

並列推論マシン PIM/i のファイングレインバラレリズム

加藤研児、佐藤正俊、武田浩一、大原輝彦

kato@okilab.oki.co.jp

沖電気工業株式会社

1 はじめに

我々は、第五世代コンピュータプロジェクト [1] の一環として並列推論マシン PIM/i の研究開発を行っている [2]。PIM/i は、ターゲット言語である並列論理型言語 KL1 を並列実行するマシンである。PIM/i の目指す並列性の抽出方法は、言語の処理方式レベルで並列処理を実現する方法とプロセッサの命令レベルで並列処理を実現する方法の 2 点である。前者は、KL1 のゴールリダクションに基づいたリダクション単位の並列実行で、マルチプロセッサ上でのコースグレインの並列性抽出を行う。一方、後者は、KL1 処理を記述した PIM/i 機械語単位の並列実行で、プロセッサ内でのファイングレインの並列性抽出を行う。

本稿では、後者のファイングレインの並列性抽出について述べる。KL1 を PIM/i で実行するには、KL1 を抽象機械語の KL1b にコンパイルし、さらに KL1b を PIM/i の機械語に変換し直す方法を取る。ここで、変換される PIM/i 機械語は、KL1 の処理方式を実現するプログラムである。ここでのファイングレインの並列性抽出は、KL1 の処理方式を実現するプログラムの記述に関する課題である。KL1 処理方式の仕様記述から機械語を生成する時の最適化の知見を得るために、アセンブラで KL1 の処理方式をテストコーディングし、並列性抽出の考察を行ったので報告する。

2 命令セットと並列性

2.1 命令セット

命令セットの特徴をまとめると以下のようになる。

1. タグ付きアーキテクチャ
データ / 命令長は、40 ビットである。データは、タグ 8 ビットと値 32 ビットで表現し、命令は、40 ビット固定長で表現する。
2. RISC 型
命令の粒度を 1 サイクルで実行できる単純な命令に限定した。

3. バイブライン・アーキテクチャ

命令の実行は、命令フェッチ、レジスタ読み出し、または、命令実行、レジスタへの書き込みの 3 段階ステージのバイブラインで、実効的に 1 命令を 1 マシンサイクルで実行する。

4. 複数命令の同時実行

命令は、最大 3 つのオペレーション (シーケンサ:S、メモリ:M、プロセッサ:P) から成り、これらのオペレーションは 1 マシンサイクルに同時実行可能である。

2.2 バイブライン実行における並列性

バイブライン実行における並列性の抽出は、3 段階のバイブラインが乱れなく充足された場合、最適であると考えられる。ここで、バイブラインの乱れが生じる時間は、分岐命令実行時の分岐先命令のフェッチに要する時間と命令やデータのキャッシュミス時のフェッチのための時間である。前者は、分岐命令実行中にフェッチした命令をディレイドスロットで実行する命令とそれを用いたプログラミングで、無駄なサイクルを軽減できる。後者については、LM を利用しキャッシュのヒット率を上げた場合と同等の効果を期待する場合の考察を述べる。

PIM/i では、ディレイドスロットで実行する命令を分岐の有無で制御する以下の 3 つの命令を用意し、分岐特性に応じたプログラミングを可能としている。図 1 にこれらの分岐命令を示す。中和分岐命令の場合 (1) を例にとると、図 1 は、分岐無 (a) の時ディレイドスロットを無視し、分岐有 (b) の時ディレイドスロットを実行することを示している。

2.3 複数命令の同時実行による並列性

複数命令の同時実行による並列性の抽出は、3 つのオペレーション (S,M,P) で 40 ビットの命令フィールドの空がなくなるようにアセンブルすることである。ここで、オペレーションの組み合わせにより命令形式を分類すると図 2 の 4 種類に大別できる。つまり、並列性の抽出は、TYPE1 から TYPE3 の命令形式の場合に、同時実行可能な M や P を、この命令フィールドに割り当てることである。

図 2 において、TYPE1 の S は、8 ビットの分岐命令

*Fine Grain Parallelism of Parallel Inference Machine PIM/i
Kenji KATO, Masatoshi SATO, Koichi TAKEDA, Teruhiko OOHARA,
Oki Electric Industry Co., Ltd.

