

並列推論マシン PIM/i における分岐特性

佐藤正俊、武田浩一、大原輝彦

masatosi@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 のターゲット言語は、記号処理向けの並列論理型言語 KL1 である。PIM/i プロセッサは、RISC・アーキテクチャ、タグ・アーキテクチャ、パイプライン・アーキテクチャ、LIW・アーキテクチャを融合したアーキテクチャを採用し、記号処理言語の高速化を狙っている。

記号処理は、データタイプが実行時に動的に決まり、かつ、データタイプ毎の処理量にバラツキがあることが知られている。これは、記号処理では分岐処理の頻度が大きく、分岐処理の影響を少なくすることが重要であることを示している。また、一般的に RISC アプローチは、CISC アプローチに比較して 1 命令当たりの機能が低いため、基本ブロックが大きくなり、パイプライン制御での遅延分岐スロットの有効利用の可能性が高まる。しかし、我々のアプローチ (LIW・アーキテクチャ) のように、命令レベルの並列性を上げることは、遅延分岐スロットの有効利用の可能性を下げる意味を持つ。このため、並列推論マシン PIM/i における分岐特性を把握することは、重要である。

本稿では、PIM/i プロセッサにおける分岐処理の設計方針と分岐の振る舞いについて、シミュレータでのデータに基づいて述べる。ここで、KL1 は KL1b を経て PIM/i の機械語に変換し [3]、データは PIM/i のレジスタトランスファレベルのシミュレータ [4] で収集した。

2 分岐処理の設計方針

我々は、PIM/i プロセッサでの分岐命令の設計方針を以下のように設定した。

- LIW に合わせた分岐命令長の選択
- 1 遅延スロットとスカッシングによる遅延分岐
- 1 命令 / 条件コード無しの場合分岐

2.1 分岐命令長

40 ビットの命令フィールドを有効に使用するために、分岐操作 (Sop) を 3 種類に分け、残りのフィールドをメモ

リ操作 (Mop)、演算操作 (Pop) に割り当てた。ここで、3 種類の Sop とは、6 ビットが符号付き相対分岐先を持つ 8 ビット長 Sop、5 ビットの間接分岐用レジスタ番号または 8 ビットの分岐時の条件 (C) を含む 16 ビット長 Sop、30 ビットの分岐先アドレスを保持する 40 ビット長 Sop である。

2.2 遅延分岐

遅延分岐スロットの利用率とパイプライン設計を考慮し、パイプラインを 3 段とし、遅延分岐スロット数を 1 つにした。このように、比較的短いパイプライン段数を採用することによって、動的に変化するデータタイプに応じた分岐処理時のパイプラインの乱れによる影響が軽減され、また、遅延分岐スロットを有効に利用できる¹。

また、遅延分岐スロットを利用できない場合、コードサイズの静的な増加が遅延分岐のコストとなる。これに対しては、分岐方向でない命令を動的に nop 化する (スカッシング) 命令を遅延分岐命令に追加し、静的なコードサイズの増加を抑えることにした。

2.3 条件分岐

条件分岐の実現は、1 命令で、条件コード無しで実現することとした。条件分岐処理に関わる命令数は、条件分岐処理を 1 命令で実現するか、または分岐の成否を決める TEST や COMPARE 等の命令と実際に分岐する BRANCH 命令との 2 命令で実現するか 2 つの選択がある。1 命令では、複雑な条件コードの実装から逃れられる点で多くのメリットがあるが、[5] によると、パイプラインを構成する上で以下のデメリットがある。

パイプラインを 1 命令で構成する場合の分岐処理は、(1) 命令デコード、(2) 比較するオペランドの読み込み、(3) 比較、(4) 分岐の決定の 4 サイクルを要し、パイプラインに 2 サイクルの空きが生じる。一方、2 命令での構成は、比較するオペランドの読み込みと比較がパイプライン上でオーバーラップでき、分岐は 3 サイクルで完了でき、パイプラインの空きは 1 サイクル削減できる。つまり、1 命令での実現が 1 サイクル不利である。

¹この利用率は遅延分岐スロット数に大きく依存し、スロット数が 1 つの場合の利用率が高く、スロット数が 3 つの場合は利用率がかなり落ちる結果がある [5]。

* Branch Behavior in Parallel Inference Machine PIM/i
Masatoshi SATO, Koichi TAKEDA, Teruhiko OOHARA,
Oki Electric Industry Co., Ltd.

