

卒業論文

題目

可変レベルキャッシュの可視化ツールの修正と
モード切り換え判定アルゴリズムの改良

指導教員

佐々木 敬泰 助教授

2015 年

三重大学 工学部 情報工学科
コンピュータアーキテクチャ研究室

刀根 舞歌(411831)

内容梗概

近年、低電力と高性能を両立したプロセッサが求められている。しかし回路の微細化に伴うトランジスタ数の増加や、高性能化のために絶縁膜を薄くした結果、リーク電力の増加が問題となっている。キャッシュはプロセッサの面積の大半を占めているため、キャッシュのリーク電力を削減することはプロセッサの低電力化に繋がる。そこで当研究グループでは低消費電力キャッシュの手法として、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、通常の活性状態と、データを保持できる最低電力状態であるスリープ状態、さらに完全に電源を落とすシャットダウン状態の3状態を使い分ける。キャッシュの容量をあまり必要としない場合に、キャッシュの一部をスリープモードやシャットダウンすることで、省電力化する。可変レベルキャッシュにはその挙動を観察できる可視化ツールが開発されたが、シミュレーションに支障をきたす問題点がいくつか発見されたため、修正と再評価が必要である。

よって本研究では、可視化ツールを修正・再評価し、その結果をもとに可変レベルキャッシュの更なる改良・評価を行った。修正した可視化ツールを用いて解析したところ、電力削減が少ないベンチマークでは、プログラム全体のミス率が高いことが判明した。そこで全体のミス率によるモード切り換えアルゴリズムを提案する。また、実行状況やベンチマークによって、適切なミス閾値は異なる。そこで、動的にミス閾値を変更できる、動的ミス閾値制御を提案する。2つの提案手法を可視化ツールに実装し評価したところ、従来の可変レベルキャッシュと比較して、平均で消費電力は28.87%，実行時間は0.53%減少した。

Abstract

Processors are required to achieve both low-energy and high-performance at the same time. However, increasing leaked energy has been becoming a major concern because of increasing of transistors with miniaturization of circuits and thinner insulating films to achieve high-performance. Reducing leaked energy consumed in a cache memory results low-energy processor since cache accounts for most of an energy in a processor. We propose Variable Level Cache (VLC) as a method to achieve low-energy cache.

VLC uses three states, active, sleep, and shutdown. Active is a same state as normal cache. Sleep can hold data, and consume very low energy than active. Shutdown cannot hold data, but consume no energy. When VLC does not need capacity, the energy are reduced by changing parts of cache to sleep or shutdown. We also propose a visualizer for VLC to watch the real behavior of VLC. However, some problem are discovered in this visualizer because of incorrect source codes. So that, visualizer needed to fix and re-evaluation.

In the present study, we aimed to fix and re-evaluate the visualizer and improve and evaluate VLC. The miss rate of a whole program is high in benchmarks which reduce little energy by analysis using a fixed visualizer. Thus, we propose the algorithm for mode changing of VLC using whole miss rate. Also best miss bounds are different between benchmarks or state of cache. Therefore, we propose Dynamic Miss Bound Control (DMBC), which can change miss bounds dynamically.

We evaluated VLC introduced two proposed method. Comparing previous VLC, proposed VLC can reduce the total energy by about 28.90 %, and the executive time by 0.41 % in average.

目次

1	はじめに	1
2	低電力キャッシュ手法に関する研究	4
2.1	DRI キャッシュ	4
2.1.1	DRI キャッシュの概要	4
2.1.2	DRI キャッシュの問題点	5
2.2	可変レベルキャッシュ	6
2.2.1	可変レベルキャッシュの概要	6
2.2.2	可変レベルキャッシュの問題点	11
2.3	可視化ツール	11
2.3.1	可視化ツールの概要	11
2.3.2	可視化ツールの問題点	12
3	可視化ツールの修正・再評価	14
3.1	修正内容	14
3.2	性能の再評価	15
3.3	修正した可視化ツールを用いた解析	17
4	全体のミス率によるモード切り換えアルゴリズムの提案	18
4.1	全体のミス率によるモード切り換えアルゴリズムの概要	18
4.2	全体のミス率によるモード切り換えアルゴリズムの利点	20
5	動的ミス閾値制御の提案	22
5.1	動的なミス閾値の概要	22
5.2	動的ミス閾値制御の利点	23
6	提案手法の性能評価	24
6.1	評価結果	24
6.2	考察	25
7	まとめ	27
	謝辞	29
	参考文献	29

A プログラムリスト	31
B 評価用データ	31

図目次

2.1	DRI キャッシュの概念図	4
2.2	可変レベルキャッシュの概念図	6
2.3	ヒット時の動作	8
2.4	ミス時の動作	8
2.5	再アクセス・交換時の動作	10
3.6	消費電力比 (再評価)	17
3.7	実行時間比 (再評価)	17
4.8	全体のミス率によるモード切り換えアルゴリズムのフロー チャート	18
4.9	従来の可変レベルキャッシュと提案手法の動作 (1/2)	20
4.10	従来の可変レベルキャッシュと提案手法の動作 (2/2)	21
5.11	動的ミス閾値制御のフローチャート	23
6.12	消費電力比	25
6.13	実行時間比	25

表 目 次

3.1	評価に用いた可変レベルキャッシュの仕様	16
6.2	提案手法を導入した可変レベルキャッシュの仕様 (追加要素のみ)	24

1 はじめに

近年、スマートフォンやノートパソコンなどの携帯端末の高性能化に伴い、プロセッサの消費電力が増加している。その結果、バッテリーの駆動時間が短くなってしまうと懸念されている。そのため、プロセッサの性能を維持しつつ、消費電力を減らすことが要求されている。

