待機ラインへの参照密度に基づく低リーク・キャッシュの動的制御

小宮, 礼子
福岡大学工学研究科電子情報工学専攻

井上, 弘士
九州大学大学院システム情報科学研究院

村上, 和彰
財団法人九州システム情報技術研究所 | 九州大学大学院システム情報科学研究院

出版情報：情報処理学会研究報告, 2005-ARC-164. 2005 (80), pp.73-78, 2005-08-03. 情報処理学会
バージョン：

待機ラインへの参照密度に基づく低リーク・キャッシュの動的制御

小宮礼子[†‡]　井上弘士[§]　村上和彰[‡§]

福岡大学　工学研究科　電子情報工学専攻[†]
財団法人九州システム情報技術研究所[‡]
九州大学大学院　システム情報科学研究院[§]
arch-ccc-lpc”at”c.csce.kyushu-u.ac.jp

概要

これまでに多くの低リーク・キャッシュが提案された．しかしながら，これらの手法は待機状態ラインのデータを破棄するため，ミス回数が増加し必然的に性能が低下する．そこで本稿では低リーク･キャッシュにおける性能低下抑制方式として，常活性ライン方式を提案する．具体的には，性能低下の原因となる待機状態ライン・アクセスの局所性を考慮し，アクセスが集中するラインは常活性ラインにする．これまでに提案された Cache decay 方式では，15.1%程度の性能低下をもたらす事で 92.7%のリーク削減率を達成した. これに対し, 本稿で提案する方式を適用すると, 同程度のリーク削減率 90.6%を維持しつつ，性能低下を 5.0%に抑制することができた.

Dynamic Performance Optimization for Low-Leakage Caches based on Increase-Miss Density

Reiko Komiya[†‡]　Koji Inoue[§]　Kazuaki Murakami[‡§]

Department of electronics engineering and computer Science, Fukuoka University[†]
Institute of Systems & Information Technologies/KYUSHU[‡]
Department of Informatics, Kyushu University[§]
arch-ccc-lpc”at”c.csce.kyushu-u.ac.jp

Abstract

A number of techniques to reduce cache leakage energy have so far been proposed. However, in these techniques, flushing the data of a turning off line causes a new cache miss. And, the increase miss degrade processor performance. We have analyzed the detail of cache-access behavior, and have found that there is a locality of accesses to the turning-off lines. Based on this observation, we propose a cache management technique to alleviate the negative effect of low-leakage caches. In our approach, cache lines having high degree of increase-miss locality are forced to stay in the high-speed but high-leakage mode. In our evaluation, the proposed scheme worsens the performance by only 5.0% with the same degree of energy reduction of the Cache decay approach.

1. はじめに

携帯電話やノート型 PC といったバッテリー駆動型機器の普及に伴い，マイクロプロセッサ・システムの低消費エネルギー化が重要視されるようになった．一般に，CMOS 回路の消費エネルギーは, 動的消費エネルギーと静的消費エネルギーの 2 つに大別される. 前者は回路負荷容量の充放電によって，また，後者はトランジスタの漏れ電流によって消費されるエネルギー(リーク消費エネルギー)である．従来の CMOS 回路では動的消費エネルギーが多くの割合を占めていた. しかしながら，微細化加工技術の進歩に伴い，リーク消費エネルギーによる影響が大きくなってきた. 例

えば, Pentium4 プロセッサでは全消費電力の 20%がリークに起因しており, 更なるプロセス技術の進歩に伴いこの割合はより大きくなると予想される[3]．特に，大量のトランジスタで構成されるキャッシュ･メモリにおいては，リーク消費エネルギーの削減が極めて重要となる．0.07μm プロセスを想定した場合, キャッシュ全消費エネルギーの 70%はリークに起因するとの予測もある[3].

