卒業論文

題目

ライン・タイル両アクセス対応新キャッシュ置換アルゴリズムの提案

指導教員

近藤 利夫

2017年度年

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

藤原 晃紫 (414853)

内容梗概

近年，コンピュータの性能が飛躍的に向上しているにも関わらず，2 次元画像処理や科学技術計算に代表される 2 次元データ処理の高速化の要求は相変わらず強い．このような状況が続いている原因の一つは，従来の汎用的なコンピュータでは膨大な 2 次元データを効率的に処理できないことにある．具体的には，並列化により実現されているプロセッサの演算速度向上にメモリのアクセス速度が追いつけず，性能向上のボトルネックになっているためである．このメモリアクセスネックの主因は，従来のキャッシュメモリが 1 次元データアクセス優先の構成であり，2 次元データのアクセスにおいて参照の局所性が活かされず，データアクセスが非効率になることにある．この問題を解決するために，当研究室では，2 次元データのアクセス速度を向上させるための行・列両方向への並列アクセスを可能にするライン・タイル両アクセス対応新構成キャッシュメモリが提案された．このキャッシュメモリは，従来のラスタラインアクセスと 2 次元データアクセスに特化した階層的なタイル形式によるタイルアクセスの 2 つを切り替えることで，従来の 1 次元データ処理の性能を維持したまま，2 次元データの処理性能を向上できる．先行研究のキャッシュメモリはセットアソシアティブ方式を採用しており，データ転送を行う際に，キャッシュメモリをいくつかに分けたウェイの一つに格納する．更に，先行研究でタグメモリのハードウェア規模増大を防ぐために，ラインアクセスにおいて整列されたタイルセットを同一ウェイに格納する RATS(Replaces multiple tiles with an Aligned The Set) 法が提案された．しかし，従来のキャッシュ置換アルゴリズムではラインアクセスで整列されたタイルセットが同一ウェイに格納されるとは限らない．そのため，行方向優先アクセスであるラインアクセスにおいて，行方向へのデータの局所性が不十分となり性能を低下させている可能性がある．本研究では，整列されたタイルセットを同一ウェイに格納することと性能向上を目的として，ライン・タイル両アクセスで使用できるキャッシュ置換アルゴリズムとして，連続アクセスにおいてタグが同一かつインデックスが連続の場合に同一ウェイに格納する手法を提案し，Simplescalar に組み込むことで性能を評価した．その結果，ラインアクセス時に行方向の連続データが同一ウェイに格納されない問題点は解決できたものの，総実行サイクル数が先行研究と比較して平均 0.005%落ち，性能の改善は確認できな

かった.

Abstract

Despite the many breakthroughs in computer performance in recent years, there is still a strong need to speed up two-dimensional data e.g. two-dimensional image processing and scientific computing. One of the reasons why the situation continues is that efficiently current general-purpose computers can't process enormous amounts of two-dimensional data. Specifically, the speed of memory access can't keep up with the processing speed improved by parallelization. And it is a bottleneck in performance improvement. The main reason of this bottleneck of memory access is that the current cache memories are designed for one-dimensional data access , so the locality of reference is efficiately used for two-dimensional data, that becomes the data access becomes inefficient. In order to solve this problem, our laboratory proposed A Cache Memory with Line and Tile Data Accessibility that enables parallel access in both rows and columns to improve the access speed of 2D data. The cache memory can switch between the conventional raster line access and the tile access by the hierarchical tile format specialized for 2D data access. Thereby the cache memory can maintain the performance of the conventional one-dimensional data processing, and improve data processing performance. The cache memory of the previous research adopts the set associative method and stores a data one of the cache way when transferring data. Furthermore, in previous research, in order to prevent the hardware of the tag memory , a RATS (Replaces multiple tiles with an Aligned The Set) method which stores tilesets aligned in line access in the same cache way was proposed. However, with the conventional cache replacement algorithm, tile sets aligned by line access are not always stored in the same way. Therefore, in the line access which is the row direction priority access, there is a possibility that the locality of the data in the row direction becomes insufficient and the performance is degraded. A purpose of this study is storing a tileset in the same cache way and improving performance. Therfore I proposed a method to store in same cache way when the tags are the same in continuous access and the index is continuous and evaluated its performance by incorporating it

into Simplescalar. As a result, although the problem of continuous data in the row direction not being stored in the same way at the time of line access could be solved, the total number of execution cycles fell an average of 0.005 % compared with the previous research, and improvement in performance could not be confirmed .

