4W-2

PIE64 のネットワーク・インタフェス・プロセッサのハードウエア構成

清水 剛, 小池 汎平, 田中 英彦
{shimizu,koike,tanaka}@mtl.t.u-tokyo.ac.jp
東京大学 工学部*

1 はじめに

現在、我々の研究室では、並列推論マシン PIE64[1] の開発をすすめている。PIE64 は 64 台の推論ユニット (Inference Unit: IU) が 2 系統の自動負荷分散機構付き多段ネットワークによって結合された構造を持ち、並列論理型言語 Fleng および、その上位言語である Fleng++ を実行する。

各 IU には、IU 内部と相互結合網とのインタフェスを行なうネットワーク・インタフェス・プロセッサ (Network Interface Processor: NIP) が用意される。NIP は PIE64 で行なわれる並列推論処理において、2 つの IU 間でのデータ転送、Fleng のプロセス間同期、PIE64 の一括ガベージ・コレクション [2] といった並列処理機能をハードウエア・レベルで直接支援する [3]。

本稿では、NIP の詳細な内部データ・パスと、その上での処理動作、及び、PIE64 の相互結合網を介して行われる 2 つの NIP 間での通信のプロトコルについて述べる。

2 NIP の内部ブロック構成

NIP の内部ブロックは、大きく分けて以下の 4 つのブロックから成る。

- コマンド・バス・インタフェス部
- メモリ・バス・インタフェス部
- データ転送処理部
- プロセス間同期処理部

コマンド・バス・インタフェス部は IU 内の他のプロセッサ (管理プロセッサ SPARC, 推論プロセッサ UNIRED [4], 他の NIP) との間で、コマンド / リプライの送受信を行なう。また、性能測定用および、タイムアウト等の監視用のレジスタを持つ。

メモリ・バス・インタフェス部は、IU 内部の 3 本のメモリ・バスと接続され、そのうちの 1 本に対しては IU 内ローカル・メモリに対する読み書きを行ない、他の 2 本に関しては、ロック・アドレスの監視をしている。また、負荷データやメッセージ受信に使用される ヒープ・メモリ管理用のレジスタを持つ。このヒープ・メモリ管理用のレジスタは 1 組の予備ヒープ・メモリ用のレジスタを含み、2 組が用意されている。

データ転送処理部は、ネットワークを介した データ転送処理、及び、一括ガベージ・コレクション支援の コマンドを実行する。また、ネットワークの自動負荷分散機能に対応して、負荷分散情報を管理するためのレジスタを持つ。

プロセス間同期処理部は、サスペンション・レコードの管理のためのリスト処理を行なう。

これらのブロックはそれぞれ独自にシーケンサを持ち、協調動作をするとともに、並行処理を可能としている。

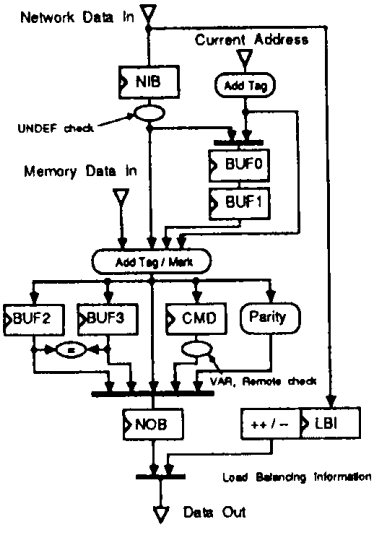

図 1: データ転送処理ブロックのデータパス

3 データ転送処理ブロック

データ転送処理ブロックのデータ・パスを図 1 に示す。データ転送処理においては、

1. 最大効率時に、1 クロック 1 ワードのデータ転送が可能となること
2. IU 内ローカル・メモリはパイプライン的にアクセスされること
3. ネットワークに約 50 ns の遅延が存在すること
4. 構造データ内に埋め込まれた未定義変数領域の一意性を保証するため、この変数領域をポインタに変換して転送できること

等の要求及び制約を満たすために、データ読みだし側の NIP で 3 段、ネットワークを 間に はさんで データ書き込み側の NIP で 3 段のパイプラインを構成する形になっている。

*Hardware Design of the Network Interface Processor of PIE64
Takeshi SHIMIZU, Hanpei KOIKE, Hidehiko TANAKA,
the University of Tokyo

