Vol.2010-ARC-190 No.12
2010/8/3

可変レベルキャッシュ用モード切換手法のマルチコア環境への適用と評価

城田幸利[†1]　佐々木敬泰[†1]
大野和彦[†1]　近藤利夫[†1]

現在，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リークエネルギーが増加しているため，リークエネルギーを削減することが重要である．そこで我々は，キャッシュの消費エネルギーを削減する手法の一つ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とはキャッシュの容量を動的に変化させる手法で，性能が必要なときには通常の容量，必要でないときにはキャッシュ容量の半分をスリープモードに移行し，1つ下位レベルの Exclusive cache として動作させることで，低消費エネルギー化を目指す手法である．これまでに我々はシングルプロセッサ環境においてその有効性を示している．

しかし，現在汎用プロセッサにおいて主流となっているのはマルチコアプロセッサである．そこで，本稿では可変レベルキャッシュをマルチコア環境へ実装し評価を行う．しかし，従来手法そのままマルチコア環境に適用した場合，性能が劣化する場合がある．そこで，キャッシュ容量切換え制御の新たな手法を提案し，その評価を行った．その結果，提案手法はマルチプロセッサ環境において従来手法を単純に適用した場合よりも，電力遅延積において 7%程度改善された．

Introducing Variable Level Cache into Multiprocessor environments and its Evalution

Yukitoshi Shirota,[†1] Takahiro Sasaki,[†1]
Kazuhiko Ohno[†1] and Toshio Kondo[†1]

Power dissipation is a 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. It is paticularly important to reduce leakage energy consumed in a cache memory because power dissipation by leakage energy current is dominant factor in deep submicron technologies and a cache memory consists of a large number of transistors. So, we propose Variable Level Cache to achieve both low energy consumption and high performance simultaneously. Variable Level Cache analyzes cache performance dynamically and if it detects that the currents running program does not need so large capacity of cache memory, half of the cache memory is put into standby mode, and is treated as a lower level exclusive cache. Variable Level Cache successed in reducing leakage energy. However, general-purpose processor shifts into the mainstream multiprocessor. So, this paper, introduces Variable Level Cache technique into multiprocessor environments. This paper also improves mode swiching approach for Variable Level Cache to be suitable in multiprocessor environments. According to the simulation results, the performance of Variable Level Cache is about 7% superior to that of conventional Variable Level Cache in the energy-delay product.

1. はじめに

現在，ノートパソコン，PDA，携帯電話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．

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って容量が増大した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．

高性能と低消費エネルギーの両立を目指すキャッシュシステムは様々なものが提案されているが，その手法の一つとして我々は可変レベルキャッシュ[1)2)] を提案している．可変レベルキャッシュとは，アプリケーションのキャッシュへの負荷が高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，負荷が低い場合はキャッシュ容量の半分をスリープモードに移行し，擬似的に1つ下位レベルの Exclusive cache として動作させる低

†1 三重大学大学院工学研究科情報工学専攻
Graduate School of Engineering, Mie University

消費エネルギーモードとして動作させる．可変レベルキャッシュはこの 2 つの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 2 つのモードを切換えることによって高性能と低消費エネルギーを実現しているため，モード切換手法によって可変レベルキャッシュの性能が大きく影響される．現在までに我々は，シングルプロセッサ環境での有効な切換手法を提案してきた[2]．しかし，近年汎用プロセッサにおいてマルチプロセッサが注目を集めている．そこで，可変レベルキャッシュをマルチプロセッサ環境へ適用し評価を行った．その結果，マルチプロセッサ環境に文献[1] の切換手法を単純に適用した場合，性能の悪化が見られた．よって，本稿ではマルチプロセッサ環境に適したモード切換手法を提案し，その有効性を示す．

2. 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換える単位で，大きく 2 つに分離することができる．

1. ライン単位の状態切換

   １つ目は，ライン単位で通常状態と待機状態を切り替えるものである．[3][4][5][6]

