

マルチスレッドプロセッサにおける メモリアクセスレイテンシ隠蔽の一手法

松崎 隆哲† 富安 洋史‡ 大庭 直行‡ 雨宮 真人‡

†九州大学 大学院 システム情報科学府

‡九州大学 大学院 システム情報科学研究所

〒816-8580 福岡県 春日市 春日公園 6-1

{takanori, tomiyasu, ohba, amamiya}@al.is.kyushu-u.ac.jp

本稿では、マルチスレッドプロセッサにおけるメモリアクセスレイテンシの隠蔽の一手法について述べる。筆者らは複数のスレッド実行ユニットとメモリを一つのチップに搭載した FUCE プロセッサを提案している。FUCE プロセッサは、オンチップメモリを利用することでレイテンシの小さいメモリアクセスを実現する。また、スレッド先読み機構とプリロード機構を利用することで、メモリアクセスレイテンシの隠蔽を実現する。

A technique of hiding memory access latency for Multi-threading Processor

Takanori Matsuzaki, Hiroshi Tomiyasu, Naoyuki Ohba, Makoto Amamiya

Graduate School of Information Science and Electrical Engineering, Kyushu University

6-1 Kasuga-koen, Kasuga, Fukuoka, Japan, 816-8580

{takanori, tomiyasu, ohba, amamiya}@al.is.kyushu-u.ac.jp

On this paper, we describe a technique of hiding memory access latency for multi-threading processor. So, we evaluate performances of the memory access latency with the model of FUCE processor. FUCE processor achieves low latency memory access with on-chip memory. Also, FUCE processor achieves hiding memory access latency by the thread context pre-fetch technique and the pre-load technique. Thus, FUCE processor can achieve thread-execution without the memory latency.

1 はじめに

近年の高性能マイクロプロセッサにおける大きな問題の一つは、プロセッサとメモリの速度の乖離である。2000年にはマイクロプロセッサの動作速度が、1.5 GHzに達したのに対し、メモリは依然として低速である。表1に主なメモリの速度を示す。ここで、メモリのレイテンシについて着目すると、現在および今後予定されている高速メモリに関してもメモリのレイテンシが全く改善されていないとわかる。このメモリのレイテンシはプロセッサの性能に多大な悪影響を与えるため、メモリのレイテンシ

を隠蔽することは、プロセッサにとって大変重要な問題である。

そのため、現在のマイクロプロセッサは、キャッシュを利用することで平均メモリアクセスレイテンシを下げている。しかし、大量のデータを扱う場合やデータの再利用性が少ない場合は、キャッシュのヒット率が大きく下がり、平均メモリアクセスレイテンシを下げるができないという問題がある。

そこで、我々はデータキャッシュを利用する代わりに、オンチップメモリ、スレッドコンテキスト先読み機構、そしてプリロード機構を利用するこ

表 1: 主なメモリの動作速度

規格	throughput	latency
SDRAM(PC133)	1064 M byte/sec	53 n sec
DDR SDRAM	2.128 M byte/sec	53 n sec
RDRAM	0.8~1.6 G byte/sec	60~ n sec

とでメモリアクセスレイテンシを隠蔽する手法を FUCE(FUision of Communication and Execution) プロセッサ [1][2] に採用した。オンチップメモリを利用することにより、FUCE プロセッサのメモリアクセスレイテンシは 4 サイクル程度となる。これは、キャッシュメモリのアクセスレイテンシ (1 サイクル) よりも大きい、FUCE プロセッサの場合は後述するメモリアクセスレイテンシ隠蔽機構を利用することで、メモリのレイテンシを隠蔽することができる。

本稿では、マルチスレッドプロセッサにおけるメモリアクセスレイテンシの隠蔽手法を述べると共に、FUCE プロセッサの実行モデルを用いて隠蔽の効果について評価をおこなう。評価により、FUCE プロセッサは、メモリのレイテンシが 4 サイクル程度の場合、十分にメモリアクセスレイテンシを隠蔽する事ができるとわかった。

2 FUCE プロセッサの概要

FUCE プロセッサは、計算機における内部情報処理と通信の融合を目指した計算機システムである FUCE アーキテクチャ [3] にもとづいたプロセッサである。FUCE アーキテクチャは、細粒度マルチスレッド実行方式にもとづくメモリ混載プロセッサ FUCE プロセッサと、細粒度マルチスレッド実行方式にもとづくオペレーティングシステム CEFOS(Communication-Execution Fusion Operating System)[4][5][6] から構成される。