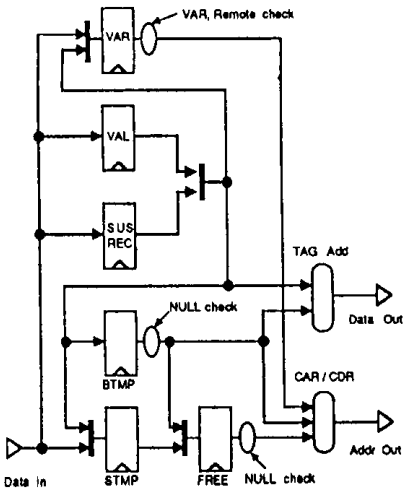

図 2: プロセス間同期処理ブロックのデータパス

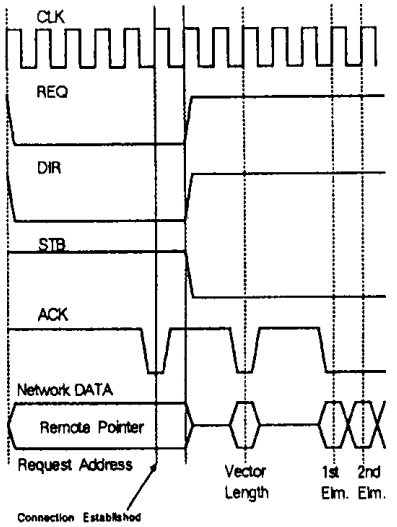

図 3: readn コマンドによるベクタ転送時の通信プロトコル

読みだし側での 3 段は、図 1 中の $BUF0 \rightarrow BUF1 \rightarrow NOB$ のパスで、前 2 段はメモリ・リクエスト及びデータ・フェッチのステージに相当し、その時のリクエスト・アドレスが前 2 段で保持され、前記 4 の変換処理に利用される。このポインタとデータの置き換えは、3 段目のネットワークへの出力段に移行する $BUF1 \rightarrow NOB$ の段階で行なわれる。

書き込み側での 3 段は、図 1 中の $NIB \rightarrow BUF2 \rightarrow NOB$ のパスで、それぞれ、ネットワークからのデータ受信、メモリ・リクエスト、データ書き込みのステージに相当する。

その他のレジスタは、書き込み側でメモリバスのアクセス権を確保できなかった場合に生じるパイプラインの乱れを解消するためのバッファとして利用される。このバッファリングは、送信側で 2 ワード、受信側で 3 ワード保持される形になる。また、一括ガベージコレクション支援時のリモート・ポインタのマークと、コンパクション後のポインタ書き戻し処理の場合には、一部のレジスタがハッシング及びキャッシングのために利用される。

4 プロセス間同期処理ブロック

プロセス間同期処理ブロックのデータ・パスを図 2 に示す。

このブロックでは、プロセス間同期処理の内の、suspend 処理における サスペンション・レコード・リストへの コンテクストの 追加登録、および、activate 処理における サスペンション・レコード・リストの 回収を 主として行なう。そのための、引数レジスタ (VAR, VAL)、リスト処理時の テンポラリ・レジスタ ($SUSREC, BTMP, STMP$)、フリーリストを 示す レジスタ ($FREE$) を持つ。また、リスト処理時にポインタに対して必要に応じてタグの書き換えを行なう。

5 通信プロトコル

ネットワーク上での通信プロトコルの例として、readn コマンドによるベクタ読みだし転送時のタイミングを図 3 に示す。

ネットワークを介した 2 つの NIP は、STB と ACK の 2 本の信号線によりお互いの状態を伝えあい、データの転送を行なうが、1 クロック 1 ワードの転送をするために、1 ワードごとのハンドシェークをするのではなく、これら 2 本の信号は、そのサイクルで、データを送信していることと、データの受信が可能な状態であることを表す信号となっている。

図 3 の例では、マスタ NIP 側からの接続要求がネットワークを通り、スレーブ側で受け取られ、スレーブ側から ベクタ長が転送された後に、パイプライン的な転送状態に入る様子を表している。

6 まとめ

PIE64 の NIP の内部データ・パス及び処理動作、並びに、ネットワークを介した通信プロトコルについて述べた。

現在、シミュレーションによる 最終チェックを 行なっている。

今後の課題としては、サンプルの性能評価、および、PIE64 の IU の設計・実装に伴って、実際の相互結合網を使用した接続試験があげられる。

なお、本研究は文部省の特別推進研究の一環である。

参考文献

[1] 小池, 田中, " 並列推論マシン PIE64 の概要 ", 情報処理学会第 37 回全国大会　5N-4, Sep. 1988.

[2] Lu Xu and Hidehiko Tanaka: *Distributed Garbage Collection for the Parallel Inference Machine: PIE64*, 情報処理学会第 38 回全国大会　5U-7, Mar. 1989.

[3] 清水, 小池, 島田, 田中, " 並列推論マシン PIE64 のネットワーク・インタフェス・プロセッサ ", 並列処理シンポジウム '89 A2-2, 情報処理学会, Feb. 1989.

[4] 島田, 下山, 清水, 小池, 田中, " 推論プロセッサ UNIRED II の命令セットの概要 ", 本大会.