   代表的な研究として，Drowsy Cache[4][5] やウェイ予測キャッシュ[6] 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．ウェイ予測キャッシュは，アクセス開始前に最も最近アクセスされたウェイ情報 (MRU) を利用して，参照データが存在するウェイを予測し，選択的に活性化する事で低消費エネルギー化を行う手法である．

   これらのようなライン単位で切り換える手法は，キャッシュの状態を細かく切り換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電力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ラインと待機状態の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．

2. バンク単位の状態切換

   もう１つは，DRI キャッシュ[7] 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，バンク単位で通常状態と待機状態を切り換えるものである．このタイプは，ライン単位で切り換える手法のように状態の細かな切り換えが出来ない反面，待機状態時にセンスアンプ等のライン以外の電力削減が可能となる．

   バンク単位で切り換えを行う手法は，バンクを構成する際にセット単位，ウェイ単位のどちらでも可能であるが，本稿ではセット単位でバンクを構成する事を考える．ウェイ単位で切り換える事はライン単位で切り換えるのと同様に実効的に連想度を低下させてしまうが，セット単位の切り換えの場合，各セット内の連想度が通常状態と同じに保たれるという利点があるからである．次項で説明する DRI キャッシュにおいても同様に扱う．

2.1 DRI キャッシュ

2.1.1 DRI キャッシュの概要

図 1 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トする (miss counter)．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．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．

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，センスア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メリットがある

2.1.2 DRI キャッシュの問題点

DRI キャッシュでは，キャッシュ・サイズを縮小する際，縮小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処理を行っている．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際には，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している．DRI キャッシュはこれらの処理が性能へ悪影響を与えているという点で問題がある．そこで，我々は DRI キャッシュを改良し，待機状態への切換え時の書き戻しを抑制する可変レベルキャッシュを提案する．

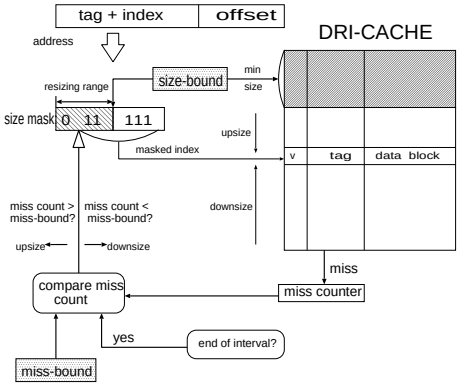

図 1 DRI のブロック図

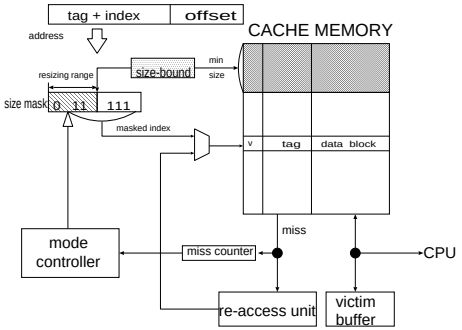

図 2 可変レベルキャッシュのブロック図

3. 可変レベルキャッシュ

3.1 概　　要

可変レベルキャッシュ[1)2)] 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の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 Exclusive cache 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．しかし，可変レベルキャッシュは DRI キャッシュと同様に，通常モードへの切換条件を満たした場合，通常モードと低消費エネルギーモードではキャッシュメモリ内のデータ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書き戻し，マスクのビットを全て 1 にし，インデックスの制限を解除する．Exclusive cache については次項で説明する．

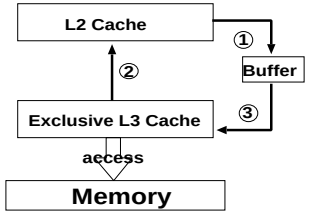

図 3 Exclusive Cache

可変レベルキャッシュの例をあげる．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，L3 キャッシュとして動作する．この時，L3 キャッシュは Exclusive cache として動作させる．

