卒業論文

題目

キャッシュの可視化ツールを用いた
可変レベルキャッシュの改良

指導教員

佐々木敬泰

2014年

三重大学 工学部 情報工学科
計算機アーキテクチャ研究室

近藤　舞佳 (410819)

内容梗概

近年，プロセッサには高性能と低消費エネルギーの両立が求められている．特に，回路の微細化にともないキャッシュメモリ内でのリークエネルギーが年々増加しており，これを削減することが重要である．そこで我々は，キャッシュのリークエネルギー削減手法として可変レベルキャッシュを提案している．可変レベルキャッシュは，動的にキャッシュの要求性能を判断し，あまり性能が必要ないと判断したときにキャッシュメモリの半分をスリープモードに移行する．スリープ領域を 1 つ下位レベルの排他的キャッシュとして動作する事で，消費エネルギーの削減と性能の維持を両立させる手法である．しかし，従来の可変レベルキャッシュは実行した際のキャッシュミス率を基準に電力削減を行っており，キャッシュの挙動の詳細を考慮していなかった．そこで本研究では，キャッシュの動作を解析し可視化出来るツールを開発し，可変レベルキャッシュの解析を行う．またその解析結果を基に可変レベルキャッシュの改良を行ったところ，提案手法は従来手法よりも消費電力が平均約 14%，最大で約 37%削減された．また，性能低下は通常キャッシュと比べても平均 1%未満となった．

Abstract

Power dissipation is one major concern not only for mobile computing but also high performance computing, and achieving both low energy and high performance at the same time is required. Especially, it is important to reduce leakage energy consumed in a cache memory because power dissipation by leakage current is dominant factor in deep submicron technologies' and a cache memory consists of a large number of transistors. In order to reduce the problem, we proposes Variable Level Cache (VLC) which analyzes required cache performance dynamically and if it detects that the running program does not need so large cache capacity, half of the cache memory is put into standby mode and is treated as a lower level exclusive cache. Previous VLC switches by miss rate without considering in detailed behavior. We develop a visualizer that can analyze the behavior of the cache and evaluate VLC. Our improvement can reduce the total energy consumed in L2 cache using analysis result by 14% in average without significantly performance degradation compared to previous VLC.

目 次

図 目 次

表 目 次

1　はじめに

1.1　背景

現在，ノートパソコンやPDA，携帯電話等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，バッテリによる駆動時間が短くなってきている．そこで，これらの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．

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大別出来る．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主に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い動的消費エネルギーが削減される一方，MOSトランジスタのサブスレッショルドリークやゲートリークが増加する傾向にある．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ともない増大したキャッシュシステムのリークエネルギーの削減が重要となっている．そのため当研究室では低電力キャッシュ手法の一つとして可変レベルキャッシュ[1]を提案している．

1.2 研究目的

可変レベルキャッシュは，キャッシュサイズを要求性能により動的に変化させる手法である．あまり性能が必要ないと判断した際にはキャッシュの半分をスリープモードへと移行し 1 つ下位レベルの排他的キャッシュとして動作させることで電力削減と性能維持の両立させる．しかし，従来の可変レベルキャッシュは実行した際のミス率を基準に電力削減を行っており，キャッシュの挙動の詳細を考慮していなかった．そこで本研究では，キャッシュの動作を解析し可視化出来るツールを開発し，可変レベルキャッシュを評価する．またその解析結果を基に可変レベルキャッシュの改良を行ったところ，改良手法は従来手法よりも消費電力が平均約 14%，最大で約 37%削減された．

2 関連研究

2.1 DRI キャッシュ

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．例えば DRI キャッシュ[2] は，一定間隔（interval）でキャッシュミス数をミスカウンタ（miss counter）によりカウントする．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事が出来る．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）を掛ける事で参照するインデックスのビット幅を変更し，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 キャッシュは，実行プログラムにおけるキャッシュへの要求性能を自動的に検出し，キャッシュサイズを変更する事でリークエネルギーを削減している．

また，DRI キャッシュは電源を切る部分のキャッシュメモリを 1 つのバンクとして電源管理を行う事で制御を簡単にし，汎用の SRAM モジュールを用いることが出来るという利点もある．