現在、オンチップマルチプロセッサやメモリ混載プロセッサの研究として SMT[7]、MP98[8]、Hydra[9] などが提案されている。SMT ではいくつかの独立したスレッドを、複数の実行ユニットに動的に割り当てて実行することで、スレッドレベルの並列性を実現している。しかし、FUCE プロセッサでは、一つの実行ユニットに割り当てるスレッドは一つとし、スレッドの並列性は多数の実行ユニットでスレッドを同時に実行することによって実現してい

る。MP98 では、レジスタ継承、データ依存解消、投機実行などを実現している。これらの機構を実現することで、スレッドの実行や生成を効率良くおこなうことができるが、ハードウェア構造が複雑となるため、FUCE プロセッサではそれらの機構を利用しない。FUCE プロセッサは、半導体上の面積の小さい単純な実行ユニットを多数同一チップ上に搭載するマルチスレッドプロセッサである。単純な実行ユニットで多数のスレッドを同時に実行することで、スレッドレベルの並列度を上げることを主眼としている。

また、Hydra は、プロセッサ上にメモリを混載している点で FUCE プロセッサと似ているが、FUCE プロセッサではプロセッサ上のメモリをキャッシュではなく、メインメモリとして利用する点で、Hydra と異なっている。FUCE プロセッサは、同一チップ上にプロセッサ、通信ユニット、メモリを搭載することで、命令実行、通信処理、メモリアクセス間の性能格差を縮小する。これにより、プロセッサの命令実行(内部処理)とプロセッサ外との通信(通信処理)とを融合して扱うことを目標とする。

図 1 に、FUCE プロセッサの概略を示し、以下に特徴をあげる。

- スレッドを実行するユニットを複数持つ
- スレッド化されていない命令を実行するユニットを持つ
- 同一チップ上に命令実行ユニット、通信ユニット、メモリを搭載する
- 命令実行ユニット、通信ユニット、メモリ間は、高バンド幅内部バス (256 G byte/sec) で接続する
- データキャッシュを持たない
- スレッドコンテキストの先読みをおこなう
- チップ外にメモリを持たない

FUCE プロセッサは、2005~10 年頃を想定している。そこで、2005~10 年頃に利用できる半導体技術を検討し、表 2 のようにプロセッサの仕様を予定している。

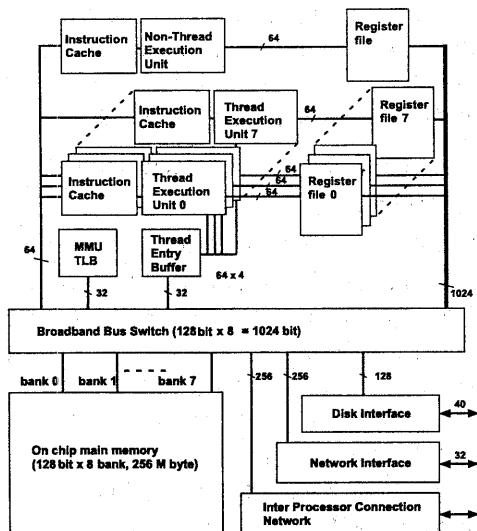


図 1: FUCE プロセッサ構成図

表 2: FUCE プロセッサの予定スペック

	2005 年	2010 年
プロセッサ動作周波数	2 G Hz	10 G Hz
オンチップメモリ容量	256 MB	1 GB
オンチップメモリ動作速度	1 G Hz	5 G Hz
内部バスデータ転送速度	256GB/sec	2.5TB/sec

3 FUCE プロセッサの内部構造

3.1 命令実行ユニット

FUCE プロセッサは、スレッド実行ユニットと非スレッド実行ユニット (コントロールユニット) の 2 種類の命令実行ユニットを持つ。

非スレッド実行ユニットは、OS のカーネル、デバイスドライバ、割り込み処理、即時応答性を要求する処理をおこなうユニットである。

スレッド実行ユニットは、一つのスレッドを中断なく実行するユニットである。スレッド実行ユニットは単純な構成とし、多数の実行ユニットで並列実行性能を上げる。また、スレッドが中断なく実行されるため、スレッド実行ユニットはスレッドのコンテキストの退避や回復をおこなう機能は持たない。

以下にスレッド実行ユニットの特徴を説明する。

2 命令同時発行 命令は 2 命令ずつ in order で発行。

ブロックロード/ストア レジスタを複数本まとめ、ブロック単位で転送をおこなう。

ノンブロッキングロード データをメモリからレジスタへとロードする際に、パイプラインを止めずにデータロードをおこなう。その際、レジスタ競合の確認はスコアボードを用いて行い、後続命令がレジスタを利用する場合に、はじめてパイプラインを停止する。