図 2 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 Exclusive cache 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を「通常モード」，キャッシュの半分をスリープモードにし，Exclusive cache として動作するときを「低消費エネルギーモード」とする．

3.2 Exclusive Cache

Exclusive cache 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，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，Exclusive cache では，図 3 のように，L1 キャッシュから L2 キャッシュへと書き戻

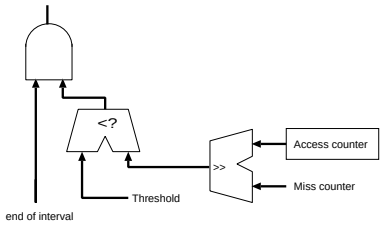

図 4 切換制御のブロック図

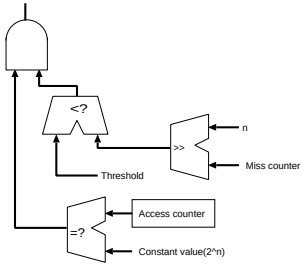

図 5 シフト演算を用いた切換制御のブロック図

キャッシュ容量	
L1 命令-cache	32KB(64B/entry, 1way, 512entry)
L1 データ-cache	32KB(64B/entry, 2way, 256entry)
L2 cache	512KB(64B/entry, 4way, 2048entry)
ヒット・レイテンシ	
L1 cache	1 cycle
L2 cache	16 cycle
主記憶	250 cycle
モード切換のオーバーヘッド	
レイテンシ	5cycle

表 1 シミュレーションのキャッシュに関するパラメータ

されるデータを一度バッファに移し (1)，その後 L2 キャッシュから必要なデータをロードし (2)，最後にバッファのデータを L2 キャッシュに書き戻している (3)．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=384K バイトとなり，キャッシュサイズを有効に活用できる．

3.3 従来のモード切換手法

可変レベルキャッシュでは，通常モードと低消費エネルギーモードがあり，この 2 つのモードを切換えることにより高性能と低消費エネルギーの両立を実現している．文献[2] で提案されているモード切換制御のブロック図を図 5 に示す．

図 5 のように一定アクセス回数 (Constant value) ごとにキャッシュミス率を測定し，閾値 (Threshold) と比較している．閾値以上だった場合通常モード，閾値未満だった場合は低消費エネルギーモードへ遷移する．また，一定のアクセス回数を 2^{13} とすることによってキャッシュミス率を測定する際に，図 4 のように除算ではなくシフト演算で測定できるようにし，ハードウェアコストを削減するものが提案されている[2]．

4. マルチプロセッサへの可変レベルキャッシュの適用時の問題点

我々はこれまで可変レベルキャッシュをシングルプロセッサ環境でのみ評価してきた．しかし，近年汎用プロセッサにおいてマルチプロセッサが注目を集めている．そこで，可変レベルキャッシュをマルチプロセッサ環境に適用し，評価を行った．

4.1 実 験 環 境

今まで可変レベルキャッシュの性能評価には，SimpleScalar を用いてきた．しかし，SimpleScalar はシングルプロセッサのプロセッサシミュレータであり，マルチプロセッサでのシミュレーションを行うことができない．また，シミュレーション速度を重視して設計されているため，プログラムが複雑になっており，マルチプロセッサ環境のシミュレーションを行えるように改造するのは困難である．そこで，本稿では東工大で開発された SimMc[8] をベースにして，共有メモリ型マルチプロセッサ環境対応に改良して使用する．SimMc にはキャッシュが搭載されていないため，L1 分離 L2 共有のキャッシュを実装し，L2 共有キャッシュに可変レベルキャッシュを実装した．プロセッサ構成は表 1 に示す．

