修 士 論 文

データ圧縮を用いたキャッシュメモリの消費電力削減に関する研究

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

松田 愛子

2006 年 3 月

修 士 論 文

データ圧縮を用いたキャッシュメモリの消費電力削減に関する研究

指導教官　田中清史 助教授

審査委員主査　田中清史 助教授
審査委員　日比野靖 教授
審査委員　井口寧 助教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

410109 松田 愛子

提出年月: 2006 年 2 月

概 要

近年のマイクロプロセッサの特徴として消費電力の増大が上げられる．また，プロセッサと主記憶の速度差を隠蔽するためにキャッシュメモリが増加してきている．その結果，プロセッサの全体の消費電力に対してキャッシュメモリの消費電力が 50% に達する状況になっている．本研究ではプロセッサの消費電力の大部分を占めるキャッシュメモリに着目し，低消費電力化を達成するキャッシュアーキテクチャの提案を目的とする．

目 次

第1章　はじめに

1.1　研究の背景

1.1.1　マイクロプロセッサの新たな問題

近年，マイクロプロセッサの消費電力増大が大きな問題となっている．消費電力の増大はバッテリー駆動型モバイル機器の駆動時間に大きな影響を及ぼし，バッテリー駆動型モバイル機器の特徴である携帯性に制限がでる．また，消費電力の発熱は機器へ負荷を与え機器の寿命を早めてしまう．特に，機器内部が高密度化されているものであれば熱を放出することは難しく機器への負担がかなり大きくなる．発熱による機器の負荷を減らすために冷却装置を備えているものはあるが，消費電力による発熱を抑えられるが冷却装置の消費電力がかかるので，結果的に機器全体の電力消費量が増える．

このまま消費電力が増大していくと，消費電力による電気代がハードウェアコストを容易に上回る可能性がありユーザへの金銭的負担は大きくなる．また，マイクロプロセッサは今や社会に必要なもであり多く出回っているので，一つ一つのマイクロプロセッサの消費電力量がたいしたことがなくとも，全世界で使われているプロセッサで見でみるとその消費電力の増大は地球環境に悪い影響を与える．

1.1.2　消費電力問題の発端

一昔前はスーパーコンピュータの処理が現在は家庭の PC で達成されており，プロセッサの性能向上は著しい．プロセッサの性能向上は，動作周波数の上げることで達成されてきた．これは，半導体技術と製造技術の向上によりプロセスルールの細微化とトランジスタの集積化が可能となりその結果，動作周波数とトランジスタ数は増大しプロセッサの処理能力が向上した．

しかし，どんどんプロセスルールを細微化していくことで，リーク電流をの存在が無視できない状況になった．マイクロプロセッサの消費電力はトランジスタのスイッチングに要する動的消費電力と常に発生しているリーク電流による静的消費電力に分けられるが，静的消費電力の占める割合が高くなっている．静的消費電力の増大はプロセッサの動作周波数向上の妨げとなり，性能に影響してくる．

1.1.3 キャッシュメモリと消費電力

一方では，機器上で実行するソフトウェアは大規模化および複雑化してきており高速化が求められている．プロセッサ・主記憶の高速化はされているが，主記憶はプロセッサの動作周波数向上と同様な高速化まではいかない．ITRS(International Technology Roadmap for Semiconductors) ロードマップによれば，2007 年までに 1 年ごとにトランジスタのパフォーマンスは 21% 向上し，一方で DRAM のレイテンシは 10% の向上と予測されている．したがって，プロセッサと主記憶の速度差は増大の傾向にある [1]．これはプログラムを実行している際，主記憶のアクセスは実行速度に大きな影響を与える．

そのため，主記憶へのアクセス回数を減らすためキャッシュメモリが設けられ，実行プログラムの実行速度への影響を軽減することが可能となった．また，トランジスタの小型化と集積化はチップ上にキャッシュメモリを多く載せることが可能となり，増加の一途をたどっている．これは，キャッシュメモリがプロセッサの面積の大部分を占めるようになり，その消費電力の占める割合はプロセッサ全体の 50% に達すると報告されている [2]．近年は，キャッシュの消費電力を削減する手法が研究されている [3, 6, 7]．

1.2 研究の目的

本研究では，プロセッサに大きな影響を及ぼす消費電力の削減を狙う．その中でもプロセッサの消費電力を大きく占めるキャッシュメモリに注目し，オーバヘッドで低消費電力化を達成するキャッシュアーキテクチャの提案を目的とする．低消費電力化を実現するための手段に，データ圧縮と電圧制御を用いる．

一般的なプロセッサである１次・２次キャッシュとライトバッファを持つオンチップキャッシュを想定し，２次キャッシュへ送られるデータに対してデータ圧縮を行う．そして，圧縮により空き領域となっている部分に対して電圧を制御することによりキャッシュメモリの低消費電力化を実現する．本方式では，２次キャッシュの有効な容量を制限していないため，ヒット率は通常のキャッシュと変わらない．また近年は，チップ上の L2 キャッシュは大容量化しており，L2 キャッシュのリーク電流を削減することはプロセッサの全体の消費電力削減につながる．

本研究の提案方式の評価をシミュレーションにより行い，その際には電圧制御により電圧がオフ状態となっていたキャッシュメモリがオンとなったときに生じる Setup Time を考慮し評価を行い，本研究の有効性を示す．

1.3 本論文の構成

本論文は．６章からなる．第２章では，世の中に出ているプロセッサで実用化されている消費電力削減法と本研究の関連研究の紹介をする．第３章では，本研究で提案するキャッシュ消費電力削減法の基本方針，第４章と第５章で提案方式で用いるデータ圧縮ア

ルゴリズムと電圧制御について説明する．第６章で，提案方式の評価を行う．最後に第７章をまとめとする．

第2章　プロセッサの消費電力削減法

プロセッサの消費電力への注目は大きくなっている．ここ数年で，消費電力削減技術を採用しているプロセッサが登場している．ここでは，商用プロセッサで採用されている技術の紹介と，キャッシュの消費電力削減に関する研究を紹介する．

2.1　動作周波数と消費電力

プロセッサの消費電力は，動作電圧の２乗に比例し動作周波数に比例する．動作周波数を向上させることは消費電力の増加に影響を与えるが，動作電圧を下げることでプロセッサの消費電力を抑えることが可能である．これまでは，消費電力は増加してはいたが深刻に受け止める必要がなくプロセッサ性能向上に力を注げた．

しかし，どんどんプロセスの細微化を行っていくことは動作電圧を下げることを難しくしさせ消費電力を抑えることが困難にした．そして，リーク電流の存在を際立たせ，リーク電流による消費電力が無視できない状況となった．また，プロセスルールの細微化に伴いトランジスタ数が増大しているので，消費電力の増大は深刻である．このまま，消費電力が増大してしまうと，消費電力による発熱は数年後には太陽の表皮と同じレベルまで達するといわれており，プロセッサの消費電力削減は重要事項である．プロセッサの大部分を占めているキャッシュに注目した本関連研究を挙げる．

2.2　実用さている消費電力削減法

プロセッサの消費電力削減が重要な課題となっている今日，消費電力削減されたプロセッサがでている．実用化されている消費電力技術の例を次にあげる．

- Clock Gating
  マイクロプロセッサ内で動作していない回路ユニット単位でとめる技術．マイクロプロセッサ内部やその周辺回路は常に動作しているわけではないので，必要がない回路の電源供給を止めることで，消費電力を削減する．

- 回路サイズの最適化
  高速動作が求められる部分はサイズが大きくともドライブ能力が高いトランジスタを，逆に高速動作が要求されない部分には，小型で消費電力の少ないトランジスタ

を利用する．各部分にあうトランジスタを採用することで回路全体を最適化をし消費電力削減を可能にする．

Clock Gating と回路サイズの最適化は Pentium M プロセッサで用いられている．

- 動作周波数を動作電圧の両方を切り替え
  一般に動作周波数と動作電圧の両方を下げれば性能は下がるが消費電力は下げることができる．特に消費電力は動作電圧の二乗に比例するため、動作電圧を切り替えることは消費電力の削減に大きく貢献することになる。

  例えば Transmeta の Crusoe では，ソフトウェアの実行中にプロセッサの負荷の大きさに応じて動作周波数を動的に変更を行う．インテルの Pentium3 では AC アダプタから電源が供給されているときは高電圧・高クロックで動作し，バッテリ動作に切り替わると電圧・クロックを低下させ動作時間を延長する仕様になっており，pentium 3-M 以降では電源によらず高負荷ならばパフォーマンスを上げ，低負荷ならばパフォーマンスを下げるといったように柔軟に対応することが可能となっている．

2.3 関連研究

プロセッサの大部分はキャッシュで占められている．したがって，キャッシュが占める消費電力の割合も大きい．近年は，キャッシュの消費電力削減に関する研究がされている．ここでは，本研究に関連するキャッシュの消費電力削減の研究について述べる．

2.3.1 DRI cache [6]

DRI cache [6] は動的にキャッシュサイズを変更することでキャッシュの消費電力削減を行う．図 2.1 にダイレクトマップ DRI i-cache の詳細を示す (同様にセットアソシアティブキャッシュでも適応する)．この提案方式で用いられるのはキャッシュのミス率である．実行アプリケージョンの決められた時間間隔 (interval) におけるキャッシュミス数をカウントする．キャッシュミス数は miss counter に保持される．interval の終わりに前もってセットされているキャッシュミス数の閾値 miss-bound と miss counter に保持されているキャッシュミス数とを比較することで，キャッシュサイズの変更を決定する．閾値よりキャッシュミス数が多ければキャッシュサイズを増大させ，キャッシュミス数が少なければキャッシュサイズを縮小する．そして，使用されていないキャッシュ領域に対して，Gated-Vdd によりキャッシュの電圧をオフの状態にしキャッシュの消費電力を削減する．