この問題を解決するため, これまでに様々なキャッシュ･リーク消費エネルギー削減手法が提案された．これらの手法は, 以降参照されないと予測されたキャッシュ･ラインを動的に活性状態から待機状態へ切替えることによってリークを削減する．待機ラインの実現方法には状態破棄([2][5])と状態保存([1][3])の 2 通りがある. 前者は待機状態へと移行するラインの SRAM セルに対して電源電圧の供給を停止することによりリーク消費エネルギーを削減する．そのため SRAM セルに記憶されていた情報は失われる．一方, 後者は SRAM セルに記憶されたデータを失わない程度まで電源電圧を下げることで実現される. 状態破棄と比較して, ライン当たりのリーク消費エネルギーは大きくなるものの, 従来型キャッシュと同じヒット率を維持できる．文献[4]において, 待機状態の実現方法は下位記憶階層への参照クロック･サイクルが長いならば状態保存，短いならば状態破棄が適していると報告されている. しかしながら, トランジスタの微細化加工技術の進歩に伴い, 電源電圧値は今後降下していくと予測される．この場合, データが損なわれない程度の低い電源電圧と通常の電源電圧を頻繁に切替えることは困難である. そのため, 状態破棄による待機状態の実現方法が今後主流になると予想される.

状態破棄によって待機ラインを実現した場合, 待機ラインのデータは破棄されるため当該ラインへの参照は必ずキャッシュ･ミスとなる．再参照されるデータを破棄した場合, リーク削減手法を用いない従来型キャッシュ･メモリと比較してミス回数が増加する. その結果, プロセッサの性能は低下する．これは，増加したミスに伴い下位記憶階層への参照が発生するからである. 本稿では，このミスを**“増加ミス”**と呼ぶ．高い性能と低消費エネルギーを両立させるためには, 低リーク･キャッシュにおける増加ミスによる性能の低下を抑制することが重要となる.

そこで本稿では, 低リークキャッシュにおける性能低下の抑制を目的として, 待機ラインへの参照の局所性を活用した動的キャッシュ制御方式を提案する．また，ベンチマーク･プログラムを用いたシミュレーションを行い, 提案手法の有効性を評価する．本手法では, 増加ミスが集中して発生するラインを活性状態で動作させる. これにより，増加ミスを回避し，性能低下を抑制する．本方式が最も効果的に動作した結果, 従来の低リークキャッシュと同程度のリーク削減率を維持しつつ，性能低下を 10%抑制できた.

以下，第 2 節では実験環境を定義し，第 3 節では増加ミスの振る舞いを詳細に解析する．次に，第 4 節で性能低下を抑制するキャッシュ制御方式を提案し，第 5 節でベンチマーク･プログラムを用いた定量的評価を行う. 最後に第 6 節で簡単にまとめる.

2. 評価環境

キャッシュ･メモリのリーク消費エネルギー(*LEtotal*)は，プログラム実行時間(*CC*)，クロックサイクル当りの 1 ライン平均リーク消費エネルギー(*LEline*)，全キャッシュ容量における活性ライン数の割合(*AR*)，ならびに，キャッシュ全体のライン数(*Nline*)によって近似できる.

$$LEtotal = CC * LEline * AR * Nline \quad (1)$$

$$CC = CCconv + CCextra \quad (2)$$

ここで，*CCconv* はリーク削減手法を用いない場合のプログラム実行時間(クロックサイクル数)であり，*CCextra* はリーク削減手法の採用に伴う実行時間オーバヘッドを表す. リーク削減手法を用いないキャッシュの場合は *CCextra* が 0 となり, *AR* は 1 になる．一方，リーク削減手法を適用すると *CCextra* が増加し，*AR* は減少する．なお，本評価では，キャッシュ･アクセスやモード変更に伴う動的消費エネルギーは考慮しない.

CCconv, *CCextra*, および *AR* を測定するため，マイクロプロセッサ･シミュレータである SimpleScalar[6]を利用した．本実験で想定したプロセッサ構成を表 1 に示す. 評価対象アプリケーションは SPEC CPU 2000 ベンチマーク･セット[7]より 4 個の浮動小数点プログラムと 6 個の整数プログラムを用いた．なお，本実験では各ベンチマークにおいて，実行開始時から 10 億命令をフォワードし，続く 5 億命令を評価対象とした.

