

修士論文

題目

可変レベルキャッシュの  
低電力化に関する研究

指導教員

近藤 利夫 教授

平成 25 年度

三重大学大学院 工学研究科 情報工学専攻  
計算機アーキテクチャ研究室

渡部 功 (412M528)

## 内容梗概

現在、ノートパソコンや携帯電話、スマートフォンなどのモバイル端末の高性能化にともない消費エネルギーが増大し、バッテリーによる駆動時間が短くなるという問題が発生している。そこで、モバイル端末の性能を落とすことなく低消費エネルギーを実現することが要求されている。現在、高性能かつ低消費エネルギーを実現する様々な手法が提案されている。その手法の一つとして可変レベルキャッシュが提案されている。

可変レベルキャッシュは、アプリケーションが多くのキャッシュ容量を必要とする場合はキャッシュ容量を通常の容量で使用する通常モードとして動作し、必要とするキャッシュ容量が少ない場合にはキャッシュ容量の半分をエネルギー消費の少ないスリープモードにする低消費エネルギーモードとして動作する。低消費エネルギーモードにおいて、スリープモードに移行した部分は擬似的に1つ下位レベルの排他的キャッシュとして利用される。可変レベルキャッシュは通常モードと低消費エネルギーモードを動的に切換ることによって高性能と低消費エネルギーの両立を実現する手法である。

可変レベルキャッシュは他手法と比較して性能維持に優れた手法であることが明らかにされている。しかしその反面、消費エネルギー削減効率があまり高くないという問題点があった。そこで本論文では、可変レベルキャッシュの消費エネルギー削減効率を高める為に2つの改良を行った。1つ目は可変レベルキャッシュの新たな制御アルゴリズムを考案し、キャッシュの50%だったスリープモード領域を従来では制御出来なかった75%まで拡張させるモードを追加することで、消費エネルギー削減効率を平均約18%高めることが出来た。

しかし、単純にスリープモード領域を拡張した場合、通常モードにおいて可変レベルキャッシュの挙動により、本来とは異なる箇所に移動されたデータへの追加アクセス回数が増大した結果、動的エネルギーが増加するという問題が発生する。この問題を解決する為に2つ目の改良を行った。追加アクセス時に不必要なキャッシュアクセスが発生している事に着目し、目的のデータの場所を記憶させる機構を考案した。この結果、追加アクセス時のキャッシュアクセスが必要最小限になり、動的エネルギーを平均約18%削減することが出来た。

これらの改良を行った結果、シミュレーション評価によると性能評価指

標であるエネルギー遅延積 (ED 積) で比較して、従来の可変レベルキャッチュと比べて平均約 34%性能向上することが明らかとなった。

# Abstract

Increasing power consumption has been becoming a major concern not only for mobile computing but also high-performance computing, and processors are required to achieve both low-energy and high-performance at the same time. In particular, it is important to reduce leakage energy consumed in a cache memory because power dissipated by leakage current is a dominant factor in deep submicron technologies and the cache memory dissipates a large amount of leakage energy. To achieve high-performance and low-energy simultaneously, a variable level cache (VLC) is proposed. The VLC dynamically varies the cache capacity according to required cache capacity by running program. If the VLC detects that the current running program does not need large cache memory, half of the cache memory is put into standby mode, and is virtually treated as a lower level exclusive cache. In this way, the VLC succeeded in reducing leakage energy without performance degradation.

To reduce more energy consumption this paper proposes two novel approaches for VLC. As the first approach, this paper devises a novel algorithm to control VLC. The improved VLC can expand sleep mode area to 75% of the cache. As a result, this approach can reduce 18% energy consumption compared with the previous VLC.

However, if the improved VLC incurs increase in its dynamic energy consumption. The increase in dynamic energy consumption is caused by increasing the number of extra access which accesses the data moved to incorrect place due to the behavior of VLC. To solve this problem, this paper proposes a second approach. As the second approach, this paper focuses on unnecessary accesses which is put on incorrect place. This approach memorizes detailed place that has the data in the extra access. As a result, this second approach further reduce 18% dynamic energy consumption compared with the first approach because it minimizes the dynamic energy of extra access.

According to simulation results of these two approaches, proposed VLC can improve 34% energy-delay product compared to the previous VLC.

# 目次

<b>1</b>	<b>まえがき</b>	<b>1</b>
<b>2</b>	<b>キャッシュメモリにおけるリークエネルギー</b>	<b>3</b>
2.1	キャッシュメモリ	3
2.2	キャッシュメモリの動作	4
2.3	リークエネルギー	5
2.4	キャッシュメモリにおけるリークエネルギーの影響	5
<b>3</b>	<b>関連研究</b>	<b>7</b>
3.1	DRI キャッシュ	8
3.1.1	DRI キャッシュの概要	8
3.1.2	DRI キャッシュの問題点	9
3.2	従来の可変レベルキャッシュ	10
3.2.1	可変レベルキャッシュの概要	10
3.2.2	排他的キャッシュ(Exclusive Cache)	12
3.2.3	可変レベルキャッシュの動作詳細	13
3.2.4	書き戻しペナルティ軽減手法	14
<b>4</b>	<b>評価方法</b>	<b>17</b>
4.1	エネルギー遅延積	17
4.2	消費エネルギー	17
4.3	シミュレータ	20
4.4	実験環境	20
<b>5</b>	<b>改良1：モード追加</b>	<b>22</b>
5.1	評価結果	24
<b>6</b>	<b>改良2：動的エネルギーの削減</b>	<b>25</b>
6.1	動的エネルギーの増加	25
6.2	二次アクセス時のアクセスバンク制限	27
6.3	評価結果	29
<b>7</b>	<b>結論</b>	<b>30</b>
	<b>謝辞</b>	<b>33</b>



## 目次

2.1	メモリ階層構造	3
2.2	キャッシュメモリの概念図	4
2.3	プロセッサ内における消費エネルギー内訳	6
2.4	トランジスタの消費電力傾向予想	6
3.5	DRI キャッシュ	9
3.6	DRI キャッシュのサイズ遷移	9
3.7	DRI キャッシュと可変レベルキャッシュの違い	11
3.8	可変レベルキャッシュ	12
3.9	通常キャッシュと排他的キャッシュ	13
3.10	通常モードへ切換わる前のキャッシュ	14
3.11	低消費エネルギーモードでの動作	16
3.12	通常モードでの動作	16
4.13	実行時間と消費エネルギー	18
5.14	可変レベルキャッシュのモード遷移	22
5.15	提案手法:拡張 NoD カウンタ	23
5.16	実行時間	25
5.17	消費エネルギー	26
5.18	ED 積	26
6.19	提案手法:VSL Bitmap	28
6.20	実行時間	30
6.21	消費エネルギー	31
6.22	動的エネルギー	31

## 表 目 次

4.1 シミュレーションのキャッシュに関するパラメータ . . . . .	21
4.2 ベンチマーク . . . . .	22



# 1 まえがき

現在、ノートパソコンや、携帯電話、スマートフォンなどのモバイル端末の高性能化にともない消費エネルギーが増大し、バッテリーによる駆動時間が短くなるという問題が発生している。そこで、モバイル端末の性能を落とすことなく低消費エネルギーを実現することが要求されている。中でも、モバイル端末の性能に大きく貢献しエネルギーを大量に消費するのがプロセッサである。

一般にプロセッサで消費されるエネルギーは動的消費エネルギーと静的消費エネルギーに分けられる。動的消費エネルギーはトランジスタのスイッチングによって消費されるエネルギーである。一方、静的消費エネルギーはトランジスタの漏れ電流(リーク電流)によって引き起こされ、トランジスタのスイッチングに関係なく消費されるエネルギーで、リークエネルギーともいう。近年、回路の微細化にともなって、動的消費エネルギーが削減される一方、リークエネルギーが消費するエネルギーの割合が増加している。リークエネルギーはトランジスタ数に比例するため、プロセッサの高性能化に伴い容量が増大しているキャッシュシステムのリークエネルギー削減が重要である。そこでキャッシュのリークエネルギー削減手法の一つとして、可変レベルキャッシュが提案されている。

可変レベルキャッシュは、アプリケーションが必要とするキャッシュ容量が多い場合はキャッシュ容量を通常の容量で使用する通常モードとして動作し、必要とするキャッシュ容量が少ない場合はキャッシュ容量の半分をエネルギー消費の少ないスリープモードに移行し、擬似的に1つ下位レベルの排他的キャッシュとして使用する低消費エネルギーモードとして動作する。可変レベルキャッシュは通常モードと低消費エネルギーモードを動的に切換ることによって高性能と低消費エネルギーの両立を実現する手法である。

従来の可変レベルキャッシュは代表的な低電力キャッシュ手法と比較して、性能維持に優れた手法であることが示されている。[6][7]しかしその反面、消費エネルギー削減効率あまり高くないという問題点があった。そこで本論文では、可変レベルキャッシュの消費エネルギー削減効率を高める為に以下の2つの改良を行った。

1つ目は可変レベルキャッシュの新たな制御アルゴリズムを考案し、キャッシュの50%だったスリープモード領域を従来では制御出来なかった75%まで拡張させるモードを追加することで、消費エネルギー削減効率を平均約18%高めることが出来た。しかし、単純にスリープモード領域を拡張

した場合、通常モードにおいて可変レベルキャッシュの挙動により、本来とは異なる個所に移動されたデータへの追加アクセス回数が増大する為、動的エネルギーが増加するという問題が発生する。この問題を解決する為に2つ目の改良を行った。追加アクセス時に不必要なキャッシュアクセスが発生している事に着目し、目的データの場所を記憶させる機構を考案した。これにより目的のデータが存在する可能性のある個所のみを追加アクセスを行う事を可能にした。この結果、追加アクセス時のキャッシュアクセスが必要最小限になり、動的エネルギーを平均約18%削減することが出来た。これらの改良を行った結果、シミュレーション評価によると性能評価指標であるエネルギー遅延積(ED積)で比較して、従来の可変レベルキャッシュと比べて平均約34%性能向上することが明らかとなった。

本論文の構成については以下の通りである。第2章では、キャッシュシステムの概要を述べ、第2章で関連研究と従来の可変レベルキャッシュについて述べ、第4章で性能と消費エネルギーの評価方法を述べ、第5章、第6章で可変レベルキャッシュの二種類の改良手法を提案する。そして最後に第7章でまとめる。