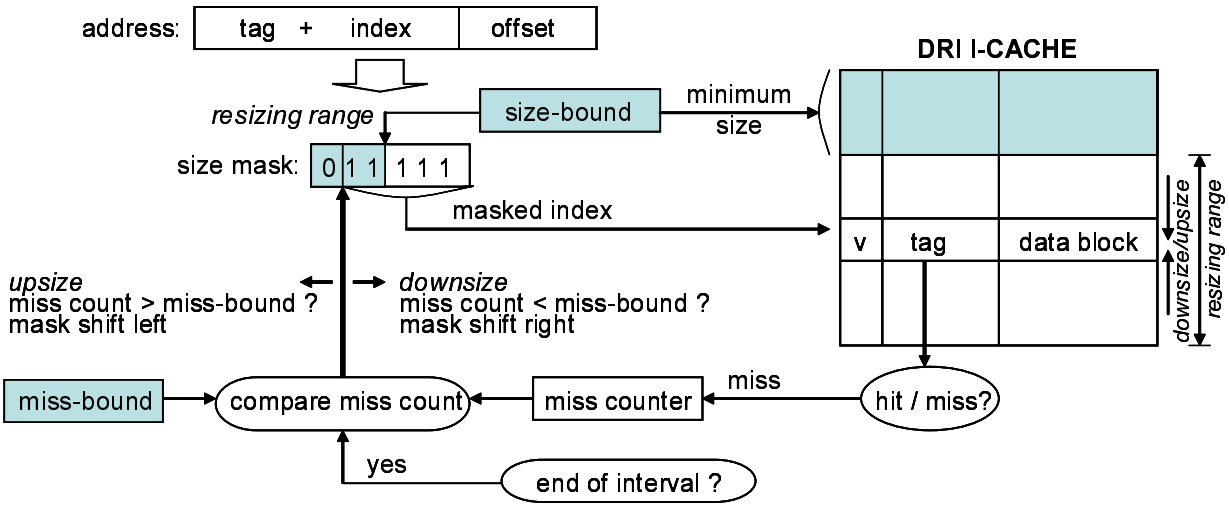

図 2.1: DRI i-cache

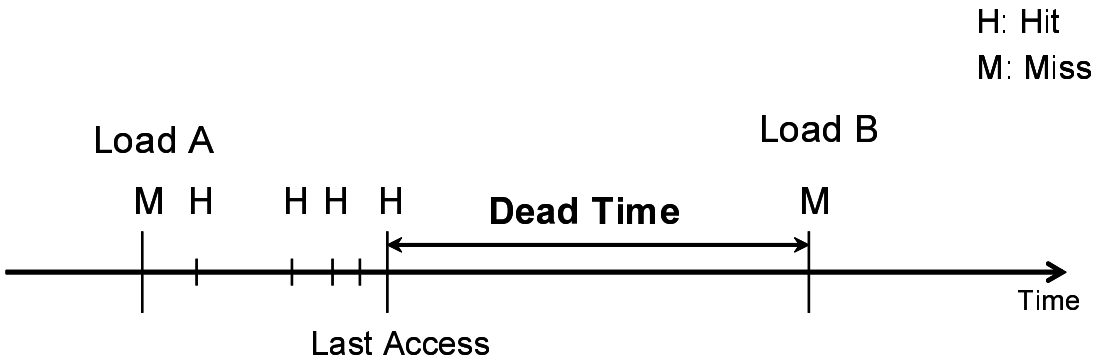

図 2.2: あるエントリーにおけるキャッシュ参照

2.3.2 Cache Decay [7]

dead time というキャッシュの時間情報を用いて，dead time に入ったと判断されたキャッシュブロックに対し Gated-dd によりキャッシュブロックの電圧をオフ状態にし消費電力削減をする．

図 2.2 にあるエントリーのキャッシュブロック参照の流れの例を示す．dead time はある時刻に格納されているキャッシュブロック (cache block A) が最後にキャッシュヒットしてから，新たなキャッシュブロックの参照 (cache block B) のため追い出されるまでの時間のことである．もしキャッシュブロックがキャッシュブロックが長い間参照されずに追い出されてしまうなら，非動作時のトランジスタによるリーク電流により多くの電力が消費されてしまう．deat time 中のキャッシュブロックを電力削減対象とすれば性能を落とさずに消費電力の削減可能となる．

2.3.3 Drowsy Cache [3]

キャッシュラインの電源電圧をデータが失わない程度まで低くすることによってキャッシュの消費電力を削減する．DRI cache [6] と Cache Decay は消費電力削減の対象となったキャッシュブロックの電圧を完全に落とし，保持されているデータは失われる．よって，データが失われたことによりキャッシュのミス率が増大する可能性がある．Drowsy Cache は，リーク電流の削減率は他の二つに比べると低くなるが，データは保持されているのでキャッシュのヒット率は従来のキャッシュと同等である．低電圧状態でキャッシュヒットとなった場合，電圧を元の状態に戻してからデータにアクセスされるので，アクセスレイテンシは従来のキャッシュに比べると長くなるが，下位の記憶階層へのアクセスがないと考えると，実行サイクル数への影響は大きくない．

第3章　キャッシュ低消費電力化

本章ではキャッシュ低消費電力化の基本方針について述べる．

3.1　メモリ階層

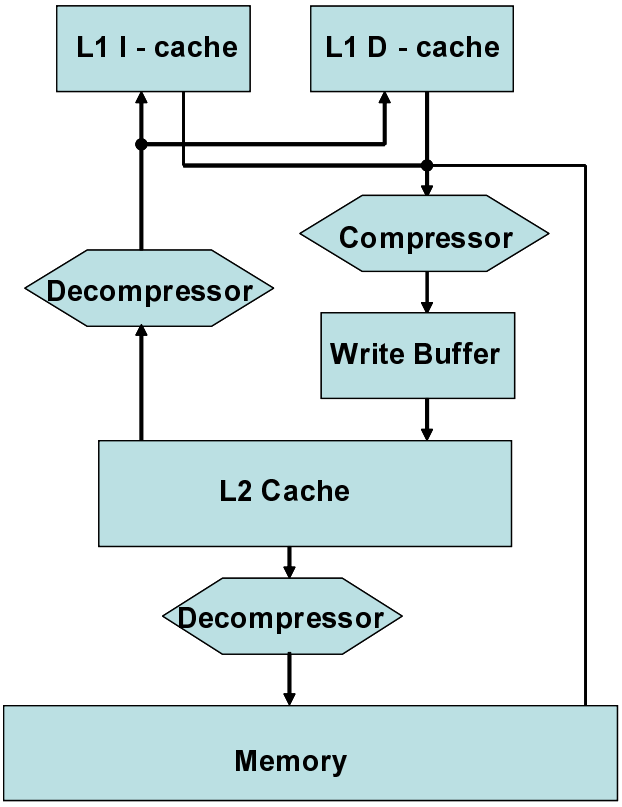

図 3.1: メモリ階層

本研究では L1 命令キャッシュ・データキャッシュ，L2 キャッシュそしてライトバッファがオンチップ上にあるアーキテクチャを想定している．

図 3.1 に提案するメモリ階層を示す．本研究では消費電力削減を行うために，データ圧縮を行う．そのために，想定するアーキテクチャに新たに，データ圧縮・復元を行うためのハードウェア Compressor と Decompressor を各一つず設ける．圧縮ターゲットは L2

キャッシュとする．L1 命令キャッシュ・データキャッシュまたはメモリから L2 キャッシュへデータを送る場合,compressor により圧縮を行なう．compressor を通過したデータはライトバッファへ送られ，順次 L2 キャッシュへデータを格納する．L2 キャッシュから L1 命令キャッシュまたはメモリへデータを送る際，圧縮されたデータの場合は decompressor によりデータを復元する．圧縮されていないデータは復元によるオーバヘッド削減のため decompressor を通さずに直接送る．データの書き込みが発生した場合，L1 キャッシュから L2 キャッシュへ，そして L2 キャッシュからメモリへの書き込みは write-back 方式を用いる．

本研究では L2 キャッシュのみを圧縮対象としている．データアクセスが頻繁に行われる L1 キャッシュを圧縮対象にするのはプロセッサのパフォーマンスに大きく影響を与える．圧縮・復元サイクルは L1 のアクセスレイテンシに比べ非常に大きく，その実行サイクルに与える影響は計り知れない. L1 キャッシュよりもアクセス数が少なくアクセスレイテンシが大きい L2 キャッシュならば，圧縮・復元サイクルが加わったとしても実行サイクルに与える影響は少なくすむ．そして，L2 キャッシュの面積は L2 キャッシュよりも大きいため，圧縮による消費電力削減は効果的である．

3.2 消費電力削減法

データ圧縮により空いた領域に対して電圧制御によりキャッシュの低消費電力化を行う．compressor による圧縮はキャッシュブロック単位で行い，ブロックサイズの 1/2 以下にに圧縮できれば対象 L2 キャッシュブロックへ圧縮された形で格納し，できなければそのまま非圧縮でデータを格納する．

圧縮により空いている領域には，ブロックサイズ 1/2 の単位で電圧制御を行い消費電力削減を行なう．L2 キャッシュブロックがすべて圧縮されると，キャッシュの 50% が消費電力削減できることになる．また，従来のキャッシュと同等のヒット率を維持することが可能である．

図 3.2 にあるように，圧縮状態を示す compression bit を設けることにする．L2 キャッシュブロックが圧縮された場合には compression bit に 1 を立てる．圧縮されたキャッシュブロックに対して電圧制御を行い消費電力削減を行なうので，compression bit に 1 が立っている場合は電圧制御を行なっているキャッシュブロックを示すことになる．