2.2 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．したがって，電源を落とした部分に格納されていた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の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 可変レベルキャッシュ

DRI キャッシュの問題点を解決する手法として，当研究室ではメモリスタンバイモード [3]（以降スリープモードと呼ぶ）を利用し，かつ待機状

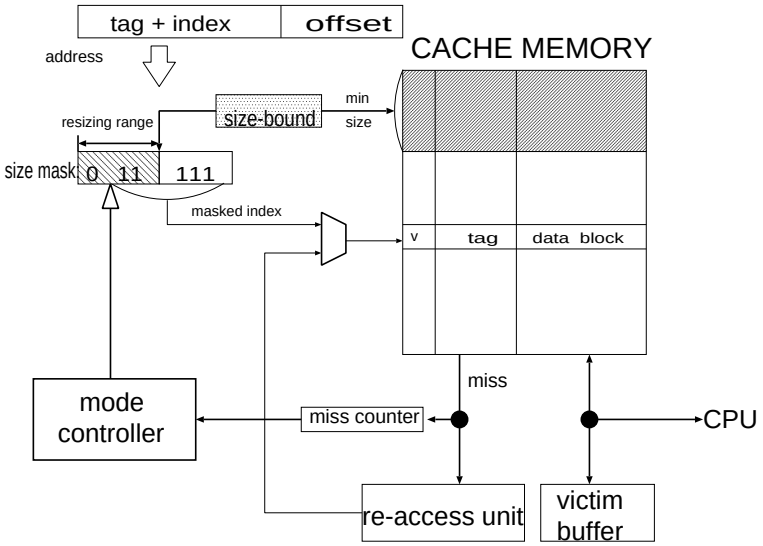

図 3.1: 可変レベルキャッシュのブロック図

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を提案している．可変レベルキャッシュはDRIキャッシュと同様に，ある一定間隔でキャッシュの必要としている容量を動的に判断し，キャッシュメモリの容量を増減させる．しかしながら，DRIキャッシュのように単純に容量を減少させると，キャッシュミス回数が増加してしまう．そこで，可変レベルキャッシュでは電力の供給を完全に停止するのではなく，スリープモードに移行する．しかし，スリープ領域にアクセスする場合，アクセス前に当該領域をアクティブにする必要があるため通常のアクセスより時間がかかる．そのため，スリープ領域にアクセスが頻発し性能が悪化する恐れがある．そこで，スリープ領域を１つ下位レ

ベルの排他的キャッシュ[4][5] とする事でスリープ領域へのアクセスを極力減らし，消費エネルギーを削減する．例えば，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時は，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させる．この時，この擬似 L3 キャッシュは排他的キャッシュとして動作させる．図 3.1 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) とビクティブバッファ(victim buffer) を加えた形で構成される．再アクセスユニットとビクティブバッファは共にキャッシュを分割した場合に利用する．再アクセスユニットは上位部分でキャッシュミスした場合に下位部分をスリープモードから通常モードに切換え，アクセスするために用いる．ビクティブバッファは排他的キャッシュとして上位部分内のデータを下位部分に書き戻す時に用いる．

すべてのラインがアクティブの時を Mode1，キャッシュの半分をスリープモードにしている時を Mode2，キャッシュの 3/4 をスリープモードにしている時を Mode3 とする．本手法では DRI キャッシュと同様にキャッシュサイズを 1/2，1/4 以外にも 1/8，1/16 と更に減少させることも可能であるが，現在は Mode3 迄のみ実装している．

この手法では，これまでは実行した際のミス率を基準にモード切換えを行っており，キャッシュの詳細な挙動を考慮していなかった．そこで本研究では可視化ツールを用いてプログラム実行中の挙動を解析し，従来の評価手法では発見出来なかった改善の余地を発見・提案を行う．

