

FUCE プロセッサにおけるメモリシステムの提案と評価

田中 康嗣† 松崎 隆哲‡ 棚林 拓也‡ 大庭 直行* 雨宮 真人*

†九州大学 工学部 電気情報工学科

‡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

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

オンチップメモリと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の削減をおこなう FUCE プロセッサのメモリシステムを提案する。FUCE プロセッサでは、オンチップメモリを利用することで、高速なメモリアクセスを実現するとともに、メモリアクセスレイテンシ隠蔽機構によって、メモリアクセスレイテンシの削減を実現している。本稿では、FUCE プロセッサのメモリシステムを紹介し、レイテンシ隠蔽機構について簡単な評価をおこなう。

The proposal and evaluations of a memory system for FUCE processor

Koji Tanaka†, Takanori Matsuzaki‡, Takuya Tanabayashi‡,

Naoyuki Ohba‡, and Makoto Amamiya‡

†Department of Electrical Engineering and Computer Science

‡Graduate School of Information Science and Electrical Engineering,

Kyushu University

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

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

This paper describe mechanisms of hiding memory access latency in FUCE processor. By using an on-chip memory, FUCE processor cuts down a memory access latency. In addition, FUCE processor hides memory access latency. In this paper, we describe the memory system of a FUCE processor, and evaluate mechanisms of hiding memory access latency.

1 はじめに

プロセッサの動作周波数は飛躍的に向上し、1GHzで動作するプロセッサが広く使われている。しかし、プロセッサチップ外の動作周波数は、依然として100MHz 前後と遅く、チップ内外の速度差は、ますます広がっている。そのため、チップ外に接続したメインメモリとの速度差が拡大し、プロセッサ性能の足枷となっている。

そこで、我々は今後の集積回路技術の進歩を見据え、メモリをプロセッサと同一チップに搭載し、細粒度マルチスレッド実行方式を採用した FUCE(FUision

of Communication and Execution) プロセッサを提案してきた [2] [3]。プロセッサと同一チップにメインメモリを搭載することで、チップ外にメインメモリを接続しているこれまでのプロセッサよりも高速なメモリアクセスを実現することができる。

現在の半導体集積回路技術は、1 平方センチメートルのシリコンチップ上に約 1 億トランジスタを搭載した回路が実現可能であり、その集積密度の向上は約 3 年ごとに約 4 倍の速度となっている。この速度は今後も継続すると予測 [1] されており、2010 年には、7 億トランジスタ以上を 1 平方センチメートル

ル程度のシリコンチップ上に実現できるようになると予想される。したがって、オンチップメモリとして利用できるトランジスタ数も増加し、メインメモリとして十分な容量が確保できると考えている。

オンチップメモリをメインメモリとすることによって、メモリアクセスレイテンシを短くすることができるが、それだけではプロセッサ性能の足枷を取り除くのに十分とは言えない。そこで、FUCE プロセッサでは、メモリアクセスレイテンシ隠蔽のための機構を提供する。

以下本稿では、2章にてFUCE プロセッサの概要を述べ、3章においてFUCE プロセッサのメモリシステムについて述べる。4章では、レイテンシ隠蔽機構について述べ、5章でその評価を行う。最後に6章で本稿のまとめと今後の予定について述べる。

2 FUCE プロセッサの概要

FUCE プロセッサは、同一チップ上にプロセッサ、メモリ、通信ユニットを搭載したオンチップマルチプロセッサである。図1にFUCE プロセッサの概要を示し、その特徴を以下に述べる。

- 同一チップ上に命令実行ユニット、メモリ、通信ユニットを搭載する
- 命令実行ユニット、メモリ、通信ユニット間は高バンド幅の内部バスで接続する
- 命令実行ユニットは、スレッド実行ユニットと、非スレッド実行ユニットからなる
- スレッドとは逐次実行の基本単位で、実行中のスレッドは、他のスレッドにより中断されない
- スレッド実行ユニットは、スレッド化された命令のみを実行する。
- 非スレッド実行ユニットは、スレッドに適しない処理を行なう。スレッドに適しない処理としては、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などがあげられる
- 各命令実行ユニットは、2命令同時発行する。同時発行する組み合わせはコンパイル時に決定し、動的なスケジューリングは行わない