3.3 データの書き込み

3.3.1 基本的な書き込み方式

キャッシュの基本的な書き込み方式には，以下の二つがある．

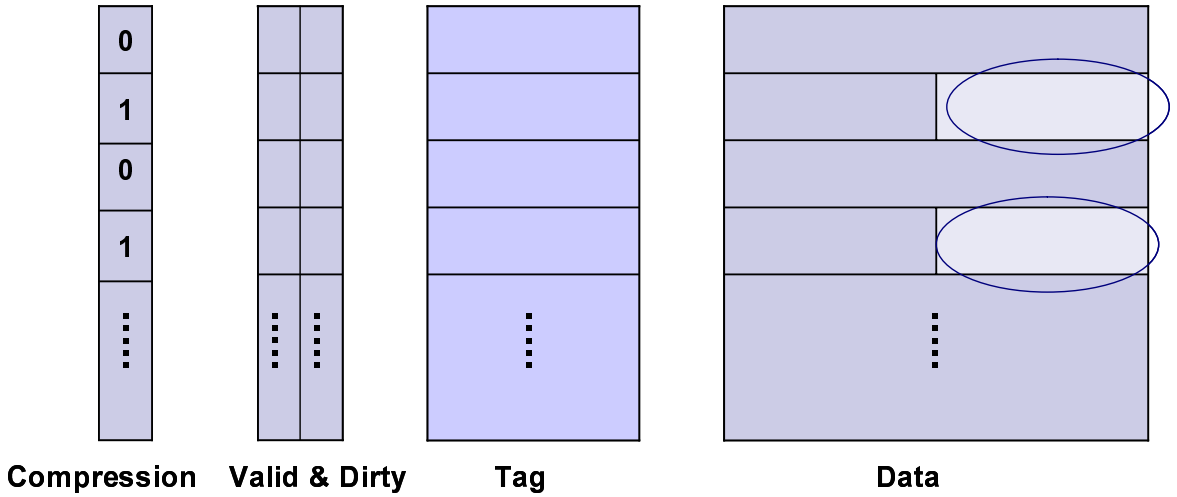

図 3.2: L2 キャッシュ詳細図

- write-through 方式
  キャッシュメモリの書き込みを行う際，同時に対応する下位の記憶階層に書き込みを行う．上位の記憶階層と一貫性を持つが，下位の記憶階層の書き込みを完了を待つ必要がある．

- write-back 方式
  キャッシュメモリに書き込みを行う際，下位の記憶階層には同時に書き込みを行わず, ある時点で上位のメモリの内容を反映させる方式である．書き込み自体の速度は高速となるが，上位の記憶階層の内容の整合性がとれなくなる面がある．キャッシュブロックが追い出される場合，そのブロックが書き換えられていたならば下位の記憶階層にも最新のデータを反映させなければならない．よって書き換えられたことを示すビット dirty bit を設け，キャッシュブロックの追い出しの場合は dirty bit が立っていたならば対応する下位の記憶階層に書き込みを行う．

書き込み方式の例として，L1 キャッシュと L2 キャッシュがオンチップのアーキテクチャの場合，チップの外から見えるデータは最新のデータであることが望ましいので，L1 キャッシュから L2 キャッシュへの書き込みを行う際は write-througt 方式，L2 キャッシュからメモリへの書き込みは write-back 方式が用いられる．

3.3.2 本研究における書き込み方式

本提案方式では，書き込みのデータが非圧縮データで書き込み対象となっているブロックが圧縮状態でキャッシュブロックの 1/2 が電圧制御されていたならば，立ち上げにかかる時間 Setup Time が発生する．また，ある特定のキャッシュブロックへの書き込み動作が

何回も行われているような状況であれば，write-through 方式を用いると Setup Time の影響を大きく受けやすい．write-back 方式であれが一度ですみ，書き換え対象が圧縮状況であれば電圧制御により消費電力を削減し続けることが可能となる．本研究ではデータの書き込み方式として，L1 データキャッシュから L2 キャッシュへの書き込みそして L2 キャッシュからにメモリへの書き込みに write-back 方式を用いる．

第4章　圧縮と復元

データ圧縮に用いる圧縮方法と復元方法について説明する．

4.1　圧縮方法

4.1.1　圧縮パターン

本研究では，FPC(Frequent Pattern Compression)[4] という圧縮アルゴリズムを用いる．圧縮はキャッシュブロック単位が基本となる．その際，キャッシュブロックを 1word(32bit) に分割する．各 word を決められたパターンに当てはめていきデータを圧縮する．

圧縮パターンは表 4.1 に示す通りである．各圧縮パターンを示す prefix と復元の，復元の際に必用な data size 分のデータにより圧縮データを表す．

表 4.1: 圧縮パターン表

prefix	pattern encode	data size
000	Zero Run	3bits(for runs up to 8 zeros)
001	4-bit sign-extended	4 bits
010	One byte sign-extended	8 bits
011	halfword sign-extended	16 bits
100	halfword padded with a zero halfword	The nonzero halfword(16bit)
101	The halfwords, each a byte sign-extended	The two bytes(16bits)
110	word consisting of repeated bytes	8 bits
111	Uncompression word	Original Word(32bit)

８通りの圧縮パターンの詳細は次の通りである．

- Zero Run
  - 1word またはそれ以上の word が連続してゼロである場合に用いる．data size の 3bit により word 数を表し，最大 8word 分連続したゼロであるデータを圧縮できる．

- 4-bit sign-extended
  - 1word データが 4bit 符号拡張のデータとして表現できる場合に用いる.
- One-byte sign-extended
  - 1word データが 1byte 符号拡張のデータとして表現できる場合に用いる.
- halfword sign-extended
  - 1word データが halfword(16bit) 符号拡張のデータとして表現できる場合に用いる.
- halfword padded with a zero word
  - 下位 16bit がゼロであるデータの場合に用いる.
- The halfwords, each a byte sign-extended
  - 上位 16bit，下位 16bit 共に 1byte の符号拡張のデータとして表されるような場合.
- word consisting of repeated bytes
  - 1byte ごとにデータパターンが繰り返されるような場合.
- Uncompression word
  - 上記７通りのパターンにどれにも当てはまらない場合．1word データがそのまま用いられる.

4.1.2 圧縮データ表現

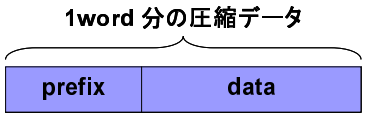

図 4.1: 1word 分の圧縮データ

表 4.1 に示す圧縮パターンにより圧縮されたデータは 1word の場合，図??のように上位ビットに prefix，つづいて data をあわせることにより圧縮データを表現する.

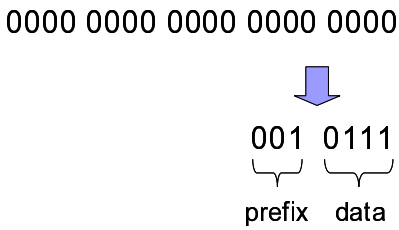

図 4.2: データ圧縮の例 (4-bit sign-extended の場合)

例えば図 4.2 にあるようなデータパターンである場合，この図のデータは 4bit 符号拡張されたデータと見ることができるので表 4.1 より 4-bit sign-extended の prefix は”001”，data size の 4bit のデータ部分は”0111”となる．よって上位に prefix 続いて data を組み合わせて，図 4.2 の例で圧縮データは”0010111”と表される．

圧縮されたキャッシュブロックを構成するには，例えば図 4.3 に示すようにキャッシュブロックが 64byte の場合 16word に分割され 16word 分の prefix が上位ビットに，そして 16word 分の data が続く．全体で 32byte 以下の圧縮データが出来る．

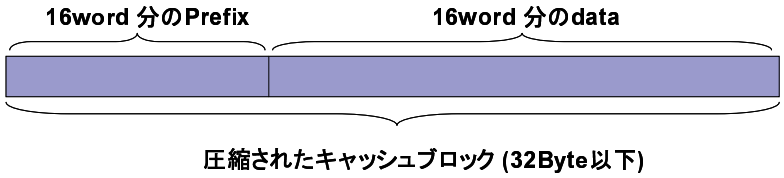

図 4.3: 圧縮されたキャッシュブロックの例

4.2 復元方法

圧縮されたキャッシュブロックを復元するときには，図 4.3 の prefix フィールドを 3 ビットずつ見ていく．prefix より圧縮パターンとデータサイズが決まっているので圧縮キャッシュブロックの data 部分の上位ビットからデータを拾い上げていく．それを word の数だけ繰り返すことにより，圧縮キャッシュブロックはもとのキャッシュブロックに復元できる．

第5章　電力削減法

キャッシュの電力削減に用いるGated-Vddと，それを用いたキャッシュメモリ電力削減法について説明する．

5.1　Gated-Vdd [5]

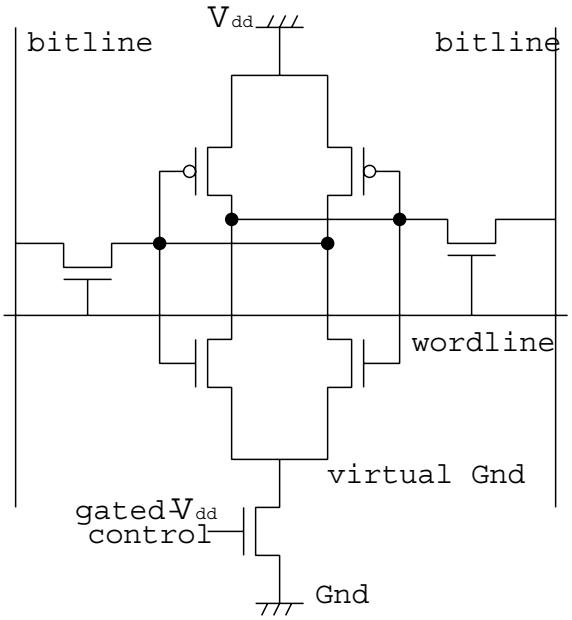

図 5.1: NMOS Gated-Vdd

動作電圧を下げるだけではトランジスタのスイッチングスピードが低下するので，しきい値電圧を下げることにより回避する．しかし，しきい値電圧を低下していくごとにリーク電流の増加を招き消費電力に影響を及ぼす．リーク電流による消費電力を防ぐための手法としてGated-Vddがある．

Gated-Vddは，図5.1のようにSRAMのセルの供給電圧からGNDへリーク電流が流れるパスに特別なトランジスタを設ける(Gated-Vddトランジスタを呼ぶことにする)．使用されるSRAMの部分であればGated-Vddトランジスタをオンとし，使用されない部分はオフとするとしリーク電流を抑える．リーク電流を抑えるのに新たにトランジスタを設

けた代わりに，SRAM の面積の増加が発生する．しかし，Gated-Vdd トランジスタは複数の SRAM セルに対して共有することが可能なので面積の増加は抑えられる [5].