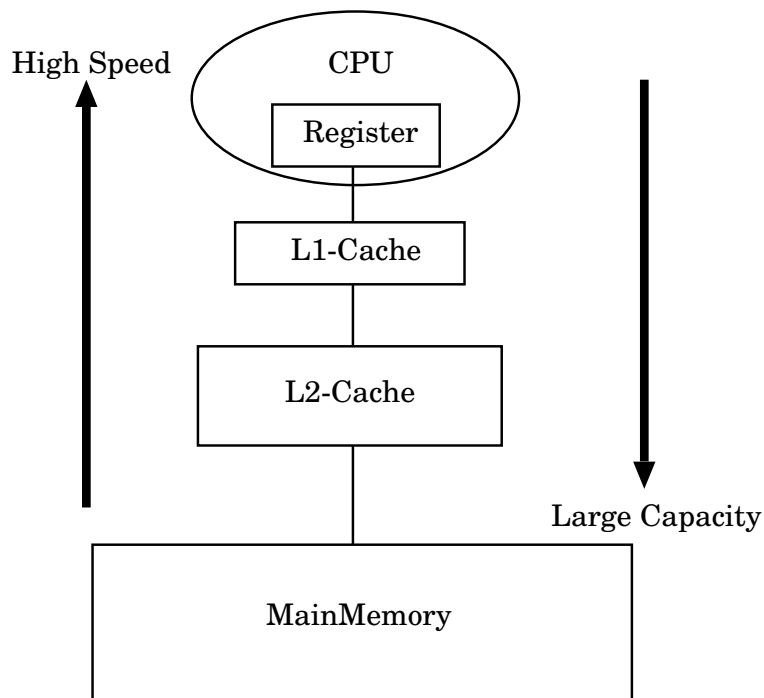


図 2.1: メモリ階層構造

## 2 キャッシュメモリにおけるリークエネルギー

### 2.1 キャッシュメモリ

キャッシュメモリとは、プロセッサに併設されるメモリユニットのことである。キャッシュメモリはプロセッサの約50%程度の面積と消費エネルギーを占め、低電力プロセッサの研究において重要なユニットの一つである。

キャッシュメモリはプロセッサとアクセス速度の遅いメインメモリとのデータのやり取りを中継する小容量高速メモリである。一般的にキャッシュメモリは2, 3回層に分かれており、上階層のキャッシュメモリ程アクセス速度が高速かつ小容量である。メモリ階層構造図を図 2.1 に示す。本研究はキャッシュシステムにおいて最大面積・最大エネルギー消費量をもつ最下層キャッシュ(図 2.1 の L2 キャッシュ)を対象としている。また、本論文で対象とするキャッシュメモリはウェイセットアソシアティブ方式である。これは最も一般的に用いられている方式のキャッシュであり、

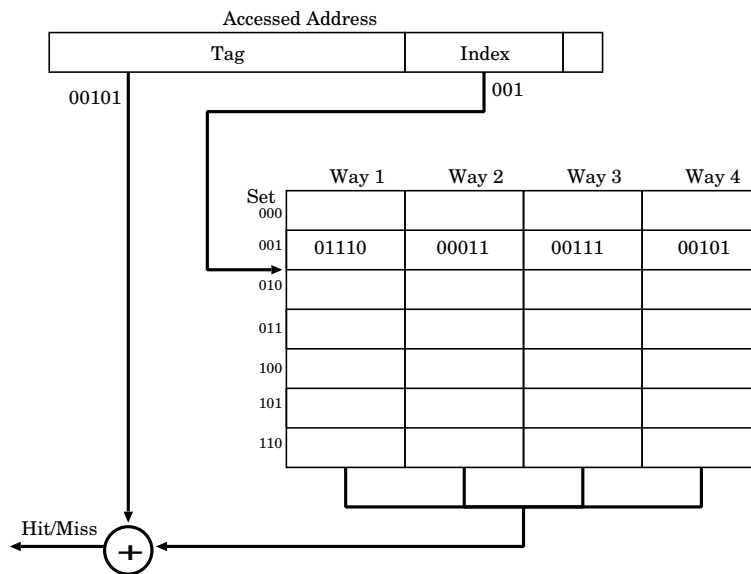


図 2.2: キャッシュメモリの概念図

キャッシュを複数のバンク化したSRAMで構成することで、アクセス効率を向上させるものである。例えば、数バンクで構成されたキャッシュはセットアソシアティブキャッシュと呼ばれる。本論文で評価に用いる4-ウェイセットアソシアティブキャッシュの概念図を図 2.2 に示す。キャッシュメモリは上位モジュールから転送されてくるアクセス先アドレスをタグとインデックスに分離し、目的のデータへアクセスする。4-ウェイセットアソシアティブの場合、同一セット内にデータラインが4ライン存在するため、アドレスから分離したタグを用いてどのラインが目的のデータかを判別する。他にも、1つメモリユニットのみで構成するダイレクトマップ方式や1データ分の最小容量SRAMでバンク構成するフルセットアソシアティブ方式が存在するが、一般的に用いられる事は少ない為、本論文では議論しない。

## 2.2 キャッシュメモリの動作

図 2.2 を用いてキャッシュメモリのアクセス動作を説明する。CPU等、上位ユニットからアクセス先アドレスが転送されると、アドレスから Tag(00101) と Index(001) を算出する。キャッシュは Index を用いて Set を引き、Set(001) 内に存在する全データの Tag(01110, 00011, 00111, 00101) とアドレス

から算出した Tag(00101) を比較する。Set 内のデータの Tag とアドレスの Tag が一致した場合、キャッシュヒットとなる。反対に、Set 内のデータの Tag とアドレスの Tag が一致しない場合、キャッシュミスとなる。

## 2.3 リークエネルギー

一般にプロセッサで消費されるエネルギーは動的消費エネルギーと静的消費エネルギーに分けられる。動的消費エネルギーはトランジスタのスイッチングによって消費されるエネルギーである。一方、静的消費エネルギーはトランジスタの漏れ電流(リーク電流)によって引き起こされ、トランジスタのスイッチングに関係なく消費されるエネルギーで、リークエネルギーともいう。近年、回路の微細化にともなって、動的消費エネルギーが削減される一方、リークエネルギーが消費するエネルギーの割合が増加している。

## 2.4 キャッシュメモリにおけるリークエネルギーの影響

リークエネルギーはトランジスタ数に比例する。プロセッサの高性能化に伴って容量が増大しているキャッシュシステムのリークエネルギー削減が重要である。図 2.3 はプロセッサ内における各ユニットの消費エネルギー内訳を示しており、キャッシュメモリが全体の約 44% を占めていることが分かる。また、図 2.4 はトランジスタの消費電力今後の傾向予想であり、リークエネルギーが約半分と大きな割合を占めており、消費エネルギーが今後も増加していくことが予想されている。これらのデータから、プロセッサの消費エネルギー削減において、キャッシュメモリのリークエネルギー削減が重要であることが分かる。

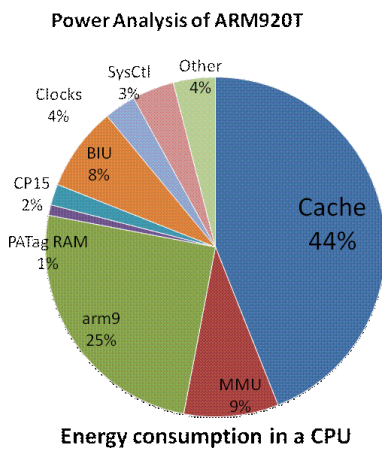
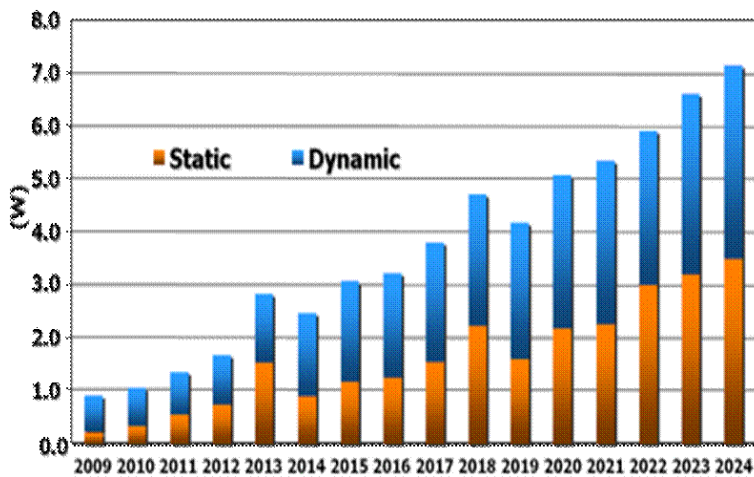


図 2.3: プロセッサ内における消費エネルギー内訳



SoC Consumer Portable Power Trend [Source: ITRS, 2010]

図 2.4: トランジスタの消費電力傾向予想

### 3 関連研究

これまでにキャッシュの様々なリークエネルギー削減手法が提案されてきた。これらの手法は通常状態と待機状態を切替える単位で、大きく以下の3つに分類する事ができる。

#### 1. ライン単位の状態切替

1つ目は、キャッシュライン単位で通常状態と待機状態を切替えるものである。代表的な研究として、Drowsy Cache [1][2] が挙げられる。Drowsy Cacheは定期的に全てのキャッシュラインへの電源電圧を下げ、アクセスが発生したキャッシュラインに対してのみ電源電圧を回復する事でリークエネルギーを削減する手法である。ライン単位で切替える手法は、キャッシュメモリの状態を細かく切替えることが可能である一方、センスアンプ等のライン以外にかかる消費エネルギーを削減できないといった問題がある。また、1つのセット内で通常状態のキャッシュラインと待機状態のキャッシュラインが混在する状態となる、すなわち連想度が落ちてしまうという問題がある。更に、ライン単位での電源制御が必要なため、実装するのに、通常のメモリマクロやメモリコンパイラが利用できないため、設計コストが非常に高くなる。よって本論文ではこれ以上議論しない。

#### 2. ウェイ単位の状態切替

2つ目は、NUAL キャッシュ[3]のようにウェイ単位で通常状態と待機状態を切替えるものである。NUAL キャッシュはウェイ予測キャッシュ[4]を基に作られた低リークキャッシュ手法であり、参照データが存在するウェイをLRUを利用して予測し、選択的に活性化することで低消費エネルギー化を行う手法である。活性化するウェイは常に1ウェイのみで、別のウェイへアクセスする場合は現在活性化されているウェイを待機状態に落としてから別のウェイを活性化させる。NUAL キャッシュは非活性のウェイへのアクセスに時間がかかり、実行時間が増大するという問題点がある。