FUCE プロセッサは2005～2010年を想定している。そこで2005～2010年に利用できる半導体技術を検討し、プロセッサの仕様を考えている。

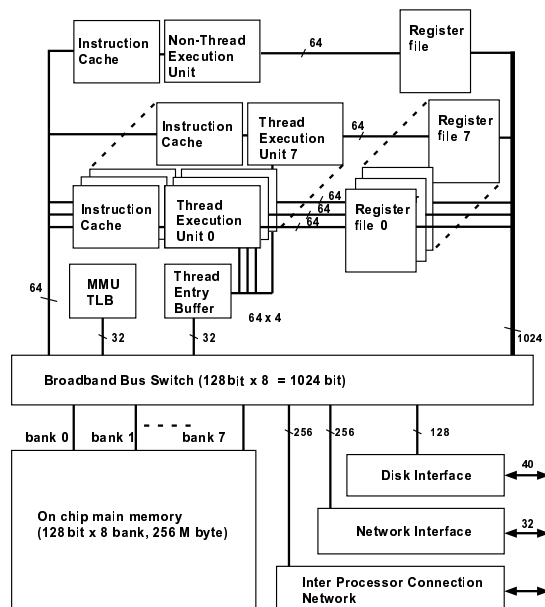


図1: FUCE プロセッサ構成図

3 FUCE プロセッサのメモリシステム

現在のプロセッサで性能向上の大きな障害となっているのは、パイプラインのストールである。パイプラインのストールの主な原因は、分岐命令とロード命令である。ロード命令を実行すると、対象とするデータがレジスタに読み込まれるまで遅延が生じる。その間、依存する後続命令を実行することができないので、パイプラインストールが生じる。したがって、データの読み込みにかかる遅延(レイテンシ)を削減することは、プロセッサ性能を向上する上で重要である。

しかし、チップ外のメモリをアクセスする際のレイテンシは増加している。これは、プロセッサチップの動作周波数が飛躍的に向上し、1GHz以上で動作することも珍しくなくなったのに対し、チップ外の動作周波数は依然として100MHz程度と遅いため、チップ外メモリの速度が相対的に低下しているからである。

そのため現在のプロセッサは、メモリアクセスレイテンシを改善するため、チップ内にキャッシュメモリを搭載している。これにより、チップ外メモリ

にアクセスする頻度を削減し、平均アクセスレイテンシを削減している。

しかしながら、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には、キャッシュが有効に機能せず、平均アクセスレイテンシを削減することができないこともある。

また、キャッシュを用いる場合、アクセスレイテンシの予測が難しくなる。キャッシュにヒットした場合のレイテンシは数サイクルから数十サイクルであるが、キャッシュミスが発生すると、そのペナルティは数百サイクルに及ぶこともある。

そこで、FUCE プロセッサでは、オンチップメモリをメインメモリとし、チップ外のメモリはメインメモリとはしない。メインメモリをチップ内に置くことによって、低レイテンシでのアクセスを可能にする。またオンチップメモリは、帯域の点でも有利である。オフチップメモリの場合、バス幅がチップ内外を接続するピン数によって制限されるが、オンチップの場合はそのような制限はなく、バス幅の拡張が容易なためである。

表 1 に既存のプロセッサのメモリ階層を、表 2 に FUCE プロセッサのメモリ階層を示す。

FUCE プロセッサでは、メインメモリにはオンチップメモリのみ使用し、チップ外のメモリは使用しない。そのため、用途によってはオンチップメモリではメモリ容量が不足する場合も考えられる。その場合、プロセッサ間接続ネットワークにメモリ、及びメモリコントローラを接続することによって、リモートプロセッサ上のメモリとして利用する (図 2)。

4 メモリアccessレイテンシ隠蔽機構

メモリアccessレイテンシ隠蔽を考える場合、FUCE プロセッサではメモリアccessを次の 2 つに分類して考える。

1. 予測可能で、レイテンシの小さな場合
2. 予測不可能、またはレイテンシの大きな場合

(1)「予測可能で、レイテンシの小さな場合」とは、データが必要となるより前に、まえてアクセスする場所が予測可能であり、かつ、その領域がオンチップメモリ内に存在することがわかっている場合である。(2)「予測不可能、またはレイテンシの大きな場合」とは、アクセスする場所がデータを