プロセッサの消費電力は、主に動的消費電力と静的消費電力の2つに分けることができる。動的消費電力は回路の動作によって消費される電力である。一方、静的消費電力はトランジスタから漏れ出てしまう電流、すなわちリーク電流により、回路の動作に関係なく消費されるのでリーク電力ともよばれる。今日、回路の微細化に伴うトランジスタ数の増加や、高性能化のために絶縁膜を薄くした結果、リーク電力が増加傾向にある。リーク電力はトランジスタ数に比例するため、面積が大きいほどリーク電力も増加する。キャッシュはプロセッサの面積の大半を占めているため、プロセッサの消費電力に占める割合も最も大きい。よって、キャッシュのリーク電力を削減することはプロセッサの低消費電力化に繋がる。そのため、さまざまな低消費電力キャッシュ手法が提案されている。当研究グループでも、低消費電力キャッシュ手法の1つとして、可変レベルキャッシュ[1]を提案している。

可変レベルキャッシュはキャッシュミス率に応じて、キャッシュの一部をデータを保持できる最低電力状態であるスリープ状態 [2] にし、キャッシュの容量を変更する手法である。また当研究グループでは、実際のキャッシュの挙動を確認できる、可視化ツール [3] も提案している。可視化ツールによって不要なアクセスが発覚し、シャットダウンが導入された。しかしこの可視化ツールには問題点があるため、正しい結果を得られていない。そこで本研究では、可視化ツールを修正し解析を行い、可変レベルキャッシュの更なる改良を目標とした。

修正した可視化ツールを用いて解析したところ、削減効率の悪いベンチマークではプログラム全体のキャッシュミス率が高いことが判明した。これは従来の可変レベルキャッシュが、一定サイクルごとのミス率しか考慮しないためである。

そこで全体のミス率も利用するモード切り換えアルゴリズムを提案する。またベンチマークや実行状況によって適切なミス閾値は異なる。それに対応するため、ミス閾値を動的に変更できる、動的ミス閾値制御を提案する。2つの提案手法を導入した可変レベルキャッシュを、可視化ツールに実装し性能を評価した。その結果、通常のキャッシュと比較して消費電力が平均 51.24 % 減少し、実行時間が平均 1.81 % 増加した。また従来

の可変レベルキャッシュと比較して消費電力が平均 28.87 %，実行時間が平均 0.53 %減少した.

2 低電力キャッシュ手法に関する研究

2.1 DRI キャッシュ

2.1.1 DRI キャッシュの概要

可変レベルキャッシュのもととなった低電力キャッシュ手法の1つに、DRI キャッシュがある。DRI キャッシュは命令キャッシュ向けに開発された手法である。DRI キャッシュの概念図を図 2.1 に示す。

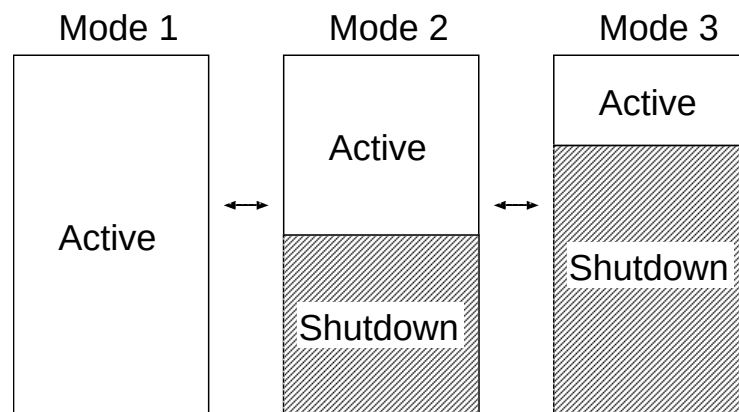


図 2.1: DRI キャッシュの概念図

最初は通常のキャッシュと同じ活性状態である。キャッシュのミス率が低い場合、キャッシュ容量をあまり必要としていないと判断し、現在のキャッシュ容量の半分をシャットダウンする。シャットダウンした領域は完全に電源を落とすため、その部分の消費電力が削減できる。反対にキャッシュミス率が高い場合、キャッシュ容量を多く必要としていると判断し、シャッ

トダウンした領域を通常のキャッシュと同じ状態に戻し、現在のキャッシュ容量を倍にする。

このようにキャッシュミス率でキャッシュ容量を切り換えることにより、不要な領域の電力削減をはかる。しかし、この手法をそのままデータキャッシュに適用すると、2つの問題が発生する。

2.1.2 DRI キャッシュの問題点

1つめはキャッシュに書き込んだデータが消失する問題である。シャットダウンすると電源を落とすため、その領域ではデータを保持できない。そのため書き込んだデータが消失する可能性がある。データの消失を防ぐために、データキャッシュではメインメモリにデータを書き戻すライトバックが必要となる。そのため、レイテンシが増加してしまう。

2つめは保持できるデータ量が減少する問題である。キャッシュ容量を減らすため、当然保持できるデータ量が減少する。そのため、すぐに使用するかもしれないデータを保持できず、キャッシュミス率が増加してしまう。

2.2 可変レベルキャッシュ

2.2.1 可変レベルキャッシュの概要

そこで当研究グループでは、可変レベルキャッシュ(Variable Level Cache: VLC)を提案している。可変レベルはDRI キャッシュと同様に、L2 キャッシュのミス率に対応して、L2 キャッシュの容量を動的に変化させる手法である。しかし、可変レベルキャッシュでは完全に電源を落とすシャットダウンの代わりに、データを保持できる程度の低電力状態であるスリープ状態を使用する。スリープ状態は、完全に電源を落とすシャットダウン状態よりは消費電力が大きいが、通常の状態よりはるかに消費電力が小さくてすむ。可変レベルキャッシュの概念図を図 2.2 に示す。