5.2 電圧制御の粒度

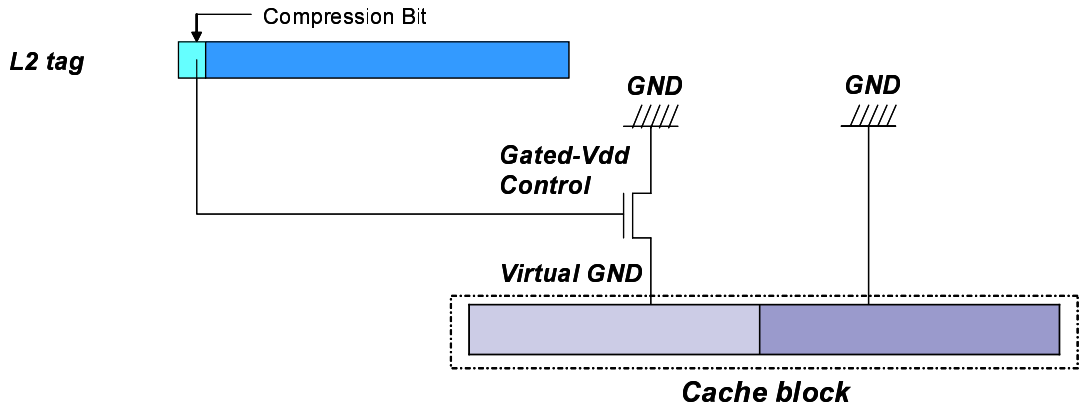

図 5.2: キャッシュブロックへの電圧制御

本研究では Gated-Vdd を用いることで，キャッシュブロックの 1/2 の単位で電圧制御を行い消費電力削減を行なっていく．キャッシュの電力制御の粒度はより細かくできるが，ハードウェア構成の複雑化を回避するためにキャッシュブロックの 1/2 の単位で電圧制御をおこなっている．また，図 5.2 に示すように L2 キャッシュのタグに圧縮状態を示す compression bit を設け，そのビットを Gated-Vdd への入力信号としキャッシュブロックの消費電力を削減していく．

5.3 Setup time の影響

圧縮された形で格納されているキャッシュブロックは，キャッシュブロックの 1/2 が Gated-Vdd によりオフの状態になっている．そのキャッシュブロックに非圧縮データが格納される場合が出てくる．その場合，オフとなっている部分のキャッシュブロックの電圧を通常の状態にもどす必要がある．キャッシュブロックの電圧が元に戻り安定したところで，データを格納することができる．電圧が立ち上がって安定するまでの時間 Setup Time はプロセッサの実行サイクルに重大な影響を及ぼしてしまうので最小限にとどめる必要がある．この Setup Time は従来のプロセッサで設けられている Write Buffer の使用で隠蔽する．Write Buffer は下位層のメモリへの書き込みが完了していなくとも，プロセッサの動作を続行することができる．これにより，Setup Time の影響を最小限に抑えていく．

第6章　評価

6.1　ベンチマークプログラム

表 6.1: SPECint95

ベンチマーク	詳細	入力ファイル
099.go	An internationally ranked go-playing program	ref
124.m88ksim	A chip simulator for the Motorola 88100 microprocessor	ref
129.compress	A in-memory version of the common UNIX utility	ref
130.li	Xlisp interpreter	ref
132.ijpeg	Image compression/decompression on in-memory images	ref
147.vortex	An object oriented database	ref

シミュレーションの対象プログラムに SPECint95 の 099.go，124.m88ksim，129.compress，130.li，132.jpeg，147.vortex の６つのプログラムを用いる．いろいろな種類のアプリケーションにより評価を行うことで本研究の効果を検証するベンチマークの内容とシミュレーションに用いた入力ファイルを表 6.1 に示す．各ベンチマークプログラムは，準備として最初の 10 億命令を実行し，その直後からの 10 億命令分を測定する．

6.2　評価方法

表 6.2: 基本的なシミュレータパラメータ

L1 I-cache&D-cache	16KB，4-way(LRU)，1 cycle latency，64byte cache line
L2 cache	512KB，2-way(LRU)，10 cacle latency，64byte cache line
Memory access latency	100 cycle latency

本研究の提案方式の性能を評価するために，CPU シミュレータを用いて評価を行う．

CPU シミュレータは命令セットとし，MIPS64 Architecture を使用し，C 言語で記述されたプログラムをコンパイルし生成されたバイナリコードを入力とする．シミュレータで用で基本的なパラメータは表 6.2 に示すとおりである．キャッシュブロックサイズは，L1 キャッシュ・L2 キャッシュともに 64byte で，シミュレータは一命令実行を１クロックサイクとする．また，復元にかかるレイテンシを 5 サイクルとする [4].

表 6.3: 1cell あたりのリークエネルギー

Implementation Technique	base low-Vt	Gated-Vdd
Active Leakage Energy	1740	1740
Standby Leakage Energy	N/A	53

本研究では，命令キャッシュとデータキャッシュそして L2 キャッシュがオンチップ上にありライトバッファをもつ一般的なプロセッサを想定しているので，そのプロセッサを評価対象とする．そして，実行サイクル数と L2 キャッシュの静的消費電力を比較し提案方式の検討を行う．L2 キャッシュの静的消費電力は [5] より，SRAM １セルあたりのリークエネルギーの値を用い計算をする．提案手法では NMOS gated-Vdd(dual-Vt, wide , with a charged pump) を，評価対象となる従来のキャッシュは標準的な低い閾値の SRAM を仮定する．表 6.3 仮定したリークエネルギーのパラメータを示す.

L2 キャッシュの静的消費電力を求めるときは次に示す式をもとに算出する

$$\begin{aligned} Leakage\ energy = &\ Active\ fraction \times Active\ LeakageEnergy \times cycle \\ &+ Standby\ fraction \times Standby\ Leakage\ Energy \times cycle \end{aligned} \tag{6.1}$$

ここで Active fraction とはセルの情報を維持している部分つまり gated-Vdd によってオフになっていない SRAM cell のことである．そして，Standby fraction は Gated-Vdd によりオフとなった SRAM cell のことである.

次に，第３章で述べた提案手法と，本研究で想定しているオンチップキャッシュ(命令・データキャッシュ，L2 キャッシュがオンチップでライトバッファを持つ) とを比較し Write Buffer と Setup Time の関係を中心に比較する．シミュレーションでは，Write Buffer のエントリ数を 16，32，64 entry とする．Setup Time はキャッシュメモリの電圧の状態により変化する．例えば Drowsy Cache のように完全にオフ状態ではなく，途中状態から電圧を安定状態に持っていく手法もある．今回はキャッシュメモリの電圧がどの程度でも対応可能とするために，Setup Time を 500，1000，5000，10000，50000 cycle とし比較する.

Write Buffer がどの程度 Setup Time に対応できているか L2 キャッシュの静的消費エネルギーと実行サイクル数が従来のキャッシュに比べてどの程度となっているのか検証する.

6.3 結果

SPEC95int の 099.go，124.m88ksim，129.compress，130.li，132.ijpeg ,147.vortex ベンチマークプログラムによる結果を示す．表6.3 に各ベンチマークプログラムの L2 キャッシュのセル稼動率，図 6.1〜図 6.12 と表 6.3〜表 6.3 に Normalaized cycle time，Normalaized leakage energy，Buffer stall ratio をまとめた．Normalaized cycle time，Normalaized leakage energy，Buffer stall ratio は以下のことを意味する．

- Normalaized cycle time

  (提案方式を用いたプロセッサの実行サイクル数)/(比較対象のプロセッサの実行サイクル数)

  従来のプロセッサの実行サイクル数を基準とした場合，提案方式を用いた際の実行サイクル数の割合を示すことにより提案方式が実行サイクルに与える影響をみる．

- Buffer stall ratio

  (提案方式で発生した $Write\ buffer\ stall$ サイクル数)/(提案方式の実行サイクル数)

  提案方式の実行サイクル数に対して，Write buffer stall の比率を示す．

- Normalaizad lekage energy

  (提案方式の $L2$ キャッシュの $leakage\ energy$)/(比較対象の $L2$ キャッシュの $leakage\ energy$)

  データ圧縮の効果が L2 キャッシュの静的消費電力量にどの程度影響しているかみる．

  また，各ベンチマークプログラムの圧縮の傾向を表 6.13〜表 6.18 に示す．横軸を (compressor により圧縮されたデータサイズ)/(ブロックサイズ) とし，圧縮データがブロックサイズの何倍になるかを示している．縦軸は実行中に圧縮された回数を示す．

099.go	124.m88ksim	129.compress	130.li	132.ijpeg	147.vortex	平均
0.831	0.852	0.501	0.948	0.902	0.731	0.794

