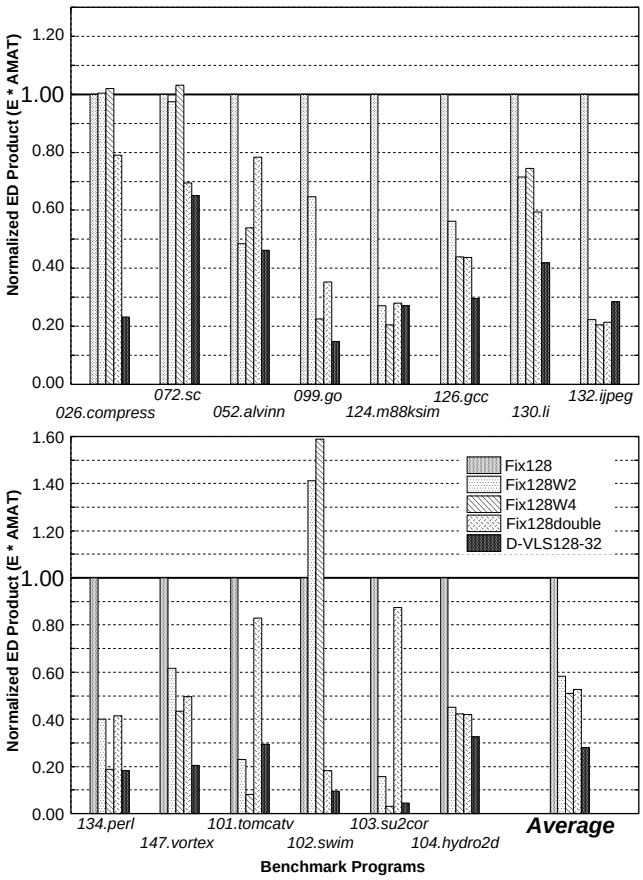

図 5: ED 積 ($E \times AMAT$)

に発生する．その結果，オンチップ DRAM アクセスで消費されるエネルギーが増大し，キャッシュ・サイズ拡大による性能向上効果は打ち消されてしまう．

16KB セット・アソシアティブ方式 (Fix128W2/W4)： 従来型キャッシュにおいて，オンチップ DRAM 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．そのため，高いキャッシュ連想度の採用によるミス率の低下に伴い，オンチップ DRAM アクセスでの消費エネルギーを大幅に削減できる．しかしながら，セット・アソシアティブ方式の採用は，キャッシュ・アクセス時間 (ヒット時間) を長くする．その結果，低ミス率であるにも関わらず，平均メモリアクセス時間は長くなり，セット・アソシアティブ方式の採用による低消費エネルギー効果は打ち消されてしまう．

16KB ダイレクトマップ方式 D-VLS(D-VLS128-32)： D-VLS キャッシュは，プログラムの特性に応じて動的にラインサイズを変更する事で，ヒット時間オーバヘッドを伴う事無くコンフリクト・ミスを削減する．つまり，D-VLS キャッシュにおいては，達成される高性能効果および低消費エネルギー効果が互いに打ち消し会う事はない．また，ラインサイズの縮小により更なる低消費エネルギー化が可能となる．従って，D-VLS キャッシュの採用により，ED 積を大幅に改善する事ができる．

4 おわりに

本稿では，DRAM/ロジック混載 LSI 向けキャッシュ・アーキテクチャである動的可変ラインサイズ・キャッシュ (D-VLS キャッシュ) について紹介した．また，D-VLS キャッシュを採用したオンチップ・メモリパスに関して，その性能およびオンチップ DRAM アクセスでの消費エネルギーを評価した．さらに，D-VLS キャッシュの有効性を明らかにするため，ED 積 (オンチップ DRAM アクセスでの消費エネルギー×平均メモリアクセス時間) を求め，従来型キャッシュ方式との比較を行った．

実験の結果，128 バイトの固定ラインサイズを有する従来型ダイレクトマップ・キャッシュ(Fix128) と比較して，セット・アソシアティブ方式の採用 (Fix128W2，Fix128W4)，もしくは，キャッシュ・サイズの拡大 (Fix128double) による ED 積削減率は約 40%〜50%であった．これに対し，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 16K バイト・ダイレクトマップ D-VLS キャッシュ (D-VLS128-32) は，70%以上の ED 積削減を達成した．

本評価では，キャッシュ・アクセスで消費されるエネルギー，ならびに，オンチップ DRAM のリフレッシュで消費されるエネルギーは考慮していない．今後これらを含めた，総合的なオンチップ・メモリパス消費エネルギーを評価する予定である．

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科 安浦寛人 教授に感謝致します．また，折りに触れ貴重なご意見を頂く，岩井原瑞穂 助教授に感謝します．本研究の遂行にあたり，ご協力頂いた PPRAM グループ関係者各位，ならびに，安浦研究室の諸氏に感謝する．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「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), 富士通研究所奨学寄付金「プロセッサアーキテクチャ」による．

参考文献

[1] 井上弘士， 甲斐康司, 村上和彰, “DRAM/ロジック混載 LSI の高オンチップ・メモリバンド巾を活用する動的可変ラインサイズ・キャッシュ方式の提案,” 信学技報, pp.109–116, 1998 年 4 月.

[2] Hill, M. D., Larus, J. R., Lebeck, A. R., Talluri, M., and Wood, D. A., “WARTS: Wisconsin Architectural Research Tool Set,” *http://www.cs.wisc.edu/~larus/warts.html*, University of Wisconsin - Madison.

[3] Inoue, K., Koji, K., and Murakami, K., “High Bandwidth, Variable Line-Size Cache Architecture for Merged DRAM/Logic LSIs,” *IEICE Transactions on Electronics*, Vol.E81-C, No.9, pp.1438–1447, Sep. 1998.

[4] Inoue, K., Koji, K., and Murakami, K., “Dynamically Variable Line-Size Cache Exploiting High On-Chip Memory Bandwidth of Merged DRAM/Logic LSIs,” *Porc. of the 5th International Symposium on High-Performance Computer Architecture*, pp.218–222, Jan. 1999.

[5] 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.

[6] Patterson, D., Anderson, T., Cardwell, N., Fromm, R., Keeton, K., Kozyrakis, C., Thomas, R., and Yelick, K., “A Case For Intelligent RAM,” *IEEE Micro*, vol.17, no.2, pp.34–44, March/April 1997.

[7] Saulsbury, A., Pong, F., and Nowatzyk, 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.

[8] SPEC (Standard Performance Evaluation Corporation), *URL: http://www.specbench.org/osg/cpu92, http://www.specbench.org/osg/cpu95.*

[9] 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.