

PIEの構造メモリ試作ハードウェアの

5B-2

マイクロプログラム

平田 圭二[†] 角田 好隆^{††} 田中 英彦[†]([†] 東京大学 工学部、^{††} 富士通研究所)1. はじめに

我々は高並列推論エンジンPIEの研究、開発を行っている[1]。現在、構造メモリ試作ハードウェアのデバッグ中である。構造メモリ試作ハードウェアは推論ユニット部と構造メモリ部がネットワークで接続された構成になっている。本稿では、本試作ハードウェアの内、特に構造メモリ部のソフトウェアについて述べる。

2. 構造メモリ試作ハードウェア

構造メモリ部には、同一クロックで独立に動作するシーケンサ (Am2910A) が2つあり、各々により制御されるハードウェア部分をSHLF、SMAN と呼ぶ。SHLFとSMAN は別々の内部バスを持ち、そこから1つの共有メモリにアクセスを行う。

Lazy Fetch Network (LFN)上のデータを受信するハードウェアは常にネットワークのアドレス線をモニタしている。相手が自分のアドレスを送出したらストロープ信号のタイミングでデータ線からデータをFIFOに読み込む。FIFOは非同期に読み書きできるものを使用したので、上記の動作は構造メモリの内部動作とは全く独立に行える。またこの時、ネットワークからデータを受信したことを知らせるために、シーケンサに対する条件分岐信号の1つをactiveにする。ハードウェア内部の詳細については文献2を参照されたい。

3. 本試作機のソフトウェア3.1 マイクロプログラム

本試作ハードウェアのマイクロプログラムの1ステートメントは次のような構文になっている。マイクロアセンブラの作成を容易にするため、推論ユニット部と構造メモリ部のマイクロプログラムの構文は同一のものとした。

```
op1(arg11,...),...,opN(argN1,...);
```

op1 ~ opN : 操作名

ステートメント中での個々の操作はカンマ(,)で区切られ、1ステートメントは1マイクロステップで実行される。即ちステートメントは同時に実行すべき操作を指定する。ステ

ートメント間はセミコロン(;)で区切られる。ステートメントにはラベル付けが可能であり、ラベル名のかわりに数字を書くくと、そのマイクロ命令がWCSの何番地にロードされるかを指定できる。以下に構造メモリ部の操作として指定できるものをいくつか挙げる。

① movea(s, d1, ..., dN) $N \geq 0$

abusを経由してデータを転送する命令で、abusに対するソースとデスティネーションを指定する。s やd1~dNはレジスタやメモリの名称。デスティネーションは無くても良い。movedはdbus経由のデータ転送命令である。

例 movea(alu, lfr, sm)

これはabus経由で、alu からlfr とsmにデータを転送する命令である

② tag(arg) arg : abus, dbus

SMAN 側にあるTagCheckerに対してabus、dbusの内、どちらのバスのタグ部(上位8ビット)とタグ照合を行うかの指定。照合したいタグは予め、ホスト計算機によって設定しておかねばならない。

③ lfn(arg) arg : req, data, eod

LFNに要求を出し使用権を確保しデータを送出するという一連の処理を行う命令。reqはLFNを要求するために必要である。reqに対するackは、シーケンサへの条件分岐信号の1つとして入力される。dataを実行すると、その時点でdbus上に存在しているデータ(32bit)が送出される。実際にデータがLFN上に出るタイミングは、lfn(data)を実行した次のマイクロサイクルである。eodはLFNのeod線をactiveにし、相手にデータの送出が終了したことを知らせる。

④ alu(ins, bw, s, d, ...)

aluに対する操作の最初の4引数については、insが演算の指定、bwがバイト演算かワード演算かの指定、sとdが各々演算のソースとデスティネーションを表し、順序は固定である。第5引数以下は順不同で次のような引数が使われる。

ram(Num) : 演算のソースやデスティネーションがALUの内部レジスタファイルの場合、そのレジスタファイルの番地を指定する

⑤ seq(ins, ...)