目 次

図 目 次

表 目 次

1　はじめに

1.1　研究背景

近年のコンピュータの 2 次元データ処理の高速化の要求に答えるための，ライン・タイル両アクセス対応新構成キャッシュメモリが当研究室で提案されている [1]．このキャッシュメモリでは，ハードウェア面積増大を防ぐための，ラインアクセスにおいて整列されたタイルセットを同一ウェイに格納する RATS 法が採用された．しかし，従来のキャッシュ置換アルゴリズムではラインアクセスで整列されたタイルセットが同一ウェイに格納されるとは限らない．そのため，行方向優先アクセスであるラインアクセスにおいて，行方向へのデータの局所性が不十分となり性能を低下させている可能性がある．

1.2　研究目的

先行研究は, ハードウェアのシミュレータである Simplescalar を改造することにより性能評価を行なっている．本研究では，整列されたタイルセットを同一ウェイに格納することと性能向上を目的として，ライン･タイル両アクセスで使用できるキャッシュ置換アルゴリズムを提案し，Simplescalar に組み込むことで性能評価を行う．

2　先行研究

当研究室では，2次元データのアクセス速度を向上させるための行・列両方向への並列アクセスを可能にするライン・タイル両アクセス対応新構成キャッシュメモリの開発が進められている．このキャッシュメモリは3つの基本的なアイデアに基づいている．一つ目は，2次元的な参照局所性の利用に適した空間アクセス法であるZ-orderに基づいた階層的なタイル形式での主記憶のアドレスの割り当て．二つ目は，キャッシュメモリのマルチバンク化，そしてバンクへのスキューアレイ形式でのデータ格納．三つ目は，階層的なタイル形式を外部に隠蔽するための，プロセッサとキャッシュメモリの間のアドレス変換ユニットとなっている．

2.1　空間アクセス法 (Z-order)

2次元の参照局所性のあるキャッシュアクセスに対する性能向上の手段として，空間アクセス法が研究されている．空間アクセス法とは，2次元データをいくつかの子空間に分け，1次元の配列データにマッピングを行うことで，規則的および不規則的なキャッシュアクセスにデータの2次元的な局所性を生かすようにする手法である．ここでは空間アクセス法の中でも，本研究に関係のあるZ-orderについて説明する．図2.1に通常の

アドレス割り当て，図 2.2 に Z-order に基づいたアドレス割り当てを示す．通常のキャッシュメモリでは図 2.1 のようにラスタ順でデータを格納しているため，行方向への局所性は高いが，列方向への局所性は低い．Z-order では図 2.2 のように N × N の正方形の順でデータを格納している．図 2.2 のような 8 × 8 の 2 次元配列データの場合，まずそれを 4 つの 4 × 4 のサブブロック単位に分け, 更にそれを 4 つの 2 × 2 のサブサブブロックに分ける．ここで,8 × 8 の 2 次元配列データは 2 × 2 のサブサブブロックのラスタ走査順の並びからなり,4 × 4 のサブサブブロックデータと 8 × 8 のサブブロックが, 更にラスタ走査順の並びである．各構成のアドレス割り当てがサブサブブロック内，サブサブブロック間，サブブロック間のそれぞれでラスタ走査順に行われる．2 次元の参照局所性のあるアクセスに対し，Z-order の走査順は任意サイズのブロック内がラスタ走査順の並びとなっているため，連続するアクセスの割合が高くなり，ブロック転送が効率的に行われる．しかし，隣接するブロック間では連続するアドレスの距離が遠いため，隣接要素間のデータの局所性の利用は非効率という問題もある．