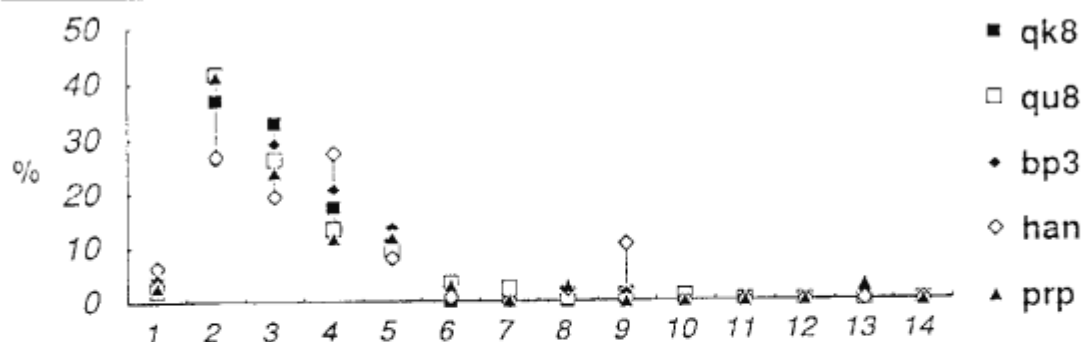


図 1: 分岐距離の分布

表 1: 遅延分岐スロットの使用率 (%)

	qk8	qu8	prp	bp3	han	av.
使用率	56.6	61.5	63.4	69.9	81.6	66.6

ここで、我々は、命令デコードと比較するオペランドの読み込みを同一ステージで実現するようにパイプラインを設計することで、パイプラインの空きを1サイクルのみで1命令分岐を実現した。これにより、2命令での実行サイクルと同じサイクルで条件コードを無くすることができ、コンパイラの設計、LSI設計時のコンディションコード用のバスの追加、パイプラインでの同期問題の考慮、コンディションコードを変化させない命令の設計等の条件コードから引き起こされる複雑さを回避した。

3 PIM/iにおける分岐特性

PIM/iにおける分岐特性の評価では、5つのベンチマーク (qk8, qu8, prp, bp3, han) を用いた。

図1に分岐距離 N の分布を示す。ここで、横軸は $\log_2 N$ を、縦軸はベンチマーク毎の全実行命令数に対する出現の割合を示している。これによると、分岐のほぼ9割以上が6ビットの符号付き相対分岐命令の範囲内であり、命令フィールドを他の操作に割り当てることは有効である。

表1に、遅延分岐スロットの使用率を示す。使用率とは、遅延分岐スロット中でスカッシュせずに使用した率である。これによると、2/3の遅延分岐スロットを有効に使用しており、遅延分岐命令を有効に使用していると言える。

表2に、スカッシングを考慮して導入した3つの分岐命令の分岐命令に対する使用率を示す。ここで、遅延分岐命令とは分岐の有無でスカッシュせず、通常分岐命令とは分岐時に遅延分岐スロットをスカッシュし、中和分岐命令とは分岐無し時に遅延分岐スロットをスカッシュする命令である。これによると、通常分岐命令が分岐命令中2/3程度であり、最も多く使用されていることが判る。ここで、通常分岐命令での遅延分岐スロットをスカッシングする割合は、ほぼ1/2程度であった。

分岐命令の種類を、表3に示す。ここでは、全実行命令

表 2: 遅延分岐の使用率 (%)

	qk8	qu8	prp	bp3	han	av.
分岐率	40.1	27.8	44.5	38.4	30.0	32.5
通常	70.7	68.3	58.3	65.9	55.3	63.7
遅延	25.4	26.8	36.0	26.6	28.5	28.6
中和	3.9	4.9	5.7	7.6	16.2	7.7

表 3: 分岐の種類 (%)

	qk8	qu8	prp	bp3	han	av.
無条件	9.3	6.4	12.0	11.4	10.1	9.8
条件	11.6	10.0	14.7	9.5	4.4	10.0
タグ	19.2	11.4	17.8	17.5	15.5	12.7

に対する無条件、条件、タグ分岐の割合を示している。これによると、タグ分岐が最も多く、タグ条件分岐命令は重要である。

4 おわりに

PIM/i プロセッサにおける分岐処理の設計方針と分岐の振る舞いについて述べた。

最後に、日頃、御助言を頂く 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 208 - 229, Nov. 1988.
- [2] 大原他: 並列推論マシン PIM/i の概要、情報全大第 40 回、pp 1185-1186(1990).
- [3] 佐藤他: 並列推論マシン PIM/i の開発支援環境 - 新訳系 -, 情報全大第 41 回、pp 5-371(1990).
- [4] 吉田他: 並列推論マシン PIM/i の開発支援環境 - シミュレーター -, 情報全大第 41 回、pp 5-367(1990).
- [5] J.A. DeRosa and H.M. Levy. An Evaluation of Branch Architectures. In 14th International Symposium on Computer Arch. May 1987.