#### 3. セット数増減による状態切替

最後の1つは、DRI キャッシュ [5]のようにキャッシュシステム内を複数のバンクに分割し、セット数を増減させて通常状態と待機状態を切替えるものである。この手法は、ライン単位で切替える手法のように状態の細かな切替が出来ない反面、待機状態時にセンスアン

プ等のライン以外の消費エネルギー削減が可能となる。セット数増減による切換を行う手法は、バンクを構成する場合にセット単位、ウェイ単位のどちらでも可能であるが、本論文ではセット単位でバンクを構成する事を考える。なぜなら、ウェイ単位で切換える事は前述したように実効的に連想度を低下させてしまうが、セット単位の切換の場合、各セット内の連想度が通常状態と同じに保たれる利点があるからである。本論文では、実行時間を維持しながら消費エネルギーを削減することを目的としているため、DRI キャッシュを基とした可変レベルキャッシュのモード切換手法の提案を行っている。これは、DRI キャッシュはキャッシュメモリの連想度を保つことが出来、実行時間に及ぼす影響は前述した他手法と比べ比較的少ないと考えられるからである。

### 3.1 DRI キャッシュ

#### 3.1.1 DRI キャッシュの概要

図 3.5 に DRI キャッシュの概要図を示す。DRI キャッシュはある一定時間間隔 (interval) でキャッシュミス数をカウンタ (miss counter) によってカウントする。そして、ミス数がある閾値 (miss-bound) より小さい場合には、キャッシュサイズを縮小しても性能には大きな影響を与えないと判断する。一方、ミス数が閾値よりも大きい場合にはキャッシュサイズを増大して性能低下を防ぐ。キャッシュサイズを減らす場合はその時の容量の半分にし、逆にキャッシュサイズを増やす場合は倍にする。これを複数段階に分けて実装を行う事で、キャッシュサイズを必要に応じて変更する。例えば、動的に 256KB, 128KB, 64KB, 32KB のキャッシュサイズに変更することができる。図 3.6 に DRI キャッシュのサイズ遷移図を示す。キャッシュラインへのアクセスは、アドレスにマスク (size mask) を掛ける事でキャッシュサイズの変化に対応させている。また、容量を小さくすると参照するインデックスのビット幅が減少する。すなわちタグフィールドのビット幅が増加するため、タグのビット幅を冗長に確保している。

このようにして DRI キャッシュでは、キャッシュサイズを縮小した場合の未使用領域の SRAM セルに対して電源電圧の供給を停止することによってリークエネルギーの削減をしている。また、DRI キャッシュは電源を切る部分のキャッシュを 1 つのバンクとして電源管理を行う事で、セン



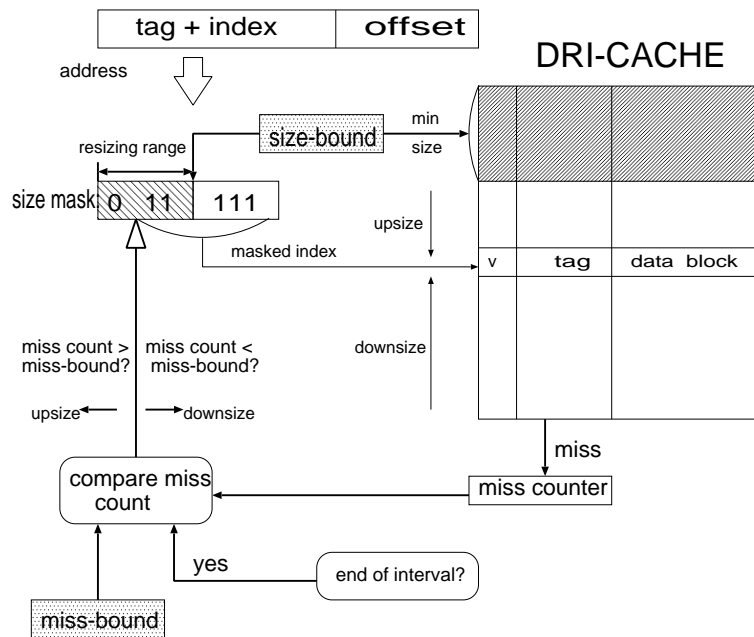


図 3.5: DRI キャッシュ

スランプやビット線などの，SRAMセル以外の回路の電源も切る事が出来る利点がある．

### 3.1.2 DRI キャッシュの問題点

従来のDRIキャッシュでは，キャッシュサイズを縮小した場合，未使用領域のSRAMセルに対して電源電圧の供給を停止することでリークエネルギーを削減している．従って，電源を落とした部分に格納されていた

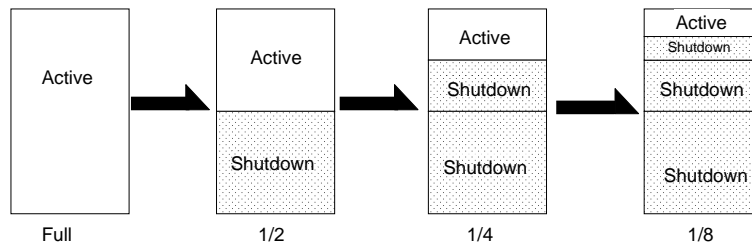


図 3.6: DRI キャッシュのサイズ遷移

データは破棄されるためキャッシュヒット率が低下するという問題がある。

更に、DRI キャッシュは命令キャッシュ用に開発された手法であり、当該手法をデータキャッシュに適用する場合、縮小する部分への電力供給を完全に停止し、データを破壊するため、その部分にあるデータを下位の記憶層へと書き戻す必要がある。また、キャッシュサイズによってデータの配置が異なるために、キャッシュサイズを増大させる場合には、現在キャッシュに入っているデータを下位の記憶層へと書き戻さなければならない。DRI キャッシュをデータキャッシュに適用するとこれらの処理によって性能へ悪影響を与えるという問題がある。

## 3.2 従来の可変レベルキャッシュ

DRI キャッシュの上記の問題点を解決する手法として、メモリのスリープモードを利用し、かつ待機状態への切替時の書き戻しを抑制することで性能を向上させる可変レベルキャッシュが提案されている。 [6] [7]

### 3.2.1 可変レベルキャッシュの概要

可変レベルキャッシュは、DRI キャッシュと同様に、ある一定時間間隔でキャッシュミス数をカウントする事で、キャッシュへの要求性能をキャッシュミス率から動的に判断し、キャッシュ容量を増減させる。しかし、DRI キャッシュのように単純に容量を減少させるだけでは、キャッシュミス回数が増加してしまう。そこで、容量の半分の電源供給を遮断するのではなく、スリープモードとする。スリープモードとは、電源の共有を完全に停止するのではなく、データの内容が破壊されない程度に電源電圧を下げた状態の事を言う。スリープモードにはデータの内容が保存されるというメリットがある。また、電源供給を停止する場合よりはリークエネルギーの削減率が低くなるが、通常モードよりは大幅にリークエネルギーが削減できるというメリットもある。しかし、単純にスリープモードにしては、スリープモードとなっている領域へのアクセスは通常アクセスより時間がかかるため、スリープモードになっている領域へアクセスが多発する場合、性能が悪化する恐れがある。そこで、スリープモードの領域を1つ下位レベルの排他的キャッシュ [8] [9] とすることでスリープモードでのキャッシュアクセスを減らし、消費エネルギーを削減する。DRI キャッシュと可変レベルキャッシュの違いを図 3.7 に示す。例えば、

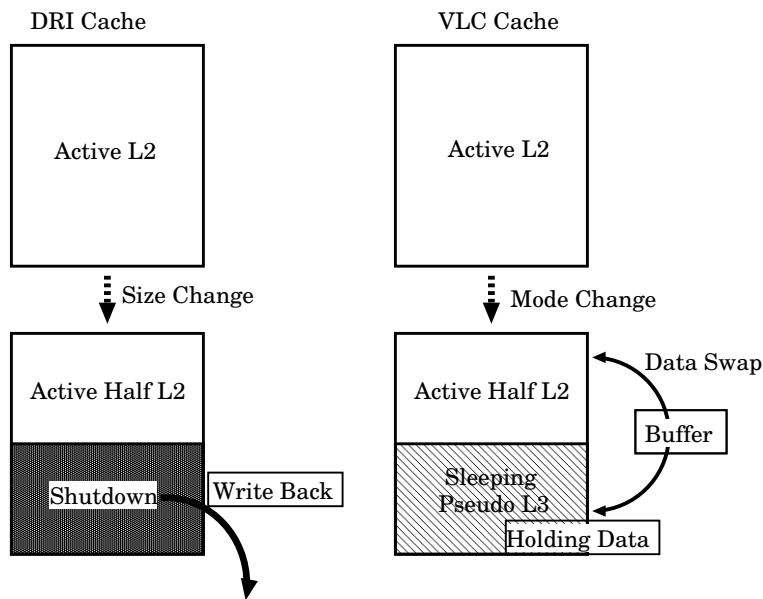


図 3.7: DRI キャッシュと可変レベルキャッシュの違い

256KBのL2キャッシュに可変レベルキャッシュを適用した場合、性能があまり必要でないと判断したときは、半分の128KBはスリープモードへと移行し、擬似的なL3キャッシュとして動作する。この時、擬似L3キャッシュは排他的キャッシュとして動作させる。

図 3.8 に可変レベルキャッシュの概要図を示す。可変レベルキャッシュは主に、DRIキャッシュの回路に再アクセスユニット(re-access unit)とバッファ(buffer)を加えた形で構成される。再アクセスユニットとバッファは共にL2とL3にキャッシュを分割したときに利用し、再アクセスユニットはL2キャッシュでキャッシュミスをした場合に、L3キャッシュ領域をスリープモードから通常モードへ切換え、L3キャッシュにアクセスする際に用いられる。バッファは排他的キャッシュとしてL2キャッシュのリプレイスデータをL3キャッシュへ書き戻すときに用いる。

以降では、全てのラインがアクティブの時に「通常モード」、キャッシュの半分をスリープモードにし、排他的キャッシュとして動作するときを「低消費エネルギーモード」と呼ぶ。