交替レジスタファイル FUCE プロセッサは、一つのスレッド実行ユニットに、レジスタファイルを 2 個持っている。一方をスレッド実行に利用し、他方は後述するスレッドコンテキスト先読み機構に利用する。

スレッドエントリバッファ スレッドエントリバッファは、ハードウェアで管理するスレッドエントリを保持するバッファである。

3.1.1 スレッドコンテキスト先読み機構

細粒度マルチスレッド実行方式を利用することによって、スレッドの切り替えが多くなる。そのため、スレッドコンテキスト切り替えに伴うオーバーヘッドが生じる。そこで、FUCE プロセッサでは、交替レジスタファイルとスレッドエントリバッファを利用して、スレッドコンテキストの先読みをおこなう。これにより、このオーバーヘッドの隠蔽をおこない、スレッドの切り替えの高速化を実現する。

具体的には、先行スレッドがスレッド実行ユニットで実行開始されるとともに、後続スレッドをスレッドエントリバッファからスレッドエントリを読み込む。そして、後続スレッドのスレッドエントリをもとにして、メモリから交替レジスタファイルの空いているレジスタファイルへ、後続スレッドのコンテキストを読み込む。これにより、先行スレッド終了後、レジスタファイルを切り替えることで、直ちに後続スレッドの実行を開始することができ、スレッド切り替えを高速化できる。

3.2 メモリシステム

3.2.1 オンチップメモリ

既存のメモリをチップ外部に持つプロセッサは、ピン数がボトルネックとなり、メモリのバス幅を広げることができない。しかし、メモリ混載プロセッサはプロセッサのピンボトルネックが生じないた

め、メモリのバス幅を大幅に拡張できる。また、オンチップメモリを採用することによって、レイテンシの小さいメモリを実現することができる。つまり、FUCE プロセッサはメモリに関して、低レイテンシ(4サイクル程度)と高バンド幅(256 G byte/sec)という二つの特徴を持つ。

3.2.2 データキャッシュレス構造

メモリに関する特徴と細粒度処理をおこなうことを考慮して、FUCE プロセッサはデータキャッシュを持たない。これは、レイテンシが小さくて高バンド幅であるオンチップメモリの特徴やスレッド実行ユニットの機構を利用することで、メモリアクセスレイテンシを十分隠蔽することができるためである。また、細粒度マルチスレッド処理をおこなうことにより、キャッシュに入っているデータを再度利用する可能性が低いためである。このように、キャッシュを利用することによる効果よりもオーバヘッドの方が大きいと考えるからである。

ただし、命令コードは局所性が高いので、命令キャッシュは利用する。

3.3 メモリアクセスレイテンシ隠蔽機構

FUCE プロセッサでは、オンチップメモリの特徴であるレイテンシの小さなメモリとスレッド実行ユニットの機構を利用することでメモリアクセスレイテンシを隠蔽することができる。これは、ブロックロード/ストアやノンブロッキングロードのスレッド実行ユニットの機構を利用することによって実現する。また、前述したスレッドコンテキスト先読み機構を利用することによって、スレッド切り替え時に発生するメモリアクセスを隠蔽することができる。これにより、細粒度処理において問題となる、スレッドコンテキスト切り替えのオーバヘッドを隠蔽することができる。

FUCE プロセッサは、メモリアクセスレイテンシ隠蔽機構を実現するために、ブロックロード/ストアやノンブロッキングロードを用いる。これは、コンパイラが命令を最適化することによって、必要とされる前にデータをレジスタへとロードするプリロード機構で実現される。これにより、データハザードによるパイプラインストールを避けることができる。また、ブロックロード/ストアを利用して、データ転送命令の回数を減らすことができる。

以上のことで実現されるメモリアクセスレイテンシ隠蔽機構を利用することによって、FUCE プロセッサはメモリアクセスによってパイプラインを停止せずにスレッド実行することを実現する。

4 メモリアクセスレイテンシ隠蔽の効果

4.1 評価環境

FUCE プロセッサにおけるメモリアクセスレイテンシ隠蔽機構の効果を評価するために、各要素が単精度浮動小数点データである 1000×1000 の行列積の計算をおこなった。評価は、実行クロック数とパイプラインストールの割合に関して、以下の4種類の方式において、メモリアクセスレイテンシを4, 6, 8サイクルと変化させることによっておこなった。図2に実行クロック数を示し、図3にパイプラインストールの割合を示す。

