

並列推論マシン P I M / i の
要素プロセッサのアーキテクチャ

武田浩一 大原輝彦 佐藤正樹
沖電気工業株式会社

The Architecture of a Processing Element of PIM/i
Koichi TAKEDA, Teruhiko OOHARA, Masaki SATO
Oki Electric Industry Co., Ltd.

1 はじめに

我々は、第5世代コンピュータ・プロジェクトの一環として並列推論マシンPIM/i⁽¹⁾の研究開発を行っている。

本稿では、PIM/iの要素プロセッサとして新規に設計を行った核言語第一版(KL1)向きプロセッサのアーキテクチャの概要について述べる。

2 設計方針

KL1で書かれたプログラムは、抽象KL1マシンの命令セットKL1-B⁽²⁾の命令列にコンパイルされる。KL1-B命令には複雑な高機能命令が存在するが、各命令をマイクロプログラムによって解釈実行するのではなく、低レベルの機械命令列に展開して実行する方式を採用することにした。これは次の理由による。

- (1) マイクロプログラムへのディスパッチ時間が節約できる。
- (2) 低レベルの機械命令列の最適化を行うことができる。
- (3) バイブライン処理方式などの高速化技術が容易に利用できる。

また、KL1-B命令では、Lisp、Prolog等の処理を行うときと同じように、データタイプの動的判定を行っている。したがって、このためのアーキテクチャ支援が必要である。

以上のことから、KL1を効率良く実行させるために、タグアーキテクチャを採用するとともに、RISC型命令セットを採用し、バイブライン処理、複数命令の並列実行、遅延分岐命令の利用などで処理の高速化を図った。

3 特徴

以下に要素プロセッサの特徴を挙げる。

3. 1 タグアーキテクチャ

KL1-B命令では実行時にデータタイプの判定と操作を行うため、ハードウェアにデータタイプを保持する機構およびデータタイプの判定と操作を支援する機能が必要となる。このため、タグアーキテクチャを採用した。汎用レジスタおよびメモリは、図1に示すように1語が40ビットタグ付きデータ（タグ8ビット、値フィールド32ビット）を格納する。このタグの操作とデータタイプの判定のために次のような命令を用意した。

(1) タグ判定分岐命令

タグ条件にしたがい、相対分岐する。

(2) タグ操作命令

タグの、任意の連続する数ビットを抽出したり、連続する数ビットに任意の値をセットできる。

3. 2 RISC型命令セット

KL1-B命令には、複雑な高機能命令が存在するが、レジスタ間転送、即値の代入、加減算、メモリアクセス、分岐などの単純な命令の組合せで実現される。このことから、命令セットを単純なものにし、各命令を高速化することが有効であると考え、RISC型命令セットを採用した。すなわち、次のような特徴を持つ。

(1) 命令長は40ビット固定長である。

(2) 32語×40ビットの汎用レジスタをもつ。

(3) 演算は3アドレス方式でレジスタ間のみに限られる。

(4) メモリアクセスはレジスタ-メモリ間の転送のみに限られる。

また、すべての命令は、図2に示すように3段パイプラインで、ハードワイヤードロジック制御により1サイクル毎に実行される。このため、命令コードを毎サイクル供給する必要がある。そこで、デー

タのアクセスが命令の供給を妨げることをないように、データバスとコードバスを分離したハーバードアーキテクチャを採用した。

この方式では、個々の命令の高速化によってプログラムの実行が高速化されるほか、コードの最適化を行うことによってプログラムの高速化が可能である。

3.3 複数命令の同時実行

KL 1-B 命令を展開した低レベルの機械命令列の最適化を行う場合、いくつかの命令が同時に実行可能である場合がある。例えば、演算命令と同時に被演算データのデータタイプによるタグ判定分岐命令が実行できる場合などが考えられる。このような命令をひとつの複合命令として実装するという選択もあるが、コンパイラによる最適化が行いやすい方法として、命令レベルの並列性を導入し、複数の命令を同時に実行できるようにして高速化を図ることにした。命令は図3に示すように3つのフィールドに分かれており、それぞれ命令を指定できる。以下に示すように最大3命令の同時実行が可能である。