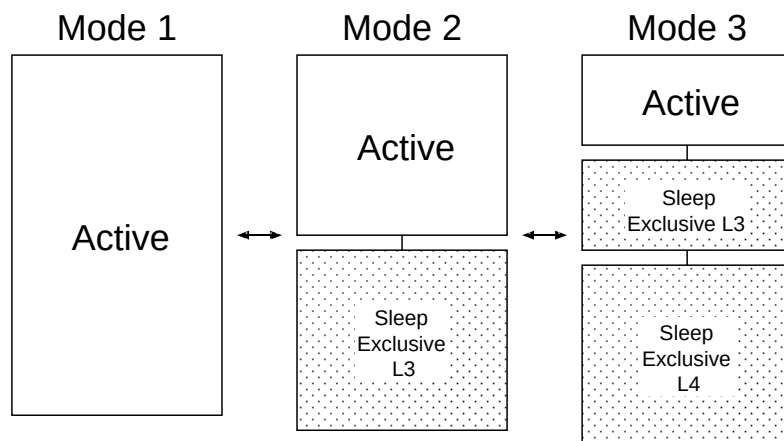


図 2.2: 可変レベルキャッシュの概念図

現在、可変レベルキャッシュには3つのモードがある。モード1は、通

常のキャッシュと同じ状態である。キャッシュミス率が低い場合、すなわちキャッシュ容量をそれほど必要としていない場合は、モード2へ移行する。モード2では、L2キャッシュの下位50%の領域をスリープ状態に移行させる。さらにキャッシュミス率が低い場合は、モード3へ移行する。モード3ではL2キャッシュの下位75%の領域をスリープ状態に移行させる。反対にキャッシュミス率が高い場合、すなわちキャッシュ容量を多く必要とする場合は1つ上位のモードに戻し、スリープ状態から復帰させて活性状態にする。

このようにモードを切り替えることにより、データを保持しつつ電力消費を抑えることができる。

次に、可変レベルキャッシュの動作について説明する。プログラム開始時、最初のモードはモード1である。モード1でアクセスが発生したとき、データが存在する場合は図2.3のようにヒットとなり、データが存在しない場合は図2.4のようにミスとなる。

ミスの場合は、メインメモリからデータを取得しなければならない。

モード1において、一定サイクルごとに計測したキャッシュミス率が、下位のモードへ移行するための閾値を下回った場合、モード2に移行する。可変レベルキャッシュでは、スリープ領域は擬似的に下位の排他的

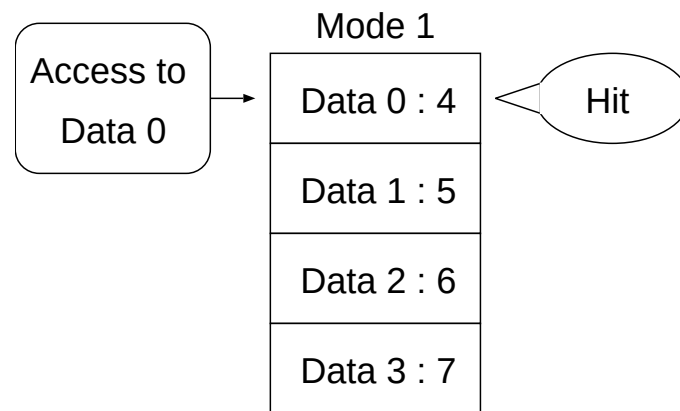


図 2.3: ヒット時の動作

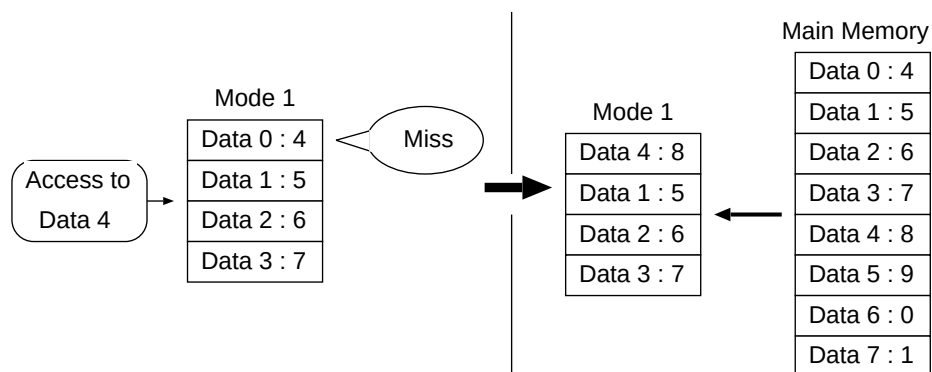


図 2.4: ミス時の動作

キャッシュ[5]として扱われる。モード2では、キャッシュの上位50%の領域をL2キャッシュ、スリープ状態になる下位50%の領域をL3キャッシュとみなす。モード2でアクセスが発生したときの動作を図2.5に示す。まずL2キャッシュにアクセスする。データが存在する場合はヒットとなる。データが存在しない場合、スリープモード状態に移行したL3キャッシュ

を、一旦通常のキャッシュ状態に戻す。それから L3 キャッシュに再アクセスする。そこにデータが存在する場合はヒットとなる。このとき、ヒットしたラインを、L2 キャッシュの対応するラインと交換する。これにより、良く使用するデータを L2 キャッシュに保持しておくことができる。