本実験において，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通常モードと，L2 キャッシュが 256KB，L3 キャッシュが 256KB である Exclusive cache として動作する低消費エネルギーモードの 2 種類を動的に切換える．また，可変レベルキャッシュのモード切換方法は一定アクセス回数ごとにシフト演算を用いてキャッシュミス率を測定し，閾値以下ならば通常モード，閾値を超えていれば低消費エネルギーモードに切換える．今回は，アクセス回数，閾値，そして L3 キャッシュの通常モードとスリープモードの切換時間をそれぞれ文献[2] と同様に 8192(2^{13}) 回，25%，5cycle とする．

ベンチマークプログラムは SPEC2000[9] より，SPECint2000 から 164.gzip, 175.vpr, 181.mcf, 256.bzip2 の 4 種類，SPECfp2000 から 183.equake, 188.ammp の 2 種類，計 6 種類を使用し改良した SimMc でデュアルプロセッサ環境で，2 種類のプログラムを組み合わせてシミュレーションを行った．

4.2 第一次評価

図 6 に実行時間を示す．「normal」が通常キャッシュ，「level」が可変レベルキャッシュである．また，縦軸は実行時間，横軸は使用したベンチマークを表している．縦軸の実行時間は，通常キャッシュで正規化してある．結果をみると，ベンチマークの半数において性能の悪化が見られた．可変レベルキャッシュは，キャッシュアクセスの局所性が高ければ，キャッシュヒット率が高くなり，低消費エネルギーモードで動作することが多く，局所性が低けれ

Vol.2010-ARC-190 No.12
2010/8/3

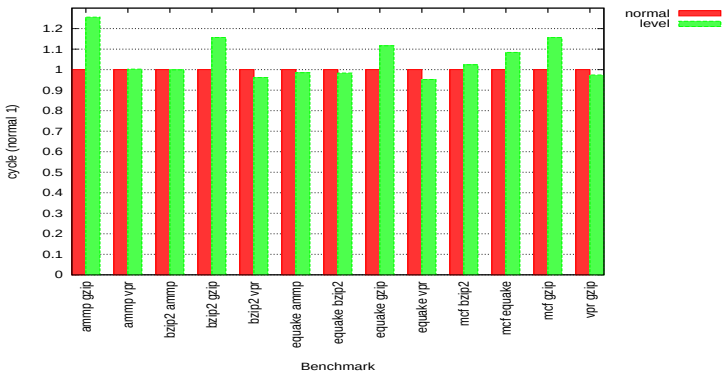

図 6 実行時間

ばキャッシュミス率が高くなり，通常モードで動作することが多くなるという特性をもつ．マルチプロセッサ環境において各プロセッサで別々のプログラムが実行している場合，それぞれのキャッシュ必要容量が異なるため，L2 キャッシュにアクセスが多いプログラムの局所性が高い場合，キャッシュヒットが多くなり，もう一方のプログラムの局所性が低く，キャッシュミスが多い場合でも低消費エネルギーモードになってしまう．上記のことが性能悪化の原因だと考えられる．

5. マルチプロセッサ向けモード切換手法の提案

5.1 従来のモード切換手法の問題点

モード切換は可変レベルキャッシュの性能を左右する重要な部分である．従来のモード切換手法を単純にマルチプロセッサに適用した場合，L2 共有キャッシュに可変レベルキャッシュを適用しているため，全 CPU のキャッシュミス率の平均値でモード切換を判定してしまう．しかし，マルチプロセッサの場合各プロセッサで実行されるプログラムによってキャッシュの必要容量が異なるため，キャッシュアクセスが多いプログラムに比重が偏ってしまい柔軟に対応ができない．

