

2C-7

PIEにおける
構造メモリの内部構成について

平田 圭二、相田 仁、田中 英彦、元岡 達
(東京大学 工学部)

1. はじめに

PIEはゴール書き換えモデルに基づき、Prologに代表される論理型プログラミング言語をOR並列に直接実行するマシンである。我々はPIEを設計する第一段階として、完全コピー方式によりゴールを生成するモデルPIE-Iの検証を行なった。この時、PIEの実働時には大きな構造データのコピーがボトルネックになることが指摘された。それに対し、構造データ共有方式を提案し、オーバーヘッドをどの程度まで減少させることができるかをシミュレーションにより測定した。その結果1桁程度の高速化が達成可能であることが分かったので、我々は構造データ共有方式を採用したPIE-IIを設計した。PIE-IIではネットワークを階層化し構造データを格納する構造メモリを設けた[1]。

現在まで行なって来たソフトウェアシミュレーションによって、構造データ共有方式による効率化の概略は把握できている。さらにオーバーヘッド等を含めた処理速度の精密な測定を行なうには、実際のマシンハードウェアの細かなレベルまで考慮に入れる必要がある。そのため、我々は現在PIE-IIにおける構造メモリ及び推論ユニットの設計、試作を進めている。本発表では、今回試作するハードウェアの仕様、機能、内部構成について述べる。

2. 試作ハードウェアの構成

PIE-IIの構造メモリ(SM)に要求される機

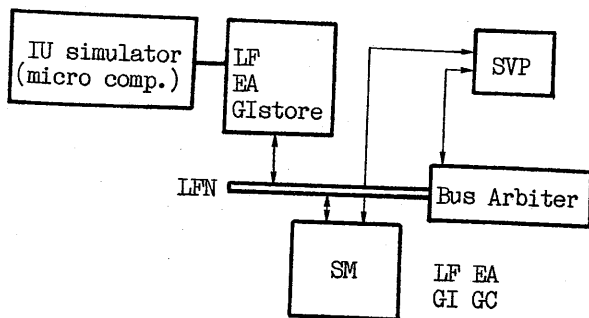


図1. 試作ハードウェアの概要

能にはLazy Fetch(LF)、Ground Instance(GI)格納、空きアドレス(EA)管理、ガーベジコレクション(GC)がある。推論ユニット(IU)側にも構造データ共有方式を支援する機構が必要である。

2.1 全体構成

今回試作するハードウェア部分を図1に示す。IU1台、SM1台の構成とし、LFネットワーク(LFN)で結合する。IU側について、我々はUNIRED[2]やIUシミュレータ[3]の試作経験があるので、今回の試作部分は現在までハードウェア化していない部分、即ち構造データ共有方式を支援する部分に限った。該当しない部分は適当なマイクロコンピュータシステムで実現する。SM側については、すべて専用ハードウェアで構成しマイクロプログラム制御を行なう。LFNはバスで実現し、アービタで集中管理する。尚クロックは5MHz、1ワードは4バイトの大きさにとる。

2.2 SMのシーケンサ

LFを行なう時、高速な応答を保証する為SMでは独立な2系統のマイクロシーケンサを同一クロックで動作させる。1系統はLFの実行を行ない、もう一方はGI格納、EA管理、GCを行なう。これら2系統のシーケンサから同じメモリへのアクセス競合が生じた場合は、シーケンサのレベルでLF側に優先権を持たせる。

2.3 SMのメモリ構成

SMの内部には、リスト構造のみを格納するリストメモリ(LM)と、複合項格納専用のアドレス変換テーブル(ATT)及び可変長セルメモリ(VC M)の3つのメモリがある[4][5]。メモリ全体はインタリーブを行なうイメージで8バンクに分け、2系統のシーケンサからのアクセス競合をなるべく低く抑えるようにした。参照カウント(RC)方式でGCを行なうので、RCを保持するためのメモリ(RCM)も必要である。これらのメモリの仮想化は行なわない。