4　可視化ツール

4.1　関連研究

本研究と同様にキャッシュの挙動を解析する為のツールが研究･開発されている．例えばYijun Yuらが提案している可視化ツール [6] では，キャッシュミスがプログラムのどこで起こったか，どのような種類のミスが起きたか等の情報を得られる．これによりプログラマがキャッシュを意識してプログラミングしやすくなる．しかし，このツールにはキャッシュ情報の描画機能が無く，ミスがキャッシュ内のどの位置で起こったのか等，キャッシュに関する研究をする上で必要としている情報を得ることが出来ない．また，Boris Quaingらが提案しているYACO[7] はボトルネックとなっているプログラムのホットスポットの抽出，プログラム改善の提案も行う．しかし，このツールにおけるキャッシュ情報の描画はキャッシュの一部，もしくはプログラムの一部のみに限られており，キャッシュ全体を

一望する事が出来ない．そこで本研究ではよりキャッシュの動作解析に適した可視化ツールとして，以下の特徴を有した可視化ツールを開発する．

- 高速な動作
- キャッシュ全体の状態が一見して把握できるインターフェース
- シークバー機能

それぞれの必要性，及び実現方法について次項以降で説明する．

4.2 高速化手法

画面描画を行う本ツールは，高速動作するキャッシュシミュレータである必要がある．そのため，本研究で作成する可視化ツールはL2キャッシュの動作のみを模倣するトレースドリブン型シミュレータとして作成する．トレースドリブン型シミュレータはプログラムを読込み走らせるのではなく，あらかじめ他のプロセッサシミュレータを動作させ，L2キャッシュへのアクセス命令をダンプファイルとして作成し，それを読込み動作する．メモリアクセスのみをダンプファイルから読み出してシミュレートする事で，プロセッサの動作を模擬する必要が無く，高速にキャッシュの評価を行うことができる．本研究で作成する可視化ツールに入力するダンプ

ファイルの形式は，図 4.2 のようになっている．コンマで区切られた各数字は順にサイクル数, アクセス先アドレス, リード/ライトを示している．

```
8764608,7994688,0
8764620,7994752,0
8764632,7994816,0
8764644,7994880,0
```

図 4.2: ダンプファイル

4.3 インターフェース

本研究で作成する可視化ツールにはキャッシュ内部のどこでどのような動作が起きているかを見るために，キャッシュ全体の状態が一見して把握できるインターフェースが必要となる．実際の可視化ツールのインターフェースを図 4.3 に示す．各短形１つがキャッシュの１ラインの様子を色で示している．数値等を表示するよりも色で区別する方が直観的にわかりやすいためである．ｓキーを押すことシークバーが出現し，クリックで指定する事で任意のサイクルへ跳ぶ事が出来る．