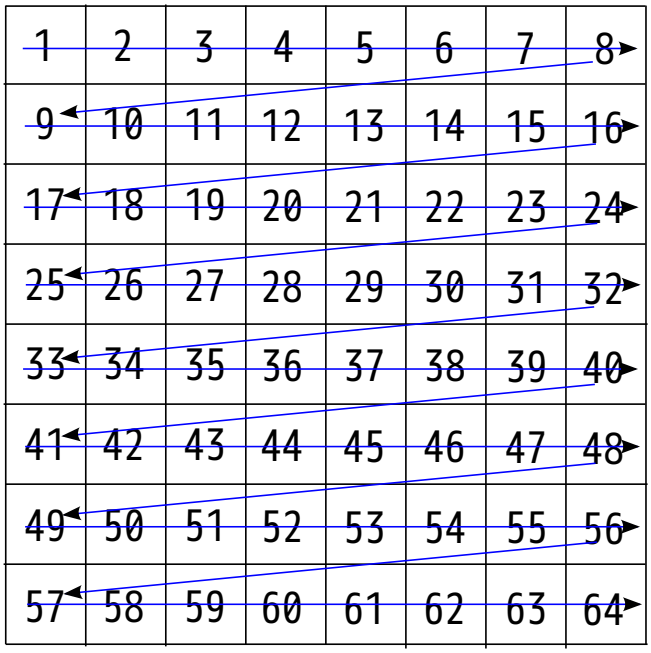

図 2.1: ラスタ形式でのアドレス割り当て

2.2 階層的なタイル形式でのアドレス割り当て

2 次元データアクセスにおいて，従来のラスタライン形式でのデータアクセスはラスタ走査方向に連続するアクセスの局所性が短い場合，ブロックサイズを拡大しても，競合性ミスの増大や不要なデータ転送による容量性ミスの増加につながる．これらの問題を解決するために主記憶から階層的なタイル形式としてアクセスされるようにする．アドレス変換機構により，プロセッサから渡された従来のラスタ順のアドレスをタイル形式のアドレスに変換する．アドレス変換機構を設ける理由は，階

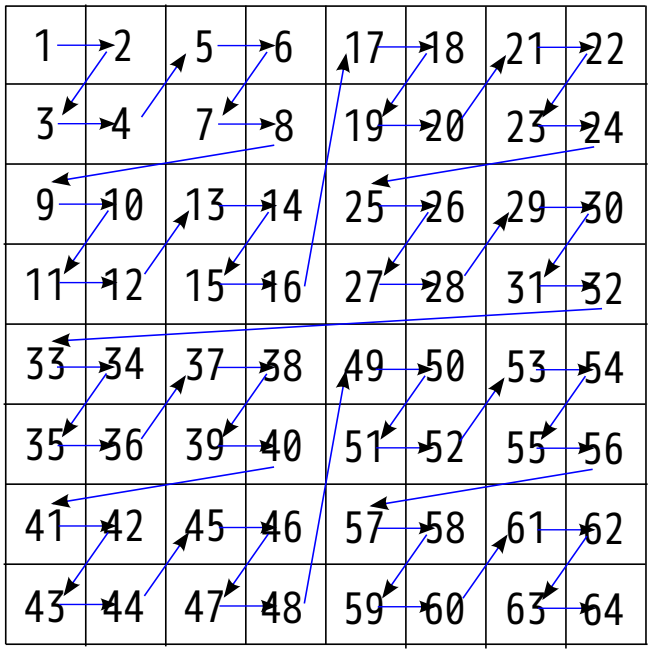

図 2.2: Z-order でのアドレス割り当て

層的なタイル形式で割り当てられた 2 次元配列データ全体を，一般的なラスタ走査順の 2 次元配列データとしてアクセスできるようにすることで，従来のアドレス計算が複雑化するのを防ぐためである．変換されるアドレス割り当てを図 2.3 に示す．このアドレス割り当ては Z-order を改良した 3 レベルの階層的なタイル形式となっている．最下層のレベルは，8 バイトからなるサブラインデータが縦方向に四つ並ぶ 32Byte のデータで，スモールタイルと呼ぶ．このスモールタイルが, 一般的なキャッシュメモリにおけるブロックサイズとなっている．スモールタイルとブロックサイズを同じにすることで，データアクセスの行方向とラスタ走査方向

に垂直な列方向にデータの局所性を持つことができる．その一つ上のレベルは，キャッシュサイズと同じ大きさに設定された，スモールタイルのラスタ走査順の並びから構成されるデータの集まりで，ラージタイルと呼ばれる．ラージタイルとキャッシュサイズを同じにすることで，各ラージタイル内はアドレスが連続して割り当てられる．更に，ラージタイル内のデータが複数の仮想記憶のページに分散しないため，2次元データアクセス時に，競合性ミスや主記憶・2次記憶間のページスワップが頻発する問題が解決できる．そして，最後のレベルが主記憶全体となる．アクセスに2次元の局所性がある場合，階層的なタイル形式はスモールタイル内，ラージタイル内のそれぞれでラスタ走査順の並びであるため，置き換え対象のブロックデータのアドレスが連続する割合が高くなるので，行列両方向への2次元的な局所性の範囲を越える不要なデータ転送の割合を低減できる．