表 6.4: 各ベンチマークプログラムにおける L2 キャッシュのセル稼動率

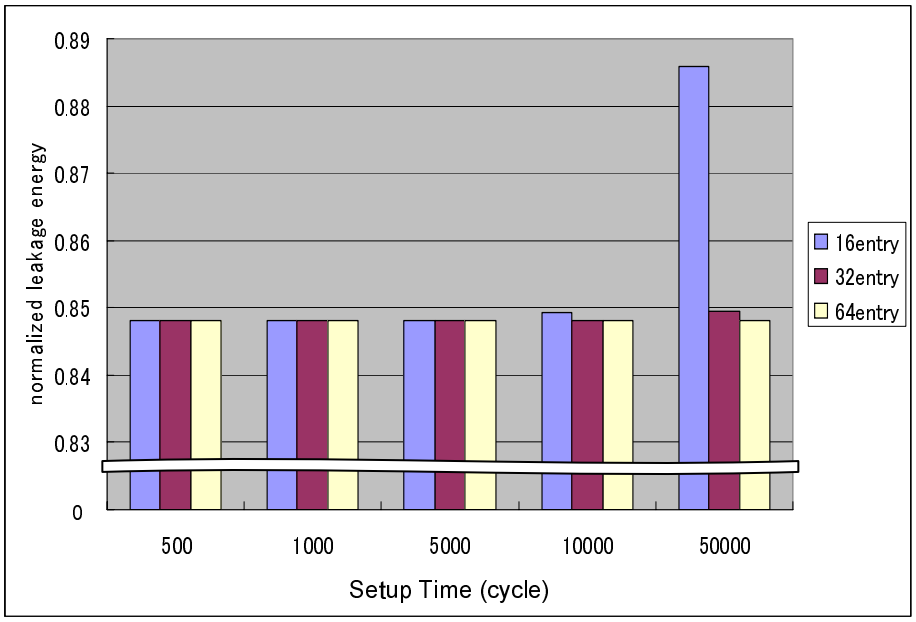

図 6.1: 099.go normalized leakage energy

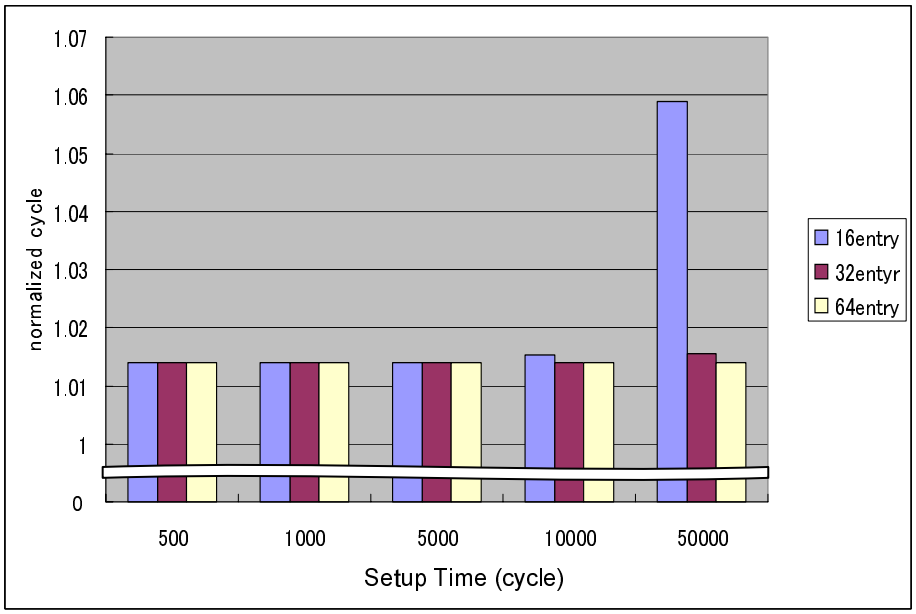

図 6.2: 099.go normalized cycle

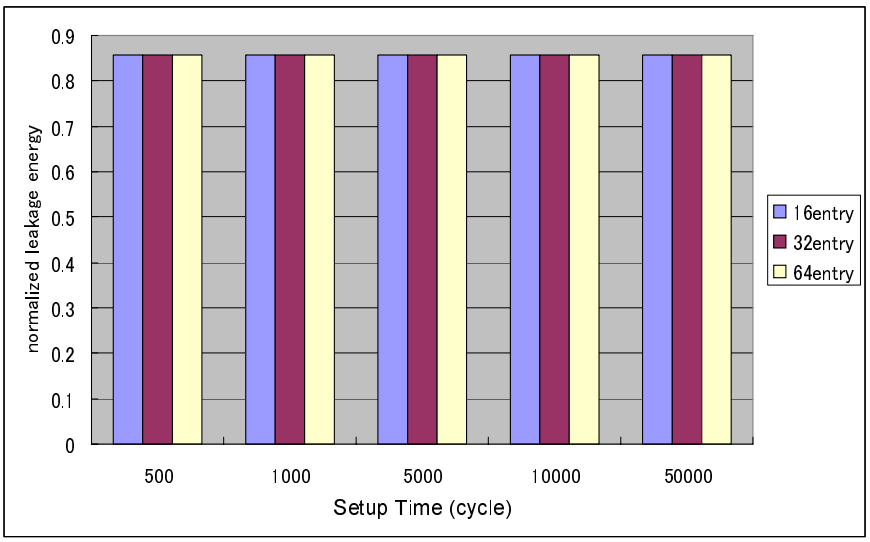

図 6.3: 124.m88ksim normalized leakage energy

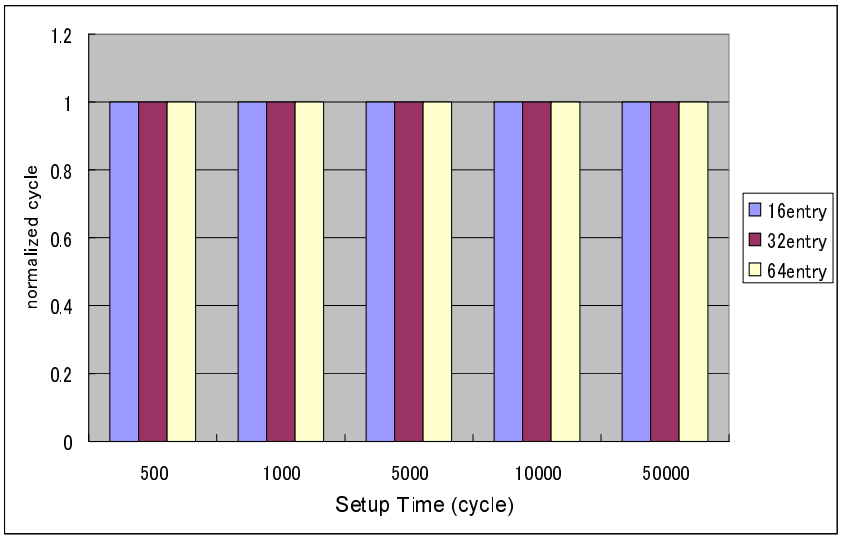

図 6.4: 124.m88ksim normalized cycle

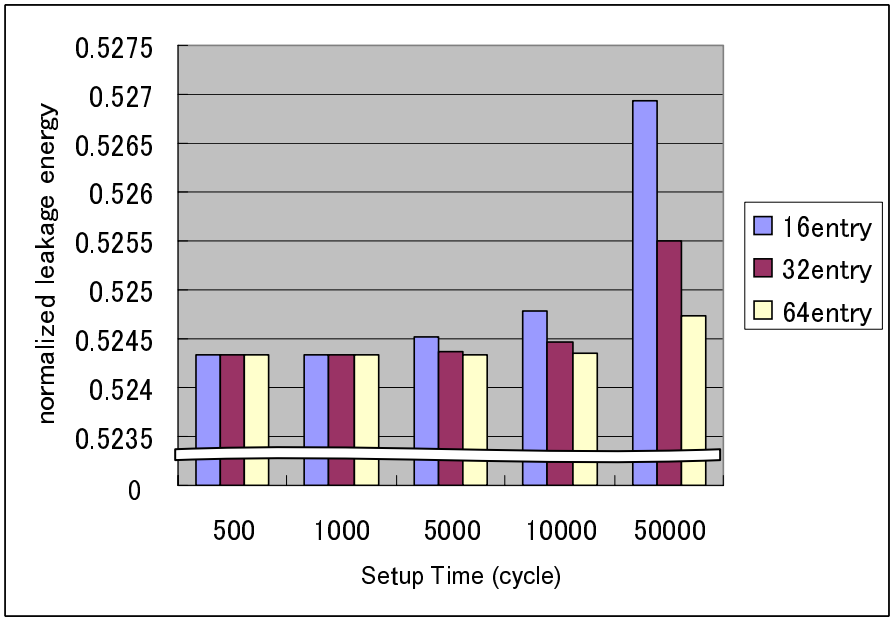

図 6.5: 129.compress normalized leakage energy

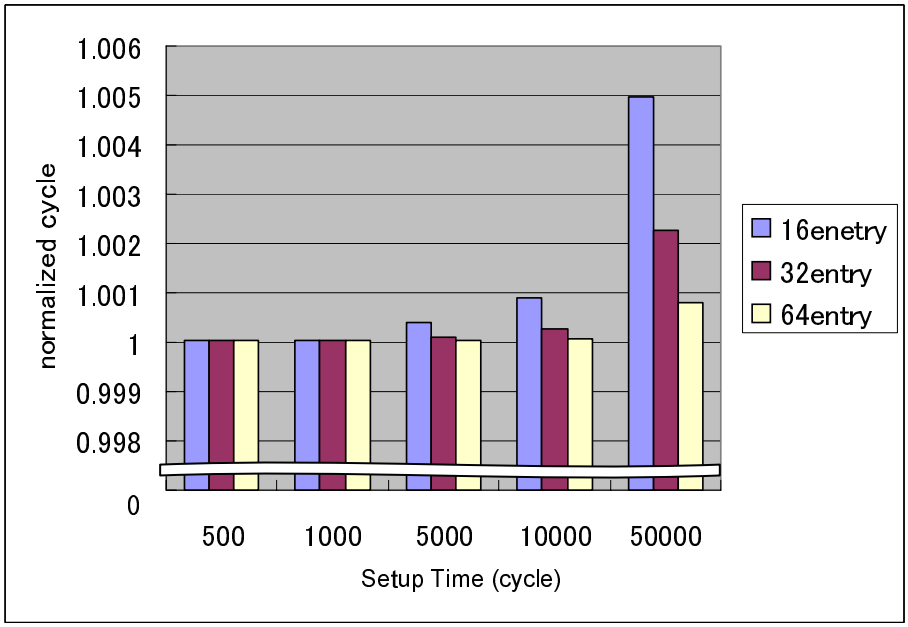

図 6.6: 129.compress normalized cycle

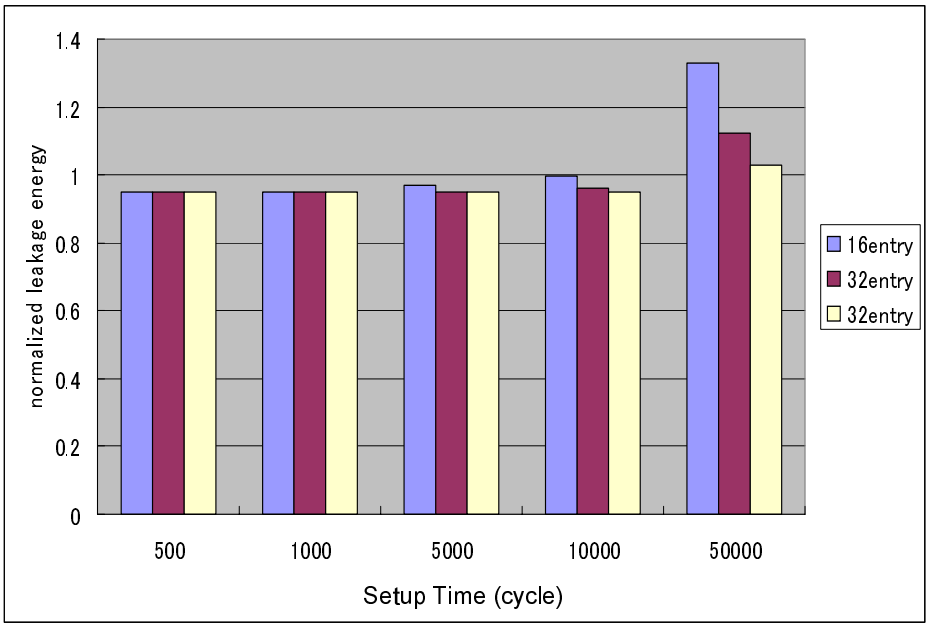