2.4 推論ユニット

LFの為にLFバッファ(LFB)とLFバッファテーブル(LFBT)を設ける。LFBTにはSMのアドレスとLFBのアドレスが対として登録さ

れており、LF時には高速に検索しなければならない。従ってLFBTは連想メモリ(CAM)で実現する。またRC命令を最適化する専用ハードウェアにもCAMを用い、検索のオーバーヘッドを極力少くした。

2.5 ネットワーク

IU-SM間では、非同期にコマンドが転送される。LFNではデータ線に4バイト、タグ線に1バイト、制御線に3ビットが必要である。

3. 試作ハードウェアの機能及び特徴

3.1 LF・GI格納・EA管理

IU-SM間のコマンドパケット転送でLFを実現すると高速な応答は望めない。そこでLFNをバスそのものとして用い、IU-SM間でDMAのようにデータを転送する。即ちIUはバスを占有してSMに対して直接アドレスを送出し、それに対してSMはノードの実体をブロック転送する。こうしてLFの制御はIU側で行なうので、LFによるたぐりがSM中で生じることはない。

リストノードはATTを通さず直接参照し、ゴミのノードは自由リストで管理する。また複合項を格納する場合は、ATTにVCM上のアドレスと複合項のサイズを格納し、ノードの実体のみVCMに格納する。これらの手法によって、より高速なノードの読み書きが可能となる。

GIを切り分け・格納する場合、IU側のオーバーヘッドをできるだけ低く抑えるため、構造データの末端のノードだけを切り分けの対象とした。従ってGIは1回につき1ノード分だけ格納される。また予めSM中のEAがIUに配られており、GI格納コマンドと対になってSMに送出されるのでGI格納命令のAckは不要である。

3.2 LFN

LFNは2つの異なるモードで動作する。1つはLFの時であり、この場合LFNはあたかもバスのように振る舞い、SMのアドレスやSM中のデータを転送する。もう1つのモードはGI格納・RC・EA要求コマンドとEAが転送される時である。これらのコマンドは1つのパケットにまとめられ、適当なタイミング(GIが生じた時等)でIUからSMに転送される。SMからは、複数個のEAが要求を出したIUへまとめて送出される。

3.3 SMのGC

GCはRC方式で行なう。不要になったATTのセルやLMのノードは固定長であり、バンク間にま

たがるような自由リストで管理する。これはシーケンサ間のアクセス競合を減らす為、各バンクに均等に分散したEAを集めてIUに送出する必要があるからである。RC方式の場合、コストのかかるコピー・圧縮といった操作で、循環構造を回収し、自由リストを作り直す必要が時折生じる。この操作をできるだけ生じないようにし、コピー自体のコストを下げるために、VCM上のゴミノードは、サイズ毎メモリバンク毎の自由リストを設けてそれぞれ管理する。

3.4 SMのシーケンサ

LFNからコマンドパケットを読み込む時は、LFNアービタからGC等を管理しているシーケンサに割り込みがかり、パケットをコマンド入力バッファに取り込む。GI格納コマンドは最も処理の優先度が高く、RCのゼロ伝搬中でも処理せねばならない。

4. おわりに

以上、構造メモリ等の構造データ共有方式を支援するハードウェアの主な特徴について述べた。このハードウェアシミュレータでより詳細なPIE-IIの性能評価が可能となる。今後はマイクロプログラムやデバッグツールの作成等を行なう予定である。

《参考文献》

- [1] Moto-oka, T., Tanaka, H., Aida, H., Hirata, K., and Maruyama, T., "The Architecture of A Parallel Inference Engine - PIE -", FGCS '84, ICOT.
- [2] Yuhara, M., Koike, H., Tanaka, H., Moto-oka, T., "A Unify Processor Pilot Machine For PIE", LPC '84, ICOT.
- [3] 濱中, 田中, 元岡, "PIEのハードウェアシミュレータ - 推論ユニット -", 第30回情報全大2C-2.
- [4] 尾内, 清水, 麻生, 益田, 松本, "リダクション方式並列推論マシンPIM-Rのアーキテクチャ", LPC '85, ICOT.
- [5] 平田, 田中, 元岡, "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