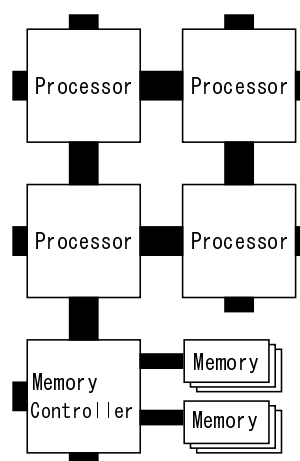


図 2: FUCE プロセッサのマルチプロセッサ構成

必要とする命令の直前までわからない場合や、チップ外のメモリをアクセスして、レイテンシが大きくなる場合などである。

FUCE プロセッサにおけるレイテンシ隠蔽機構の適応範囲を図 3 に示し、説明する。

FUCE プロセッサでは、(1)「予測可能で、レイテンシの小さな場合」、つまり、オンチップメモリにアクセスする場合、コンパイル時の命令スケジューリングによりレイテンシを隠蔽する。また、(2)「予測不可能、またはレイテンシの大きな場合」は、マルチスレッディングを用いて、スレッドの動的スケジューリングによりレイテンシを隠蔽する。

FUCE プロセッサでは、具体的には次のようなレイテンシ隠蔽機構を持つ。

1. Non Blocking Load 命令
2. Block Load/Store 命令
3. スレッドコンテキスト先読み機構

それぞれについて、以降、説明する。

Non Blocking Load 命令とは、メモリからデータをロードする場合、読み込みが完了するまでパイプラインをストールさせるのではなく、実行可能な後続命令は実行を続けるロード命令で、読み込みが完了する前にデータを利用しようとする、そのときにパイプラインがストールする。また、Block Load/Store 命令は、複数のデータを一括して読み込む命令で、アドレス指定を一括してメモリに送ることによってレイテンシを削減する。

表 1: 既存のプロセッサのメモリ階層

Level	1	2	3	4
名称	レジスタ	キャッシュ	メインメモリ	ディスク
実装技術	専用メモリ	オンチップ SRAM	オフチップ DRAM	磁気ディスク
管理	コンパイラ	ハードウェア	ソフトウェア	ソフトウェア

表 2: FUCE プロセッサのメモリ階層

Level	1	2	3	4
名称	レジスタ	メインメモリ	リモートメモリ	ディスク
実装技術	専用メモリ	オンチップメモリ	オンチップメモリ	磁気ディスク
管理	コンパイラ	ソフトウェア	ソフトウェア	ソフトウェア