図 6.7: 130.li normalized leakage energy

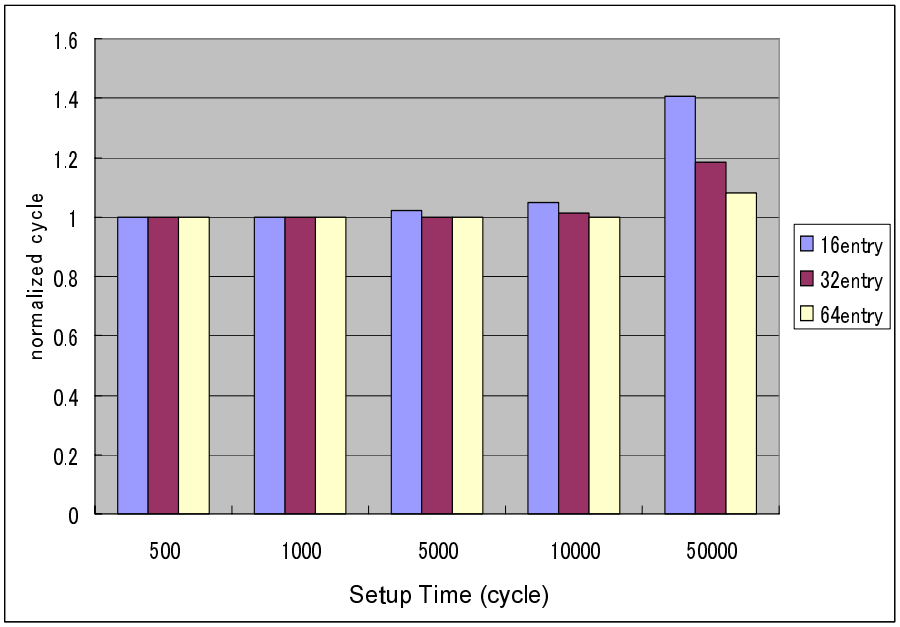

図 6.8: 130.li normalized cycle

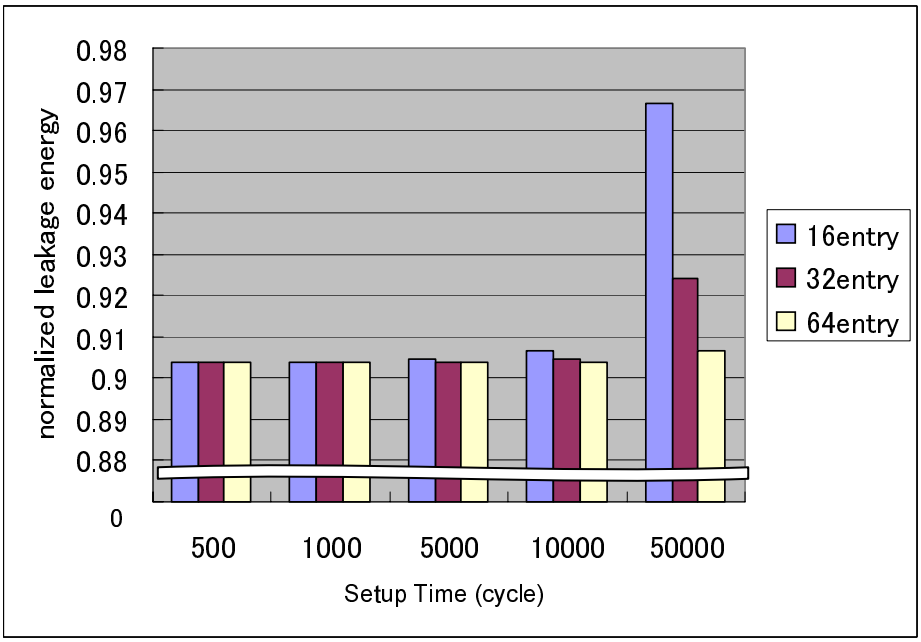

図 6.9: 132.ijpeg normalized leakage energy

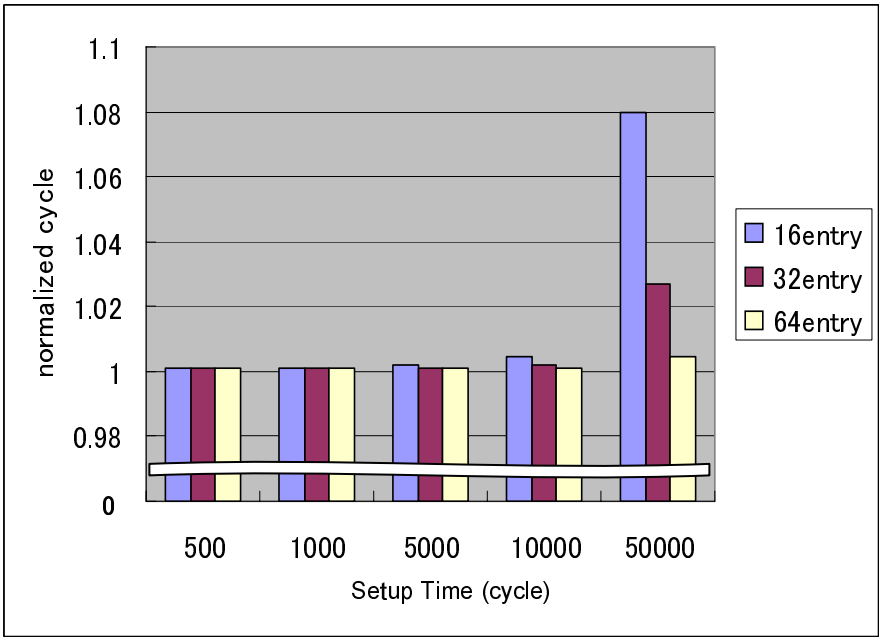

図 6.10: 132.ijpeg normalized cycle

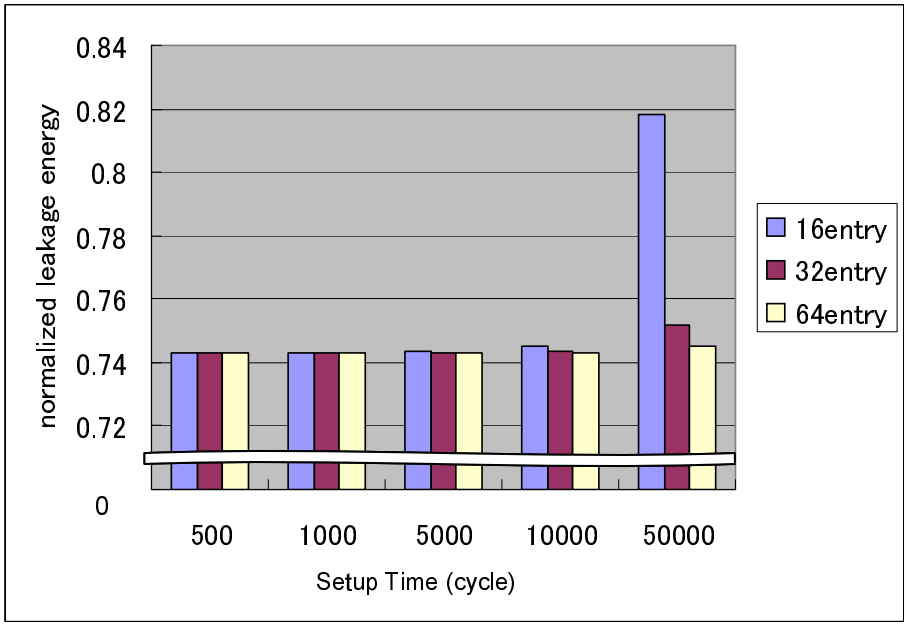

図 6.11: 147.vortex normalized leakage energy

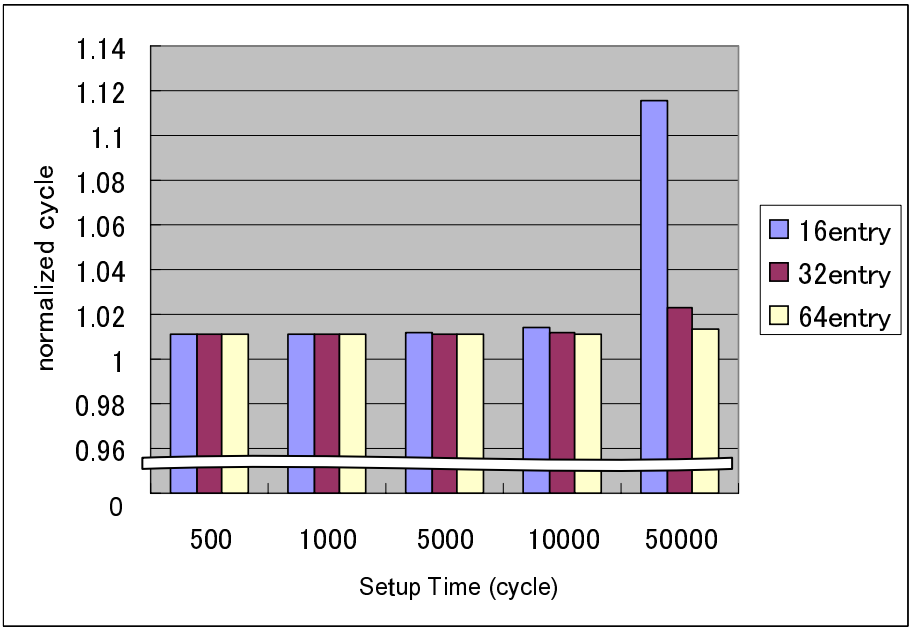

図 6.12: 147.vortex normalized cycle

Setup cycle	500	1000	5000	10000	50000
16entry	0	0	0.000726	0.00283	0.093597
32entry	0	0	0.000019	0.00054	0.011741
64entry	0	0	0	0.000008	0.002419