シーケンサに対する操作の第1引数insは演算の指定であ

る。第2引数以下は、順不同で次のような引数が指定できる。

l(Label) : ジャンプ先のラベルや番地を指定

cn(Condition Signal) : どの条件で分岐するかの指定

alu や seq のニーモニックはAMDASHに準拠した [3]。

我々は上記のマイクロプログラムのためのマイクロアセンブラ及びシミュレータをC言語で作成し、マイクロプログラムの作成、デバッグに用いている。

3. 2 構造メモリの各操作のアルゴリズム

ここでは構造メモリで行う操作のアルゴリズムについて述べる。尚、推論ユニットが構造メモリに送信するコマンドパケットは、Ground Instance 格納、参照カウンタ(Reference Counter) 更新命令、空きアドレス(Empty Address) の要求のサブコマンドから成る。

① Lazy Fetch

SMLFはbusy wait で、LFBufにデータが到着していることを知らせる条件分岐信号がactiveになるのを待つ。LFBuf中のデータのタグ部を見てリストかベクタの判断を行い、各々のルーチンへ飛ぶ。LFR は自動インクリメント可能なので、メモリの読み出しは1語/1ステップで行える。リストノードの場合は9ステップ、サイズnのベクタノードの場合はn+8ステップで処理が終了する。

② 参照カウンタ(RC)の更新

SMAN はReference Count Queue(RCQ)から1つのコマンドを取り出し処理を完了する毎に、コマンドパケットが到着しているか否かの検査を行う。もしコマンドパケットが到着していない場合はRC命令の処理を行い、そうでない場合はコマンドパケットの処理に移る。従ってコマンドパケットの処理をしている間はRC命令の実行は中断する。RCの更新操作は必ずRCQ から取り出したコマンドによって行われる。零伝搬が生じた場合でも子ノードに対するRC命令は一旦RCQ にpushする。

③ Ground Instance(GI) の格納

格納すべきGIはSMの空きアドレスと共に送られて来る。リストノードが送られて来た場合、List Memory(LM) の所定の番地にGIを書き込み、RCの更新命令を発行しRCQ にpushしておく。これに対してベクタノードの場合は、GIと共に送られて来るアドレスはAddress Translation Table(ATT)の空きアドレスである。従ってVari-sized Cell Memory (VCM)の領域は、ベクタノードが送られて来た時点で動的に割り当てられる。VCM の空き領域はサイズ毎の自由リストで管理されている。リストリードの場合は6ステップ、サイズnのベクタノードの場合は自由リストが空でない限りn+14ステップで処理が終了する。

④ 空きアドレス(EA)の管理

EAの要求があった場合は、リスト型かベクタ型に従い、

空きアドレスバッファ(EAOB)より一定個数のEAを取り出し、パケットの形で要求元の推論ユニットに返送する。EAOBにはガーベジコレクションによって回収されたEAの一部が、常にある個数以上バッファリングされている。例えば、30個のEAを送出する場合、EAOBに十分なEAがバッファリングされていれば、40ステップで処理は終了する。

4. 構造メモリの処理速度

上で述べたアルゴリズムを用い、本構造メモリの平均稼働率をステップ数から概算する。平均稼働率とは、ハードウェアが実際に動作している時間の割合とする。文献4、5に従い、適当な仮定を置く。即ち構造メモリ1台に対して推論ユニットが16台接続されると、最繁忙時、構造メモリではLazy Fetchは20マイクロステップに1回、GIの格納は25ステップに1回、EA要求は750ステップに1回、RC更新命令は25ステップに1回生じる。これより、SMLFの平均稼働率は59%、SMAN の平均稼働率は116%となり、この内64%をRC命令が、52%をGI格納とEA管理が占めている。構造メモリではオーバーヘッドとなるRC命令処理に一番低い優先度を与えているので、SMAN の最繁忙時の平均稼働率が単純計算で100%を越えてもRCQ にバッファリングされているRC命令の個数が多少増加するだけである。SMAN が比較的idleな時にRC命令が処理されて行く。

5. おわりに

現在作成中のマイクロプログラムを基に、構造メモリの動作効率を概算した。今後さらにマイクロプログラムの改良を重ねて行く予定である。

<< 参考文献 >>

- [1] Moto-oka, T., Tanaka, H., et al, "The Architecture of A Parallel Inference Engine - PIE - ", FGCS'84, ICOT, 1984.
- [2] 平田, 猪股, 他, "PIEの構造メモリ試作ハードウェアの設計について", 信学技報, EC85-64, (1986). [3] "Am2900 Family 1985 Data Book", Advanced Micro Devices, INC. (1985).
- [4] 平田, 他, "PIEにおける構造メモリの構成について", アーキテクチャワークショップインジャパン'84, 情報処理学会, (1984).
- [5] 平田, 他 "高並列推論エンジンPIEの構造データ共有方式", 信学論, Vol. J69-D, No. 7, (1986).