4.4 シークバー機能

シミュレート中に，特徴的な挙動があった場合や特に目立った挙動がない場合等，任意のサイクルからシミュレートを再開したいという場面

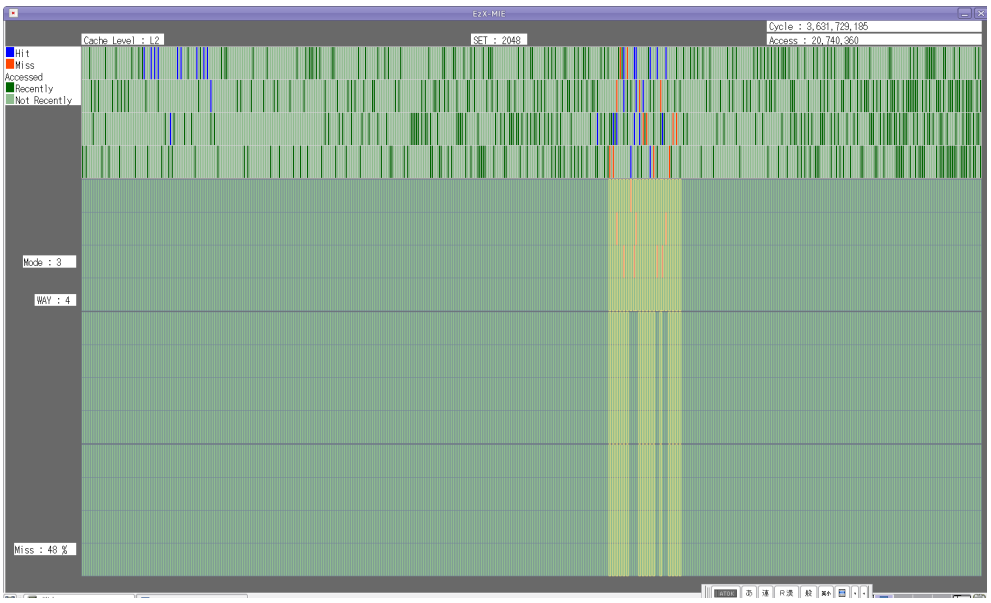

図 4.3: インターフェース

は起こりやすい．そこで本節では，現在の処理個所に関係なく任意のサイクルへ処理を移すシークバー機能を実装する上で起こる問題と解決法について述べる．

4.4.1 実装における問題点

トレースドリブン型シミュレータではダンプファイルからの値読込みと値の反映を順次行っていく．値の反映をする際に変化した個所は以前の状態を保持しないため，シミュレートは不可逆な物となる．本研究で作成する可視化ツールにおいても，キャッシュアクセス毎にキャッシュ状態が上書きされてしまうので，任意のサイクルからシミュレートを行う

為には直前のキャッシュ状態が必要となる．しかし，シミュレート中に全サイクルのキャッシュ状態を保持するのは情報量が多過ぎるため現実的ではない．また，ダンプファイルの先頭から描画処理のみを省略し読み飛ばすにしても跳び先サイクルが遠い程時間がかかってしまう．このような問題があるため一般に不可逆なシミュレータは任意のサイクルからのシミュレートは不得手である．

4.4.2 チェックポイントを用いたシークバー機能の実装

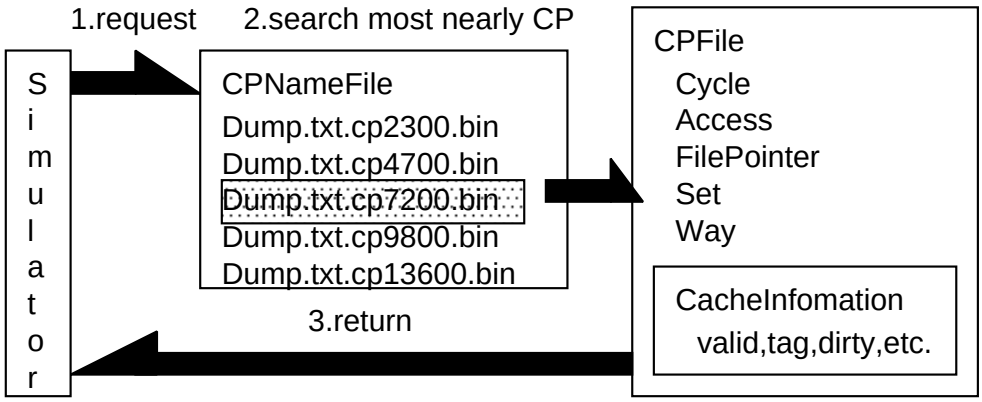

図 4.4: シークバーとチェックポイント

本シミュレータでは一定サイクル毎にチェックポイントを作成する事で，既にシミュレートしたサイクルに戻れるようにする．チェックポイントの概要を図 4.4 に示す．各チェックポイントではその時点のキャッシュ状態をチェックポイントファイル (CPFile) として出力し，チェックポイントファイル名をテキストファイル (CPNameFile) に保持しておく．シーク

バーにて現在のサイクル以後の地点が指定された場合，その地点までシミュレート処理及び未作成ならばチェックポイントの作成を繰り返し指定されたサイクルのキャッシュ状態のみを描画する．順次描画を進める場合と比べて描画回数が少なくなるため高速となる．また，現在のサイクルよりも以前の地点が指定された場合は，図 4.4 のように跳び先サイクルから直前のチェックポイントのキャッシュ状態を復元し再シミュレートを再開できるようになる．ダンプファイルの先頭からシミュレートする必要が無いため，短時間でキャッシュ状態の描画処理に復帰できる．チェックポイントの個数はダンプファイルが膨大になり過ぎない程度の数に設定されており，作成間隔は読み込むダンプファイルの行数に応じて自動で決定される．