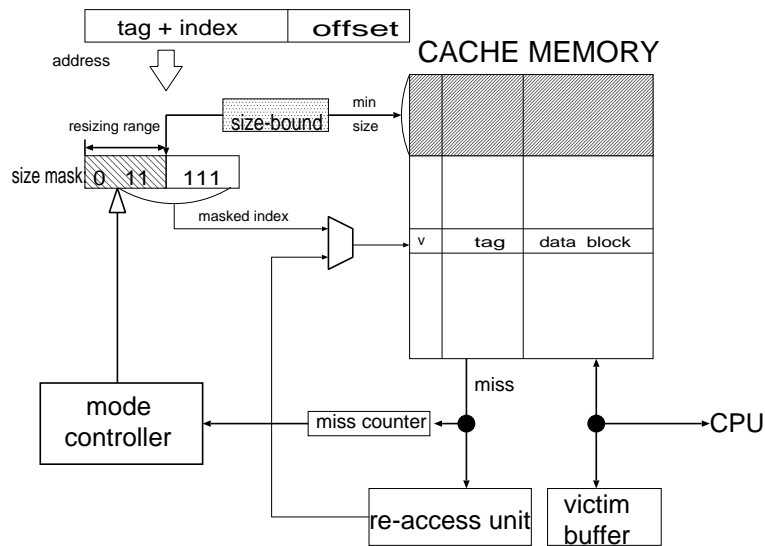


図 3.8: 可変レベルキャッシュ

### 3.2.2 排他的キャッシュ(Exclusive Cache)

排他的キャッシュは AMD 社が開発したキャッシュアーキテクチャで、L1 キャッシュと L2 キャッシュのデータを排他的にすることでキャッシュメモリを有効に利用する手法である。

以降の説明で、キャッシュの構成を L1 キャッシュは 128KB、L2 キャッシュは 256KB と想定する。

従来のキャッシュでは、すべてのデータはまず L2 キャッシュに格納され、その後 L2 キャッシュから L1 キャッシュへとコピーされる。そのため図 3.9 の Normal Cache のように、データ K の読み出しが合った場合でも、L1 キャッシュ内の LRU を追い出し、L2 キャッシュからコピーするだけである。そのため、L1 と L2 の割り当てがあっても全体的なキャッシュ・サイズは L2 キャッシュに相当する 256K バイトであるといえる。それに対し、排他的キャッシュでは、図 3.9(B) のように、L1 キャッシュから L2 キャッシュへと書き戻されるデータを一度バッファに移し、その後図 3.9(C) に示すように、L2 キャッシュから必要なデータをロードし、最後に図 3.9(D) バッファのデータを L2 キャッシュに書き戻している (図 3.9(4))。このように、データを L1 と L2 との間で交換する事 (排他的に処理する事) で全体的なキャッシュ・サイズが  $128+256=384$ K バイトとなり、キャッシュサイズを有効に活用できる

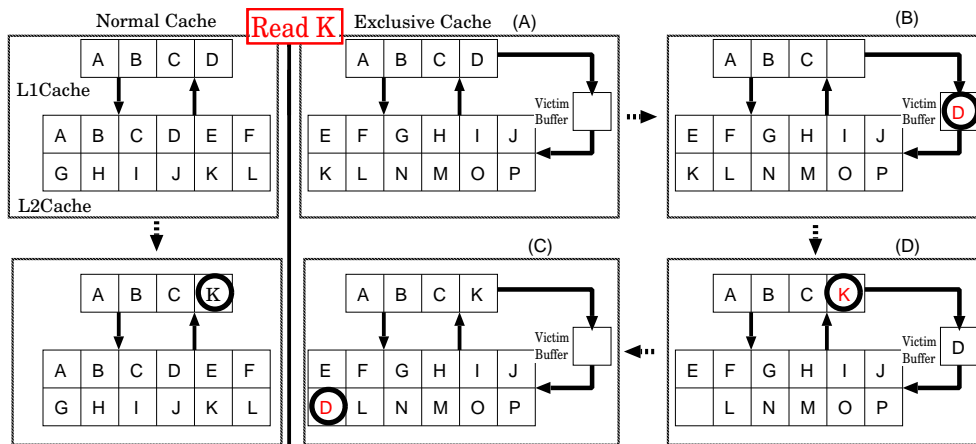


図 3.9: 通常キャッシュと排他的キャッシュ

### 3.2.3 可変レベルキャッシュの動作詳細

通常モード時は DRI キャッシュと同様、通常のキャッシュとして動作する。そして低消費エネルギーモードへの切替条件を満たした場合は、低消費エネルギーモードへと切替わる。その際、図 3.8 に示すマスクのビット幅を変更し、参照するインデックスに制限をかけ、キャッシュメモリの半分をスリープモードへと切替える。また、可変レベルキャッシュは DRI キャッシュと異なりスリープモードに切替える部分のデータを書戻さなくてよいという利点がある。低消費エネルギーモード時はマスクによってキャッシュメモリへのアクセスに制限をかけ、容量が通常モード時の半分になっている L2 キャッシュの部分へアクセスを行う。キャッシュミスをした場合、L3 キャッシュへと書戻すデータを図 3.8 に示すバッファへと移動し、同じく図 3.8 に示す再アクセスユニットにより L3 キャッシュの部分へとアクセスを行う。このとき、L3 キャッシュへのアクセスはスリープ状態の L3 キャッシュを活性状態に起こしてからアクセスされるため、L3 キャッシュのアクセス・レイテンシは L2 キャッシュへのアクセス・レイテンシにモード切替レイテンシを加算したものとなる。L3 キャッシュでキャッシュヒットした場合は、ヒットしたデータを L2 キャッシュへと移動し、L3 キャッシュの部分でもキャッシュミスした場合はメモリにアクセスを行う。バッファに入れられたデータはキャッシュアクセスの完了後、バックグラウンドで下位の記憶層に書戻しされる。

通常モードへの切替条件を満たした場合は、低消費エネルギーモードか

ら通常モードへと切換えられる。このとき、キャッシュメモリ内のデータの配置が変わってしまうためキャッシュメモリ内のデータをメモリへと書戻し、マスクのビットをすべて1にし、インデックスへの制限を解除する。可変レベルキャッシュはこのように動作を行い、低消費エネルギーと高性能を両立させる。

### 3.2.4 書き戻しペナルティ軽減手法

可変レベルキャッシュは動的に要求性能を判断し、通常モードと低消費エネルギーモードを切換えている。通常モードから低消費エネルギーモードへと切換える際はデータを下位層に書き戻す必要はないが、低消費エネルギーモードから通常モードに切換える際は、キャッシュ内のデータの配置が通常モードでの本来の位置と異なるため、データを下位層に書き戻す必要がある。

index	tag	
000	01101	← LA
001	10100	
010	00010	
011	11000	
100	00000	← LB
101	10101	
110	00111	
111	10011	

L2

図 3.10: 通常モードへ切換わる前のキャッシュ

そこで、低消費エネルギーモードから通常モードに切換える際のペナルティを低減する手法が提案されている [10]。

モードを切換えた時にダーティビットの立っている不良配置ラインの書戻しを行わない方法である。その代わりに、通常モードに切換えた後も不良配置ラインがキャッシュ内に存在しているため、キャッシュミスが発生したときは他方のバンクもアクセスする必要がある。しかし、キャッシュ

ミスの際に他方のバンクもアクセスすることはキャッシュアクセスのレイテンシと消費エネルギーの増大につながる。

そこで、可変レベルキャッシュの各セットに他方のバンクの「対応するセット (corresponding set)」に格納されている不良配置ラインの個数を記録する NoD カウンタを追加する。「対応するセット」とは、それぞれのバンクの最上位ビットを除いたインデックスが一致するセットを指す。NoD カウンタの値の最大値は連想度と等しくなるため、1つの  $n\_different$  につき  $\log_2(\text{連想度})$  のビット数が必要となる。そのため、キャッシュメモリのハードウェアサイズが大きくなるが、従来の可変レベルキャッシュと比較して 16way の場合でも 1つのセット、すなわち 16 ラインにつき 4ビットであるため、1セット当たりのビット数と比較すると十分に小さい。本手法を具体的な例を用いて説明する。図 3.11 は低消費エネルギーモードで動作して

おり、不良配置ラインが混在している状態を示している。図 3.11 の A0-1 ~ A1-4 は bankA に存在すべきライン、B0-1 ~ B1-4 は bankB に存在すべきラインである。つまり、この例では四角で囲った「B0-1, A0-2」が不良配置ラインである。この状態で通常モードへの切換条件を満たした場合、bankA, B が統合し、通常モードに遷移する。必要なラインが不良配置ラインである「A0-2」の場合、次の手順でアクセスを行う。

ステップ 1 : インデックス 00 にアクセスするが、キャッシュミスとなる。  
ステップ 2 : インデックス 00 の NoD カウンタを参照する。値が 0 でない為、「A0-2」が他のバンクにある可能性がある。そこで、バンク B の対応するセット (インデックス 10) に二次アクセスをし、ヒットとなる。

NoD カウンタが 0 になれば、他方のバンクの対応するセットに不良配置ラインがなくなったことになり、2次アクセスを行うことなく、通常のキャッシュと同様に下位層へアクセスを行う。このようにして、配置が混ざったデータもプログラムの実行時間と共に本来の配置へと移動し、徐々に通常の状態へと戻っていく。

また、1次アクセスのキャッシュミスの際に下位の記憶層へとリプレースするラインはアクセスされたセット内の不良配置ラインを優先し、不良配置ラインが無ければ通常通り LRU アルゴリズムで行う。これはできる限り早く不良配置ラインを追出し、通常のキャッシュの動作に戻すためである。

このようにして可変レベルキャッシュはキャッシュ必要容量を判断し、動的に通常モードと低消費エネルギーモードを切換えることによってシ

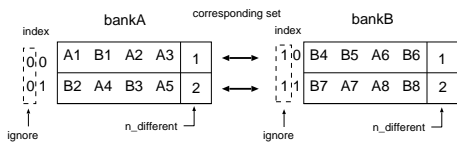


図 3.11: 低消費エネルギーモードでの動作

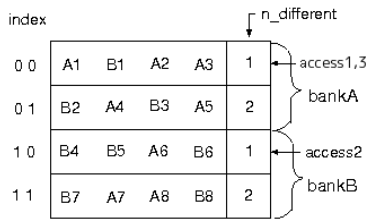


図 3.12: 通常モードでの動作

シングルプロセッサでは通常キャッシュと比較して約 30%の性能向上が得られた。本論文では，更なる低電力化を図る為，可変レベルキャッシュに改良を行った。行った 2 種類の改良を次々章より述べる。