2.3　マルチバンク化とスキュードアレイ形式

スモールタイルをキャッシュラインとしてアクセスされるようにすると，2次元データアクセスに対する不要なデータの転送や格納を低減できる効果があるが，従来のラスタ走査順の処理方法との整合性が低い欠点

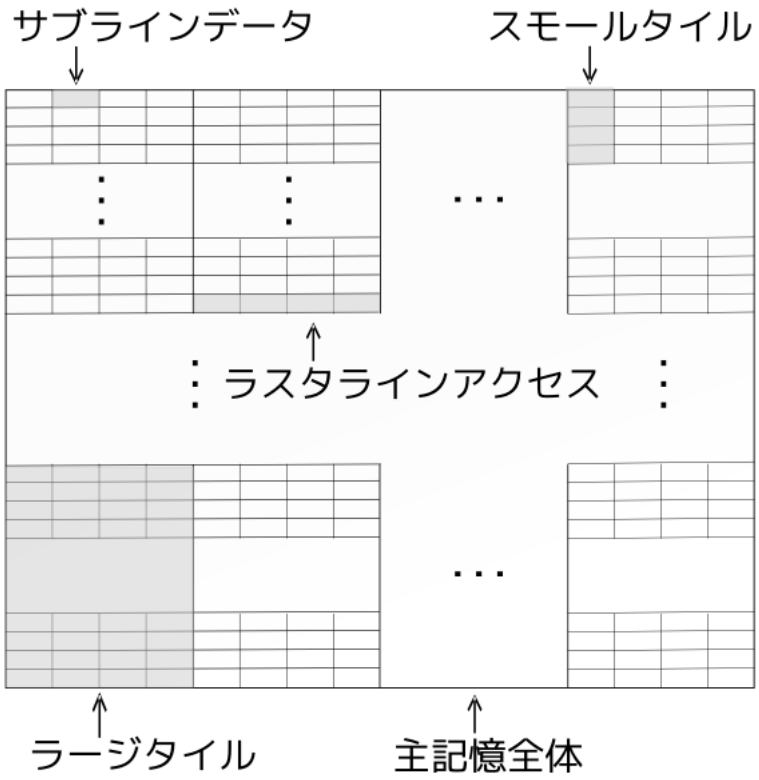

図 2.3: 階層的なタイル形式でのアドレス割り当て

がある．また，行方向のデータの並列アクセス数が減ってしまうため，ラスタ形式データを前提としたSIMDプロセッサの並列演算機能が活かせない．この問題を解決するため，従来のラスタ形式のアクセスと2次元的な参照局所性を活かすスモールタイル形式のアクセスを両立するマルチバンク化とスキューアレイ形式が採用されている．マルチバンク化とは，キャッシュメモリを複数のバンクに分けてデータを格納することで，SIMD演算機構により複数のデータを同時に転送することを可能にする

手法である．マルチバンク化は，主に階層的なキャッシュメモリにおいて採用される手法である．この場合，行方向への連続データはキャッシュメモリの異なるバンクに格納されているため，SIMD 演算機構により同時にデータを転送できる．しかし，列方向への連続データはキャッシュメモリの同一のバンクに入るため，SIMD 演算機構を用いても同時にデータを転送することはできない．この問題を解決するために，キャッシュメモリのバンクに格納する際に，列方向のデータをバレルシフタにより一つずつずらすスキューアレイ形式が提案された．このスキューアレイ形式により，列方向への連続データもキャッシュメモリの異なるバンクに格納されるため，SIMD 演算機構により同時にデータを転送することができる．

2.4 RATS(Replaces multiple tiles with an Aligned The Set) 法

従来のキャッシュメモリでは，データ毎にタグデータとバリッドビットを付加する構成を採用している．先行研究のキャッシュメモリにおいて，仮にサブラインデータ (8Byte) 毎にタグを付与した場合，ラスタラインとスモールタイルの両アクセスに対応できる．しかし，この場合従来のキャッシュメモリに比べ，タグ容量がタグバンク数倍増加し，ハードウェ