5　可視化ツールを用いて発見した改良点

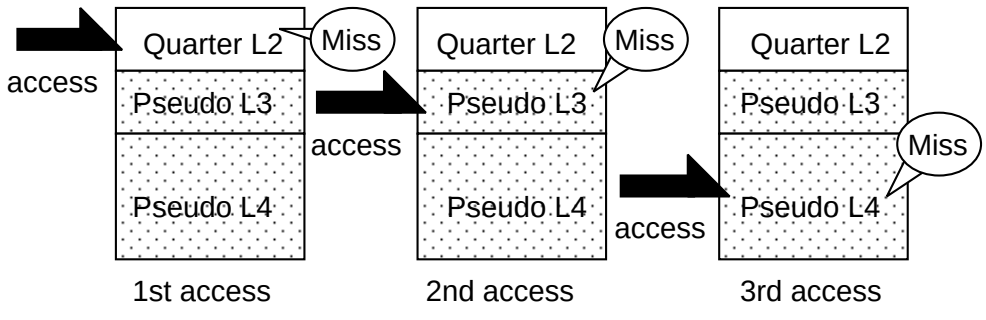

図 5.5: Mode3 にてミスが起こった場合のアクセス

VLC がスリープ領域を持つ際には，1 度のデータ要求に対し複数回のアクセスが発生する可能性がある．実際に Mode3 にて複数回のアクセスが発生する場合を図 5.5 に示す．データ要求が来たときにはまず起きている部分（Quarter L2）へアクセスする．キャッシュミスが起きた場合，更に下位のキャッシュにアクセスを行う．その際はスリープ領域（Psendo L3）を起こすオーバーヘッドと 2 度目のアクセスのオーバーヘッドが生じる事となる．再度キャッシュミスが起きた場合，更に下位半分（Psendo L4）へとアクセスを行う．これらの通常キャッシュでは起こりえない 2 度目以降のアクセス (以下リアクセス) は VLC の性能低下を防ぐための機構である．しかし L2 キャッシュ全体に目的のデータが無い場合のリアクセスは完全な無駄である．無駄なリアクセスはアクセス数が増える事によるダイナミックエネルギーの増加を引き起こす．またスリープ領域を起こす事によるダイナミック及びリークエネルギーの増加も引き起こしてしまう．可視化ツールを用いて VLC の動作を追った結果，一定期間内のリアクセスはヒット又はミスに偏りやすい事が判明した．これは要求されるデータの時間的局所性，空間的局所性によると思われる．この結果を用いて，消費電力及び実行時間のボトルネックとなっている無駄なリアクセスを削減する方法を考える．

6　可変レベルキャッシュの改良

6.1　シャットダウンとスリープモードの併用

スリープ領域でも特にデータが必要とされていない部分をシャットダウンする事で電力削減と実行時間短縮を目指す．一定サイクル毎にスリープ領域の各部をシャットダウンするかどうかの判断を行う．閾値としてスリープ領域でのヒット数を用い，シャットダウンをするかどうかを判定する区域はそれぞれキャッシュ総容量の 1/4 とする．イメージ図を図 6.6 に示す．シャットダウン領域にはリアクセスする必要が無いため不要なリア