## 4 評価方法

本論文では、可変レベルキャッシュについて、実行時間と消費エネルギーの評価を行った。本章ではその評価に用いた方法について述べる。尚、他手法との比較については、文献 [6][7] において可変レベルキャッシュの優位が既に示されているため、本論文では行わない。

### 4.1 エネルギー遅延積

NUAL キャッシュ、DRI キャッシュとの比較は既に文献 [10] で有用性が示されている。また、DRI キャッシュは L1 命令キャッシュに適用させるために提案されているため、L2 キャッシュに適用させる可変レベルキャッシュとは性質が違う。そのため、今回比較は行わなかった。

評価指標として通常、高性能なプロセッサを評価する場合には実行時間を見比べればよく、低消費電力プロセッサにおける評価では主に消費エネルギーを見比べればよい。しかし、今回は高性能かつ低消費エネルギーのプロセッサを作成することを目標としているため、実行時間と消費エネルギー両方について見比べる必要がある。そのため、高性能かつ低消費エネルギーのプロセッサ評価を行う場合によく用いられるエネルギー遅延積を用いる。エネルギー遅延積は消費エネルギー  $E$  と実行時間  $T$  を用いて以下の式で求めることができる。

$$\text{エネルギー遅延積} = E \times T \quad (1)$$

このエネルギー遅延積が小さいほど低消費エネルギーと高性能の両立が達成できている。エネルギー遅延積では性能に依存しない消費エネルギーに性能に依存する実行時間をかけているため、消費エネルギーの指標とは異なり、性能に依存する値となる。また、エネルギー遅延積は消費エネルギーに実行時間の 2 乗をかけても求めることができる。そのため、消費エネルギーと性能について評価する場合は性能の方が重要視された指標となるが、今回は消費エネルギーと性能についての評価であるのでエネルギー遅延積を用いた。

### 4.2 消費エネルギー

消費エネルギーについては、文献 [11] [12] を参考に以下の計算式を用いて計算した。

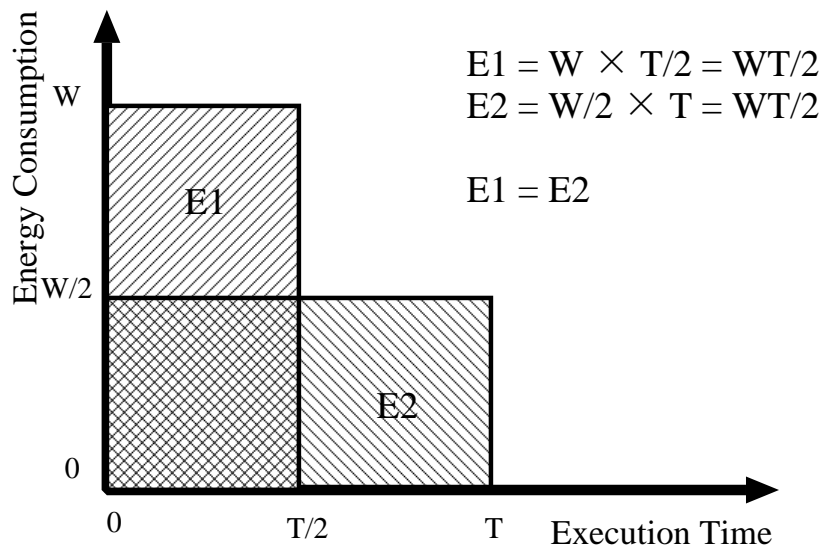


図 4.13: 実行時間と消費エネルギー

可変レベルキャッシュにおける動的エネルギーはL2 キャッシュへの通常アクセスとスリープモードからの復帰の2種類に分類される。よって可変レベルキャッシュの動的エネルギーは

$$E_{dynamic} = E_{access} + E_{wakeUp} \quad (2)$$

となる。式(2)中の  $E_{access}$  は下記のように計算される。

$$E_{access} = E_{one\_access} \times N_{access} \quad (3)$$

ここで、 $E_{one\_access}$  は1回のアクティブ状態のキャッシュアクセスに必要なエネルギー、 $N_{access}$  はキャッシュアクセスの回数である。スリープモード中の可変レベルキャッシュにおいて1つのメモリアクセス命令を処理する場合、下記のとおり複数回のアクセスが必要になる。

1. アクティブ状態のセットにアクセス。
2. 活性化させたセット(疑似L3)にアクセス。
3. 疑似L3でヒットした場合、ラインをスワップするために2回のアクセス。

$N_{access}$  はこれらのアクセス回数の総和である。

$$E_{wakeUp} = E_{one\_wakeUp} \times N_{way} \times N_{wakeUp} \quad (4)$$

式(2)中の  $E_{wakeUp}$  は上記の様に計算され、この時  $E_{one\_wakeUp}$  は1回スリープ状態のセットを活性化するために必要なエネルギーと蓄積された電荷の持つエネルギーの総和である。 $N_{way}$  は可変レベルキャッシュの連想度、 $N_{wakeUp}$  はスリープ状態のセットを活性化させた回数である。本節末にも記述するが、今回の電力評価においては主な周辺回路以外の回路の電力は評価に入れていない。また、活性化状態とスリープ状態との遷移時の電荷をチャージ・ディスチャージする際にかかるエネルギーは算出が困難なため、SRAMの最大周波数での動作時の動的エネルギーを用いた。次に静的エネルギーは下記の計算式で計算される。

$$E_{static} = E_{set\_leak} \times N_{set} \times N_{way} \times Cycles \div Freq \quad (5)$$

ここで、 $N_{set}$  は可変レベルキャッシュのセット数、 $Cycles$  はプログラム実行にかかったクロックサイクル数、 $Freq$  はCPUのクロック周波数である。また、1セットにおけるリークエネルギー  $E_{set\_leak}$  は下記の計算式で計算される。

$$E_{set\_leak} = R_{sleep} \times E_{sleep\_leak} + (1 - R_{sleep}) \times E_{active\_leak} \quad (6)$$

ここで、 $R_{sleep}$  は可変レベルキャッシュのスリープモードに落ちているラインの平均割合、 $E_{sleep\_leak}$ 、 $E_{active\_leak}$  はそれぞれスリープ状態と活性状態での1ラインにおけるリークエネルギーである。

この評価式において、通常キャッシュにおいてはスリープモードが存在しないため  $E_{dynamic} = E_{access}$  とし、 $E_{static} = E_{active\_leak}$  となる。最後に、可変レベルキャッシュの電力については可変レベルキャッシュ実装に必要な追加回路の電力も考慮に入れるため、消費エネルギーの総和に以下のエネルギーを加算する。

$$E_{P\_circuit} = E_{controller} \times N_{access} + E_{buffer} \times N_{wakeUp} \quad (7)$$

ここで、 $E_{P\_circuit}$  はL2キャッシュに追加した周辺回路のエネルギーの総和、 $E_{controller}$  はモード切替コントローラが1回の動作に必要なエネルギー、 $E_{buffer}$  はビクティムバッファへのデータ待避及び読出しを1回ずつ実行する際に必要なエネルギーである。周辺回路のリークエネルギーは非常

に小さく、個別の評価が困難なため、考慮しないものとする。その他の追加回路、具体的にはミスカウンタ、再アクセスユニット、および改良案 2 で必要になる 4 ビットカウンタについては、Verilog HDL を用いたハードウェア設計を行い、トランジスタレベルでの電力シミュレーション評価の結果、コントローラ部分とビクティムバッファの消費エネルギーと比較して 0.1% 未満であることがわかったため、電力評価モデル式では考慮しないものとする。

なお、L2 キャッシュからメインメモリへの書戻しの際の DRAM やバスの電力は測定が困難であるため評価に入れないものとする。また、評価式中のスリープ状態と活性状態との遷移に必要なエネルギーは算出が困難なため、仮定値として CACTI で求められる最大周波数での動作時の動的エネルギーの値を用いた。ただし、この仮定値を 10 倍、100 倍、1000 倍に増やしたことによる結果全体への影響は 0.01% 以下とごく微少であり、電力見積りに与える影響は十分に小さいといえる。

以上の式で求めた、実行サイクル数と消費エネルギーの積を求め比較を行う。

### 4.3 シミュレータ

本論文では独自に開発した L2 キャッシュ・シミュレータを用いて評価を行う。本シミュレータはプロセッサシミュレータ M5[14] を利用して、各ベンチマーク実行中の L2 キャッシュへのアクセス情報のみをダンプさせたファイルを用いる。ダンプさせた L2 キャッシュアクセス情報を用いて L2 キャッシュの動作のみをトレースするトレース・ドリブン・シミュレータである。

### 4.4 実験環境

可変レベルキャッシュに関して性能と消費エネルギーの評価を行うため、開発した L2 キャッシュシミュレータに可変レベルキャッシュを実装した。ここで可変レベルキャッシュは L2 の統合キャッシュに実装する事を想定した。プロセッサ構成を表 4.1 に示す。

今回の評価において、可変レベルキャッシュは、1 つの 512KB の L2 キャッシュとして扱うモード 1 と、L2 キャッシュが 256KB、L3 キャッシュが 256KB である排他的キャッシュとして動作するモード 2、L2 キャッシュ

表 4.1: シミュレーションのキャッシュに関するパラメータ

キャッシュ容量	
L1 命令-cache	32KB(64B/entry, 1way, 512entry)
L1 データ-cache	32KB(64B/entry, 1way, 512entry)
L2 cache	512KB(64B/entry, 4way, 2048entry)
ヒット・レイテンシ	
L1 cache	1 cycle
L2 cache	10 cycle
主記憶	250 cycle
スリープ復帰のオーバーヘッド	
レイテンシ	10cycle

が 128KB, L3 キャッシュが 128KB である L2 キャッシュの排他的キャッシュ, L4 キャッシュが 256KB である L2, L3 キャッシュの排他的キャッシュとして動作するモード 3 の 3 種類を動的に切替える. 可変レベルキャッシュは 8192( $2^{13}$ ) サイクル毎にキャッシュヒット率を測定し, キャッシュミス率が閾値以下ならば小さなモードに (モード 2 ならモード 1 へ), 閾値を超えていれば大きなモードに (モード 2 ならモード 3 へ) 切替えるものとする. また, キャッシュミス率の閾値は 30% とする. なお, これらの閾値は各手法についてシミュレーション評価により実験的に求めた最適値である. 通常のキャッシュと可変レベルキャッシュを組み込んだシミュレータ上でベンチマークプログラムを実行し, それぞれの手法の性能を測定した. ベンチマークプログラムは表 4.2 に示すように, SPEC2000 [15] の 7 種類を使用し, 評価を行った.