3. 増加ミスの振る舞い解析

状態破棄により待機ラインを実現する低リーク･キャッシュには様々な手法がある．それらの中で代表的な手法として“Cache decay”が挙げられる．我々が提案する低リーク･キャッシュの制御方法はどの低リーク･キャッシュに対しても適

表 1：プロセッサ構成

命令発行方式	インオーダ
分岐予測器	
種類	bimodal
サイズ	テーブル:128 エントリ, 1 ウェイ
命令デコード幅	2 命令/サイクル
命令発行幅	2 命令/サイクル
IFQ サイズ	2 エントリ
RUU サイズ	16 エントリ
LSQ サイズ	16 エントリ
キャッシュ･メモリ	
L1 データ	32KB (32B/エントリ, 32 ウェイ, 1K エントリ)
L1 命令	32KB (32B/エントリ, 32 ウェイ, 1K エントリ)
L2 共有	なし
ヒットレイテンシ	
L1 キャッシュ	1 クロック
L2 キャッシュ	なし
主記憶	32 クロック
メモリ･バンド幅	8B
メモリ･ポート数	1
ITLB，DTLB	
エントリ数	1M エントリ (4KB/エントリ, 32 ウェイ，32 エントリ/ウェイ)
ミスペナルティ	30 サイクル
整数演算器 (装置数, 実行レイテンシ, 発行レイテンシ)	
ALU	1, 1 クロック, 1 クロック
乗算器	1, 3 クロック, 1 クロック
除算器	1, 20 クロック, 19 クロック
浮動小数点演算器 (装置数, 実行レイテンシ, 発行レイテンシ)	
ALU	1, 2 クロック, 1 クロック
乗算器	1, 4 クロック, 1 クロック
除算器	1, 12 クロック, 12 クロック

用可能であるが，本稿では Cache decay を基準として性能低下抑制手法を提案，評価する．

3.1 増加ミスが性能に与える影響

待機ライン数が多い場合，高いリーク削減効果を得られる一方，増加ミス回数も増すため極端に性能が低下する．つまり，ラインの状態がリーク削減効果と実行時間に大きな影響を与える．したがって，待機/活性間の動作状態切替え制御は非常に重要となる．Cache decay の場合，アクセスされること無しにキャッシュ内に滞在する期間が閾値期間(decay-interval)以上となった時点で待機状態へと変更し，アクセスが発生したラインに対して活性ラインに変更する．本稿では decay-interval を 4K クロック･サイクル，また，リーク削減手法を L1 データ･キャッシュに適用すると仮定する．