データが存在しない場合はミスとなる。モード 2 において、一定サイクルごとのミス率が、上位のモードへ移行するためのミス閾値、すなわち上方のミス閾値を上回った場合、モード 1 に移行する。反対に、一定サイクルごとのミス率が下位のモードへ移行するためのミス閾値、すなわち下方のミス閾値を下回った場合、モード 3 に移行する。モード 3 では、キャッシュの上位 25 % の領域を L2 キャッシュ、スリープ状態の領域のうち、上位 25 % の領域を L3 キャッシュ、残りの下位 50 % の領域を L4 キャッシュとみなす。モード 3 でアクセスが発生したとき、まず L2 キャッシュにアクセスする。データが存在する場合はヒットとなる。データが存在しない場合、スリープ状態モードに移行した L3 キャッシュを、一旦通常のキャッシュ状態に戻す。それから L3 キャッシュに再アクセスする。そこにデータが存在する場合はヒットとなる。このとき、ヒットしたラインを、L2 キャッシュの対応するラインと交換する。データが存在しない場合、今度は L4 キャッシュを、一旦通常のキャッシュ状態に戻し再ア

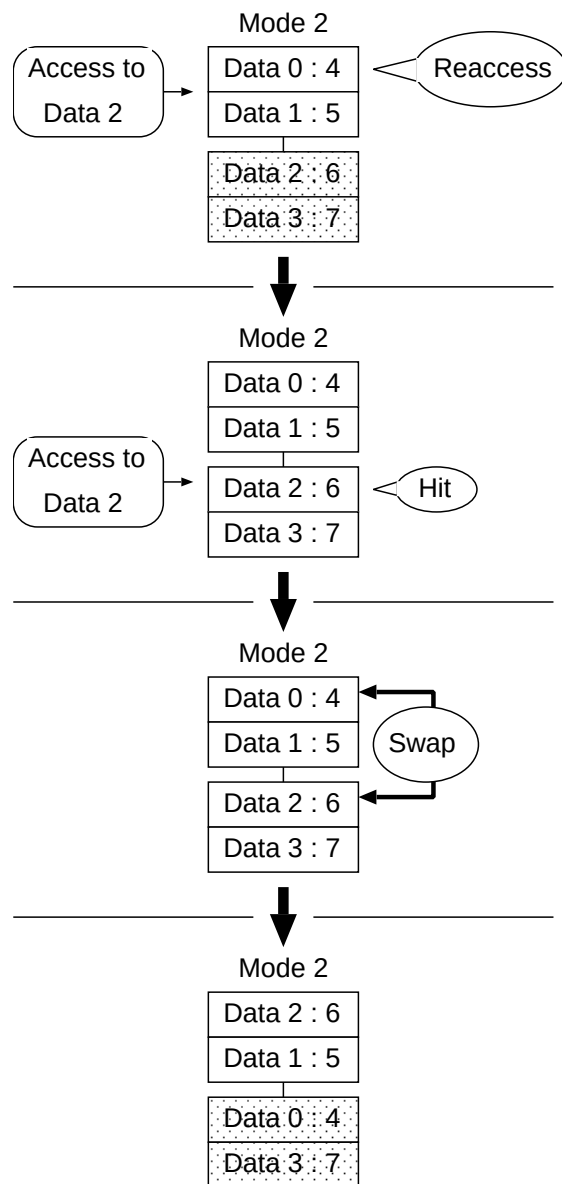


図 2.5: 再アクセス・交換時の動作

クセスする．そこにデータが存在する場合はヒットとなり，存在しない場合はミスとなる．モード3において，一定サイクルごとのミス率が，上位のモードへ移行するための閾値を上回った場合，モード2に移行する．

2.2.2 可変レベルキャッシュの問題点

ここまでに説明した可変レベルキャッシュは実際のキャッシュの挙動を考慮せず，単純に一定サイクルごとのミス率でモード切り換えを行う．そのため，必要なキャッシュ容量に合わず，余分なミスや再アクセス，交換が発生しやすい．

2.3 可視化ツール

2.3.1 可視化ツールの概要

そこで実際の可変レベルキャッシュの挙動を可視化する，可視化ツールが提案されている．可視化ツールはC言語で記述されたプログラムで，トレースドリブン型のシミュレータである．トレースドリブンとはあらかじめプログラムを走らせ，その結果の命令群を使う方法である．分岐結果がすでに出ているため，分岐予測をするハードウェアが必要ない，プログラムの好きな場所からシミュレーションできる，高速にシミュレーションできるといった利点がある．可変レベルキャッシュの可視化ツール

は、現在のサイクル数、アクセス回数、ミス率、そして現在のキャッシュの様子を見ることができる。

この可視化ツールによる解析により、アクセスが一部のスリープ領域に集中していたり、目的のデータが存在しないのに再アクセスを繰り返していることが判明した。

そこで、スリープ状態モードのほかに、シャットダウンも用いる可変レベルキャッシュが提案された。これにより、従来の可変レベルキャッシュと比較して、消費電力を平均 18 %削減できたとされていた。しかし、この可視化ツールにはシミュレーションに支障をきたす問題点がいくつかあることが分かった。

2.3.2 可視化ツールの問題点

可視化ツールには次のような問題点があった。

- 巨大ファイルの末尾まで読み込めていない。
- 全く同じ条件でも結果が異なる場合がある。
- アクセス回数、ミス回数、スリープ状態・シャットダウン時のサイクル数が間違っている。

- 大量の再帰によりセグメンテーションフォールトが発生する場合がある.
- 手動で動かす場合とチェックポイントファイルを使う場合とで結果が異なる.
- 描写サイクル間隔によって結果が変わってしまう.