表 4.2: ベンチマーク

ベンチマーク	処理内容
ammp	計算科学
bzip2	圧縮
crafty	チェスゲーム
equake	地震波電波シミュレーション
gap	群論, 通訳
gcc	C 言語コンパイラ
gzip	圧縮
mcf	組み合わせ最適化
vpr	FPGA の配置配線

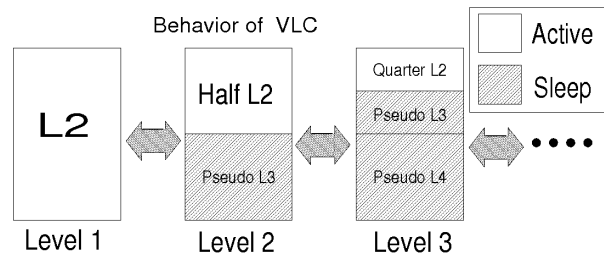


図 5.14: 可変レベルキャッシュのモード遷移

## 5 改良 1 : モード追加

可変レベルキャッシュは性能維持に優れた手法であることが明らかにされている。[10] しかしその反面，スリープモードへ移行可能な領域が L2 キャッシュの 50% であるため，諸費エネルギー削減効率は最大 50% 程度とあまり高くないという問題点がある。この問題を解決する為に，本章では書き戻しペナルティ軽減手法を拡張し，可変レベルキャッシュにより低電力なモードを追加することで，消費エネルギー削減効率の向上を図る。本章以降，可変レベルキャッシュの通常モードを「モード 1」，低消費エネルギーモードを「モード 2」，新たに追加するより低電力なモードを「モード 3」と呼ぶ。モード 3 を追加した可変レベルキャッシュ[16] のモード遷移図を図 5.14 に示す。

図 5.15 は提案する拡張 NoD カウンタとその例であり，可変レベルキャッ

		index				NoD 2	NoD 1
Bank A	0 0	A0-1	C0-1	A0-3	B0-3	1	1
Bank B	0 1	B0-1	B0-2	A0-4	B0-4	1	0
Bank C	1 0	A0-2	C0-2	C0-3	C0-4	0	1
	1 1	C1-1	C1-2	C1-3	C1-4	0	0

図 5.15: 提案手法:拡張 NoD カウンタ

シユがモード 1, モード 2 を経てモード 3 に移行した後の状態を示す. 四角で囲った 4 つのラインが不良配置ラインである. 従来の書き戻しペナルティ軽減手法との違いは, 追加された NoD カウンタ (図 5.15 中の NoD2) である. NoD1 と NoD2 はそれぞれ異なるバンクに対応している.

従来の書き戻しペナルティ軽減手法では NoD カウンタを一つしか持たない為, 不良配置ライン「A0-4」にアクセスすることが出来なかった. 従来手法を用いて「A0-4」にアクセスすると, 可変レベルキャッシュはインデックス 10 (バンク C の先頭セット) を参照してしまう. これは, NoD1 が対応しているのはバンク C (インデックス 10 と 11) であり, バンク B には対応出来ないからである. インデックス 10 は「A0-4」を保持しておらずキャッシュミスとなるが, 「A0-4」はインデックス 01 に存在している為, キャッシュ動作不良を引き起こす. 拡張 NoD カウンタは次の手順によって不良配置ライン「A0-4」へアクセスすることが出来る.

ステップ 1, 2 は従来手法と同様であり, キャッシュミスが発生する.

ステップ 1 : インデックス 00 にアクセスするが, キャッシュミスとなる.

ステップ 2 : インデックス 00 の NoD 1 カウンタを参照する. 値が 0 でない為, 不良配置ラインが他のバンクにある可能性がある. そこで, バンク C の対応するセット (インデックス 10) に二次アクセスを行うが, キャッシュミスとなる.

ステップ 3 : インデックス 00 の NoD 2 カウンタを参照する. その値が 0 でない為, 対応するバンク B のインデックス 01 へ三次アクセスを行い, ヒットとなる.

ステップ 4 : 不良配置ライン「A0-4」と「B0-3」を入れ替え, NoD 2 カウンタの値を 1 減らす.

この様にして拡張 NoD カウンタを用いることで, 可変レベルキャッシュ

はモード 3 以上のモードにも対応可能となる。NoD カウンタの数はモード数が増える毎に増加するが、NoD カウンタのハードウェアコストは小さく、インデックス毎に  $\log(\#way) + 1$  ビット必要となる。(例として、4 セットアソシアティブキャッシュの場合、インデックス毎に 3 ビットのカウンタが必要となる。)

## 5.1 評価結果

提案する改良 1 をシミュレータに実装し、評価を行った。図 5.16, 図 5.17, 図 5.18 はそれぞれ実行時間, 消費エネルギー, ED 積の評価結果を示す。図中の「Previous」はモード可変域をモード 1 とモード 2 に限定した従来の可変レベルキャッシュ, 「Proposed」はモード 3 を実装した提案手法である。各評価結果は通常キャッシュの結果で正規化したものである。図 5.16 の結果から、提案手法は従来の可変レベルキャッシュと比較して実行時間の増加は平均 1% 未満であることがわかった。しかし、ベンチマーク Mcf においては通常キャッシュと比較して約 19% の実行時間の増加がみられた。これは不良配置ラインへの二次アクセスが頻発したためだと考えられる。その他のベンチマークについては、大きく分けて 2 パターンの傾向が見られた。一つ目のパターンは実行時間の殆どをモード 1 で実行するベンチマーク (Bzip2, Gap, Gzip, Vpr) である。もう一つのパターンは実行時間の殆どをモード 3 で実行するベンチマーク (Ampmp, Crafty, Equake, Gcc) である。

図 5.17 の結果から、提案手法は従来の可変レベルキャッシュと比較して平均約 18% の消費エネルギーを削減することがわかった。この結果から、モード 3 の追加によって提案手法はより多くの消費エネルギーを削減出来ることがわかった。

図 5.18 の結果から、提案手法は従来の可変レベルキャッシュと比較して平均約 17%, 通常キャッシュと比較して平均約 22% の性能向上がみられた。しかし、その後新たな電力評価モデルを用いて行った予備評価によって、提案手法を用いてモード 3 を実装した可変レベルキャッシュの動的エネルギーが通常のキャッシュの約 170% にまで増加することが分かった。



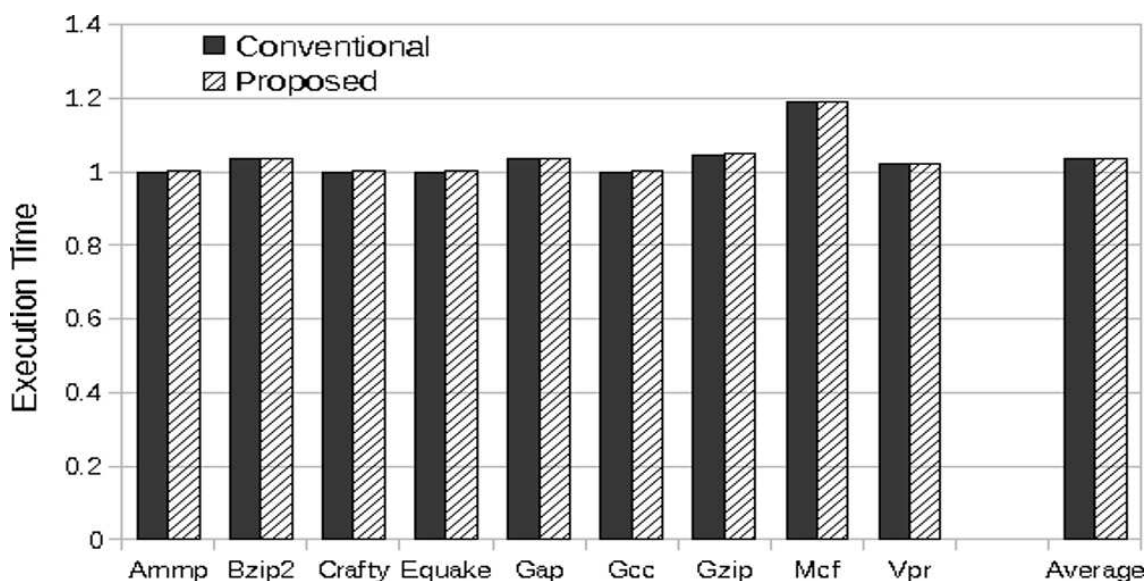


図 5.16: 実行時間

## 6 改良2：動的エネルギーの削減

本章ではモード3の追加により発生した可変レベルキャッシュの問題点と、それを解決する為に行った二つ目の改良手法である可変レベルキャッシュの動的エネルギー削減手法 [17] について述べる。

### 6.1 動的エネルギーの増加

3.2.4節で述べた書き戻しペナルティ軽減手法は下位モードから上位モードへの移行時（例えば、モード2からモード1への移行）に発生する不良配置ラインの書き戻しによる性能低下の防止に有用である。しかしその反面、モード1，モード2において二次アクセスが必要になる。特に、ヒット率の変動が激しいプログラムにおいてはモード変更が多発するため、下位モードで不良配置ラインが発生した後、上位モードへ移行し不良配置ラインへの二次アクセスを行う結果、二次アクセス数が増加してしまう。モード3を追加する以前の可変レベルキャッシュにおいては、二次アクセスによる動的エネルギー増加は問題にならなかった。これは、文献 [10] の状況ではリークエネルギーが消費エネルギーの多くを占めており、当研究室もまた、高リークエネルギー設定の電力評価モデルを用い

