

7D-8 並列オブジェクト指向トータルアーキテクチャ A-NET - PEにおけるマイクロプログラムの作成とその評価 -

岩本善行 寺岡孝司 吉永努 馬場敬信

hackun, zoon, yoshi, baba@deep.infor.utsunomiya-u.ac.jp

宇都宮大学工学部

1. はじめに

A-NET[1]では並列オブジェクト指向言語A-NETL[2]を設計し、シミュレーションを行ってきたが、現在これを実行するためにプロトタイプのア-NET計算機を試作中である。

各PEは、水平型76ビットのマイクロプログラムによって制御され、2段のパイプライン制御を行う。

PEを製作するにあたり、以下のような設計方針を立てた。

(1)メッセージ送信や構造体操作命令などのA-NETL指向高機能可変長命令セットとする。

(2)動的データ型付け、ガーベジコレクション(GC)、OS支援などのためのタグ付きアーキテクチャとする。

(3)オブジェクトの動的ロード、再配置を容易にするため、オペランドのアドレス指定はベースアドレス指定方式、分岐先アドレス指定は相対アドレス指定とする。

本稿では、はじめにPEのアーキテクチャについて述べ、それを実現するマイクロプログラムの作成とその評価について述べる。

2. PEのアーキテクチャ

2.1 機械命令定義

PEの機械命令として、16種79命令を定義する。これらは、1語40ビットのローカルメモリ上に置かれるが、各命令はバイト境界を持ち、オペコード1バイト、オペランド最大15バイトまでの可変長命令で、その長さはオペランド中に含まれる。

機械命令は、大きくデータ移動、構造体操作、メッセージ送受信、演算、条件/無条件分岐、システム特権命令に分類できる。以下に特徴的な命令について述べる。

(1)構造体操作

A-NETLで定義されている構造体に対する置換、挿入、削除などの操作を1命令で実現する。

(2)メッセージ送受信

メッセージ送信は待ち合わせにより過去、現在、未来型があり、それぞれについてマルチキャストを定義する。また、オブジェクト、メソッドの生成、メッセージ受信なども1命令で実現する。

オペランド指定のベースアドレスとしては、一時変数ベースアドレス(TBA)、メッセージ引数ベースアドレス(PBA)、リテラルベースアドレス(LBA)、状態変数ベースアドレス(SBA)を用意し、また、即値も使用可能である。これらはオペランドタグによって指定される。

データは、整数や浮動小数点数、オブジェクト識別子などの1語からなる構造を持たないもの6種類と、配列やリスト、メソッドなどの複数語からなる構造を持つもの7種類を定義する。

2.2 マイクロレベルアーキテクチャ

図1にPEのハードウェア構成図を示す。全体は大きく分けて、制御部、データ処理部、メモリ部、命令フェッチ部、及び、割り込み制御部に分けられる。以下に特徴的なユニットについて述べる。

(1)命令前処理ユニット(IPU)

データはワード単位にアドレス指定されるのに対して、機械命令はバイト単位にアドレス指定される。このようなアドレス指定方式に対応するため、プログラムカウンタ(PC)を19ビットとし、上位16ビットをワードアドレス、下位3ビットを5進数で1語中のフィールド指定を行うバイトアドレスとする。命令前処理ユニットでは1語中に埋め込まれている命令コードとオペランドを取り出す。

(2)タグ処理ユニット(TPU)

データ処理を行う際には、データ型の比較も同時に行う必要がある。このようなタグ処理によるオーバーヘッドを緩和するため、タグ処理ユニットを設ける。タグ処理ユニットは、データフラグ、データタイプのセット、比較や移動などを行う。

3. マイクロ命令定義

図2に、マイクロ命令のフィールド構成を示す。

マイクロ命令は1語76ビットの水平型で、15のフィールドからなる。1マシンサイクルは125nsecで、2段パイプライン処理を行う。次に、各フィールドの簡単な説明をする。

①バスフィールド(IBO OBO B1 B2)

B0バスへの入出力、B1,B2バスへの入力を制御する。

②演算フィールド(ALU FPU)

ALUあるいはFPUでどのような演算をするのかを制御する。

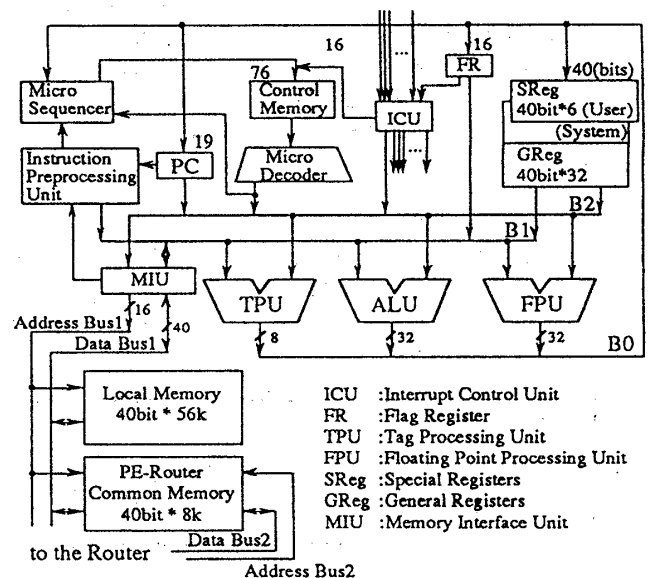


図1 PEのハードウェア構成

③タグ処理フィールド(TM BS DF DT)