3 可視化ツールの修正・再評価

そこで、可視化ツールを修正し、修正した可視化ツールを用いて解析し、可変レベルキャッシュの更なる改良を試みた。

可視化ツールの修正・変更内容は以下のとおりである。

3.1 修正内容

- Lareg File Support を正しく適用し、巨大ファイルを末尾まで読み込めるように修正した。
- ファイルから1行読み込んだ時に、アクセス数を増加させるように修正した。
- ファイルの終端に到達した時に、スリープ状態・シャットダウン時のサイクル数を数えるように修正した。
- 再帰関数を while 文に変更した。
- キャッシュへのアクセス時に、有効ビットを必ず確かめるように修正した。
- モード切り換え後のアクセス判定を修正した。

- チェックポイント機能を一時的に廃止した.
- 描画サイクル間隔を一時的に短くした.

チェックポイント機能はデータの書き込みと読み込みが正常に行われないため、一旦廃止した. また描写サイクル間隔によってキャッシュの更新のタイミングが異なるため、アクセスがあるたびにキャッシュを更新するように間隔を短くした.

3.2 性能の再評価

再評価は通常キャッシュ, シャットダウンを用いない通常の可変レベルキャッシュ, シャットダウンを用いる可変レベルキャッシュで行う. 再評価であるので, 評価項目や評価環境は文献 [3] と同じとする. 評価項目は低消費電力と高性能の両立を目標とするため, 消費電力と実行時間とする. 消費電力は文献 [3] と同じ方法で計算する. 実行時間はシミュレーションにかかった総サイクル数とする.

今回の評価に使用したキャッシュの仕様やレイテンシを表 3.1 に示す.

ベンチマークプログラムとして, SPEC2000[6] より, SPECint2000 から 164.gzip, 175.vpr, 181.mcf, 256.bzip2, 254.gap の計 5 種類を使用する.

表 3.1: 評価に用いた可変レベルキャッシュの仕様

キャッシュ容量	512 KB
セット数	2048
ウェイ数	4
ライン長	64 bit
ヒット時のアクセスサイクル	10 cycle
ミスレイテンシ	250 cycle
ライトバックレイテンシ	1 cycle
再アクセスレイテンシ	10 cycle
交換レイテンシ	30 cycle
スリープモードからの復帰レイテンシ	10 cycle
シャットダウンからの復帰レイテンシ	10 cycle
モード切り換え判定を行う間隔	8192 cycle
シャットダウン判定を行う間隔	8192 cycle
モード切り換え後シャットダウンしない間隔	16384 cycle
上位のモードへ移行するためのミス閾値	40 %
下位のモードへ移行するためのミス閾値	70 %

図 3.6 に消費電力比を，図 3.7 に実行時間比を示す．どちらも通常キャッシュの結果で正規化した．

通常キャッシュと比較して，消費電力が平均 23.26 % 減少し，実行時間が平均 2.34 % 増加した．また従来のシャットダウンを用いない可変レベルキャッシュと比較して，消費電力が平均 12.71 %，実行時間が平均 0.51 % 減少した．

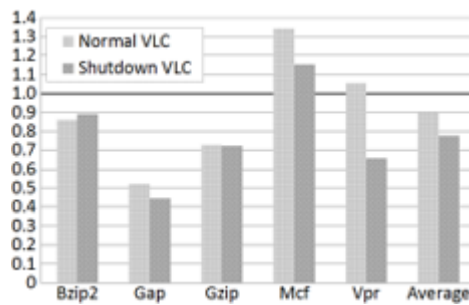


図 3.6: 消費電力比 (再評価)

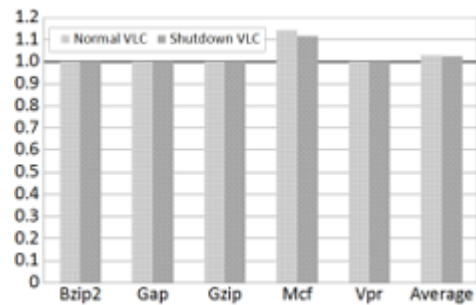


図 3.7: 実行時間比 (再評価)

3.3 修正した可視化ツールを用いた解析

修正した可視化ツールを用いて解析したところ、削減率が悪いベンチマークでは、シミュレーション開始から終了までのキャッシュミス率、すなわちプログラム全体のミス率が高いことが判明した。これは従来の可変レベルキャッシュでは、一定サイクルごとに計測したキャッシュミス率のみでモード切り換えを行うからである。全体のミス率が高く、キャッシュ容量を多く必要としていても、一定サイクルごとのミス率が低ければ、下位のモードへ移行してしまう。そのため無駄なモード切り換えやミス、再アクセス、交換が発生する。