ア規模が増大するという問題がある．そこで，先行研究のキャッシュメモリでは，スモールタイル毎にタグを付与し，整列されたタイルセットを同一ウェイに格納する RATS 法を採用している．この RATS 法により，先行研究のキャッシュメモリのタグ容量は，従来のキャッシュメモリと同容量まで低減される．

2.5 Simplescalar

Simplescalar とはオープンソースで提供されているマイクロプロセッサの命令セットアーキテクチャをシミュレートするプログラムであり，キャッシュメモリやプロセッサ等の各機能の性能を評価することも可能である．先行研究においては，この Simplescalar を改造することによってライン・タイル両アクセス対応新構成キャッシュメモリを実装し，性能評価を行なっている．

3　提案手法

3.1　先行研究の問題点

RATS 法を採用することで，列方向に連続する 4 つのサブラインデータからなるスモールタイルは一本のキャッシュラインに格納されるため，必ず同じウェイに入る．すなわち，タイルアクセスにおけるデータの格納には問題はない．しかし，ラインを構成する行方向に連続する 4 つのサブラインデータは異なるタグを持つため，必ずしも同じウェイに入るとは限らない．行方向優先アクセスである従来のラスタラインアクセスでは，行方向の連続データを同じウェイに格納する必要がある．又，ラインモードで行方向の連続データが同じウェイに入らない場合，行方向へのデータの局所性が不十分で性能が低下する可能性がある．

3.2　ライン・タイル両アクセス対応キャッシュ置換アルゴリズム

3.1 章で先行研究の問題点について記述した．この問題を解決するために，ライン・タイル両アクセス共通で使用できるライン・タイル両アクセス対応キャッシュ置換アルゴリズムを提案する．キャッシュ置換アルゴリズムとは，セット・アソシアティブ・キャッシュやフル・アソシアティブ・キャッシュにおいて用いられるプログラムであり，主記憶からデータ

をキャッシュメモリに格納する際に，どのウェイに格納するかを決定する．一般的なキャッシュメモリや先行研究のキャッシュメモリにおいては，キャッシュ置換アルゴリズムとして，LRU(Least Recently Used) が採用されている．LRU では最も使われていない期間が長いウェイにデータを置換する．しかし，LRU ではラインミス時にタイルセットを同一ウェイに入れることが出来ず，ラインアクセスにお対応していない．ラインアクセスに対応させるには行方向に連続するデータを同じウェイに格納させ，タイルセットを同じウェイに格納する必要がある．そこで，提案したライン・タイル両アクセス対応 LRU アルゴリズムでは，キャッシュアクセスがあった際に，現在のアクセスと前回のアクセスのインデックスとタグを比較し，それらのインデックスが連続かつタグが同一だった場合，行方向に連続したデータと判断し，同一のウェイに格納する．このような改造を施すことで, 行方向の連続データを同一ウェイに入れることを可能にし，ライン・タイルの切り替えに対応させる．

4　性能評価

4.1　評価環境

Simplescalar に提案手法を実装し性能評価を行う．提案キャッシュメモリは 4KB の 8 ウェイ・アソシアティブ・キャッシュとする．改造が施されていない従来のラスタアクセスのみの Simplecalr，先行研究の階層的なタイル形式が実装された Simplescalar，そして提案手法を実装した Simplescalr の 3 つに対し，総実行サイクル数，L1 キャッシュのミス率の 2 項目で性能を比較する．実行プログラムはそれぞれ 500 × 500,512 × 512,1000 × 1000,1024 × 1024,2000 × 2000,2048 × 2048 の行列乗算とする．

4.2　実行結果

行列サイズ	500	512	1000	1024	2000	2048
従来法	0.08	4.38	0.07	4.39	0.07	4.39
先行研究	0.08	0.09	0.10	0.10	0.10	0.10
提案手法	0.26	0.26	0.27	0.27	0.23	0.23