表 6.5: 099.go buffer stall ratio

Setup cycle	500	1000	5000	10000	50000
16entry	0	0	0	0	0
32entry	0	0	0	0	0
64entry	0	0	0	0	0

表 6.6: 124.m88ksim buffer stall ratio

Setup cycle	500	1000	5000	10000	50000
16entry	0	0	0.000367	0.000875	0.004926
32entry	0	0	0.000082	0.000256	0.002225
64entry	0	0	0	0.000054	0.000774

表 6.7: 129.compress buffer stall ratio

Setup cycle	500	1000	5000	10000	50000
16entry	0	0	0.018315	0.047156	0.287261
32entry	0	0	0	0.011803	0.154173
64entry	0	0	0	0	0.074735

表 6.8: 130.li buffer stall ratio

Setup cycle	500	1000	5000	10000	50000
16entry	0	0	0.00111	0.0035	0.072724
32entry	0	0	0.000119	0.001081	0.025082
64entry	0	0	0	0.000098	0.00369

表 6.9: 132.ijpeg buffer stall ratio

Setup cycle	500	1000	5000	10000	50000
16entry	0	0	0.000726	0.00283	0.093597
32entry	0	0	0.000019	0.00054	0.011741
64entry	0	0	0	0.000008	0.002419

表 6.10: 147.vortex buffer stall ratio

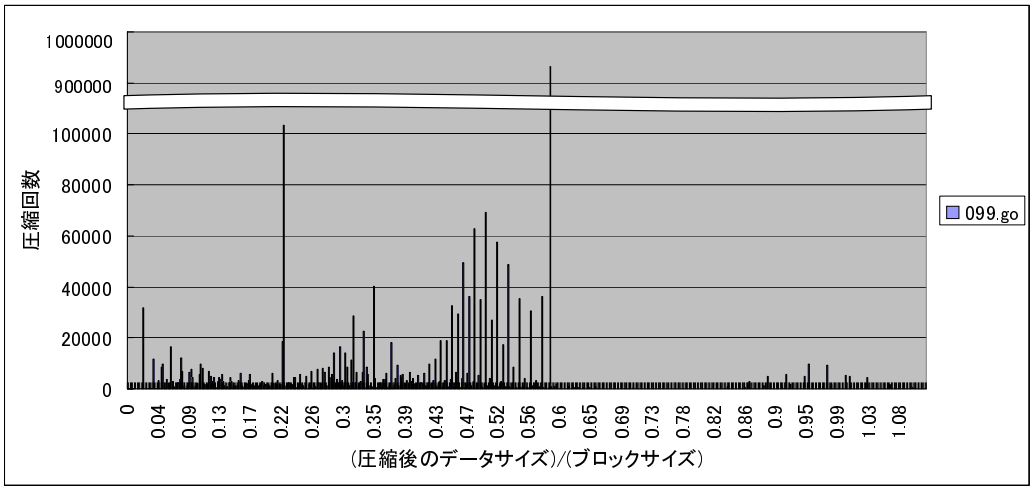

図 6.13: 099.go の圧縮傾向

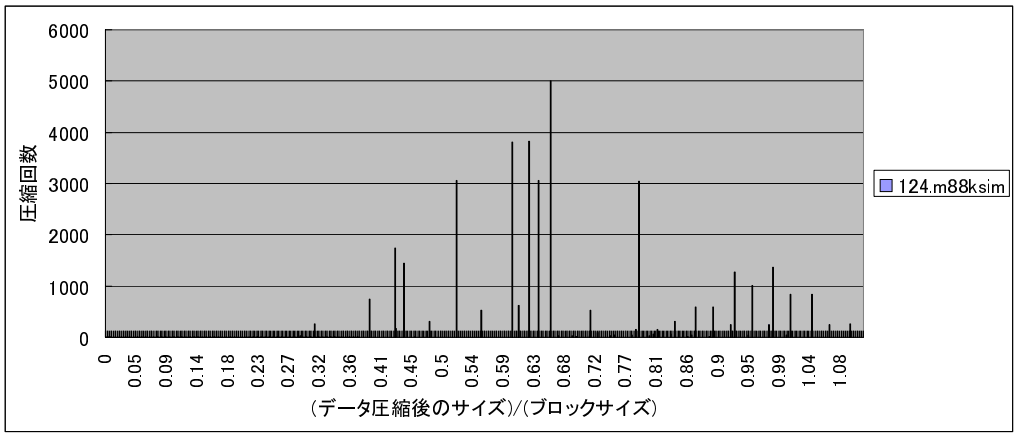

図 6.14: 124.m88ksim の圧縮傾向

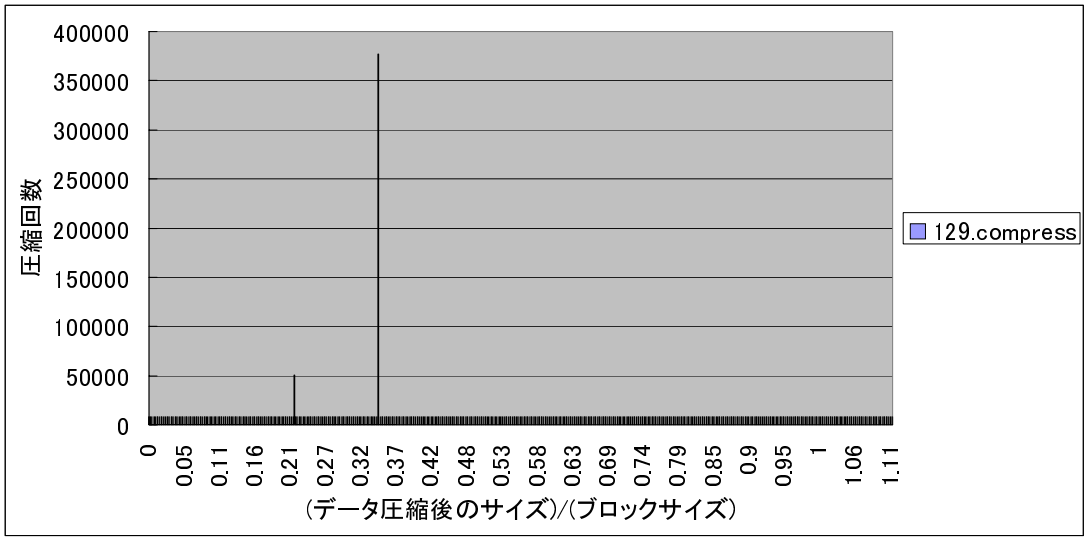

図 6.15: 129.compress の圧縮傾向

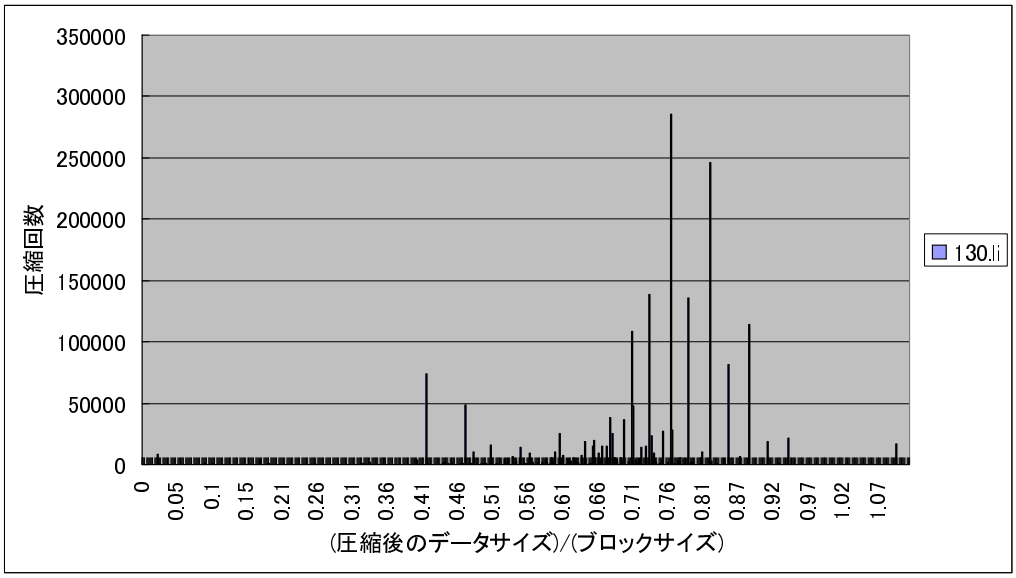

図 6.16: 130.li の圧縮傾向

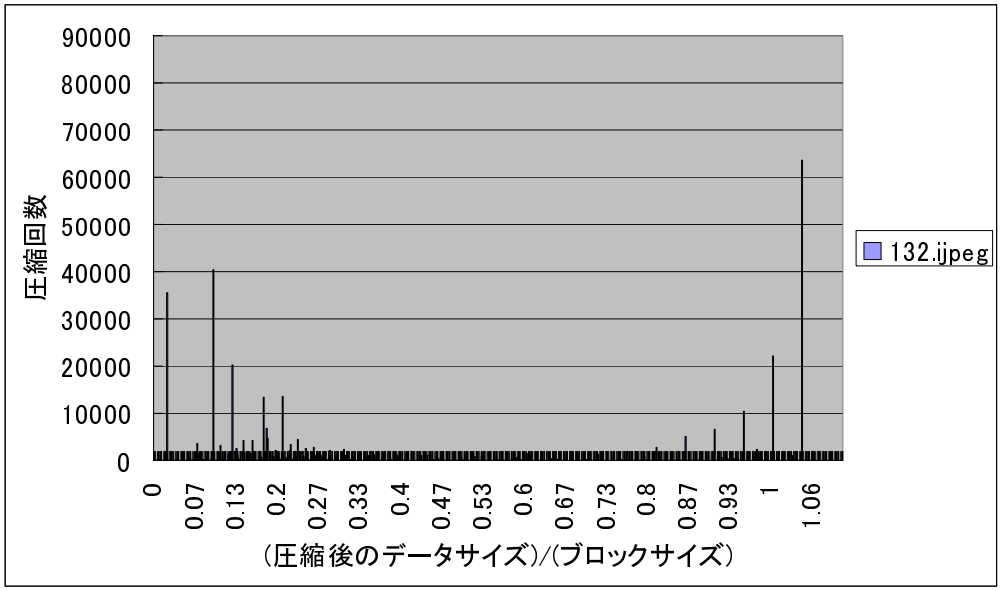

図 6.17: 132.ijpeg の圧縮傾向

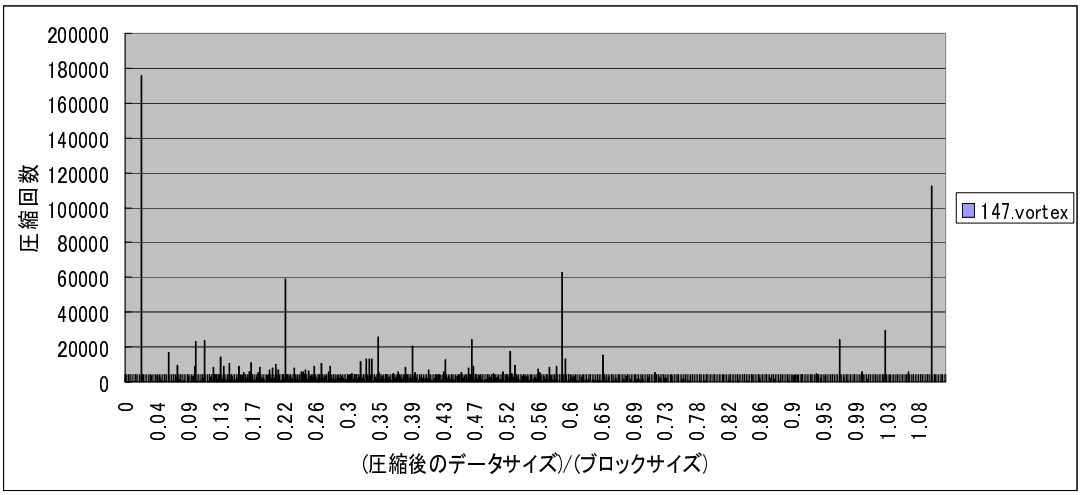

図 6.18: 147.vortex の圧縮傾向

6.4 考察

6.4.1 圧縮の効果

表 6.3 より，一番圧縮の効果がでたベンチマークプログラムは，もっともセル稼働率が低い 129.compress で，L2 キャッシュブロックほぼすべてが圧縮対象となる状態となった．反対に，一番圧縮効果がみられなかったのは 130.li で，セル稼働率は約 95%とほぼすべてのキャッシュブロックが稼動している状態となった．SPEC95int ベンチマークプログラム６つの L2 キャッシュセル稼働率の平均は約 79%となっておりキャッシュブロックの約 4 割が圧縮対象のブロックとなっている．L2 キャッシュの静的消費電力は各ベンチマークプログラムもほぼセル稼働率に対応しており，セル稼働率が低ければ低いほど，消費電力削減の効果は大きい．

復元の実行サイクルによる消費電力への影響は，各ベンチマークプログラムの Buffer stall ratio(表 6.3〜表 6.3) が 0 である Setup cycle の Normalized cycle time と Normalized leakage energy をみる．すべてのベンチマークプログラムにおいて，2%未満であり特に 124.m88ksim，129.compress，130.li，132.ijpeg は 1%にも満たない．６つのベンチマークプログラムで実行サイクル数に対して復元の影響が出ていた 099.go で，Normalized cycle は 1.013 であった．キャッシュの圧縮率は約 83.1%で Normalized leakage energy は約 84.8%であった．今回用いたベンチマークアプリケーションでは復元による影響は少ない．

6.4.2 Write Buffer の効果

Setup Time は実行サイクルに影響を大きく及ぼし，消費電力の増大に繋がる．本件急では Setup Time の影響を Write buffer を使用で隠蔽する．Setup Time が 500，1000cycle の場合，すべてのベンチマークプログラムにおいて Buffer stall がなく Setup Time の影響が隠蔽された．Setup Time が 5000cycle 以上になると，Buffer stall が発生してくる．特に 130.li は，Write buffer が 16entry で Setup Time が 50000cycle になると buffer stall が実行サイクルの約 29%となり消費電力，実行サイクル共に多大な影響を受けている．他に 099.go，132ijpeg，147.vortexSetup Time が大きくなるほど消費電力と実行サイクルに大きく影響を及ぼしている．Write buffer が 32entry，64entry と Write buffer の entry 数を増加することで Setup Time の影響を減らすことができ，消費電力と実行サイクル数を抑えられる.．一方で，124.m88ksim と 129.compress は Setup Time の影響は Write buffer が 16entry でも十分無視できる．124.m88ksim は，すべての Setup Time において Normalized leakage energy と Normalized cycle の値は変化しなかった．

