

並列オブジェクト指向トータルアーキテクチャA-NET

2L-4

— マイクロプログラムの実装 —

澤田東 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て、言語とアーキテクチャを統合的に設計したトータルアーキテクチャA-NETの開発を行なっている [1, 2]. これまでに16台のノードプロセッサが完成し、実際にその上で要素プロセッサ (PE), ルータおよびマイクロプログラムのデバッグを行ない、小規模のアプリケーションが動作している.

本稿では、ノードプロセッサを簡単に説明し、その後マイクロプログラムの実装とその評価について述べる.

2. ノードプロセッサ

ノードプロセッサは、メソッド実行を行なう PE とメッセージ送受信を行なうルータで構成される [3]. PE は1語76ビットの水平型マイクロ命令によって制御され、プロトタイプ的设计変更に対して柔軟に対応できる.

PEにはA-NETL指向の高機能命令セットが定義され、マイクロプログラムによって制御される. 命令セットは10種79命令からなり、基本的にA-NETLのメッセージ的に1対1に対応する. 表1にその例を示す.

表1: 命令セットとA-NETLの対応 (一部)

種類	ニーモニック	A-NETL記述
データ転送	mv des so	des=so
加算	add des so1 so2	des=so1+so2
論理算	blAnd des so1 so2	des=so1&&so2
配列操作	at des so1 so2	des=so1 at:so2
	atPt so1 so2 so3	so1 at: so2 put:so3
メッセージ送信	sndP obj sel so1...	obj sel: (so1,...)

PEの特徴的なユニットとして、タグ処理ユニット (TPU), プリフェッチとデコードを支援する命令前処理ユニット (IPU), システム/ユーザそれぞれのベースレジスタなどを格納する2セットの特殊レジスタなどがある.

3. マイクロプログラム

マイクロ命令は76ビット水平型で図1のように構成されており、それぞれ,ALUやTPU,シーケンサなどを制御する.

1マイクロ命令 (1ステップ) では、同時に複数のフィールドを制御することができ、これによりステップ数や語

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

図1: マイクロ命令フィールド構成

数を節約することができる. 図2, 図3に例を示す.*から次の*までが1ステップとなる. この例は,IPUフィールド以外のすべてのフィールドを制御している.ALUとFPUとで同時に加算を行なっているが、ここではALUの結果だけが有効であり,FPUの結果を要求する場合は次ステップで指定する必要がある. 参照するフラグは1ステップ前で立てておく.

```

*
:
if Z then jmp des_write
Greg.5 := Greg.1 add Greg.2
dflag Greg.2 Future
dtype Integer
*
    
```

図2: マイクロプログラムの記述例

バス				演算				タグ						
IB0	OB0	B1	B2	ALU	FPU	TM	BS	DF	DT	LM	IPU	TS	SEQ	LIT
ALU	GR.5	GR.1	GR.2	ADD	ADDF	B2.H	BUS2							

タグ		メモリ		IPU	制御部		リテラル
DF	DT	LM	IPU	TS	SEQ	LIT	
SET3	SETINT	FT	N	Z	BRCCCL	des_write	

図3: 記述例に対するフィールドごとのニーモニック

A-NETLはデータ型定義の必要ない言語であるため、機械命令もデータ型によらない総称的なものとなっている. そのため、データ型をオペランドの読み出し時にマイクロプログラムで検査する必要がある.

各機械命令を実行するマイクロプログラムでは、フェッチサイクルで命令コードを取り出し分岐を行ない、結果格納先、数値などのオペランドを取り出した後、演算、構造体操作、メッセージ作成などの各処理を行い、フェッチサイクルに戻る. また,GCもマイクロプログラムで実現する.

4. メッセージ送信

*A Parallel Object-Oriented Total Architecture A-NET - Implementation of the microprograms -,Akira SAWADA, Tsutomu YOSHINAGA,and Takanobu BABA,Utsunomiya University.