リーク削減手法を用いない従来のキャッシュでの L1 データ･キャッシュ･ミス回数によって正

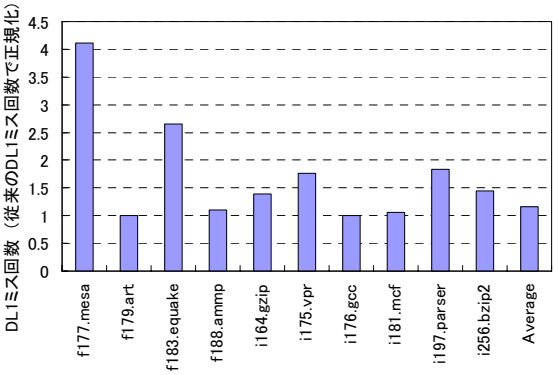

図 1：Cache decay の DL1 ミス回数

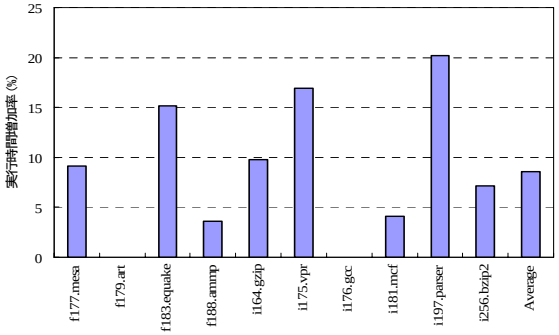

図 2：Cache decay の性能低下

規化された Cache decay のミス回数を図 1 に示す．f179.art と 176.gcc を除いた 8 ベンチマークにおいて増加ミスが発生している．次に，実行時間増加率を図 2 に示す．実行時間増加率とは，リーク削減手法を用いない従来のキャッシュに対する Cache decay の性能低下である．これらの結果から，増加ミス回数が多いベンチマークほど性能が低下することが明らかである．

3.2 Increase Miss Density

一般に，メモリ参照には局所性が存在するため，増加ミスに関しても何らかの局所性が存在すると予測される．そこで，プログラム実行における増加ミスの局所性を調査した．ここで，キャッシュ・ライン i における増加ミスの局所性 IMD_i(*Increase Miss Density*)を以下の式で表す．

$$IMD_i = \frac{N_i}{N_{avg}} \qquad (3)$$

N_i はライン i における増加ミス回数，N_{avg} は全ラインに関する増加ミス回数の平均である．例えば，IMD_2 が 1 以上の場合，ライン 2 は平均以上の増加ミスが発生していることになる．

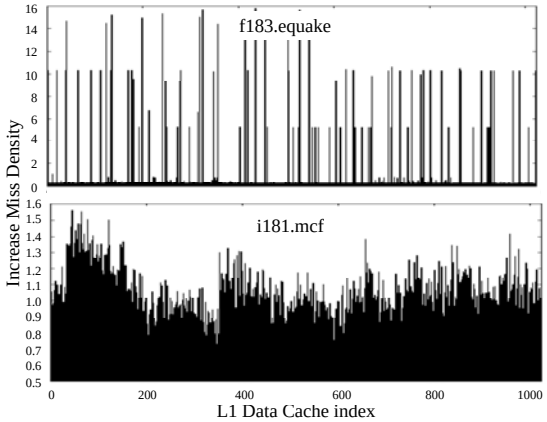

図 3：Increase Miss Density (f183.equake, i181.mcf)