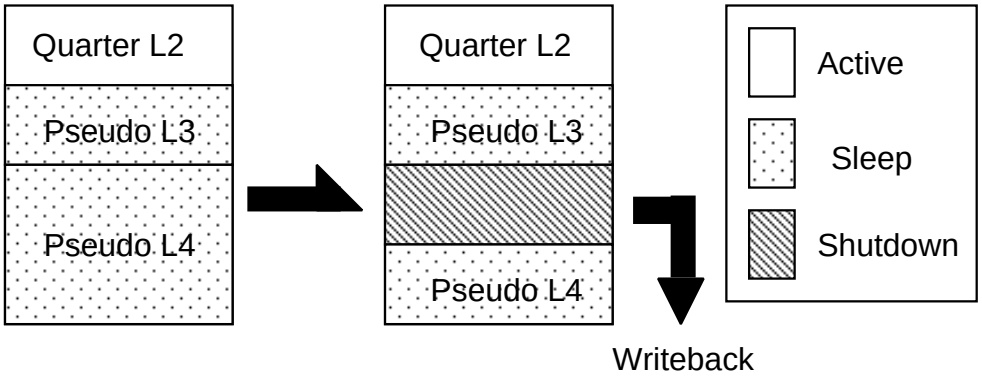

図 6.6: シャットダウンのイメージ図

クセスを削減し，ダイナミックエネルギーを削減出来る．また，リークエネルギーの削減も行える．閾値にはアクセス数よりもヒット数を用いる方が，各領域の必要性を正確に判断できる．そのため不容易なシャット

ダウンがなくなり，ミス率の増加やそれに伴うモード切替を防ぐ事ができる．

6.2 書き戻しペナルティーの軽減

シャットダウンが起きる度にその領域全体を走査し書き換えられたラインを書き戻すペナルティーは大きい．このペナルティーを軽減するためコンパクティッドダーティーフラグを用いた．コンパクティッドダーティーフラグの概要を図 6.7 に示す．8 セット毎に 1bit のコンパクティッドダーティーフラグを用意し，書きこみがあった場合対応するフラグをたてる．シャットダウンの際はまずコンパクティッドダーティーフラグを走査する．フラグがたっている場合はその部分のみさらに走査し書き戻せば良い．図 6.7 の例では下部の 8 セットはダーティーフラグの走査のみとなるのでペナルティーが軽減される．

7 評価

7.1 評価方法

今回の評価は 512KB の L2 キャッシュに関して通常キャッシュ，従来の可変レベルキャッシュ，改良を行った可変レベルキャッシュの消費電力及び実行時間をシミュレートにより求める．

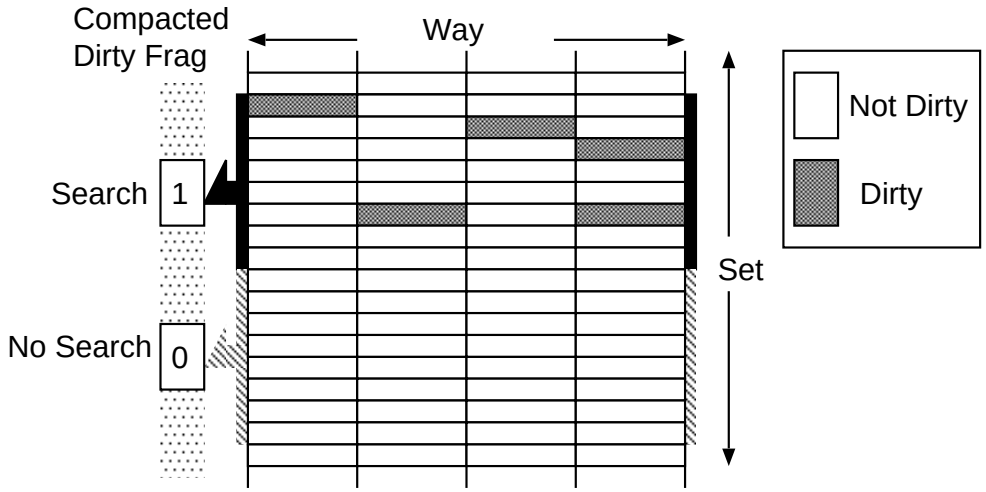

図 6.7: ダーティーフラグ

ベンチマークプログラムはSPEC2000[8] より，SPECint2000から164.gzip, 175.vpr，176.gcc，181.mcf，256.bzip2，254.gap の 6 種類，SPECfp2000 から 179.art，183.equake，188.ammp の 3 種類，計 9 種類を使用した．

その際，プログラムの実行安定時の評価を行うために，プログラム実行開始より 20 億命令実行後の 20 億命令を評価対象とした．

7.2 評価結果

評価結果を図 7.8，7.9 に示す．各図の「Nomal VLC」は従来の VLC,「Shutdown_Quarter」は6章で提案した手法を組み込んだ物,「Shutdown_Half」は 6 章で提案した手法からシャットダウンの判断区域を変化させた物である．具体的には下位半分（Mode2 の Psendo L3 及び Mode3 の Psendo