(1) 分岐とメモリアクセスと演算

近傍への分岐は、メモリアクセス、演算と同時に実行できる。

(2) タグの判定と演算

演算を行っているデータと、そのタグの判定が同時に実行できる。

(3) タグの設定と値フィールドの演算

値フィールドの演算と同時に、その演算結果に即値のタグを設定できる。

3.4 バイプラインの乱れと分岐命令

命令の実行にパイプライン方式を採用しているため、分岐命令の実行後、分岐先命令を実行できるようになるまでに1サイクルかかる。そのため、いくつかのRISC型プロセッサでは、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、いわゆる遅延分岐命令が用いられている。しかし、ディレイドスロットで有効な命令を実行できない場合、分岐しないときでも無駄なサイクルを費や

すことになる。そこで、分岐の有無でディレイドスロットの命令を実行するか否かを制御する3種類の分岐命令を用意した。

(1) 遅延分岐命令

分岐するときも、分岐しないときも、ディレイドスロットの命令を実行する。

(2) 通常分岐命令

分岐するとき、ディレイドスロットの命令を無効化し、分岐しないとき、ディレイドスロットの命令を実行する。

(3) 中和分岐命令

分岐するとき、ディレイドスロットの命令を実行し、分岐しないとき、ディレイドスロットの命令を無効化する。

これらの命令を、分岐が起きやすいかどうかを予測して使い分け、コードを最適化することにより、高速化が可能である。例えば、(2)は例外的な条件で分岐するときには有効である。(3)はループを構成するときには有効である。

4 おわりに

並列推論マシンPIM/iの要素プロセッサのアーキテクチャについて述べた。このプロセッサのLSIの設計および試作はすでに完了しており、現在ハードウェアの評価中である。評価結果については別の機会に譲る。

謝辞

日頃助言をいただくICOT第4研究室、および沖電気のPIM担当諸氏に感謝する。

参考文献

- [1] 大原他：並列推論マシンPIM/iの概要
本大会予稿集，1990
- [2] Y.Kimura, et al: An Abstract KL1 Machine
and its Instruction Set.
In Proceeding of the 1987 SLP, 1987.

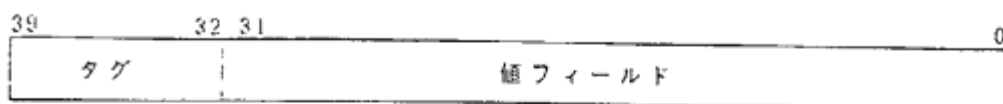
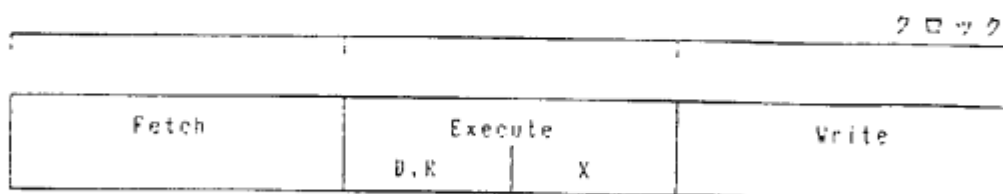
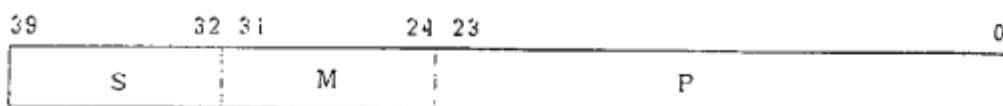


図 1 ！語の構成



Fetch: 命令フェッチステージ
 Execute: 実行ステージ
 Write: レジスタ、メモリ書き込みステージ
 D: デコード
 R: レジスタ読みだし
 X: 演算、分岐先計算

図 2 バイブライン構造



S: シーケンサフィールド
 分岐命令などを指定する
 M: メモリオペレーションフィールド
 メモリアクセス命令などを指定する
 P: プロセッサフィールド
 演算命令などを指定する

図 3 命令形式