方式(a) プリロード機構+スレッドコンテキスト先読みを利用

方式(b) プリロード機構のみ利用

方式(c) スレッドコンテキスト先読み機構のみ利用

方式(d) プリロード機構+スレッドコンテキスト先読みを共に利用しない

また、メモリアクセスレイテンシが4サイクルの際に、ベクトル長を変化させることによって、どのように影響があるかについても評価をおこなった。図4に実行クロック数を示し、図5にパイプラインストールの割合を示す。

なお、命令キャッシュは常にヒットするものとして、評価をおこなった。

表3に、評価に用いたFUCE プロセッサの命令実行ユニットの基本モデルを示す。性能評価に用いたFUCE プロセッサは、この命令実行ユニットを8個持ったプロセッサとする。つまり、8個のスレッドを同時に実行することができる。

表3: 命令実行ユニットの基本モデル

命令発行	2 命令/クロック
レジスタ本数	32 本 × 2
レジスタのブロック転送単位	4 本/1 命令
メモリアクセスレイテンシ	4, 6, 8 サイクル
浮動小数点命令の実行レイテンシ	4 サイクル

4.2 性能評価

図2と図3は、メモリアクセスレイテンシを4, 6, 8サイクルと変化させた際の実行クロック数およびパイプラインストールの割合である。メモリアクセスレイテンシ隠蔽機構について着目すると、プリロード機構の有無によって大きな性能の格差が生じている。また、スレッドコンテキスト先読み機構の有無に関してはあまり性能差が生じていない。

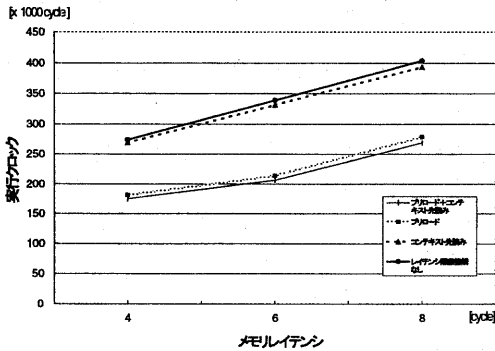


図2: メモリアクセスレイテンシを変化させた場合の実行クロック数

図3によって、メモリアクセスレイテンシ隠蔽機構の利用状態により、パイプラインストールの割合が大きく変化しているとわかる。特に、メモリアクセスレイテンシが4サイクルの際は、プリフェッチ機構+スレッドコンテキスト先読みを利用した場合と、プリフェッチ機構+スレッドコンテキスト先読みを利用していない場合を比較すると約4倍の差が生じている。また、メモリアクセスレイテンシが4,6,8と変化するに従って、パイプラインストールの割合は増加している。このことより、メモリアクセスレイテンシ隠蔽をおこなうためには、メモリアクセスレイテンシを4サイクル程度にする必要があるとわかる。

図4と図5による、ベクトル長を変化させた場合においても、プリロード機構の方がスレッドコンテキスト先読み機構よりも性能に寄与しているとわかる。また、ベクトル長が短くなると実行クロック数が増加し、パイプラインストールの割合が増加している。これは、ベクトル長が短くなったため、スレッドコンテキスト切り替えが増加したからである。

