

並列推論マシンPIM/pの要素プロセッサにおける 分岐機能の高速化のためのアーキテクチャ

篠木 剛¹、久門 耕一¹、安里 彰¹

服部 彰¹、近山 隆²、後藤 厚宏²

(¹富士通株式会社、²新世代コンピュータ技術開発機構)

1. はじめに

我々は、第5世代プロジェクトの一環として、数百台規模の要素プロセッサ(以下PEと呼ぶ)をもつ並列推論マシンPIM/pの開発を行っている。PEは、核言語KL1向けの専用アーキテクチャを新規設計し、PEの単体性能向上のためにパイプライン方式を採用することとした。本報告では、PEのアーキテクチャの、特に、条件判定機能や分岐機能について述べる。

2. データ型判定機能

KL1は、LispやPrologなどと同様に、変数のデータ型が動的に変更されるため、ハードウェアでのデータ型判定のサポートが必要である。そのため、タグアーキテクチャを採用し、汎用レジスタは1語がタグ8bit+データ32bitから構成することとした。また、以下のようなタグ判定用条件分岐命令を用意することにした。

jumpif <タグ判定条件>, <分岐先PC相対アドレス>
<タグ判定条件>には、汎用レジスタのタグ8bitと命令中の即値8bitとの間で次のような演算または各演算の否定演算が指定でき、これによってタグ判定を行う。

XOR: 各ビット間でXORを行った結果すべて1であるか

AND: 各ビット間でANDを行った結果1つ以上1があるか

OR: 各ビット間でORを行った結果すべて1であるか

XORはタグが即値で指定された値かの判定に用い、ANDは即値で指定されたビットがすべて0か、ORは即値で指定されたビットがすべて1かの判定を行うことができる(図1)。

図1 タグ判定機能(AND, OR)

タグの指定ビットがすべて0か	
レジスタのタグ	0xx0xxx0
即値tag	10010001 (AND)
	00000000
タグの指定ビットがすべて1か	
レジスタのタグ	1xx1xxx1
即値tag	01101110 (OR)
	11111111

3. パイプライン実行方式と分岐命令

命令実行ユニットの1チップ化のために、演算命令はレジスタ間に限り、主記憶アクセスはレジスタとの転送に限るいわゆるRISCの命令セットとすることにした(表1)。また、各命令は4段のパイプラインで実行され(表2)、1サイクル毎に命令をパイプラインに投入する1サイクルパイプ方式とすることとした(図2)。一般に、パイプライン計算機では、パイプラインブレイクにより分岐命令の実行は遅くなるが、本PEでも4サイクルの実行時間がかかる(図3)。

表1 命令の基本

演算命令	add <Ri>, <Rj>, <Rk>
メモリアクセス命令	read <Ri>, <offset>(<Rj>)
分岐命令	jumpif <条件>, <PC相対offset>

表2 パイプライン構造

	演算命令	メモリアクセス命令	分岐命令
D	Decode	Decode	Decode
A		Register read	Register read
		Operand address計算	分岐先命令address計算
T	Register read	Cache access (Address 変換)	Cache access (Address 変換)
B	ALU Register write	Cache access Register access	Cache access 条件判定 分岐先命令受け入れ

図2 1サイクルパイプ

	時間				
命令1;	D	A	T	B	...
命令2;		D	A	T	B
命令3;			D	A	T B

図3 分岐命令の実行

	時間				
分岐命令;	D	A	T	B	...
分岐先命令;				D	A T B

The Architecture on Branch of a Processor Element of PIM/p
Tsuyoshi SHINOBU¹, Kouichi KUMON¹, Akira ASATO¹,
Akira HATTORI¹, Takashi CHIKAYAMA², Atsuhiko GOTO²
¹FUJITSU Limited, ²ICOT

