5N-5

PIE64のネットワーク・インタフェス・プロセッサの概要

小池 汎平, 清水 剛, 島田 健太郎, 田中 英彦
(東京大学 工学部)

1 はじめに

並列推論マシンPIE64q[1]のネットワーク・インタフェス・プロセッサ (Network Interface Processor: NIP) は、各推論ユニット (Inference Unit: IU) 内部と、相互結合網[2]とのインタフェスを行ない、データ転送やプロセス間同期などのプリミティブ・オペレーションをネットワークを介して2台のIU間で実行することにより、PIE64で行なわれる並列推論処理のうち、並列処理機能を支援する。NIPはマスタ部とスレーブ部にわかれる。マスタNIPは、UNIRED及びSPARCから、並列処理用プリミティブ・オペレーション実行の要求を受け付け、相手IUまでのネットワークの経路を接続したのち、接続先IU内のスレーブNIPと協調して、プリミティブ・オペレーションを実行する。PIE64は2系統のネットワークを持ち、各IUには、それぞれのネットワークのために、2系統のNIPが用意されている。本稿では、現在検討を進めているPIE64のNIPの機能について述べる。

2 NIPの機能

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 (Network Interface Processor: NIP) は、各推論ユニット (Inference Unit: IU) 内部と、相互結合網とのインタフェスを行ない、データ転送やプロセス間同期などのプリミティブ・オペレーションを2台のIU間でネットワークを介して実行することにより、PIE64で行なわれる並列推論処理のうち、並列処理機能を支援する。図1に示すように、NIPは相互結合網と、推論ユニットの内部バスに接続され、IUのローカル・メモリを直接アクセスし、ネットワークにデータを送り出す。又、NIPは、コマンドバスを通じてUNIRED/SPARCより、プリミティブ実行のコマンドを受け取る。

NIPは、マスタ部とスレーブ部にわかれる。図2に示すように、マスタNIPは、UNIRED/SPARCより様々な並列処理用プリミティブ・オペレーション実行の要求を受け付け、相手IUまでのネットワークの経路を接続したのち、接続先IU内のスレーブNIPと協調して、各種の並列処理用プリミティブ・オペレーションを実行する。マスタNIPとスレーブNIPは、大部分のハードウェアを共通化できるので、同一チップをモード切替えして動作させる。PIE64は2系統のネットワークを持ち、各IUには、それぞれのネットワークのために、2系統のNIPが用意されている。

3 NIPのプリミティブ・オペレーション

NIPのプリミティブ・オペレーションは、基本的に、

[0]The Network Interface Processor of PIE64
Hanpei KOIKE, Tsuyoshi SHIMIZU, Kentaro SHIMADA
Hidehiko TANAKA, the University of Tokyo

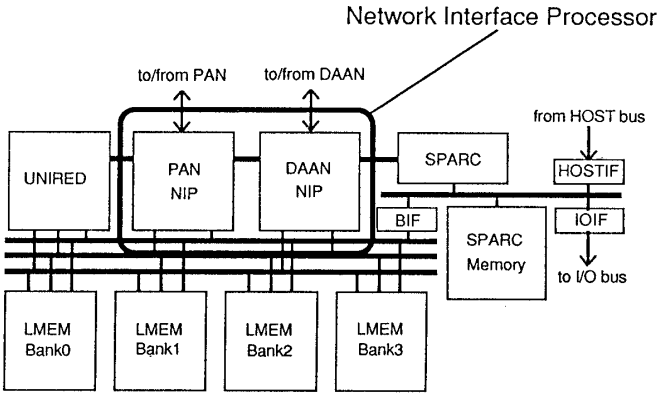

図 1: Network Interface Processor

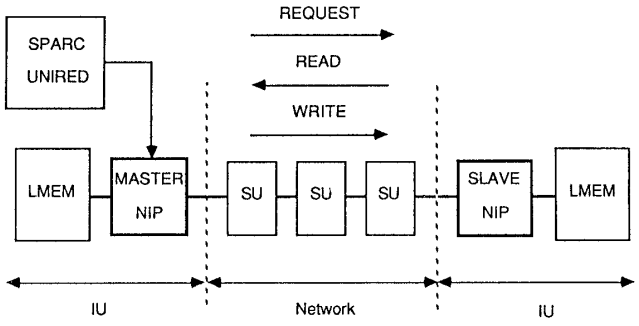

図 2: NIPの役割

- データ転送用
- プロセス間同期用
- その他

に分けられる。

データ転送用プリミティブ・オペレーションとしては、

- read1 read2 readn readx
- write1 write2 writen writex
- writel1 writel2 writeln writelx

がある。read、write、writelは、それぞれ、相手IU内データの読み出し、相手IUローカルメモリへのデータの書き込み、最小負荷IUへのデータの書き込み (負荷分散)、を表す。また、末尾の1、2、n、xは、転送するデータのサイズを表し、それぞれ、1ワード、2ワード、データの先頭からサイズを読み出す、プリミティブ・オペレーションのオペランドとしてサイズを直接指定する、ことを意味する。前3者は、それぞれ、FLENGの、変数型、リスト型、ベクタ型に直接対応する。