表 4.1: 各行列サイズでの dl1 キャッシュのミス率

図 4.1 に各行列サイズでの L1 キャッシュのミス率，図 4.2 に各行列サイズでの実行サイクル数を示す．L1 キャッシュのミス率において，行列

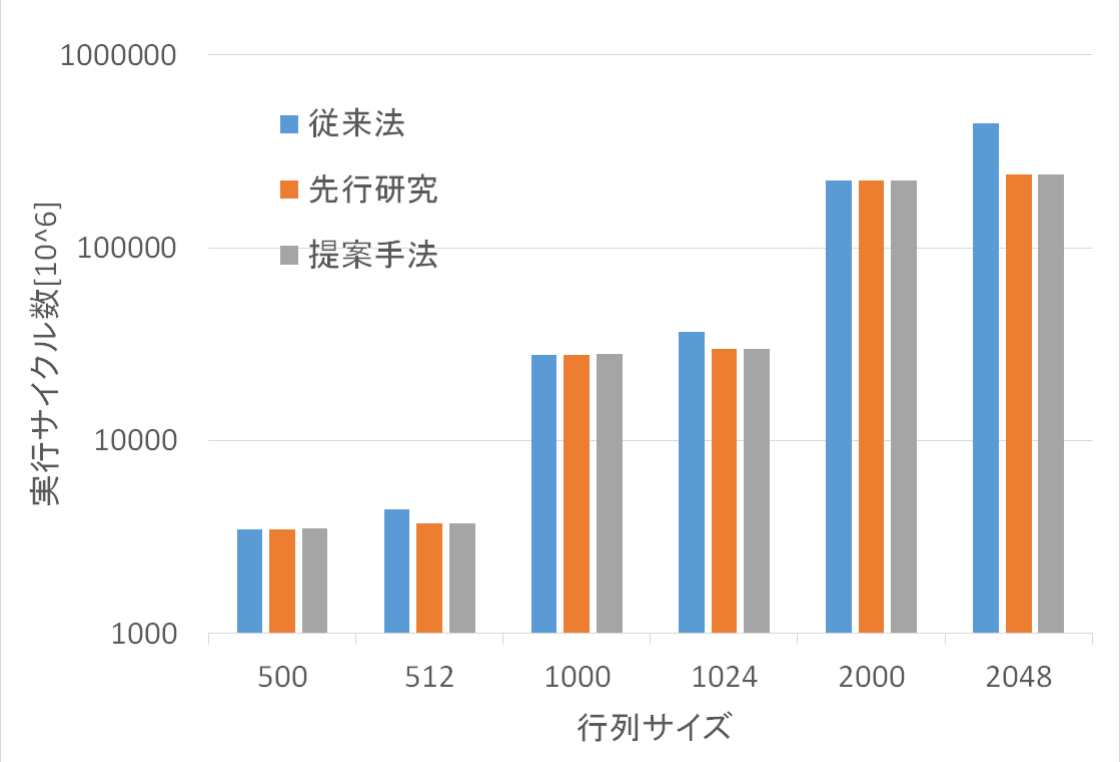

図 4.4: 各行列サイズでの実行サイクル数

サイズが 2 のべき乗の場合に従来法の性能が落ちる傾向がある．これは，行列サイズが 2 のべき乗の場合に列方向の連続データが同じキャッシュラインに格納されるため，競合性ミスが頻発していることを示す．提案手法に関しては，先行研究に比べ平均 2.6 倍高くなるという結果になった．総実行サイクル数は，従来法と比較すると行列サイズが 2 のべき乗の場合に，先行研究と提案手法が約 50%の大幅な削減に成功している．先行研究と提案手法とではほとんど値が変わらず，先行研究より提案手法が

平均0.005%高いという結果が出た.

5　おわりに

本研究では，結果的にタイルセットを同一ウェイに格納することには成功したが，今回の評価では先行研究と提案手法でサイクル数にほとんど差が出なかった．先行研究においての整列タイルセットが同一ウェイに入らない問題が，行列乗算単独のシミュレーションではあまり起こらなかった可能性が考えられる．行列乗算中に他の演算を行い，キャッシュメモリの中身のデータが頻繁に置換されるような評価環境では，先行研究において整列タイルセットが異なるウェイに格納される可能性が増えるため，提案手法の方がサイクル数が低くなると考えられる．今後の課題としては，そのような環境で改めて評価をし，提案手法の性能面での有用性を示すことが考えられる．

謝辞

本研究を行うにあたって, 常日頃から様々なご指導, アドバイスをいただきました近藤 利夫教授, 佐々木 敬泰助教, 深澤 祐樹技術員に感謝いたします. また, 研究室での共同生活等様々な面でお世話になりましたコンピュータアーキテクチャ研究室の皆さまに感謝いたします. 最後に, ここまで育ててくれた家族に最大限の感謝の意を表します.

参考文献

[1] ”タイル・ライン両アクセス機能を備えた新構成キャッシュメモリの提案”， 修士論文， 三重大学， 2014.02.