

機能メモリをベースにしたコプロセッサシステム

7G-6

路奎明 田丸啓吉

京都大学 工学部 電子工学科

1 まえがき

近年、計算機の応用分野が広がり、処理量が増えることに伴い、高速な計算機への要求が高まっている。しかしながら、従来の計算機ではプロセッサとメモリ間のフォンノイマンボトルネックが問題になっている。従って、メモリに存在するデータに対して、並列性を活かした処理を行なうことの重要性が浮かび上がってくる。例えば、データの記憶と処理の2つの機能を持つ機能メモリの場合は、データを移動させることなく並列に処理することができる。近年の集積回路技術の成果を踏まえて、機能メモリを用いた高い並列度を持った計算機システムを構成しようとする試みが始まっている [1][2]。

2 機能メモリの制御方式

大容量の機能メモリをプロセッサの記憶空間の一部に割り当て、処理データを一度機能メモリへ格納する。格納された処理データに対してはデータを移動させることなく、ほとんどデータ量に係わらず一定時間で処理できる。しかし、現在まで計算機システムの中でどのように機能メモリを組み込むかまだ明確にされていない。可能な機能メモリの制御方法を大別すると、2つの方式に分類される。

(1) 周辺デバイスとして使われる：機能メモリとプロセッサ間の関係は、簡単に言えば、機能メモリがプロセッサの周辺デバイスと見なされる。プロセッサ側から機能メモリへの制御がデータ転送命令で行われる。この制御方式には、また2つの種類がある。

(a): この方式 [3] の特徴は機能メモリをプロセッサのバックエンドプロセッサとして用いることである。プ

ロセッサ側が動作命令を機能メモリへ渡すことにより、機能メモリを利用して処理を行なう。

(b): この方式 [4] の特徴はプロセッサ側からメモリのマッピング方式で機能メモリを制御することである。アドレス上に制御情報、探索情報などを載せている。

(2) コプロセッサとして使われる：プロセッサのコプロセッサというのは、プロセッサと通信する機能を備えたデバイスの集合である。コプロセッサがプロセッサと通信しながら両者共同に仕事を処理する。

3 分析

この2つの方法を使って機能メモリをシステムの中へ組み込むことができるが、どの方式を使ったらメリットが一番あるか、次に分析した結果を述べる。

(1) 制御回路の複雑さ：メモリマッピング方式の制御回路は機能メモリの命令コードと制御信号を生成する必要がないので、制御回路がコプロセッサ方式より簡単にできる。

(2) 処理の並列性：コプロセッサ方式では、プロセッサとコプロセッサを同時に動作させることができるので、プログラムの実行効率については、周辺デバイスとして使われる場合より良い。

(3) プログラムの容易さ：周辺デバイスとして使われる場合には、機能メモリを使うための命令を生成するのが一つの問題点となる。メモリマッピング方式とバックエンド方式の場合は、プログラミングする時に、たくさんのデータ転送用コマンドを書くことになり、プログラムが理解しにくいことになる。一方、コプロセッサ方式はこういう問題が存在しない。

今回、処理の並列性とプログラミングの容易さに注目し、コプロセッサ制御方式を選んだ。

4 機能メモリをベースにしたコプロセッサの設計

今回の実験システムは、機能メモリが汎用プロセッサ MC68020 のコプロセッサとして動作する。また、実験に使われた機能メモリは NTT 社の 4k ビット CAMLSI である。この連想メモリをベースにしたコプロセッサの基本的な構成を図 1 に示す。

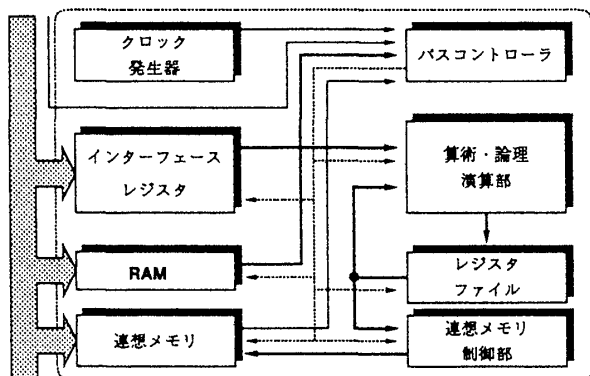


図 1: 連想メモリをベースにしたコプロセッサの構成

- (1) クロック生成部: システムのクロックは 4 相から構成される。1 サイクル時間は 200ns である。
- (2) 算術論理演算部: データの加工と条件分岐の処理として、算術論理演算部が使われる。
- (3) インターフェース・レジスタ: これらのレジスタ群を利用しながら、プロセッサとコプロセッサが共同に仕事を完成させる。
- (4) 連想メモリ制御部: 連想メモリ制御部は 2 つの役目を持つ。連想メモリが普通の RAM の時には、メモリ制御部として動作する。一方、連想メモリが連想機能を使う時に、プロセッサとして動作する。
- (5) バスコントローラ: 32 ビットの命令をデコードすると 32 ビットのデータバスを制御する。
- (6) 連想メモリ: 2k ワード容量を持つ。

5 実験結果

(1) ハードウェア: 本システムはコプロセッサ命令と連想メモリ命令を一つの命令ワードに収めることにより、パラレルで実行することができた。この方法によって全体処理ステップ数を減らすことができる。また、コプロセッサ制御方式の場合には、コプロセッサが処理中の時に、プロセッサは他の仕事を処理することができる。いわゆる、プロセッサとコプロセッサ

がパラレルに処理することにより、全体の処理効率が向上できる。今回、用意したコプロセッサ命令の実行ステップ数と直接にプロセッサから制御する場合の実行ステップ数を表 1 に示す。

(2) ソフトウェア: プロセッサのコプロセッサ命令を使うことだけで、機能メモリを利用することができる。ユーザは機能メモリの知識を知らなくても、機能メモリコプロセッサの設計者が定義した演算命令を使うことにより機能メモリの優れた並列処理機能を簡単に使える。

N: 探索動作が一致した回数, M: パターンの文字数
W: 処理のデータ数

処理種類	周辺デバイス	コプロセッサ
内部値並列加算	465	422
外部値並列加算	640	576
パターンマッチング	7+10M	6+7M
極値探索	360-N	296-N
計数ソート	4+(297+6N)W	4+(249+4N)W

表 1: ステップ数の比較

6 まとめ

機能メモリは、現在のメモリ集積回路の技術を利用して、集積規模の増大が実現できる。今回は機能メモリを計算機システムへ組み込む制御方式を検討し、機能メモリを用いた計算機システムの実例を紹介し、結果を考察した。コプロセッサ制御方式は全体処理ステップ数の減少とプログラミングの容易さから考慮すると、有効な制御方式と断定できる。

参考文献

- [1] 長沼次郎, 小倉武, 山田慎一郎: 連想メモリを用いた Prolog マシンの構成と処理アルゴリズム, 情報処理学会 (記号処理), 32-3, (1985)
- [2] 井出, 石和, 小島, 鈴木, 大附: 連想メモリを用いた図形処理装置の試作, 信学技報, CAS87-106, (1987)
- [3] 桑原, 中村, 久保田, 佐藤, 大附: 連想メモリを用いた図形処理用ハードウェアエンジン, 電子情報通信学会技術研究報告, CPSY92-9-19, 63-70, (1992)
- [4] 井田: Common Loops のためのあるメッセージ送信機構, 情報処理学会 (記号処理), 39-6, (1986)