

並列オブジェクト指向トータルアーキテクチャ A-NET

6P-7

— PE のアーキテクチャ —

寺岡孝司 鈴木充 吉永努 馬場敬信

(宇都宮大学工学部)

1. はじめに

我々は、並列オブジェクト指向を核概念とした、トータルアーキテクチャ A-NET (Actors NETWORK) [1,2] の開発を行っている。A-NET では、処理要素 (PE) とルータの対からなるノードプロセッサを基本的な構成要素とし、これを多数結合した高並列ネットワーク計算機的设计・試作をおこなっている。

本稿では、おもに A-NET における PE のアーキテクチャについて述べる。

2. マクロアーキテクチャ

最初に、マクロアーキテクチャの特徴を述べる。

① 高機能命令セット

機械命令セットは、A-NET L [4] 指向の高機能命令セットとする。A-NET L で記述されるメッセージ送信や、プリミティブメソッドはひとつの機械命令に対応する。命令によってオペランド数が異なるため、命令形式はバイト境界を持つ可変長命令となっている。

② タグ付きアーキテクチャ

各データに、データタイプや、並列に実行されるオブジェクト間の同期をとるためのフラグ、ガーベジコレクションを効率よく行なうためのフラグなどをまとめタグとして付加した、タグ付きアーキテクチャを採用している。

③ アドレッシング法

オブジェクトの動的ロードを考慮し、オペランドのアドレス指定はベースアドレス方式とし、分岐先アドレス指定は相対アドレス方式とする。ベースレジスタとして、状態変数ベースレジスタ (SBR)、一時変数ベースレジスタ (TBR)、リテラルベースレジスタ (LBR) がある。

3. PE における処理

PE のハードウェアの構成を図 1 に示す。A-NET L プログラムの高速実行を目標として次のような処理を行う。

3.1 高機能命令の前処理

命令前処理ユニット (IPU) は、命令を含む 1 語

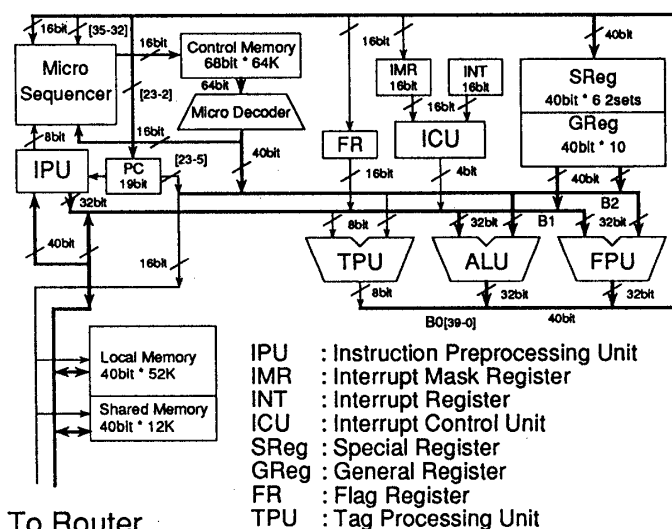


図1. PEのハードウェア構成

40ビットをローカルメモリより取り込み、その命令のオペコードを取り出しマイクロシーケンサに送る。また、その命令のオペランドの内容を示すタイプ部の削除、符号拡張、ベースレジスタ指定の処理を加え、シーケンサやバスに送り出す。

データはワードアドレスでアクセスされるのに対し、機械命令はバイトアドレスによって指定される。プログラムカウンタは、上位16ビットをワードアドレス、下位3ビットをワード中のフィールドを指定する5進数とすることによって、19ビットのバイトアドレスを示す。PCや分岐命令の飛び先を表すオペランドには、このバイトアドレスを用いる。

3.2 タグ付きアーキテクチャ支援

メモリ、レジスタ上の40ビット長のデータに対して、データ処理部 (ALU、FPU) では32ビット長のデータまでしか処理できない。このため、上位8ビットのタグデータを処理するためタグ処理ユニット (TPU) を用意する。

TPUの主な処理は、B1、B2の上位8ビットあ

A Parallel Object-Oriented Total Architecture A-NET
- Architecture of the PE -

Takashi TERAOKA, Mitsuru SUZUKI, Tsutomu YOSHINAGA, and Takanobu BABA
Utsunomiya University.