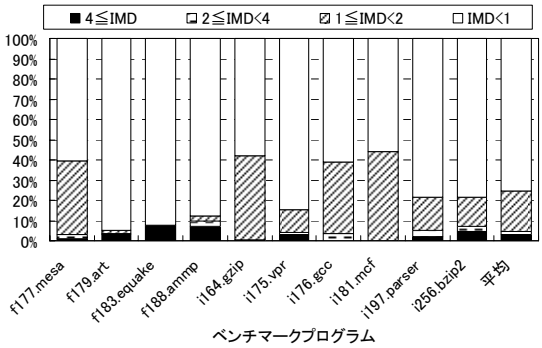

図 4：IMD_i に関するライン数の内訳

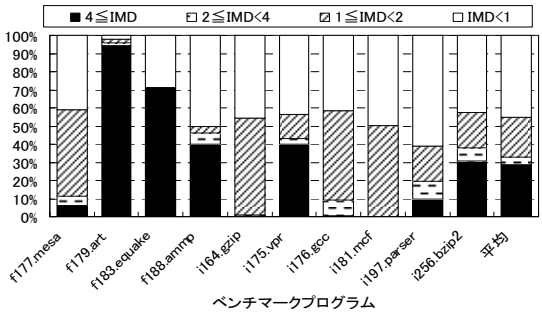

図 5：IMD_i に関する増加ミス回数の内訳

L1 データキャッシュを想定し，各ラインにおける IMD_i を測定した．特徴的な結果を示した 2 ベンチマークの結果を図 3 に示す. 横軸はキャッシュ･インデックスを表している．f183.equake の場合，増加ミスが一部のラインに集中している．したがって, 非常に局所性が高いといえる. 一方, i181.mcf の場合，多くのラインが 1 以下の IMD_i であり，全ラインが平均的に参照されている．このことから, ベンチマークごとに増加ミスの局所性は異なることが分かる. 図 4 はキャッシュ中で IMD_i の値が 1 未満, 1 以上 2 未満, 2 以上 4 未満，または，4 以上であったラインがキャッシュ全体に占める割合を示している．大多数のキャッシュ･ラインは IMD_i の値が 1 未満であり，また，ごく僅かなライン(10%以下のライン)が 4 以上の IMD_i 値を示している．次に，図 5 に各 IMD_i 値のラインにおいて発生した増加ミスが全増加ミス回数に占める割合を示す. 例えば *f179.art* の場合，全増加ミスのうち 94%が IMD_i 値 4 以上(黒の部分)のラインで起きている．多くのベンチマークにおいて, 高 IMD_i のキャッシュ･ラインが多数の増加ミスを引き起こしている. これらの結果から, ごく一部のラインが, 多数の増加ミスを引き起こしていることが明らかになった．したがって，増加ミスには局所性があると言える．例えば，IMD_i が 4 以上であるラインは平均 2.8%と非常に少ないにもかかわらず，28.6%もの増加ミスを引き起こしている．また，54.7%の増加ミスが 24.8%の IMD_i1 以上であるラインにおいて起きている．

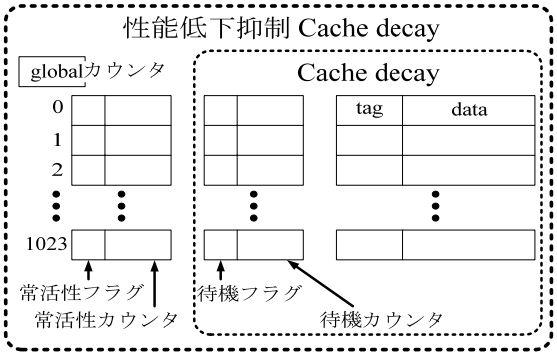

図 6：性能低下抑制キャッシュ

4. 性能低下を抑制するキャッシュ制御方式

性能低下を抑制するには, 再参照されるラインを活性状態に保つ必要がある．その手段として，集中的に増加ミスが発生するラインを常に活性ラインとして動作させ, 待機状態への切替えを防ぐ制御方式を提案する．以降このラインを“**常活性ライン**”と呼ぶ.

例として, 常活性ラインによる性能改善手法を支援する 1 ウェイ 1024 ラインのキャッシュを図 6 に示す．内部の点線で囲まれた領域が従来の Cache decay である．Cache decay はラインごとに非アクセスサイクル数を待機カウンタにて保持する．カウンタの値が decay-interval 以上になると待機ラインへと切替わる(待機フラグ=1，データ領域の供給電圧=0)．次に，常活性ラインを支援するために必要な機構を考える. 待機ラインへの参照局所性の判定には IMD_i の値を用いる．IMD_i を求めるには，式(3)より，全ラインに関する増加ミス回数の平均および, 各ラインの増加ミ