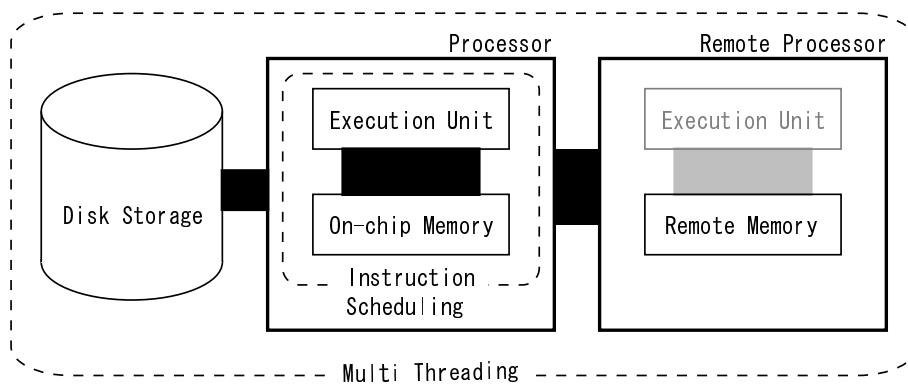


図 3: レイテンシ隠蔽機構の適応範囲

スレッドコンテキスト先読み機構とは、スレッド切り替えのオーバーヘッドを削減する機構である。スレッドコンテキスト先読みの動作を図 4 に示す。

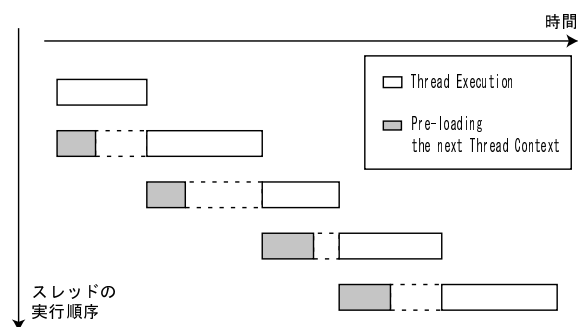


図 4: スレッドコンテキスト先読み

スレッド実行ユニットには、実行中のスレッドと、次に実行するスレッドが割り当てられる。スレッドの実行と、次に実行するスレッドのコンテキスト

の読み込みをオーバーラップさせることによって、スレッド切り替えに伴うコンテキスト読み込みのレイテンシを隠蔽する。これにより、スレッド切り替えのコストを削減し、より粒度の小さいスレッド(細粒度スレッド)の実行を可能にする。細粒度スレッドを用いることによって、きめ細かいレイテンシ隠蔽を行う。

5 評価

4章で説明した、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、次の二つのベンチマークプログラムを利用して、FUCE プロセッサにおけるレイテンシ隠蔽機構の効果を評価した。

- 1000 × 1000 の行列積の計算
- 8-Queen 問題の全探索問題

(1) 「1000 × 1000 の行列積の計算」では、4章で説明したレイテンシ隠蔽手法の内、命令スケジュー

リングによるレイテンシ隠蔽について、評価を行う。行列積の計算では、データが必要になる際に、アクセスする場所が予測可能であるため、コンパイル時の命令スケジューリングによってレイテンシを隠蔽することが可能である。(2)「8-Queen 問題の全解探索問題」では、4章で説明したレイテンシ隠蔽手法の内、細粒度スレッドによるレイテンシ隠蔽について、評価を行う。8-Queen 問題は行列積と異なり、単純な命令スケジューリングではレイテンシを隠蔽できないメモリアクセスが多く含まれる。そのようなメモリアクセスに対し、細粒度スレッド実行方式を適用し、レイテンシ隠蔽を行った。

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 ~ 16 サイクル
浮動小数点命令の実行レイテンシ	4 サイクル

5.1 行列積

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。

行列積では、メモリへのアクセスパターンの予測が容易であるため、レイテンシ隠蔽機構として、コンパイル時の命令スケジューリングによる Non Blocking Load 命令と Block Load/Store 命令を利用している。

評価は、レイテンシ隠蔽機構を利用した場合と利用しなかった場合について、メモリアクセスレイテンシを 4 ~ 16 サイクルへと変化させた際における、実行クロック数、パイプラインストール、パイプラインアイドルの割合についておこなった。結果を図 5 に示す。これより、レイテンシ隠蔽機構を利用することによって、メモリアクセスレイテンシが 10 サイクルまでは、レイテンシを十分隠蔽することができているとわかる。

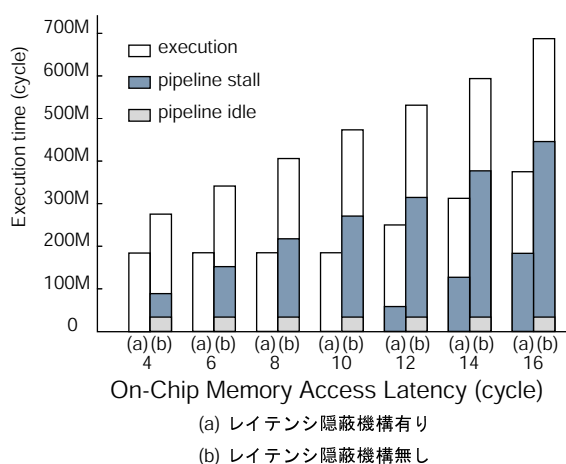


図 5: 行列積

5.2 8-Queen

メモリアクセスレイテンシが 4 サイクルの場合において、レイテンシ隠蔽機構を利用した場合と利用しなかった場合について、8-Queen 問題の全解探索問題の実行クロック数とパイプラインストールの割合について評価をおこなった。