B1,B2バス上のタグの未来フラグのチェック、データ型の比較、セットを行う。また、B1,B2バスのタグのB0バスへの出力を選択する。

④メモリフィールド(LM)

メモリに対する読み出し、書き込みを制御する。

⑤命令前処理ユニットフィールド(IPU)

IPUを制御し、オペコード、オペランドをシーケンサ、あるいはB1バスに出力する。

⑥順序制御フィールド(TS SEQ)

マイクロシーケンサへのテスト条件の入力選択と、順序制御を行う。

⑦リテラルフィールド(LIT)

16ビットのリテラルを与え、演算に使用する定数や分岐のためのマイクロアドレスを与える。

4. マイクロプログラム

4.1 作成支援環境

マイクロプログラムの作成を支援するため、プロシージャ単位に記述することができるマイクロ言語を定義し、アセンブラ、リンクをワークステーション上に作成した。これにより、マイクロプログラムは階層的な構造になり、機械語命令フェッチと256方向への多分岐命令のプロシージャと、各機械語命令ごとに1つのメインプロシージャといくつかのサブプロシージャによって構成される。

また、デバッグ支援のためにマイクロプログラムシミュレータを作成し、各機械命令についてマイクロプログラムを実際に実行して検証を行った。

4.2 マイクロプログラムの特徴

以下にA-NET PEに特徴的な処理についてそのマイクロプログラムの作成方法について述べる。

①オペランドタグ

オペランドは、4種のベースアドレスおよび即値が指定可能であり、即値以外はベースアドレスレジスタを参照し値を取り出さなければならない。また、機械語命令によりどのベースアドレスを参照できるかも異なる。これはオペランド取得時に区別され、さらにオペランドタグによる8方向の多分岐命令によって値の取り出しを行ったり、オペランドタグエラーとしてエラー処理に分岐される。

②可変長命令

メッセージ送信命令に使用される可変長命令は、そのオペランドの長さをオペコード中に含むが、オペコードはバス上に出力されず、シーケンサによって256方向の多分岐のオフセットに使用されるだけである。よってこの分岐先でオペランドの長さを与え

フィールド	OB0	B1	B2	ALU FPU	LM	DF	DT	IPU	TS	SEQ	LIT	
使用効率	58.1	61.8	36.9	45.4	7.9	16.5	7.8	10.4	0.7	15.8	55.2	63.9

図3 PE各資源の使用効率 (unit: %)

ニーモニック	Min	Typ	Max	ニーモニック	Min	Typ	Max
NOP		5		DIV	262		699
MV	14		16	SHIFT	22		26
ADD	25		32	STRUCT	17		70~
MULT	25		48	BRANCH	13		51

図4 マシンサイクル数 (unit: cycles)

なければならない。

③メッセージ送信命令

メッセージ送信命令の処理はメッセージバケットを作成し、ルータとの共有メモリ内の出力キューに書き込み、ルータに割り込みをかける。

4.3 評価

以上によって作成されたマイクロプログラムを評価する基準としてサイズ、各種資源の使用効率、実行に必要なマシンサイクル数が挙げられる。

サイズは、全マイクロプログラムサイズが2699Wordsでそのうち実効サイズが2491Words(93.3%)である。この値は2段パイプライン制御の性質上100%にすることが可能である。

各種資源の静的使用効率は、マイクロフィールドのサイズの使用効率から求めたものを、図3に示す。

マシンサイクル数は、一部を図4に示す。オペランド数、タグの種類などによりかなりの変化があり、特にメッセージ送信などは算出不可能である。

5. おわりに

現在、マイクロプログラムの記述はほぼ終了し、より詳細の検証、評価、および最適化を行っている。また、これと並行してプリント基板の配線設計・作成を行っている。

今後は、2ノードからなるプロトタイプマシンを試作し、実機上でのマイクロプログラムの動作検証を行う予定である。

参考文献

- [1] T.Baba et al.: "A Parallel Object-Oriented Total Architecture: A-NET", Proc.Supercomputing '90, pp.278-285(1990).
- [2] T.Yoshinaga and T.Baba: "A Parallel Object-Oriented Language A-NETL and Its Programming Environment", Proc.COMPSAC '91, pp.189-196 (1991).
- [3] 吉永 他: "A-NET計算機のノードプロセッサとその実行方式", JSPP '91, pp.189-196 (1991).
- [4] 鈴木 他: "並列オブジェクト指向トータルアーキテクチャA-NETの要素プロセッサ", 情報処理学会第87回計算機アーキテクチャ研究会報告, 91-ARC-87-4(1991).

フィールド名	バス	演算	メモリ	タグ	IPU	制御部	リテラル
フィールド	IB0 OB0 B1 B2	ALU FPU	LM	TM BS DF DT	IPU	TS SEQ	LIT
ビット長	2 6 6 6	9 3	2	2 1 4 5	4	4 6	16

(全76 bits)

図2 マイクロ命令のフィールド構成