5.2 改良型モード切換手法の提案

可変レベルキャッシュを L2 共有キャッシュに実装しているため，従来手法をそのまま適用した場合はプログラム実行中の全 CPU のキャッシュアクセス回数と全 CPU のキャッシュミス回数をカウントし，一定キャッシュアクセス回数に達した際に全 CPU のキャッシュミス率の平均を測定し，モード切換判定を行う．しかし，マルチプロセッサ環境で評価を行った結果，各プロセッサで別々のプログラムが実行していた場合に性能の悪化が見られた．そこで，マルチプロセッサにおいて可変レベルキャッシュに対する全 CPU のキャッシュミス率の平均を測定するのではなく，各プロセッサでキャッシュミス率を求めることにより性能悪化を低減する手法を提案する．ブロック図を図 7 に示す．提案手法では従来の可変レベルキャッシュと同じように可変レベルキャッシュに対する全キャッシュアクセスのカウントを行い，一定キャッシュアクセス毎 (constant value) に各プロセッサで可変レベルキャッシュに対するキャッシュミス率を測定し，モード切換判定を行う．

各プロセッサのキャッシュミス率を出した後の状態遷移を表 2 示す．表のように全てのプロセッサでキャッシュミス率が閾値を下回った場合のみ低消費エネルギーモードに切換え，逆に全プロセッサで上回れば通常モードに切換えを行うことで性能悪化を低減させる．その他場合は状態を維持する．

表 2 状態遷移

Core1 \ Core2	閾値以下	閾値より上
閾値以下	低消費エネルギーモード	状態維持
閾値より上	状態維持	通常モード

6. 性 能 評 価

6.1 評 価 方 法

本稿では，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う．

DRI キャッシュとの比較は既に文献[1] でシングルプロセッサにおいて有用性が示されている．また，DRI キャッシュは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，共有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．

シミュレータは先ほどと同じ改良した SimMc を使用し，パラメータも同様な物を使用した．それぞれの手法におけるプログラムの実行サイクル数を調べ，消費エネルギーについては，文献[10][11] を参考に以下のように近似した．

まず，キャッシュアクセスによる動的エネルギーの総和 DE_{total} はラインアクセス当りの平均動的エネルギー DE_{line} とアクセス回数 $Access$ の和で求められる．よって近似式は

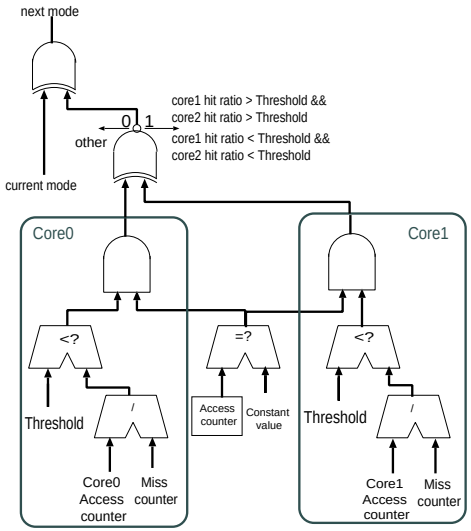

図 7 切換手法ブロック図

$$DE_{total} = DE_{line} \times Access \quad (1)$$

となる．

次にリークエネルギー LE_{total} は 1 クロックサイクルで消費するライン当りの平均リークエネルギー LE_{line} に総ライン数 $CSize$ とプログラム実行クロックサイクル数 CC の積で求まる．よって近似式は

$$LE_{total} = CC \times LE_{line} \times CSize \quad (2)$$

式 (2) の LE_{line} は，キャッシュメモリ全体のうちスリープモードとなっているラインの割合 SR とスリープモードのライン当たりのリークエネルギー LE_{sline} の積に，通常モードのラインの割合 $(1 - SR)$ と通常モードのライン当たりのリークエネルギー LE_{aline} の積を加えたものであるため，以下のように

$$LE_{line} = SR \times LE_{sline} + (1 - SR) \times LE_{aline} \quad (3)$$

となる．スリープモード時に L3 キャッシュにアクセスする場合, 通常モードに切り換える必要がある, L3 キャッシュアクセスの際，キャッシュラインをスリープモードから通常モードに切り換えるエネルギーは

$$CE_{total} = CE_{line} \times BSize \times Access_{sline} \quad (4)$$

と表される．CE_{line} はキャッシュライン当たりのモード切替エネルギーであり，$BSize$ はモードを切り換えるライン数 (バンクの大きさ)，$Access_{sline}$ は通常モードへの切換回数, つまり可変レベルキャッシュにおけるスリープモード時の L3 キャッシュに当たる部分へのアクセスの回数である．