4 全体のミス率によるモード切り換えアルゴリズムの提案

4.1 全体のミス率によるモード切り換えアルゴリズムの概要

そこで全体のミス率を利用するモード切り換えアルゴリズムを提案する。全体のミス率を利用するモード切り換えアルゴリズムのフローチャートを図 4.8 に示す。

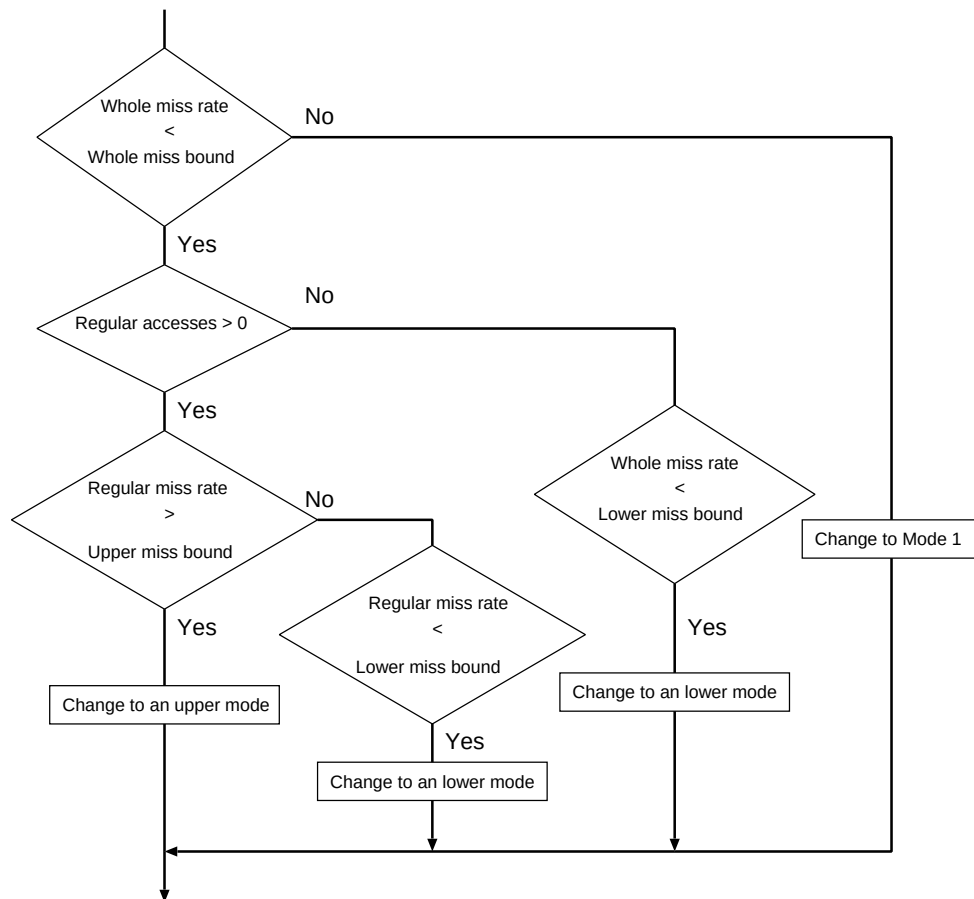


図 4.8: 全体のミス率によるモード切り換えアルゴリズムのフローチャート

全体のミス率を利用するモード切り換えアルゴリズムでは、最初に、全体のミス率と、全体のミス率用の閾値とを比較する。全体のミス率が全体のミス率用の閾値以上である場合、キャッシュ容量が多く必要であると判断し、現在のモードを強制的にモード1にする。

全体のミス率が全体のミス率用の閾値を下回っている場合、一定サイクル内にアクセスがあったかどうか確認する。一定サイクル内にアクセスが無かった場合、全体のミス率と下方のミス閾値とを比較する。全体のミス率が、下方のミス閾値以上である場合は何もしない。全体のミス率が、下方のミス閾値を下回っている場合、1つ下位のモードへ移行する。すでに最下位のモードである場合は何もしない。

一定サイクル内にアクセスがあった場合、従来の可変レベルキャッシュと同じ処理をする。つまり一定サイクルごとに計測したミス率と、下方のミス閾値とを比較する。一定サイクルごとのミス率が上方のミス閾値を上回っている場合、1つ上位のモードへ移行する。すでに最上位のモードである場合は何もしない。一定サイクルごとのミス率が、上方のミス閾値を下回っている場合、一定サイクルごとのミス率と、下方のミス閾値とを比較する。一定サイクルごとのミス率が、下方のミス閾値を下回っている場合、1つ下位のモードへ移行する。すでに最下位のモードである

場合は何もしない。

このようにして，全体のミス率を利用してモード切り換えを行う。

4.2 全体のミス率によるモード切り換えアルゴリズムの利点

図 4.9 に，全体のミス率によるモード切り換えアルゴリズムを実装した可変レベルキャッシュの具体例を示す．図 4.9 において，従来の可変レベルキャッシュでは一定サイクルごとのミス率が，下方のミス閾値を下回るため，1つ下位のモードへ切り換える．一方，全体のミス率によるモード切り換えでは，先に全体のミス率と全体のミス閾値とを比較する．図 4.9 の場合，全体のミス率が全体のミス閾値を上回っているので，モードを切り換えない。

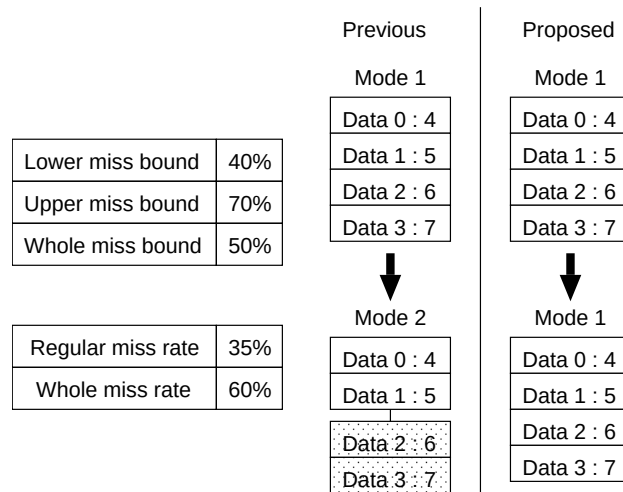


図 4.9: 従来の可変レベルキャッシュと提案手法の動作 (1/2)

その直後に、図 4.10 のようなアクセスがあった場合、従来の可変レベルキャッシュでは再アクセス・交換・ミスが発生するのに対して、全体のミス率によるモード切り換えアルゴリズムを実装した可変レベルキャッシュでは発生しない。