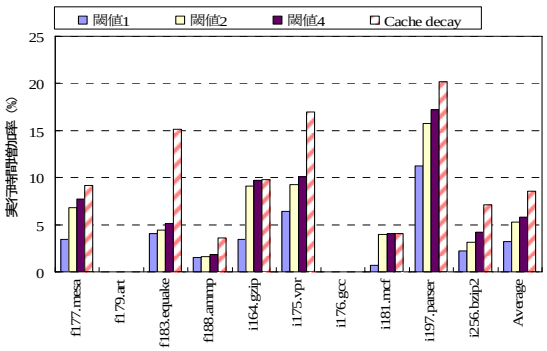

図 7：性能低下

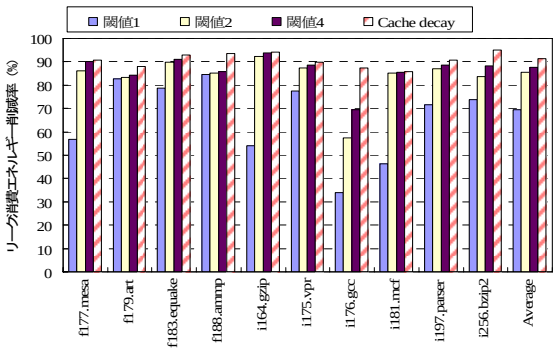

図 8：リーク消費エネルギー削減効果

ス回数が必要である．そのため，キャッシュ全体の増加ミス回数を保持する global カウンタ，また，各ラインに常活性フラグと常活性カウンタを追加する．増加ミス発生時に global カウンタと参照されたラインの常活性カウンタをカウントアップする．これらのカウンタ値を用いて IMD_i を求め，その値が閾値以上ならばそのラインは性能低下を引き起こしていると判断し，常活性ラインに切替える(常活性フラグ=1)．常活性ラインは待機フラグの状態に関わらず，常に活性ラインとして動作する．逆に，IMD_i が閾値以下の場合，当該ラインは性能低下に影響を与えていないと推測されるため，Cache decay と同様に動作する．

常活性ライン数が過剰になると，性能低下は改善されるがリーク消費エネルギーの削減効果が得られなくなる．それに対して，常活性ライン数が極端に少ない場合は従来の低リーク･キャッシュと比較して性能が改善されない．

5. 評価

本節では，性能低下抑制方式を適用した低リーク・キャッシュの評価を行う．具体的には性能オーバヘッドおよびリーク消費エネルギー削減率を測定し，従来の低リーク・キャッシュである Cache decay と比較することで本方式の有効性を議論する．

5.1 実行時間増加率

本提案方式が性能低下抑制に与える影響について評価する．評価モデルは常活性ラインへ切替える IMD_i の閾値が 1，2，4 の 3 パタン(それぞれ閾値 1 モデル，閾値 2 モデル，閾値 4 モデルと呼ぶ)と Cache decay であり，各評価モデルの実行時間増加率を図 7 に示す．f179.art，i176.gcc の全モデル，また，i164.gzip の閾値 4 モデル，および i181.mcf の閾値 2, 4 モデルは Cache decay と同等の性能低下を示している．一方，それ以外のモデルでは性能低下抑制方式の適用により，性能が改善されている．特に，閾値に反比例して性能改善効果が高まることがこの図から明らかである．全ベンチマークを平均すると，閾値 1，2，4 モデルにおいて，それぞれ 5.3%，3.3%，2.7%性能低下抑制効果が得られた．

5.2 リーク消費エネルギー削減率

次に，性能低下抑制方式がリーク消費エネルギーに与える影響を評価する．各モデルのリーク消費エネルギー削減率を図 8 に示す．閾値 2，4 モデルは i176.gcc を除く全てのベンチマークにおいて高いリーク削減効果を維持している．Cache decay に対して，閾値 2 モデルは 3.5%，閾値 4 モデルは 5.8%のリーク消費エネルギー増加に抑えている．しかしながら，多くのベンチマークプログラムにおいて閾値 1 モデルはリーク削減効果が Cache decay と比較して低下する．具体的には，閾値 1 モデルでは平均して 21.6%リーク消費エネルギーが増加する．

5.3 考察

5.1 節および 5.2 節の結果から，提案方式の性能改善とリーク削減効果について議論する．評価結果は以下の 4 つのパタンに分類できる．

1 つ目は性能，リーク削減効果ともに Cache decay と変わらないパタンである．例えば，i181.mcf の閾値 2，4 モデルがこの分類に当てはまる．図 5 から，i181.mcf は IMD_i の値が 2 以上のラインにおいて増加ミスが発生しないことが分かる．そのため，閾値 2，4 モデルでは常活性ラインへの切替えが少なくなり，Cache decay と同様の動作をする．その結果，性能は改善されず，リーク削減効果も Cache decay と等しいままであった．

2 つ目のパタンは, f179.art や i176.gcc のように性能低下が改善されず, かつ, リークが増加するパタンである. これらのベンチマーク・プログラムは Cache decay において性能が低下していない. そのようなベンチマークに対して性能低下改善手法を適用すると, 冗長な活性ラインが増加しリーク消費エネルギーも増してしまう.