よって，キャッシュ全体の消費エネルギー E_{total} は，式 (1)，式 (2)，式 (4) の和，すなわち，

$$E_{total} = DE_{total} + LE_{total} + CE_{total} \quad (5)$$

となる．

尚，通常キャッシュや DRI キャッシュでは，スリープモードのラインは存在しないため，$CE_{total} = 0$ となる．

キャッシュのエネルギーを評価する値は Cacti[12] を用い, 32nm プロセスを想定し，ラインサイズを 64B として求めた, 求めた値を以下に示す．

$$DE_{line} = 2.20E - 10(J) \quad (6)$$
$$LE_{aline} = 5.39E - 15(J) \quad (7)$$
$$LE_{sline} = 8.61E - 16(J) \quad (8)$$
$$CE_{line} = 1.92E - 16(J) \quad (9)$$

以上の式で求まった，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．

6.2 評価結果

本稿では，高性能と低消費エネルギーの両立を目的としているため，性能の指標として電力遅延積 (実行時間 T× 消費エネルギー E) を使用する．実験によって得られた可変レベルキャッシュの実行時間を図 8，電力遅延積による評価結果を図 9 に示す．図 8 では縦軸が実行時間 (cycle)，図 9 では縦軸が電力遅延積，横軸はともに使用したベンチマークを表している．縦軸はそれぞれ通常のキャッシュの結果で正規化したものである．

「normal」が通常キャッシュ,「level」が従来の可変レベルキャッシュ,「proposed」が提案手法である．実行時間を見てみると，依然通常キャッシュよりも実行時間がかかっているものがあるが，全てにおいて従来手法よりも性能が改善された．電力遅延積においても，通常キャッシュと比べて ammp gzip 以外の全てで良い結果となり，平均 26%程度改善されている．従来手法においても実行時間が下がった分良い結果となった．これは，従来手法では，全 CPU のキャッシュミス率でモード切換を行いっているのに対して，各 CPU のキャッシュミス率を求め，全 CPU がキャッシュ容量を必要としたときのみモード切換を行っているため，モード切換回数が減少し，低消費エネルギーモードから通常モードへ切換える際の書