基本的には、データ転送は、コピー元、コピー先のメモリアドレスを指定した、2つのIUのローカルメモリ間でのメモ

リ内容のブロック転送である。ただし、1ワードの読み出しでは、読み出した値が、コマンドバスを介して、読み出し要求元のUNIRED又はSPARCに直接返され、又、1ワードの書き込みでは、書き込む値は、書き込みコマンドとともに、コマンドバスを介して、直接NIPに渡される。

プロセス間同期用のプリミティブ・オペレーションとしては、

- suspend
- bind
- activate

が用意され、FLENGの持つプロセス間同期機構を、ハードウェアレベルで直接サポートする。これらのプリミティブオペレーションは、互いに密接に関連して動作し、以下に示すように、場合によってお互いに他を呼び合うことがある。

suspendは、あるIU内で、あるコンテクストの処理が、別のIU内に置かれた論理変数の値が定まっていないためにサスペンドを起こした時、その論理変数の置かれたIUに対して、サスペンドしたコンテクストのアドレスを送り付け、未束縛論理変数の保持するサスペンションリストにサスペンドしたコンテクストを登録するためのオペレーションである。もし、suspendオペレーションの実行前に、既に別のゴールによって論理変数に値がバインドされていた場合は、スレーブNIPは、送られてきたコンテクストをサスペンションリストに登録する代わりに、そのコンテクストに対するactivate動作を開始する。

bindは、あるIU内でActive Unificationが行なわれたときに、他IU内に置かれた論理変数に対し値をバインドするためのオペレーションである。値をバインドするに先立ち、スレーブNIPは、その論理変数のサスペンションリストに登録されている個々のコンテクストに対し、activate動作を行なう。もし、論理変数にバインドする値が、他の未束縛の論理変数であった場合は、2つの論理変数の保持するサスペンションリスト同士をマージする必要があり、このために、スレーブNIPは、バインドする論理変数に対し、自IU内の論理変数が保持するサスペンションリストを用いたsuspendオペレーションを起動する。

activateは、論理変数のサスペンションリストに登録されている個々のコンテクストの置かれたIUに対し、論理変数にバインドされた値を送りつけ、ゴールの管理を行なうプロセッサに対し割り込みをかけるオペレーションである。

NIPには、以上で示した基本的なプリミティブオペレーションの他に、モード設定や、タイムアウト時間などのパラメータの設定のためのコマンドが用意される。

4 NIPの通信プロトコル

NIPによる1回のプリミティブオペレーションの実行は、

- UNIRED又はSPARCからのコマンドの到着
- マスタNIPによる前処理
- ネットワークの接続
- ネットワークを介したデータのやりとり
- ネットワークの解除
- マスタNIP及びスレーブNIPによる後処理

の順で行なわれる。

ネットワークの接続時に、マスタNIPは、最初に、ネットワークに対しプリミティブ・オペレーションのオペランドである相手IU内のデータを指すポインタを送出する。この最初の1ワードの送出で、ポインタの各フィールドは、次のように用いられ、可能な限りの情報が相手プロセッサに渡される。このポインタのプロセッサ番号フィールドは、ネットワークは、経路を接続するのに用いる。ネットワークが接続されるとともに、ポインタはそのまま相手IUのスレーブNIPに転送され、ポインタのデータ型を表すタグ部とGC時に用いられるマーク部のフィールドは、実行すべきプリミティブ・オペレーションを、ローカルメモリアドレスフィールドは、スレーブNIPのアドレスレジスタに設定される。以上の操作は、ネットワークに衝突がなければ、ネットワークの接続に1-2クロック、接続の確認に1クロック、合計2-3クロック程度で実行できると考えられる。しかし、ネットワーク上で接続要求の衝突が頻繁に起きる場合は、接続要求は、一定時間でタイムアウト処理する。ネットワーク接続の後、基本的には、1ワード/クロックで、データの転送を行ない、プリミティブ・オペレーションの実行が行なわれる。この間、マスタ/スレーブ両NIP内の制御回路は、1クロック遅延のあるハンドシェークによって、同期がとられる。又、PIE64のネットワークは双方向なので、1クロックのオーバヘッドでネットワークの方向を逆転させることも可能である。最後に、ネットワークの解放が行なわれ、これには1クロック要する。

5 おわりに

本稿では、並列推論マシンPIE64のネットワークインタフェースプロセッサの機能について述べた。現在詳細な仕様の検討を進めている。今後に残された課題としては、

- データ転送エラーに対する信頼性向上のための検討

などが挙げられる。

参考文献

[1] 小池, 田中, "並列推論マシンPIE64の概要", 本大会.

[2] Koike, H., Takahashi, E., Yamauch, T. and Tanaka, H. : *The High Performance Interconnection Network of the Parallel Inference Machine PIE64*, Proc. of Computer Architecture Simposium, IPSJ, May, 1988.