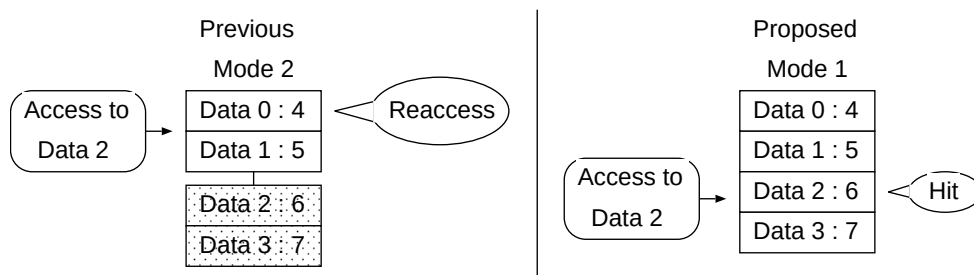


図 4.10: 従来の可変レベルキャッシュと提案手法の動作 (2/2)

このように、全体のミス率によるモード切り換えアルゴリズムを導入すると、無駄な下位モードへの移行を抑制できる。プログラム全体のミス率が高い場合、ミス率を下げることもできる。しかしプログラム全体のミス率が低い場合には、従来と同じ処理を行うことが多いため、あまり効果がない。

5 動的ミス閾値制御の提案

5.1 動的なミス閾値の概要

ここまでの可変レベルキャッシュは、あらかじめミス閾値を定めているため、ベンチマークや実行状況に最適なミス閾値を使えなかった。そのため、最適な結果を得られなかった。そこで、ヒット数に応じてミス閾値を上昇させる、動的ミス閾値制御 (Dynamic Miss Bound Control : DMBC) を提案する。

動的ミス閾値制御のフローチャートを図 5.11 に示す。

動的ミス閾値制御では、最初に一定サイクルごとに計測したアクセス回数とミス回数よりヒット数を求める。次に、一定サイクルごとのヒット数が 100 回に到達するごとに、閾値を 1 % 増加する。ミス閾値は、基本の閾値から最大 10 % まで増加できる。ヒット数分増加し終わるか、最大値に達するまで、この処理を繰り返す。図 5.11 の動的ミス閾値制御を終えたら、モード切り換え判定に移る。

動的ミス閾値制御では、一定サイクルごとのミス率が高い場合、ミス閾値はあまり増加しない。反対にミス率が低い場合、ミス閾値は急激に上昇する。現在、上方・下方・全体の 3 つのミス閾値全てに DMBC を適応している。

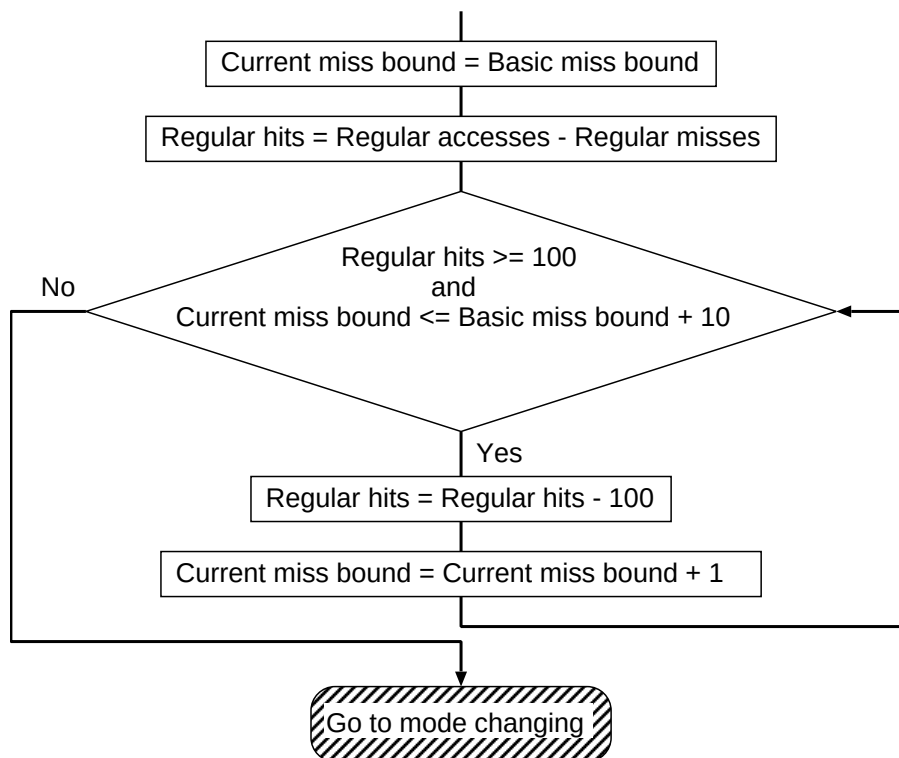


図 5.11: 動的ミス閾値制御のフローチャート

5.2 動的ミス閾値制御の利点

動的ミス閾値制御を適用すると、従来の可変レベルキャッシュではミス閾値を超えないミス率でも、増加したミス閾値を超え、モード切り換えを行う。ミス閾値を増加させると、下位のモードへの切り換えが、上位のモードへの切り換えより優位になる。その結果、省電力なモードでいる時間が長くなり、低電力化できる。また余計なモード切り替えを防ぐことができる。

6 提案手法の性能評価

6.1 評価結果

評価方法と評価環境は第3章で述べたものとほぼ同じであるため，第3章との差分のみ記載する．全体のミス率によるモード切り換えアルゴリズムと，動的ミス閾値制御に用いた数値を表6.2にまとめる．