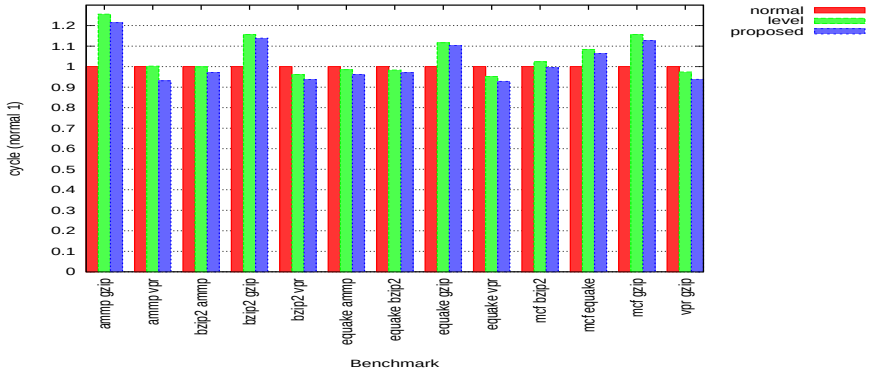

図 8 実行時間

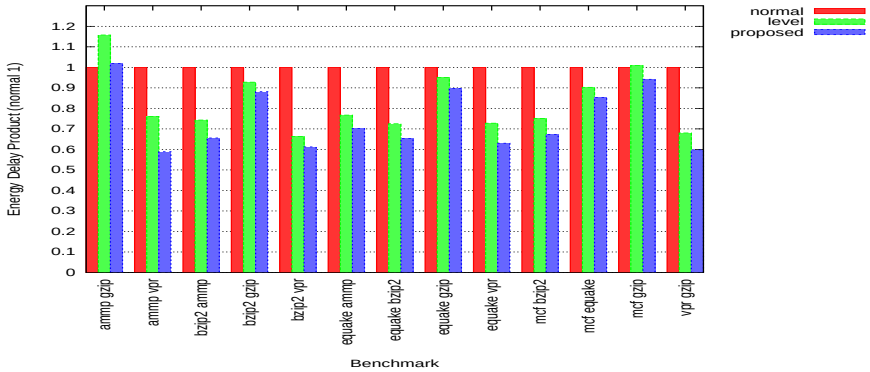

図 9 電力遅延積

き戻し回数が減ったため，良い結果となったと考えられる．しかし，ammp gzip のみ実行時間，電力遅延積ともに通常キャッシュよりも悪い結果となってしまった．また，gzip と他のベンチマークを組み合わせた結果においても、実行時間が悪化している．これは従来手法においても提案手法においても言えることであるため，従来同様にモード切換の閾値を 25%としたのがマルチプロセッサ環境での gzip の動作と相性が良くなかったのではないかと考えられる．

7. ま と め

本稿では，可変レベルキャッシュをマルチプロセッサ環境へ適用し評価を行った．その結果，可変レベルキャッシュの従来のモード切換手法ではベンチマークの半数において実行時間での性能悪化が見られた．そこで，新たなモード切換手法としてマルチプロセッサの各 CPU のキャッシュミス率を求め，全 CPU がキャッシュ容量を必要としないときのみ低消費エネルギーモード，必要としたときのみ通常モードへモードを遷移する手法を提案した．その結果，従来の可変レベルキャッシュのモード切換よりも電力遅延積において平均 7%程度改善が見られた．

今後は実際にハードウェアを設計し，詳細な評価を行う．または，ウェイ予測キャッシュや DrowsyCache のようなライン単位の切換えを行う手法との比較を行っていきたい．

謝辞 SimMc のプログラムを提供して頂き、改造についても貴重な助言を頂きました東京工業大学の吉瀬謙二先生、植原昂氏に感謝致します。

参 考 文 献

1) 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
2) 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換アルゴリズムの改良と評価 “，信学会技報，CPSY2009-44, pp.7-12，December 2009
3) S. Kaxiras, Z. Hu, and M. Martonosi, “Cache Decay: Exploiting Generational Behavior to Reduce CacheLeakage Power,” Proc. of the 28th Int. Symp. on Computer Architecture, pp.240-251, June 2001.
4) 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.
5) 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, November 2002.
6) 田中秀和，井上弘士，モシニャガ・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.
7) 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-

Vol.2010-ARC-190 No.12
2010/8/3

Performance I-Caches," Proc. of the 7th Int. Symp. on High-Performance Computer Architecture, pp.147-157, February 2001.
8) 藤枝直輝, 渡邉伸平, 吉瀬謙二, " ＳｉｍＭｉｐｓ：教育・研究に有用なＬｉｎｕｘが動く５０００行のＭＩＰＳシステムシミュレータ", 情報処理学会シンポジウム論文集, Y0978B, pp.143-150, December 2009
9) "SPEC -Standard Performance Evaluation Corporation-," URL: http://www.spec.org/.
10) 小宮礼子，井上弘士，モシニャガ・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.
11) 図子純平，冨山宏之，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.
12) CACTI 5.1 Shyamkumar Thoziyoor, Naveen Muralimanohar, Jung Ho Ahn, and Norman P. Jouppi.