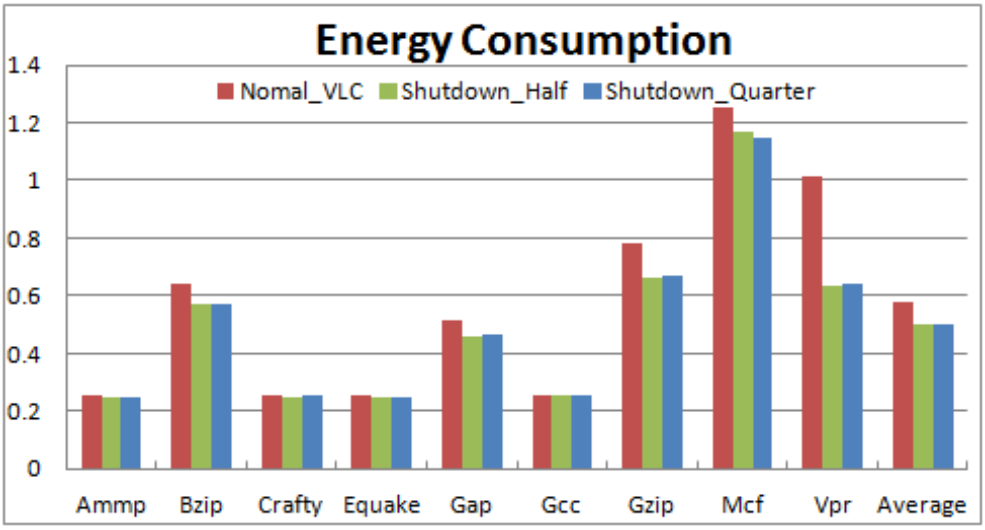

図 7.8: 消費電力

L4）を 1 つの区域として扱う．イメージ図を図 7.10 に示す．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．Shutdown_Quarter は従来手法と比較して消費電力を平均約 14%，最大で約 37%削減することが分かった．また実行時間も短縮されており，VLC でない通常キャッシュと比べても増加分は 1%未満となった．

電力削減の理由としては，キャッシュの一部をシャットダウンする事でリアクセス数が大幅に減り，ダイナミックエネルギーが削減された点が挙げられる．これにより Vpr 等のアクセス頻度が高く，従来手法でのリークエネルギー削減だけでは効果が薄かったベンチマークにおいても電力削減が行えた．また，従来手法でも大きく削減できていた Ammp 等のベ