6.4.3　圧縮サイズと電圧制御の粒度

表6.13〜表6.18 に各ベンチマークプログラムの圧縮傾向を示している．099.go と 147.vortex の場合，ほとんどの圧縮ブロックがキャッシュブロックサイズの 0.6 倍以下となっている．099.go は 147.vortex に比べキャッシュブロックの 0.5 倍〜0.6 倍のとなる圧縮データ数が多いために，セル稼働率は高くなっている．124.m88ksim と 130.li はキャッシュブロックの 0.5 倍以上となる圧縮ブロック数が多いのでセル稼働率は高いものとなっている．129.compress はほとんどの圧縮ブロックがキャッシュブロックの 0.5 倍以下であったために提案方式にうまく当てはまった．132.ijpeg は圧縮アルゴリズが効果的なデータと全く効果的ではないデータにはっきり分かれた．

どのベンチマークプログラムも，電圧制御の粒度を細かくすることにより電圧制御対象となるキャッシュブロックの増え消費電力削減の効果がより出る可能性がある．

第7章　まとめ

7.1　まとめ

本論文では，プロセッサの大部分を占めるキャッシュメモリに注目し，データ圧縮と電力制御を用いてキャッシュメモリの低消費電力化を行う手法を提案した．提案手法において，Setup Time 影響を削減のため Write buffer を使用し実行サイクル数と消費電力への影響を減らすことを狙った．

SPECint95(099.go, 124.m88ksim, 129.compress, 132.ijpeg, 147.vortex) 提案手法が Write buffer が Setup Time にどの程度対応できるか評価を行った．評価の結果，６つのベンチマークプログラムにおいて圧縮率と L2 キャッシュの消費電力削減率はほぼ等しくなった．また Setup Time と Write buffer の関係を見てみると entry 数が 16 のとき隠せなかった Setup Time の影響が 32，64entry の場合では十分隠せた．Write buffer のサイズをの観点から見ると，32entry が望ましい．

7.2　今後の課題

今後の課題として以下の点を挙げる．

- compressor と decompressor のハードウェア量
- 本提案機構をのせたプロセッサ全体の消費電力

現在のところ，L2 キャッシュの静的消費電力の割合とプロセッサの実行サイクルを評価している．電力消費量はキャッシュメモリがプロセッサの大部分を占めているが提案機構を載せたプロセッサ自体のの電力消費量の評価が必要である．

参考文献

[1] International Technology Roadmap for Semiconductors. Semiconductor Industry Association,2002

[2] Afzal Malik, Bill Moyer, Dan Cermak, A Low Power Unified Cache Architecture Providing Power and Performance Flexibility, Int Symp. on Low Power Electroinics and Design, 2000

[3] Krisztian Flautner, Nam Sung Kim, Steve Martin, David Blaauw, Trevor Mudge, Drowsy Cache: Simple Techniques for Reducing Leakage Power, Proc. of 29th Int. Symp. on Computer Architecture, 2002

[4] Alaa R. Alameldeen and David A. Wood, Freauent Pattern Compression: A Significance-Based Compression Scheme for L2 Caches Technical Report 1500, Computer Sciences Dept., UW-Madison, 2004

[5] Michael Powell, Se-Hyun Yang, Babak Falsafi, Kaushik Roy, and T.N. Vijaykumar, Gated-Vdd: A Circuit Techniqus to Reduce Leakage in Deep-Submicron Cache Memories ISLPED 2000

[6] Se-Hyun Yang, Michael D. Powell, Babak Falsafi, Kaushik Roy, and T.N. Vijaykumar, An Integerted Circuit/Architecture Approach to Reducing Leakage in Deep-Submicron High-Performance I-Caches HPCA, 2001

[7] Stefanos Kaxiras, Zhigang Hu, Margret Martonosi, Cache Decay: Exploiting Generational Behavior to Reduce Cache Leakage Power ISCA, 2001