バス結合による並列処理システムのハードウェア化

5G－1　高橋潤　黒川恭一　古賀義亮
防衛大学校情報工学教室

1．はじめに

粒度の大きい並列処理システムの課題の一つに通信や同期に係る高速化がある[1)]。この課題を解決する1つの試みとして、我々は分散並列決定方式のアービタ[2)]を使ったバス結合方式によるMIMD型並列処理システムを提案し[3)]、現在そのハードウェア化を進めている。

本稿においては、このハードウェアシステムの概要を報告する。

2．ハードウェア構成

文献3)で提案したシステムは、複数のPEをバス接続した図1のような構成であり、共有メモリや集中コントローラを持たない簡単な構造となっている。

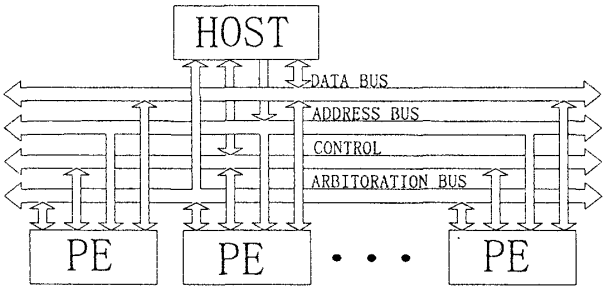

図1　システムの概要

個々のPEは、図2に示すように32ビットDSP(μPD77240:NEC)を核とし、RAM、ローカルアービタ、インタフェイス回路によって構成されている。このDSPはインストラクションとデータを別々のRAMに格納する方式を採っているため、PE内にはインストラクションRAMと、データRAMを組み込んである。

システムバスは、32ビットデータバス・15ビットアドレスバス・21本のコントロールと、バスマスタの決定（アービトレーション）に使う6ビットアービトレーションバスから成っている。コントロールは、ホストと各PE間のハンドシェイク信号（リード・ライトストローブを含む）や各PE間のハンドシェイク信号等からなる。

アービトレーションバスには、各PEのローカルアービタが接続され、これらによりシステムのアービタ（調停回路）を構成する。ローカルアービタは、64PEまで拡張可能なものを1チップのPAL(PALCE16V8HD-15PC:AMD)上に構成した。このアービタは、論理回路により構成されているため、極めて高速にバスマスタを決定できる上に、耐故障性に優れた構成も可能である[4)]。

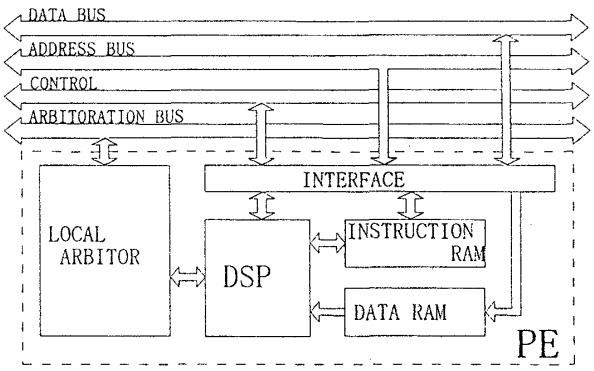

図2　PEの概要

データの入出力のためにホストを設け、これによりデータバス、アドレスバス、コントロール線の管理を行っている。このホストとしてはパソコン(PC-98:NEC)を用意した。

3．動作手順

前節で提案したハードウェアシステムの動作手順は、以下に示すように大きく3つに分けられる。

① イニシャルロード

イニシャルロードは、ホスト上に用意した各PE用のインストラクション及び初期データを、対応する各PEのインストラクションRAM及びデータRAMに転送することによって行う。なおこの時DSPは、リセット状態にしてある。

Hardware Implementation of Bus Connected Parallel Processing System
Jun Takahashi, Takakazu Kurokawa, Yoshiaki Koga
Department of Computer Science, National Defense Academy
1-10-20 Hashirimizu, Yokosuka, Kanagawa 239, Japan