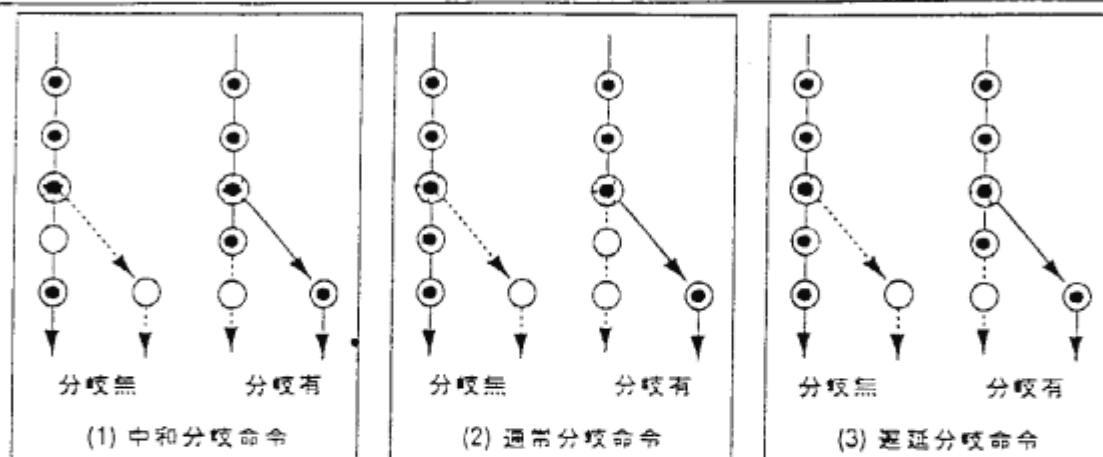


図 1: 分岐命令

TYPE1	S	M	P
TYPE2	S	(C)	P
TYPE3	S		P
TYPE4	S		

図 2: 命令形式

(6 ビットが符号付き相対分岐先を持つ) を示し、TYPE3 の S は、16 ビットの分岐命令 (5 ビットで間接分岐用レジスタ番号を指定) を示す。TYPE4 の S は、40 ビットの分岐命令で 30 ビットの分岐先アドレスを保持する。TYPE2 の C は、分岐時の条件を指定している。

3 考察

上記の並列性を考察するために、幾つかの KL1 プログラム (QK,QLAY,BUP) をシミュレータ上で実行し、データを収集した。この時のコード/データキャッシュの大きさは実験と同じ 32K ワードとした。ヒット率は、コードについては 99.9% 程度、データについては 97.5% 程度であった。また、プログラムコードのサイズは、ユーザプログラムは 1 から 1.4 K ワード、ランタイムルーチンは 4 K ワードであった。

3.1 パイプラインによる並列性

パイプラインにどのくらい空きができたかを知るために、キャッシュミス (コード/データ) のためにパイプラインが空きになった時間と分岐時にパイプラインが空きになった時間を比較する。ここで、ret/T はデータキャッシュミス時に要した時間/実行時間を表す。ref/T はコードキャッシュミスの場合、neu/T は分岐の場合を表す。

表 1 より、パイプラインを空にしている主な原因は、分岐命令によることが分かる。また、分岐命令の種類別の実行数は、通常分岐命令が 8 割を占めていた。一方、分岐命

表 1: パイプラインの空き時間 (%)

	ret/T	ref/T	neu/T
QK	4.36	0.03	15.52
QLAY	3.69	0.04	15.54
BUP2	3.87	3.94	12.25

表 2: 命令形式の出現度 (%)

	TYPE1	TYPE2-3	TYPE4
QK	86.9	5.9	7.2
QLAY	85.6	6.2	8.3
BUP2	81.4	9.4	9.2

令の 5 割程度が無駄なサイクルとならなかった。

また、コードサイズが小さかったためかランタイムを GM に移した場合はほとんど見られなかった。

3.2 複数命令の同時実行による並列性

命令形式別の出現度を表 2 に示す。この表より、TYPE1 が主に使用されることがわかる。また、同時に実行された命令数は 1 サイクル当たり 1.3 から 1.4 程度であった。

謝辞

日頃、御助言を頂く (財) 新世代コンピュータ技術開発機構 (ICOT) 第一研究室、および、沖電気の PIM 担当者諸氏に感謝する。

参考文献

- [1] A. Goto et al. Overview of the Parallel Inference Machine Architecture (PIM). In FGCS 1988, pp 203 - 229, Nov. 1988.
- [2] 大原他: 並列推論マシン PIM/i の概要、情報処理学会第 40 回全国大会, pp 1185-1186(1990).