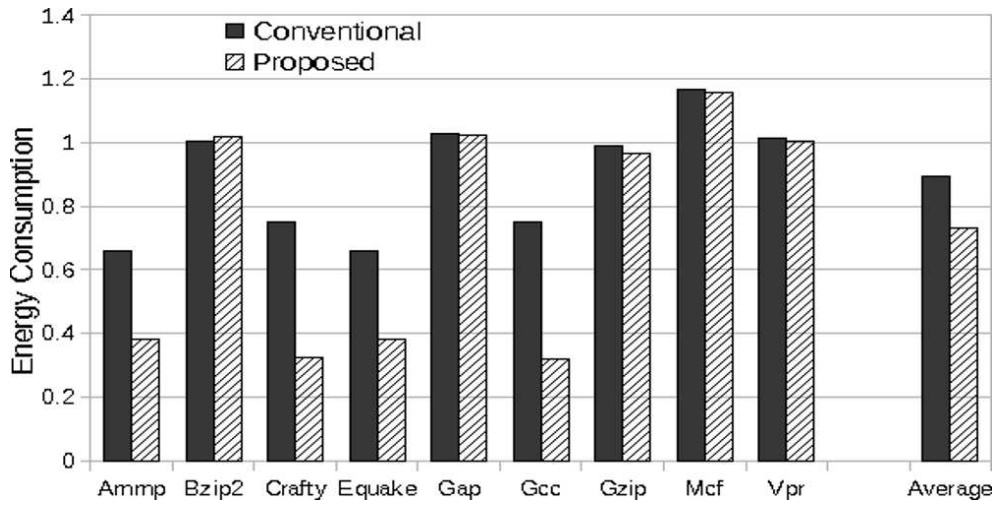


図 5.17: 消費エネルギー

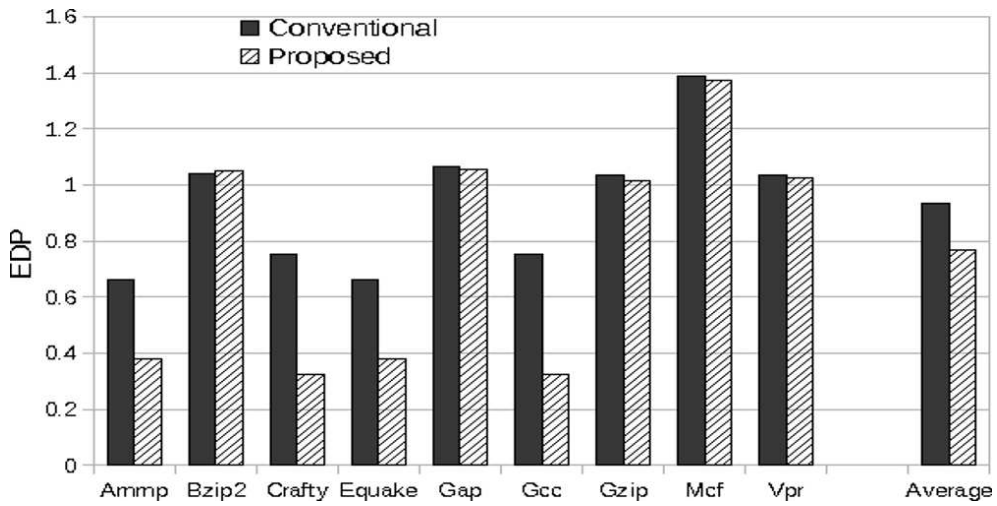


図 5.18: ED 積

て可変レベルキャッシュの評価を行っていた為である。しかし、近年の CMOS デバイス技術の進歩によってトランジスタの消費エネルギーにおけるリークエネルギーの割合が減少しつつある。よって当研究室も電力評価モデルを新しい技術に沿ったものに見直した。

予備評価を行った結果、モード 3 を実装した可変レベルキャッシュの動的エネルギーは、通常のキャッシュと比較して約 170% にまで増加することがわかった。これはモード 1，モード 2 時における二次アクセス数の増加と、モード 2，モード 3 時におけるスリープ領域へのアクセス数の増加が原因である。常時ヒット率の高いプログラムにおいては可変レベルキャッシュはモード 3 で動作し続ける為、消費エネルギー削減効果が高い。しかし、ヒット率の上下が激しいプログラムにおいては、可変レベルキャッシュはモード 1 またはモード 2 で動作することが多くなり、二次アクセス回数が増加する。その結果、可変レベルキャッシュの動的エネルギーは増加し、効果的に消費エネルギーを削減することができなくなる。この問題を解決する為に、本章では動的エネルギー増加の主な原因である不良配置ラインへの二次アクセス方法の改良を行った。

## 6.2 二次アクセス時のアクセスバンク制限

可変レベルキャッシュの動的エネルギーを削減する為に、著者はその原因である二次アクセス方法の改良を考えた。いくつかのベンチマークプログラムを用いて予備評価を行った結果、二次アクセス発生時にアクセスされたセット内に存在する不良配置ラインの数は平均 1 ラインまたは 2 ラインであることがわかった。二次アクセス時にアクセスする必要があるウェイは不良配置ラインを持つウェイのみである。これは、不良配置ラインを持たないウェイへの二次アクセスは不要であることを意味する。そこで本研究では二次アクセス時にアクセスされるウェイを限定することで、二次アクセスにかかる不要な動的エネルギーを削減することを考える。

不良配置ラインを持つウェイのみに二次アクセスする為には、どのウェイが不良配置ラインを持つのかを記憶しておく必要がある。従来の NoD カウンタでは不良配置ラインを持つセットのみを記憶する為、セット内のどのウェイが不良配置ラインを持つかは判断出来ない。そこで、ウェイ単位で不良配置ラインの有無を記憶する為の改良を NoD カウンタに施すことを考える。

	index	Way 1	Way 2	Way 3	Way 4	VSL 2 bitmap	VSL 1 bitmap
Bank A	00	A0-1	C0-1	A0-3	B0-3	0010	1000
Bank B	01	B0-1	B0-2	A0-4	B0-4	0001	0000
Bank C	10	A0-2	C0-2	C0-3	C0-4	0000	0100
	11	C1-1	C1-2	C1-3	C1-4	0000	0000

図 6.19: 提案手法:VSL Bitmap

図 6.19 は改良をした NoD カウンタであり、ビットマップテーブルを用いることで、各ウェイが不良配置ラインを保持しているか否かを判別出来るようになった。本論文ではこの提案手法を「Violated Set Lines Bitmap(VSL Bitmap)」と呼ぶこととする。

各ビットマップは各ウェイに対応しており、不良配置ラインを保持するウェイに対応したビットマップには 1 が立っている。可変レベルキャッシュはこの VLS Bitmap を用いて不良配置ラインを保持するウェイのみに二次アクセスをすることが出来る。例をあげると、図 6.19 中の不良配置ライン「A0-2」へモード 1 の可変レベルキャッシュがアクセスする場合、次の手順でアクセスを行う。

ステップ 1 : インデックス 00 にアクセスするが、キャッシュミスとなる。

ステップ 2 : VLS 1 Bitmap を参照し、Way 1 に対応する箇所の値が 1 である。これはインデックス 10 の Way 1 に不良配置ラインがある可能性を示している。

ステップ 3 : インデックス 10 の Way 1 にのみ二次アクセスを行う。

この VSL Bitmap はセット毎にウェイ数分のビットを疑似階層の数だけ必要である。しかしながら、ハードウェアコストは 512Kb 4-way セットアソシアティブの場合で、キャッシュメモリのハードウェア量と比較して 0.01%未満と非常に小さい。したがって、十分に無視できるハードウェア量だといえる。

### 6.3 評価結果

提案する改良2をシミュレータに実装し、評価を行った。図6.20, 図6.21, 図6.22はそれぞれ実行時間, 消費エネルギー, 動的エネルギーの評価結果を示す。図中の「NoD Counter」は5章で説明した改良1を実装した可変レベルキャッシュ, 「VSL Bitmap」は本章で提案するVSL bitmap (改良2)を実装した可変レベルキャッシュである。各評価結果は通常キャッシュの結果で正規化したものである。

図6.20の結果から, 提案手法であるNoD Counter, VSL Bitmap共に通常のキャッシュと皮革して実行時間の増加は平均1%未満であることがわかった。また, VSL BitmapはNoD Counterと比較して実行時間は増加しなかった。さらに, 殆どのベンチマークにおいて, 可変レベルキャッシュの実行時間は通常キャッシュより小さくなった。これは可変レベルキャッシュの疑似階層キャッシュを作る動作によって擬似的にウェイ数が増加し, その結果キャッシュのヒット率が向上したためである。特に, Gzipにおいては通常キャッシュと比較して約3.6倍のヒット率向上がみられた。

図6.21の結果から, VSL Bitmapは消費エネルギーを通常キャッシュと比較して平均約34%, NoD counterと比較して平均約11%削減することがわかった。特にBzip2, Gap, Mcf, Vprにおいては大きな効果が見られた。これら4つのベンチマークはキャッシュヒット率が低く, ウェイ数の増減に影響を受けやすい。そのため, これらのベンチマークは実行時間の殆どをモード1で実行し, 多くの二次アクセスを起こす。この動作によってNoD Counterでは消費エネルギーが最大35%増加するが, VSL Bitmapでは最大8%の増加にとどまった。

図6.22の結果から, VSL BitmapはNoD Counterと比較して動的エネルギーを平均約18%削減することがわかった。特に, Mcfにおいては37%の動的エネルギーを削減することがわかった。ヒット率の低いベンチマークであるBzip2, Gap, Mcf, Vprでは実行時間の殆どをモード1で動作し, 多くの二次アクセスを起こす。しかし, VSL Bitmapは二次アクセス時にアクセスされるウェイ数を限定することで, 二次アクセスにかかる動的エネルギーの多くを削減することが可能である。この理由により, VSL BitmapはNoD Counterよりも多くの動的エネルギーを削減する結果となった。