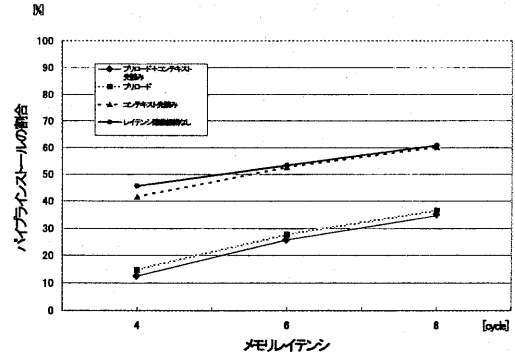


図3: メモリアクセスレイテンシを変化させた場合のパイプラインストールの割合

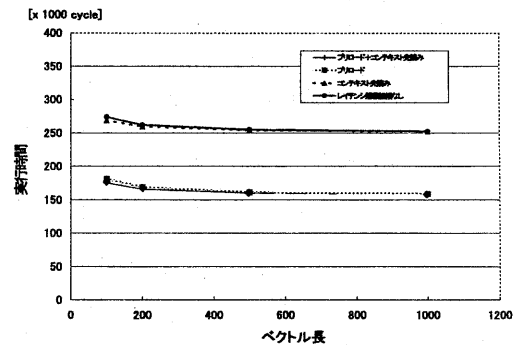


図4: ベクトル長を変化させた場合の実行クロック数

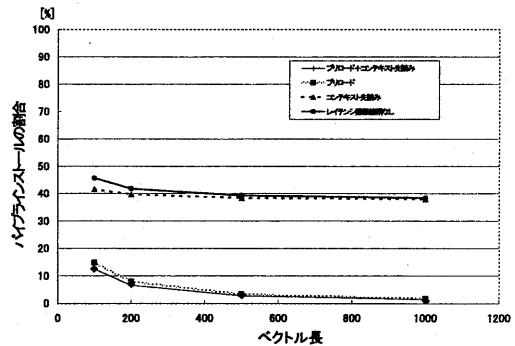


図5: ベクトル長を変化させた場合のパイプラインストールの割合

以上のことより、FUCEプロセッサはメモリアクセスレイテンシ隠蔽機構を利用することで、メモリアクセスレイテンシを隠蔽できている。特に、ブロード機構は、メモリアクセスレイテンシ隠蔽をおこなうために大変重要である。

5 おわりに

本稿では、マルチスレッドプロセッサにおけるメモリアクセスレイテンシ隠蔽機構の一手法について述べた。また、FUCEプロセッサの実行モデルにおいて性能評価をおこなうことで、メモリアクセスレイテンシ隠蔽機構の効果について述べた。

FUCEプロセッサでは、オンチップメモリシステムを利用することで、低レイテンシで高バンド幅のメモリを実現する。これにより、FUCEプロセッサの構造は、キャッシュメモリを利用する既存のプロセッサアーキテクチャとは異なる構造(データキャッシュレス構造)を採用している。そのため、FUCEプロセッサでは、メモリアクセスレイテンシ隠蔽をおこなうための機構を備えることで、データキャッシュレス構造においても、メモリアクセスレイテンシを感じさせない命令実行を実現している。

現在、プロセッサ構造の詳細を検討すると共に、FUCEプロセッサのシミュレーション環境の開発を行っている。今後、シミュレーションにより本アーキテクチャの性能評価をおこない、FUCEプロセッサ設計方針の有効性を確認する予定である。

本研究は、通信・放送機構の創造的情報通信技術研究開発推進制度に係わる研究開発課題「次世代型インテリジェント・マルチメディア情報通信網の基盤技術に関する研究」による。

参考文献

- [1] 松崎 隆哲, 富安 洋史, 大庭 直行, 雨宮 真人, “高バンド幅内部バス構造のオンチップメモリを持つFUCEプロセッサ”, 情処研報, 2000-HPC-83, Vol.2000, No.93, pp.7-12 (2000).
- [2] 松崎 隆哲, 富安 洋史, 大庭 直行, 雨宮 真人, “通信と処理との融合を行うFUCEプロセッサの提案”, 信学技報, CPSY2000-52, Vol.100, No.249, pp.1-7 (2000).
- [3] M. Amamiya, et al. “An architecture of fusing communication and execution for global distributed processing”, In SSGRR2000 Computer and Business Conference, 8 2000.
- [4] 日下部 茂, 富安 洋史, 村上 和彰, 谷口 秀夫, 雨宮 真人, “並列分散オペレーティングシステム CEFOS(Communication-Execution Fusion OS)”, 信学技報, CPSY99-50, Vol.99, No.251, pp.25-32 (1999).
- [5] 谷口 秀夫, 日下部 茂, 棚林 拓也, 中山 大士, 雨宮 真人, “CEFOS オペレーティングシステムのスレッド管理機構”, 情処研報, 2000-OS-83, Vol.2000, No.21, pp.7-12 (2000).
- [6] 棚林拓也, 中山大士, 日下部茂, 谷口秀夫, 雨宮真人, “並列分散オペレーティングシステム CEFOSにおける一括システムコールの機構”, 信学技報, CPSY2000-57, Vol.100, No.249, pp.41-48 (2000).
- [7] Dean M. Tullsen, et al. “Simultaneous Multithreading: Maximizing On-Chip Parallelism,” Proceedings of the 22nd Annual International Symposium on Computer Architecture, June, 1995.
- [8] N. Nishi, et al. “A 1GIPS 1W Single-Chip Tightly-Coupled Four-Way Multiprocessor with Architecture Support for Multiple Control Flow Execution”, In Proc. ISSCC2000, WP25.5.
- [9] L. Hammond, et al. “The Stanford Hydra CMP”, IEEE Micro, Vol. 20, No. 2, March/April 2000
- [10] The International Technology Roadmap for Semiconductors (ITRS), <http://www.itrs.net/ntrs/publntrs.nsf>
- [11] MoSys, Inc. <http://www.mosysinc.com/mhome/>