

高速信号処理向き並列アーキテクチャの提案

4B-5

水野 政治 宮田 裕行 菅 隆志

三菱電機株式会社 情報システム研究所

1 はじめに

近年の集積技術の高度化に伴い、従来は実現が困難と考えられていた技術の実現可能性が高まっている^{1), 2)}。特に画像処理、信号処理の分野では、その高速化の要求に応じて様々な計算機アーキテクチャが検討されてきており、その実用化が行なわれている。

我々は、定期的に送られてくる信号に対し、ある定められた時間内に処理し、かつ次々と送られてくる信号を滞らせることなく処理できる信号処理プロセッサの並列アーキテクチャを検討中である。

本論文では、従来システムからの大幅な性能向上、拡張性、信頼性の向上等を目指して研究・開発を進めてきた信号処理装置のアーキテクチャに関して説明する。

2 動機と目的

これまでに我々が扱ってきた信号処理装置としては、図1に示すように、専用のハードウェアを幾つか接続し、次々と入力される信号を順にパイプライン処理する手法を用いてきた。

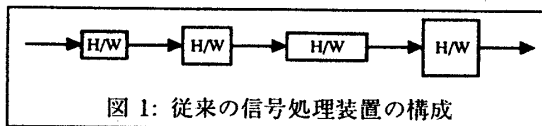


図 1: 従来の信号処理装置の構成

しかしながら、このような手法では次のような問題点があげられる。

- 予め決められた特定の処理にしか対応することができず、処理の融通性に欠ける。
- 信号処理のスペックの変更に応じて容易に性能を上げることが難しく、拡張性に欠ける。
- 故障への対処が困難。

また、定期的に入力される信号に対して、その負荷の変動に応じて使用するハードウェア資源を有効に利用する(以下、負荷分散と呼ぶ)という新たな要求もある。

そこで、我々は、上記のような問題点を解決する、融通性、拡張性、信頼性に富む高速信号処理用計算機アーキテクチャを提案する。

Proposal of Parallel Processing Architecture for High-Speed Signal Processing.

Masaji MIZUNO, Hiroyuki MIYATA, Takashi KAN

Computer and Information Systems Laboratory, Mitsubishi Electric Corporation

3 アーキテクチャの検討

我々の行なったアーキテクチャの検討内容について、その経緯の順に説明する。

3.1 マイクロプロセッサによる置き換え

まずは、単純に従来の構成において専用のハードウェアとなっている部分をマイクロプロセッサに置き換え、プログラムの変更によって処理内容を変更することができるため、ソフトウェアにより信号処理を実現することとした(図2)。

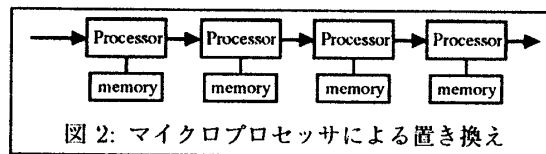


図 2: マイクロプロセッサによる置き換え

この構成においては、処理の融通性を向上できるものの、次のような問題点が考えられる。

- プログラムで処理を実現するために、各プロセッサ毎にデータをいったん蓄える必要があり、プロセッサ毎に膨大なメモリ量が必要となる。
- 各プロセッサ間でのデータ転送が性能に大きく影響を与える。

3.2 外部メモリを用いた構成

上記の構成案において問題となったデータ転送をなくすため、全プロセッサがアクセスできるメモリを外部に設ける構成を考える。例えば、図3に示すように、順次入力されるデータを用意したメモリに順に格納しておき、処理の流れに応じて、各プロセッサが外部メモリにアクセスして処理を行うこととする。

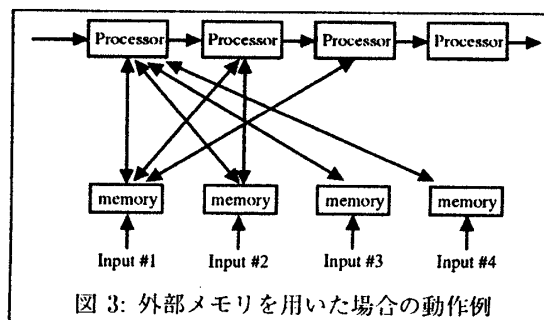


図 3: 外部メモリを用いた場合の動作例

この場合、各プロセッサが順にすべてのメモリにアクセスするため、複数のプロセッサによるメモリアクセス

が同時に発生することとなる。そこで、プロセッサとメモリとをクロスバー・スイッチで接続する構成を考える。この方式によれば、プロセッサ間で処理が移る時にデータ転送がなくなるが、一方では以下に示す問題点が残っている。