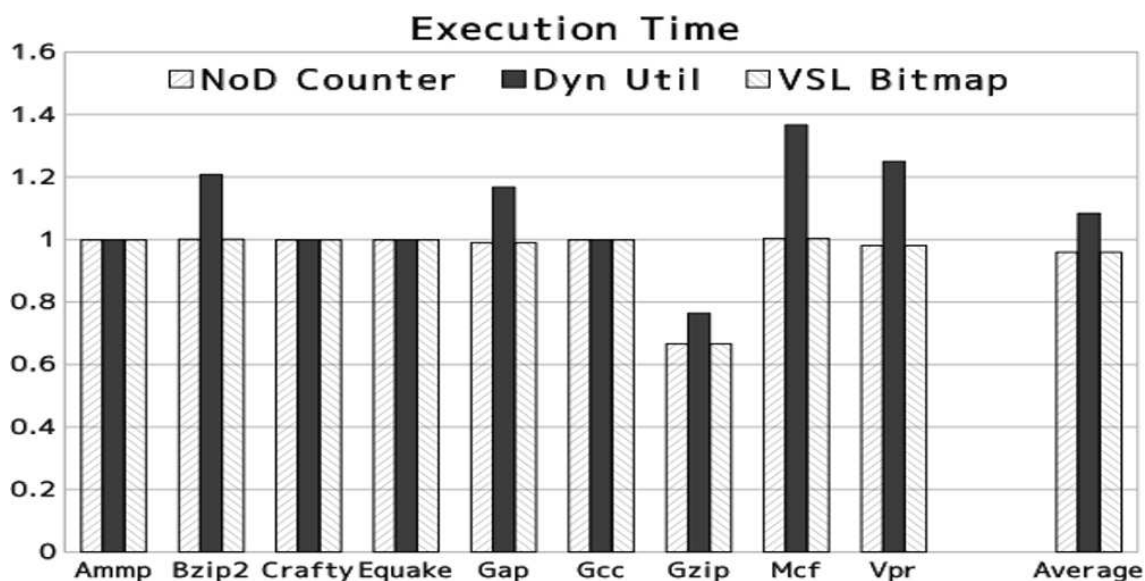


図 6.20: 実行時間

## 7 結論

本論文ではキャッシュリークエネルギー削減手法である可変レベルキャッシュの消費エネルギー削減効率を高める為に改良を行った。キャッシュの50%だったスリープモード領域を75%まで拡張させるモードを追加することで、消費エネルギー削減効率を平均約18%高めることが出来た。

しかし、スリープモード領域を拡張したことで二次アクセスの発生頻度が増大した結果、動的エネルギーが増加してしまった。この問題を解決する為に更なる改良を行った。二次アクセス時のアクセス先バンクを必要最小限に抑えることで、不必要なバンクへのアクセスを制限し、動的エネルギーを削減した。この結果、動的エネルギーを平均約18%削減することが出来た。

これらの改良を行った結果、シミュレーション評価によるとエネルギー遅延積(ED積)で比較して、従来の可変レベルキャッシュと比べて平均約34%性能向上することが明らかとなった。過去の文献から、可変レベルキャッシュは他の代表的な手法より優位である事が既に示されている為、本論文では比較評価は行わない。

今後の展望としては、可変レベルキャッシュの更なる改良、及びマルチコアプロセッサを対象とした可変レベルキャッシュの提案、実際のハード

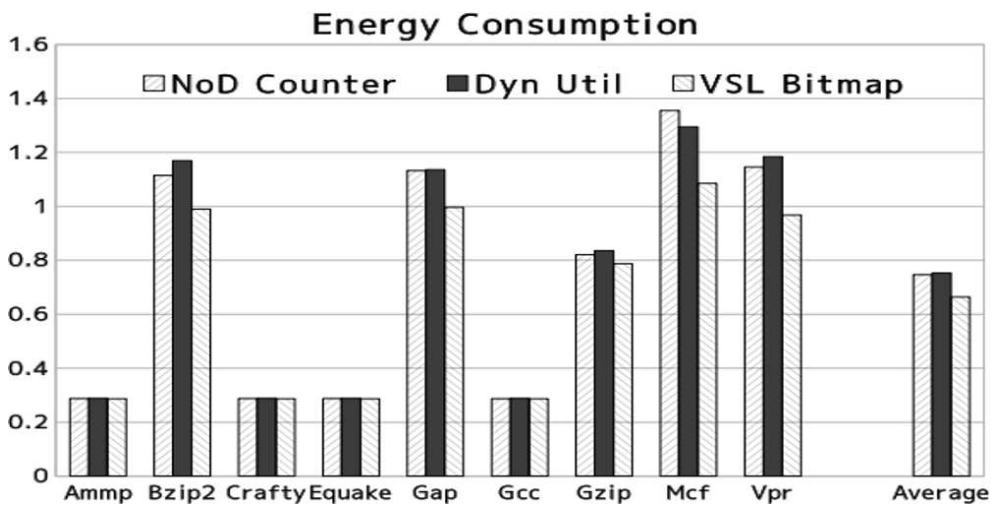


図 6.21: 消費エネルギー

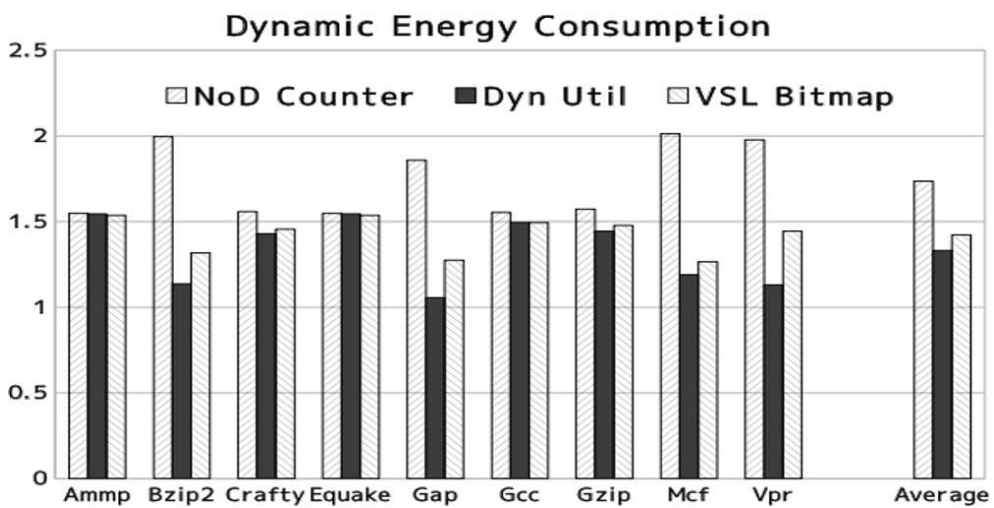


図 6.22: 動的エネルギー

ウェアを設計し，より詳細な評価を行うなどが挙げられる．



## 謝辞

本研究を行うにあたり，多くの助言をいただきました近藤利夫教授，大野和彦講師，並びにご指導，ご助言いただきました下さいました佐々木敬泰助教に深く感謝いたします。また，様々な局面にてお世話になりました計算機アーキテクチャ研究室の皆様にも心より感謝いたします。

## 参考文献

- [1] K. Flautner, N.S. Kim, S. Martin, D. Blaauw, and T. Mudge, “Drowsy Cache: Simple Techniques for Reducing Leakage Power,” Proc. of the 29th Int. Symp on Computer Architecture, pp. 148-157, May 2002.
- [2] N.S. Kim, K. Flautner, D. Blaauw, and T. Mudge, “Drowsy Instruction Caches; Leakage Power Reduction using Dynamic Voltage Scaling and Cache Sub-bank Prediction,” Proc. of the Int. Symp. on Microarchitecture, pp.219-230,
- [3] Akihito Sakanaka, Toshinori Sato, “Reducing Static Energy of Cache Memories via Prediction-Table-less Way Prediction” International Workshop on Power And Timing Modeling, September 2003.
- [4] 田中秀和, 井上弘士, モシニヤガ・ワシリー, “低消費電力を目的とした適応型ウェイ予測キャッシュとその評価,” 信学会技報, VLD2004-139, ICD2004-235, pp.13-18, March 2005.
- [5] S.H. Yang, M.D. Powell, B. Falsafi, K. Roy, and T.N. Vijaykumar, “An Integrated Circuit / Architecture Approach to Reducing Leakage in Deep-Submicron High-Performance I-Caches,” Proc. of the 7th Int. Symp. on High-Performance Computer Architecture, pp.147-157, February 2001.
- [6] 恩賀琢也, 佐々木 敬泰, 大野 和彦, 近藤 利夫, “キャッシュ階層動的切り替えによる低消費電力化”, 情処学研報, 2007-ARC-174, pp.115-120, August 2007.
- [7] 松原 伸幸, 佐々木 敬泰, 大野 和彦, 近藤 利夫, “高性能かつ低消費電力を実現する可変レベルキャッシュのモード切替アルゴリズムの改良と評価“, 信学会技報, CPSY2009-44, pp.7-12, December 2009.
- [8] Ying Zheng, Brian T. Davis, Matthew Jordan “Performance Evaluation of Exclusive Cache Hierarchies” IEEE International Symposium of Performance Analysis of Systems and Software, ISPASS, pp.89-96, September 2004.

- [9] Advanced Micro Deices, AMD, <http://www.amd.com/us-en/>. (Current June 2003).
- [10] 渡部 功, 佐々木 敬泰, 松原伸幸, 大野 和彦, 近藤 利夫 “モード切換オーバヘッドを低減した可変レベルキャッシュの提案と評価,” 情報処理学会 ACS 論文誌, pp48-60, August 2012.
- [11] 小宮礼子, 井上弘士, モシニヤガ・ワシリー, 村上和彰, “キャッシュ・リーク電力削減アルゴリズムに関する定量的評価,” 第 17 回回路とシステム軽井沢ワークショップ論文集, pp.235-240, April 2004.
- [12] 図子純平, 富山宏之, 高田広章, 井上弘士, “Drowsy キャッシュにおけるモード切替アルゴリズムの評価,” 情処学研報, 2006-ARC-170, pp.37-41, December 2006.
- [13] CACTI 5.3 Shyamkumar Thoziyoor, Naveen Muralimanohar, Jung Ho Ahn, and Norman P. Jouppi
- [14] Nathan L. Binkert, Ronald G. Dreslinski, Lisa R. Hsu, Kevin T. Lim, Ali G. Saidi, Steven K. Reinhardt “The M5 Simulator: Modeling Networked Systems, ”
- [15] “SPEC -Standard Performance Evaluation Corporation-,” URL: <http://www.spec.org/>.
- [16] Ko Watanabe, Takahiro Sasaki, Kazuhiko Ohno and Toshio Kondo “IMPROVEMENT OF WRITEBACK MECHANISM OF VARIABLE LEVEL CACHE, ” ITC-CSCC2012, July 2012.
- [17] Ko Watanabe, Takahiro Sasaki, Kazuhiko Ohno and Toshio Kondo “Reducing Dynamic Energy of Variable Level Cache, ” Int. Journal of Comp. and Elec. Eng., vol.5, no.6, pp.581-586, December 2013.