表 6.2: 提案手法を導入した可変レベルキャッシュの仕様 (追加要素のみ)

全体のミス閾値	50 %
ミス閾値の最大増加値	10 %
ミス閾値を増加する基準のヒット数	100 回

通常のキャッシュ，文献 [3] のシャットダウンを導入したキャッシュと，全体のミス率によるモード切り換えアルゴリズムと動的ミス閾値制御を導入した可変レベルキャッシュの消費電力比を図 6.12 に，実行時間比を図 6.13 に示す．なお，消費電力，実行時間ともに通常キャッシュの結果で正規化してある．

通常のキャッシュと比較して，消費電力が平均で 51.24 % と大幅に減少し，実行時間が平均でわずか 1.81 % 増加した．またシャットダウンを導入したキャッシュと比較して，消費電力が平均で 28.87 %，実行時間が平均で 0.53 % 減少した．

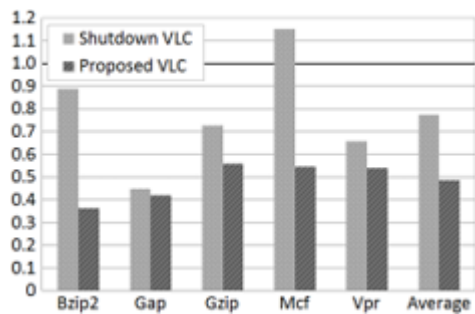


図 6.12: 消費電力比

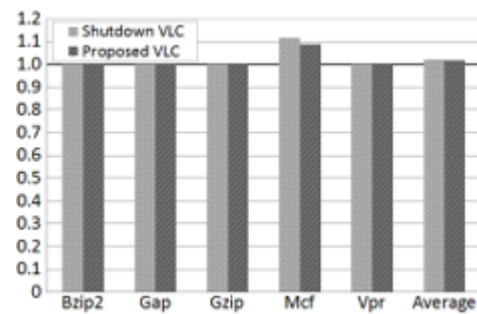


図 6.13: 実行時間比

6.2 考察

可視化ツールによる解析の結果，全てのベンチマークで，モード 3 かつシャットダウンである状態が，実行時間の大部分を占めていることが判明した．

ミス閾値を増加させると，下位のモードへの切り換えが，上位のモードへの切り換えより優位になる．かわりにキャッシュ容量が小さくなるため，全体のミス回数が増え，その分のレイテンシが増大する．しかし最下位のモードでシャットダウンしている場合，キャッシュに必要なデータがないことがわかっているので，余分なアクセスが発生せず，再アクセス回数と交換回数が減少する．

Mcf を除くベンチマークでは，アクセス間隔がミスレイテンシより長いため，ミス回数が増えても実行時間はあまり増加しない．しかし Mcf はアクセス間隔がミスレイテンシより短いため，ミス回数が増えるほど，

実行時間が大幅に増加する．その結果，Mcfを除くベンチマークでは大幅な電力削減をしつつ実行時間を保てた一方で，Mcfでは大幅な電力削減ができたが，実行時間が大きく増加してしまった．

7 まとめ

低消費電力キャッシュの手法として、新たに全体のミス率によるモード切り換えアルゴリズムと、動的ミス閾値制御の2つの手法を提案した。提案手法を導入した可変レベルキャッシュは消費電力を通常キャッシュと比較して平均 51.47 %，従来の可変レベルキャッシュと比較して平均 28.87 %削減できるとわかった。一方で実行時間は通常キャッシュと比べて平均 1.81 %増加し，Mcf ベンチマークにおいては 9.02 %も増加する。これは現在の DMBC ではミス閾値を増やすことしかできないため下位のモードに移行しやすくなり，かえってミスが増加するためである。ミスが増加するとレイテンシが増大し，実行時間が急激に増える。

また可視化ツールにはシミュレーションの実行時間が長すぎる，2回目以降のシミュレーションを素早く行うためのチェックポイント機能が正常に動作しない，描写間隔を変えると結果が変わってしまうといった問題が残っている。

今後の展望としてミス閾値を増加するだけでなく減少もさせる方法の検討，可視化ツールの完全な修正，ハードウェア記述言語での DMBC の実装があげられる。また，従来の可変レベルキャッシュはシングルスレッド環境での実装・評価に留まっているので，マルチスレッド環境におけ

る可変レベルキャッシュの実装も望まれる。さらに，マルチコア環境における可変レベルキャッシュ[7]は現在モード2までしか実装されておらず，モード3に対応していない。高性能化にマルチコア環境は不可欠であるため，こちらも実装する必要がある。

謝辞

研究のアドバイスを下さった先輩方，佐々木助教授と近藤教授にこの場を借りてお礼申し上げます。

参考文献

- [1] 渡部功，佐々木敬泰，松原伸幸，大野和彦，近藤利夫，“モード切替
オーバー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
処理学会論文誌，vol.50，No.2，pp.1234-1247，February，2006.
- [2] 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.
- [3] 近藤 舞佳，“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”，平成 25 年度 IEICE 研究発表会予稿集，March，2013.
- [4] 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 Sympo-

sium on High-Performance Computer Architecture pp.147-157 January 2001

[5] 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.

[6] "SPEC -Standard Performance Evaluation Corporation-", <http://www.spec.org/>, 参照 February, 2015.

[7] 城田幸利, 佐々木敬泰, 大野和彦, 近藤利夫, '可変レベルキャッシュ用モード切手法のマルチコア環境への適用と評価', 情処学報, 2010-ARC-190 No.12, pp.1-8, August, 2010.

A プログラムリスト

B 評価用データ