4. 動的マクロ機構

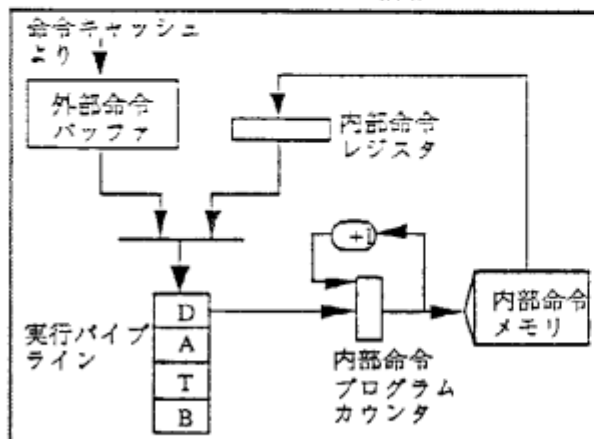
核言語KL1はコンパイラによって抽象命令セットであるKL1/Bの命令列に展開される。このKL1/Bの各命令をPEで実行するためには、2つの方法が考えられる。1つは、KL1/Bの各命令をサブルーチン呼出しとして共通ルーチンと呼び出す方法であり、もう1つは、KL1/Bの各命令をさらにマシン命令列に展開して実行する方法である。前者は、呼出し／復帰を分岐命令で行うため、そのためのオーバーヘッドが大きいという欠点を持ち、後者は、オブジェクトコードが大きくなって主記憶からのプログラムのフェッチ量が増大し、いわゆるフォンノイマンボトルネックになるという欠点をもつ。そこで我々は、マシン命令に展開すると大きなステップ数になってしまうKL1/Bの命令はサブルーチンとすることにし、少ないステップ数でできるKL1/Bの命令は展開することにした。さらに、前者のサブルーチンの呼出し／復帰のオーバーヘッドを小さくするための以下に説明するような機構(動的マクロ機構)を考えた。

分岐命令によるサブルーチン呼出しに時間がかかるのは、分岐先命令の、実効アドレスの計算やキャッシュアドレスへの変換、さらにはキャッシュミス時の主記憶からのフェッチが原因である。そこで、PE内にサブルーチン本体を格納する高速メモリ(内部メモリ)を置くことにし、サブルーチンと呼び出す命令(マクロ命令と呼ぶ)にはエントリアドレスを直接指定することにした。

mcall <entry address>, <レジスタ番号>...

マクロ命令の呼出し機構は、図4のように構成した。その結果、2サイクルでマクロ本体を呼び出すことができる。また、内部メモリ中の各命令にはサブルーチンの終了を示すビットが付随しており、さらに、外部命令バッファにはマクロ呼出し命令の次の命令が格入されているため、0サイクルでマクロ本体からの復帰が可能である。

図4 マクロ呼び出し機構



能である。分岐命令によってサブルーチンを構成すると、呼出し／復帰に各々4サイクルで合計8サイクルかかるのに対して、マクロ機構では、呼出し／復帰のオーバーヘッドが合計2サイクルに短縮することができた。また、KL1/B命令は実行時のデータ型判定を含むものも多いため、次のようなタグ判定による条件マクロ呼出し命令も用意することにした。

mcallif <タグ判定条件>, <entry address>, <レジスタ番号>...

5. スキップ命令

KL1処理系では、変数の値を読む度に変数セルの値がポインタ(reference)であるか確認し、referenceであればreferenceでなくなるまでポインタをたぐる必要がある。その確認処理のために条件サブルーチン呼出しが頻繁に用いられるが、これは一般的なプロセッサでは不得手であり、次のようなコードになるのが普通であろう。

```
jumpif <条件>, LAB;
call Deref;
LAB: -----
```

4サイクルかかる

この場合、サブルーチンと呼び出さない場合も分岐が発生してしまい、パイプラインブレイクが起こってしまう。PIM/pのPEでは、前述のように4サイクルかかる。このような場合にパイプラインブレイクを小さくするための簡単な方法として、次のようなスキップ命令を導入することにした。

skipif <タグ判定条件>

skipif 命令は、<タグ判定条件>が成立時に次の命令をNOP(NoOperation)命令として実行する。前の例は次のようになり、呼び出さない場合には2サイクルで済む。

```
skipif <条件>; 1サイクル
call Deref; 1サイクル (NOP命令として実行)
```

6. おわりに

PIM/pの要素プロセッサにおける分岐機能の高速化のためのアーキテクチャとして、データ型判定、マクロ機構、条件スキップ命令について述べた。現在、LSI開発を中心とし、PIM/pシステム全体の開発を進めている。

謝辞 B項、御指導いただくICOT 内田 第4研究室長、富士通研究所 林 人工知能研究部長、並びに、ICOTと富士通のPIM研究開発メンバーに感謝いたします。

参考文献 [1] Shinogi他: Macro-call Instruction for the Efficient KL1 Implementation on PIM, Proc. of International Conf. on Fifth Generation Comp. Systems, 1988; [2] 藤本他: 並列推論マシンPIM/pの要素プロセッサのアーキテクチャ, 情報処理学会37回全国大会, 1988; [3] 後藤他: 並列推論マシンPIM/pの概要, 情報処理学会37回全国大会, 1988; [4] 服部他: 並列推論マシンPIM/pのアーキテクチャ, 並列処理シンポジウムJSPP89, 1989