るいは下位8ビットデータのB0への移動や、データフラグのチェック、セット、リセット、データタイプのチェック、セット、B1とB2上のデータタイプの比較である。

TPUでのタグチェックと、データ処理部での2つのデータの演算を同時に行うことにより、タグ付きデータ処理のオーバーヘッドを緩和することができる。例えば整数か浮動小数点数か分からない2つのデータの加算を行う場合、TPUでタグ中のフラグチェックとデータタイプの比較をおこない、同時に、ALUとFPUで2つのデータの加算を行う。2つのデータの型が一致し、整数型である場合ALUの結果が、浮動小数点数である場合FPUの演算結果が取り出される。一致しなかった場合、整数型データをFPUで浮動小数点数型に変換して演算を行う。

また、データフラグの処理は、TPU内のデータフラグマスクで2つのデータのデータフラグの値をビット単位で論理積を取り、いずれかのデータフラグをB0に出力する。ユーザモードで、未来フラグが立っている場合、例外処理(future trap)を起動し、コンテキストチェンジを行う。

3.3 割り込み処理

システムモードで例外処理の実行中、割り込み処理の要求があると、不当命令や特権命令違反など直ちに受け付ける必要のあるものと、DMA開始やメッセージ読み込みなど、それを直ちに受け付ける必要がない場合がある。

このため、割り込み処理は、割り込みレジスタ(INT)に格納された割り込み信号に対し、割り込み制御ユニット(ICU)で、割り込みマスクレジスタ(IMR)の値によりマスクされるマスクブル割り込みと、マスクされないノンマスクブル割り込みの2レベルで行われる。

3.4 コンテキストチェンジ

メッセージの送受信や割り込み処理に伴い、コンテキストチェンジの要求が頻繁に発生し、これが処理のオーバーヘッドとなる。

これを緩和するため、システム用とユーザ用の2バンク構成となった特殊レジスタ群(SReg)を用意する。また、これにより1セットのマイクロ命令によるシステムとユーザの2つの特殊レジスタ群へのアクセスを可能にする。

4. マイクロアーキテクチャ

マイクロ命令は1語73ビット水平型で、図2に示されるようなフィールド構成となっている。次に各フィールドの制御内容について述べる。

① バス制御 (IB0、OB0、B1、B2)

B0への入出力、B1とB2への入力を、それぞれ、IB0、OB0、B1、B2フィールドで制御する。

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	5	5	5	9	3	2	2	1	4	5	4	4	6	16

(全73ビット)

図2. フィールド構成

② メモリ制御 (LM)

メモリに対する読み出し、書き込みサイクルの開始を制御する。

③ TPU制御 (TM、BS、DF、DT)

B1とB2のタグデータを認識し、フラグのチェック、セット等の処理を施し、B0に出力する。

④ IPU制御 (IPU)

IPU内の命令レジスタに格納されている1語からオペコード、オペランドを取り出し、シーケンサ、あるいは、B1バスへ出力する。

⑤ 演算制御 (ALU、FPU)

B1とB2の下位32ビットを入力として、ALUやFPUで演算指定を制御し、結果の出力は、IB0フィールドで指定されたユニットからバスB0の下位32ビットに出力する。

⑥ テスト制御 (TS)

テスト条件を選択し、その結果をシーケンサへの入力とする。

⑦ シーケンス制御 (SEQ)

マイクロプログラムの順序制御を行う。

⑧ リテラル (LIT)

16ビットのリテラルを定義し、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。

5. おわりに

現在、マイクロプログラムの記述が完了し、ハードウェアの詳細設計中である。今後、PEのプロトタイプの試作を行ない、A-NETLで記述したプログラムを実行し、A-NETのハードウェア及びソフトウェアの評価を行なう予定である。

参考文献

- [1] 馬場 他 : 「並列オブジェクト指向トータルアーキテクチャA-NET」、並列処理シンポジウム、JSPP '89、A4-1 (1989)
- [2] 吉永 他 : 「並列オブジェクト指向トータルアーキテクチャA-NETの開発方針」、情報処理学会第40回大会、1L-3 (1990)
- [3] 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におけるPEのハードウェア構成」、情報処理学会第40回大会、1L-4 (1990)
- [4] 岩本 他 : 「並列オブジェクト指向言語 A-NETLの言語処理系」、情報処理学会第38回大会、4P-1 (1989)