3 番目は性能オーバヘッドを改善し, かつ, Cache decay と同等のリーク消費エネルギー削減効果を得られるパタンである. f183.equake の閾値 2, 4 モデルがこのパタンである. このベンチマークはごく一部のラインが非常に多くの増加ミスを引き起こしている(図 4, 図 5). このような場合, 待機ラインへの参照密度に基づく本提案方式は非常に有効に働く.

最後はリーク削減効果を犠牲にして性能を改善するパタンである. i256.bzip2 がこの特徴をはっきりと示している. これは, 常活性ラインの増加に伴い増加ミスは削減出来るものの, 当該ラインで消費されるリークエネルギーが増加するためである. 閾値が小さいほど常活性ラインの数は増すため, 閾値に反比例して性能改善効果は高まり, リーク消費エネルギーも増す. したがって性能へ高い要求がある場合, 閾値を 1 として性能低下抑制方式を用いれば良い. 逆に, 高いリーク削減率が求められる場合には閾値を大きくすると良い.

6. おわりに

本稿では, 待機状態時の参照局所性が高いラインに対して常に活性状態で動作させることによって性能低下を抑制する方式を提案し, その評価を行った. その結果, 多くのベンチマークプログラムにおいて Cache decay と比較して, 性能を改善できることが分かった. しかしながら, 常に活性状態で動作させるライン(常活性ライン)数に比例して性能低下は改善されるがリーク消費エネルギーも増加することがわかった. 本手法が有効に動作した場合, Cache decay とほぼ同等のリーク削減率 90.6%を実現し, 性能低下は 5.0%に抑えることが可能である(f183.equake 閾値 4 モデル).

本稿では, エネルギーはリーク消費エネルギーのみに着目している. キャッシュ・メモリで消費されるエネルギーには動的消費エネルギーもある. それらも含めたキャッシュ全体のエネルギー評価が今後の課題である. また, 本評価ではライン状態の切替え等によって発生する動的消費エネルギー・オーバヘッドも含まれていない. 今後, これらを含んだより詳細な評価を行う予定である.

謝辞

本研究を進めるにあたり, 多くのご指導を頂いた富士通株式会社の池田正幸氏, 丸山拓巳氏, 富士通研究所の勝野昭氏, 坂本真理子氏に深く感謝致します. なお, 本研究は一部, 文部省科学研究費補助金(課題番号 : 14GS0218, 14702064, 14102027）による.

参考文献

[1] K.Flautner, N.S.Kim, S.Martin, D.Blaauw, and T.Mudge, “Drowsy Caches: Simple Techniques for Reducing Leakage Power,” *Proc. of the 29th Int. Symp. on Computer Architecture*, pp.148-157, May 2002.

[2] S.Kaxiras, Z.Hu, and M.Martonosi, “Cache Decay: Exploiting Generational Behavior to Reduce Cache Leakage Power,” *Proc. of the 28th Int. Symp. on Computer Architecture*, pp.240-251, June 2001.

[3] 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, Nov. 2002.

[4] D. Parikh, Y. Zhang, K. Sankaranarayanan, K. Skadron, and M. Stan, “Comparison of State-Preserving vs. Non-State-Preserving Leakage Control in Caches,” *Workshop on Duplicating, Deconstructing and Debunking*, pp.14-25, June 2003.

[5] M. Powell, S. Yang, B. Falsafi, K. Roy, and T. N. Vijaykumar, “Gated-Vdd: A Circuit Technique to Reduce Leakage in Deep-Submicron Cache Memories,” *Int. Symp. on Low Power Electronic and Design*, pp.90-95, July 2000.

[6] SimpleScalarLLC, http://www.simplescalar.com

[7] SPEC –Standard Performance Evaluation Corporation, http://www.spec.org/