8-Queen 問題では、メモリへのアクセスパターンの予測が困難なため、マルチスレッディングを用いて、スレッドの動的スケジューリングによってレイテンシの隠蔽をおこなった。これは、スレッドコンテキスト先読み機構と Block Load/Store 命令を利用することによって、スレッド切り替えのオーバーヘッドを削減することで実現をした。結果を図 6 に示す。これより、レイテンシ隠蔽機構を利用した場合は利用しなかった場合と比較して、2 割程度の性能向上をしているとわかる。これは、スレッド切り替えのオーバーヘッドを削減することで、マルチスレッディング環境下における性能向上ができると示している。

5.3 結論

二つの性能評価の結果より、FUCE プロセッサにおけるメモリアクセスパターンの二つの分類、「予測可能で、レイテンシの小さな場合」と「予測不可能、またはレイテンシの大きな場合」において、それぞれに対応するレイテンシ隠蔽機構を用意することによって、メモリアクセスレイテンシ削減の効果が得られているとわかる。

これは、「予測可能で、レイテンシの小さな場合」

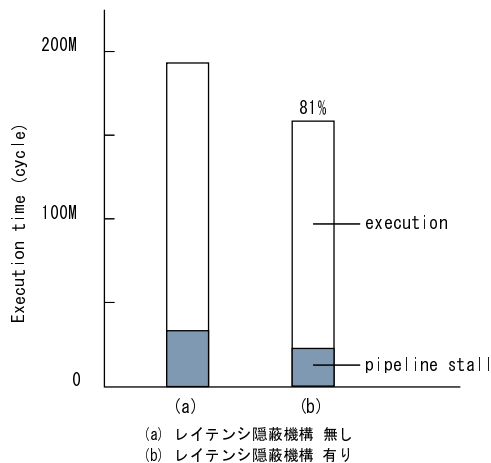


図 6: 8-Queen 問題

は、前もってデータをロードしておくことでメモリアクセス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 Block Load/Store 命令を利用することによって、データの転送に要するクロック数を削減することで、メモリアクセスに要するレイテンシを削減することができる。

一方、「予測不可能、またはレイテンシの大きな場合」は、前もってデータをロードすることは不可能である。そのため、細粒度スレッド実行方式を適用し、スレッドの動的スケジューリングによってレイテンシ隠蔽を行った。これにより、メモリアクセスの予測が困難な場合においても、レイテンシの隠蔽が可能となる。細粒度スレッド実行方式ではスレッド切り替えのオーバーヘッドが問題となるが、スレッドの実行と次に実行するスレッドのコンテキストの読み込みをオーバーラップさせることによって、スレッド切り替えのオーバーヘッドを削減し、メモリアクセスレイテンシの隠蔽に適用することを可能にした。

これらのことより、FUCE プロセッサのメモリアクセスレイテンシ隠蔽機構は、メモリアクセスレイテンシの削減を実現するとともに、細粒度スレッド実行の効率化を実現している。

6 おわりに

本稿では、FUCE プロセッサにおけるメモリシステムの概要と、レイテンシ隠蔽機構について述べた。また、レイテンシ隠蔽機構の効果について評価をおこなった。

FUCE プロセッサでは、オンチップメモリをメインメモリとし、チップ外のメモリはメインメモリとはしない。メインメモリをチップ内に置くことによって、低レイテンシでのメモリアクセスを可能とする。

また、メモリアクセスを「予測可能で、レイテンシの小さな場合」と「予測不可能、またはレイテンシの大きな場合」に分類し、それぞれに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、メモリアクセスのレイテンシを隠蔽することが可能である。

今回の評価では、FUCE プロセッサのレイテンシ隠蔽機構の効果を明らかにするため、これらの機構を適用しやすいアプリケーションを用いて評価を行っている。したがって、レイテンシ隠蔽機構を他のアプリケーションに適用した場合の評価が今後の課題である。

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] The International Technology Roadmap for Semiconductors(ITES). <http://public.itrs.net/>.
- [2] 松崎 隆哲, 富安 洋史, 大庭 直行, 雨宮 真人, “通信と処理の融合を行なう FUCE プロセッサの提案”, 信学技報, CPSY2000-52, Vol.100, No.249, pp.1-7(2000)
- [3] Makoto Amamiya, Hideo Taniguchi, Takanori Matsuzaki: An Architecture of Fusing Communication and Execution for Global Distributed Processing, Parallel Processing Letters, Vol.11, No.1, pp.7-24, 2001