② 並列処理

ホストからのイニシャルロードが完了して、リセットを解除すると、各ＰＥはイニシャルロードされたインストラクションに従ってそれぞれの処理を開始する。その処理の形態としては、(1) 各ＰＥが単独で処理するモードと(2) 相互通信するモードとを用意した。今回のシステムにおいては、取り扱うアプリケーションとして、全ＰＥが同期しつつ上記のモードを交互に繰り返すことにより処理可能な方式に限定した。

全ＰＥの同期方式を以下に示す。

(1)のモードから(2)のモードに移る時は、通信準備を完了したことを知らせる出力ポート(READYポート)をアクティブにする。各ＰＥのREADYポートは、ワイヤード論理により相互接続され、これが各ＰＥの入力ポート（GO!ポート）に入る。従って各ＰＥは、全てのＰＥのREADYポートがアクティブになったことを判定し、相互通信を開始する。

一方、各ＰＥ間の相互通信は、分散決定方式のアービタを使ったバスにより行う。これを採用したことにより、以下の３点から通信に要するオーバーヘッドを軽減できる。

第１に、出力の必要の無いＰＥを無視するため、その確認等にかかる手間を省くことができる。

第２に、ハードウェア構成上、アービトレーションの結果バスマスタが決定すると、アービトレーションバス上には、バスマスタになったＰＥの番号（の反転）が乗っている。このため、データを転送する際にそのＰＥの番号を改めてブロードキャストする手間が省ける。

第３に、今回作成したローカルアービタでは、検証の結果約２０ns程度で１回の調停が行われる。本システムのクロックサイクルが１０MHz程度なので、調停のためにシステムに負荷をかけることはない。つまり、１クロックでバスマスタを決定し、全ＰＥがそれを知ることができる。

適切なアプリケーションを採用することによって、以上の特性を十分に生かしつつ、通信のオーバヘッドに関する問題に対処できる。

③処理結果の出力

各ＰＥには、ホストとのハンドシェイク用にポート（フラグポート）を設けてある。各ＰＥのフラグポートは、ワイヤード論理により互いに接続され、これがホストの監視用ポートに入る。これによってホストは、全てのフラグポートがアクティブになったことを判定できる。

各ＰＥは処理が終了（一段落）すると、フラグポートをアクティブにする。全てのＰＥのフラグポートがアクティブになることにより、全てのＰＥの処理が終了していることが判定されるため、ホストはその時点での全ＰＥの処理結果を取り込むことができる。そのため、まずＤＳＰ－データＲＡＭ間のバスを切り離し、ホストからデータＲＡＭへのアクセスを可能な状態にする。これを受けてホストは、各ＰＥのデータＲＡＭの内容を読み込み、ハードウェアシステムによる並列処理結果の解析を行うことになる。

４．おわりに

本報告では現在作成中の、分散並列決定方式のアービタを使ったバス結合方式による並列処理システムのハードウェア構成を示した。

今後は、疎行列問題[5]、ニューラルネットワーク[6]等のアプリケーションの実装を行いつつシステムの評価を行う予定である。

参考文献

1) 所：“並列処理システムの展望”，情報処理, 27, 9, pp. 1049-1055 (1986).

2) IEEE: *"IEEE Standard Backplane Bus Specification for Microprocessor Architecture: Futurbus,"* ANSI/IEEE Std. 896. 1-1987, July 1988.

3) 高橋、黒川、古賀：”バス結合方式による並列処理システムの提案”、１９９３年電子情報通信学会秋季大会予稿集, D-71, (1993).

4) 時藤, 黒川, 古賀：“セルフテスティングバスアービタの一実現法について”，信学論, J75-D-I, 1, pp. 30-40 (1992).

5) 戸川: “マトリクスの数値計算”, オーム社 (1971).

6) Y. Takefuji: *"Neural Network Parallel Computing"*, Kluwer Academic Publishers (1992).