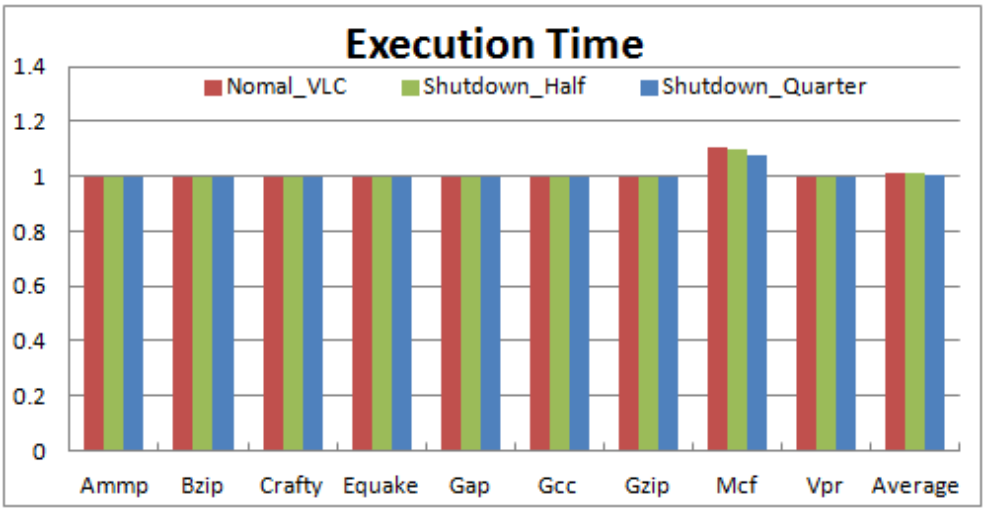

図 7.9: 実行時間

ンチマークにおいても僅かに削減されている．大部分のスリープ領域をシャットダウンした事で，リアクセス数の削減及びリークエネルギー削減が行えるためである．

実行時間短縮の理由としては，電力削減と同様に再アクセス数が減った点が挙げられる．特に再アクセス頻度が高かった Vpr や Mcf においては再アクセス数はおよそ半減している．6.2 節の書き戻しペナルティー軽減手法を用いたことで，ライトバックレイテンシを 1 サイクルから主記憶アクセスレイテンシまで増加させても実行時間の変動は平均して 0.1%未満であった．

Shutdown_Quarter よりも Shutdown_Half の方が消費電力及び実行時間に

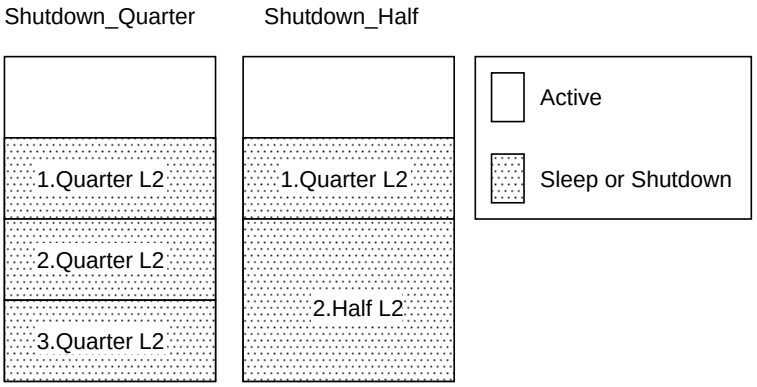

図 7.10: シャットダウンの判断区域

優れている．これはシャットダウンの判断区域が細分化される事でより正確に各区域の必要性を判断でき，不容易なシャットダウンや再アクセスが減るためである．

8 おわりに

本研究では，可視化ツールを用いてキャッシュの消費電力削減手法の 1 つである VLC の評価・改良を行った．VLC において，スリープモードで保持しているデータが不要な場合には該当部分をシャットダウンすることで更なる電力削減，及び高速化を実現した．提案手法は従来手法よりも消費電力が平均約 14%，最大で約 37%削減される．また実行時間も短縮されており，VLC でない通常キャッシュと比べても増加分は 1%未満である．

謝辞

参考文献

[1] K. Watanabe, et al. , 'Reducing Dynamic Energy of Variable Level Cache', International Journal of Computer and Electrical Engineering vol.5, no.6, pp.581-586, December, 2013.

[2] 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', International Symposium on High-Performance Computer Architecture, pp.147-157, January, 2001.

[3] Huifang Qin, Yu Cao, Dejan Markovic, Andrei Vladimirescu, and Jan Rabaey, 'SRAM Leakage Suppression by Minimizing Standby Supply Voltage', Department of EECS, University of California at Berkeley, Berkeley, CA 94720.

[4] Ying Zheng, Brian T. Davis, Matthew Jordan, 'Performance Evaluation of Exclusive Cache Hierarchies', IEEE International Sympo-

sium of Performance Analysis of Systems and Software, ISPASS, pp.89-96, September, 2004.

[5] Advanced Micro Deices, AMD, http://www.amd.com/us-en/. (Current June 2003).

[6] Yijun Yu, Kristof Beyls, Erik H.D' Hollander, 'Visualizing the Impact of the Cache on Program Execution', CCAI, vol.19, pp.3-4, July, 2001.

[7] Boris Quaing, Jie Tao, Wolfgang Karl, 'YACO: A User Conducted Visualization Tool for Supporting Cache Optimization', HPCC 2005, pp.694-703, September, 2005.

[8] "SPEC -Standard Performance Evaluation Corporation-," URL: http://www.spec.org/.

A　プログラムリスト

B　評価用データ