メッセージ送信は、マイクロプログラムで最終語に future flag を立てたメッセージを FIFO に書いた後、システム変数 QEND(0xFFFFE 番地) を最終語アドレス+1 の値に更新し、ルータに割り込みを発生させることで行なう。ルータはシステム変数 QTOP(0xFFFFD 番地) からメッセージを送信し、future flag の立っているアドレスを読み出すことによって割り込みをクリアする。

マイクロプログラムは、大きなメッセージの分割や FIFO の折り返しチェック、メッセージ通信量が増えルータにデータが溜った場合の QEND の QTOP 追い越しチェックなども行なう。

5. 評価

ここでは、実機上で実際に動作確認を終了したマイクロプログラムや、静的な TPU・IPU の効果などについて評価する。語数を表 2 に、代表的な機械命令に対するマイクロプログラムの動的ステップ数を表 3 に示す。

表 2: マイクロプログラムの語数

命令の種類	語数
通常命令	1,030
メッセージ	672
システム	684
その他	888
空アドレス	495
計	3,769

語数については、現在調整中であるためデバッグ用のものも含んでいる。空アドレスは多方向分岐を行う為のアドレス補正により生じるが、これはリンク時に調整することにより減少できる。ステップ数については、オペランドが即値かメモリ上のデータかにより異なるため、最小と最大の値を示す。基本的なレジスタ演算は 1 ステップで終了するが、ローカルメモリからのデータのロードやその型の検査を実行時に行う必要があるため、1 機械命令のステップ数は多くなっている。

表 3: 代表的な機械命令のステップ数

命令	最少	最大
mv	8	10
getType	8	10
ins	27	43
at	21	23
fst	11	12
addFst	31	33
Add	13	21
blAnd	12	20
sndP	65	

メッセージ送信命令 sndP のステップ数は、引数により大きく代わるが、最小では 65 ステップとなり、実行時間は 30MHz とすると約 10.8 μ s である。

TPU は、実行時のデータ型や同期フラグ検査のオーバヘッドを緩和するために設けた。TPU の効果を調べ

表 4: TPU の効果 (機械命令 Add の場合)

比較の種類	TPU 使用	TPU 未使用	TPU 未使用/使用
語数	23	32	1.60
ステップ数(平均)	19	27.3	1.44

るため、加算命令を例にとり、TPU 使用と未使用の場合の比較を行なった。結果を表 4 に示す。これより平均で語数は約 1.6 倍、ステップ数は約 1.44 倍になることが分かる。

IPU の効果を調べるため、IPU を使用しないマイクロプログラムを作成し、その場合のステップ数を調べた。その結果、IPU 使用時と比較し、命令フェッチで 3 ステップ、命令コード処理に 6 ステップ、オペランド 1 つにつき 10 ステップ余分に必要であった。オペランドが 3 つある整数加算を例にとると、IPU 使用時 17 ステップであったものが、未使用時には 17+3+6+10 \times 3=56 ステップとなり、IPU の顕著な効果が分かる。

次に、A-NETL で記述したユーザプログラムを実機で動作させ、マイクロプログラムの評価をする。用いた問題は、100 までの素数を求める問題である。

この実行結果は、機械命令 3,495 語、マイクロステップ数 104,231 となった。1 ステップ 5 クロックであり、1 クロックを 30MHz とすると実行時間は、17.37ms となる。

6. おわりに

現在、16 台のノードプロセッサを用いて、PE、ルータおよびマイクロプログラムのデバッグを行っており、8Queens などの GC が起こらない規模のアプリケーションプログラムが並列に動作している。

今後は、早急に GC 処理、評価ツールを完成させ、実機上での、より大規模なアプリケーションプログラムの実行、その並列度など詳しい性能評価を行っていく予定である。

謝辞

本研究は、一部文部省科学研究費、一般 (C)07680334、奨励 (A)07780225、および電気通信普及財団の援助による。

参考文献

- [1] 馬場, 吉永: “並列オブジェクト指向トータルアーキテクチャ A-NET における言語とアーキテクチャの統合”, 信学会論文誌, Vol.J75-D-I, No.8, pp.563-574(1992).
- [2] 馬場: “コンピュータアーキテクチャ”, オーム社, p.408(1994).
- [3] 吉永, 馬場: “並列オブジェクト指向トータルアーキテクチャ A-NET のノードプロセッサ”, 信学会論文誌, Vol.J79-D-I, No.2(掲載予定).