- メモリアクセスの低下による性能低下。
- クロスバー・スイッチによるハードウェア量の増加。特にプロセッサ数が大きくなった場合。
- プロセッサ数の増加に対応し難い。
- 処理内容の変化に伴う負荷分散の制御の複雑さ。

3.3 パイプライン処理から並列処理への移行

基本的に、ここまでの検討においては、複数のプロセッサを用い、それらをパイプライン的に動作させることを考えてきた。しかし、プロセッサによるパイプライン処理では機能(処理内容)毎にプロセッサを割り当てるため、負荷分散の実現は難しい(図4)。

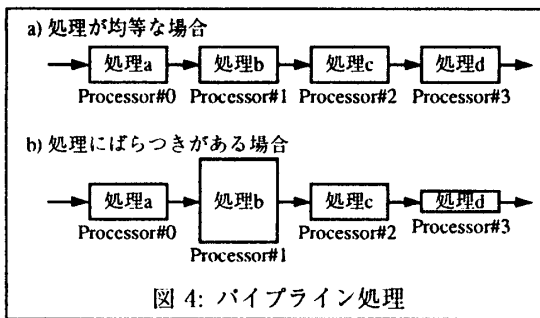


図4: パイプライン処理

幸いにして、我々の扱う信号処理では並列性が非常に高く、データ間の依存関係が少ないため、並列処理を基本とした構成を考えることとする。

並列処理のアーキテクチャについては、共有メモリか分散メモリか、プロセッサ間をどの様に接続するかなど多くの選択肢があるが、1) データの依存関係がないため、プロセッサ間のデータ転送が必要ないこと、2) 比較的实现が容易であること等を考慮した結果、図5に示す構成案を採用することとする。

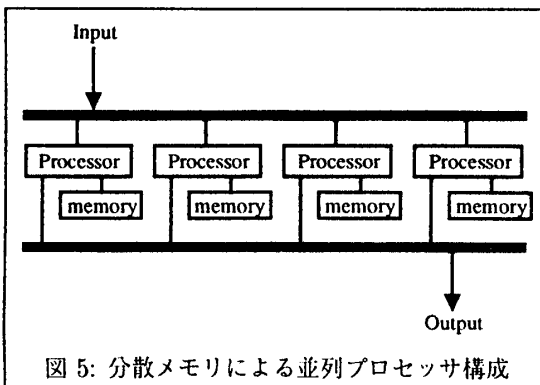


図5: 分散メモリによる並列プロセッサ構成

4 分散メモリによる並列プロセッサ構成

図5に示した構成について、動作・特徴を説明する。

図6に、本構成における基本的なデータの流れを示す。同図に示すように、定期的に入力されるデータはバ

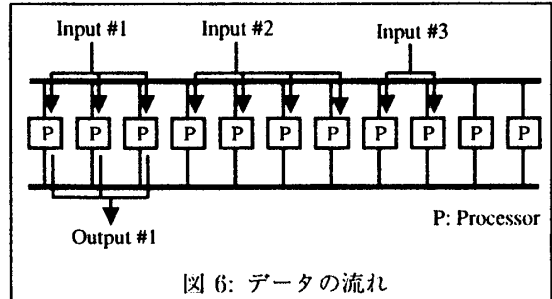


図6: データの流れ

ス1を使用し各プロセッサに転送される。プロセッサではデータに対して信号処理を行ない、その後バス2を使って処理結果を出力する。

入力されるデータの処理量は予め知ることができるため、その処理量に応じて使用するプロセッサ数を変更し、データを各プロセッサに分配する。

また、信頼性向上として、プロセッサが故障した場合、空いているプロセッサに処理を切り替えることにより対処する(図7)。

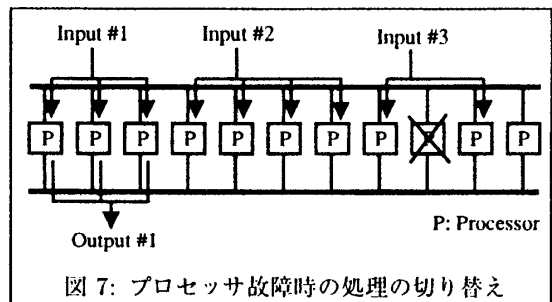


図7: プロセッサ故障時の処理の切り替え

5 おわりに

本論文では、高速な信号処理を実現するための計算機アーキテクチャについて述べた。今後は、本方式の実現及び評価を実施していく予定である。

謝辞

本研究に関して、貴重な助言と示唆を数多く頂きました。川田圭一次世代方式技術開発部長ならびに分散グループ諸氏に感謝致します。

参考文献

- 1) Reed, D. A. and Fujimoto, R. M.: *Multicomputer Networks - Message-Based Parallel Processing*, The MIT Press (1987).
- 2) 奥川峻史: 並列計算機アーキテクチャ, コロナ社 (1991).