

プログラムによるメモリ階層制御*

3B-3

電気通信大学情報システム学科†

牧 晋広‡ 曾和 将容§

1. はじめに

コンピュータは、基本的に高速で大容量のメモリを必要とする。並列コンピュータにおいてもこのことは同じである。実際に高速で大容量のメモリを用意することは原理的に不可能である。したがって、現実的には小容量で高速なメモリと大容量で低速なメモリを階層的に組み合わせ、疑似的に高速で大容量のメモリを構成している。通常これは、キャッシュメモリとして実現されている。キャッシュメモリでは、原理的にキャッシュミスが起こり、これがパイプラインの乱れを起こし、コンピュータの性能を下げている。我々は、実行しようとするプログラム（処理プログラム）を解析して、必要なデータの使用時間と場所を割だし、必要なデータを用意するためそれ専用のプログラム（メモリ操作プログラム）を作成し、このプログラムを処理プログラムと並列に動かすことによって疑似的に高速大容量のメモリを構成する方法に関する研究を行ってきた。[1][2] この方式では、キャッシュミスと言う概念が無くなり、プログラム制御によってプロセッサが必要とする時は、いつでも必要なデータを高速メモリに用意することができる。ここでは、メモリ階層を、レジスタと、キャッシュメモリに相当する高速メモリと、主メモリによって構成した場合の、プログラムによる階層メモリ制御 (PCHMS: Program Controled Hierachical Memory System) について述べる。特に今回は、その基本原理と基本構成、シミュレーションによる結果について述べる。

2. 基本原理と構成

2.1 PCHMS の構成

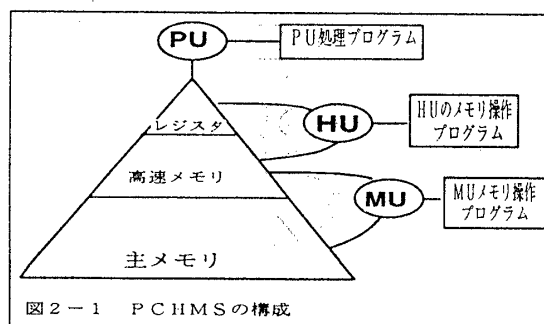
図2-1に本プログラムによるメモリ階層制御システム (PCHMS) の構成を表している。PCHMS は、独立のアドレスをもつレジスタ (超高速小容量)、高速メモリ (高速小容量)、主メモリ (低速大容量) の3階層のメモリで構成され、それぞれのメモリは、3つのプロセッサにより制御される。これらは、PU、HU、MU と呼ばれている。それぞれのプロセッサは、独立したプログラムを持ち、以下のように働く。

・PU：一般の処理を実行するプロセッサで、レジスタのみを対象に処理を行う。

・HU：レジスタと高速メモリにアクセスすることができるプロセッサで、レジスタと高速メモリ間のデー

タの転送を行なう。

・MU：高速メモリと主メモリをアクセスすることができるプロセッサで、高速メモリと主メモリ間のデータの転送を行なう。



この構成で、PUが必要とするデータは、必要とされる時間までに主メモリから高速メモリへMUメモリ操作プログラムの実行によって移され、高速メモリに置かれたデータはHUメモリ操作プログラムの実行によって、レジスタに格納される。必要でなくなったデータは、この逆の手続きを経て、主メモリの格納される。

2.2 プログラムによるメモリ階層制御システム (PCHMS) の基本動作

図2-2に、本コンピュータのプログラムを示す。ここで、円が命令、矢印が命令の先行関係を表し、複数本の矢印が入力されている命令は、その上の命令の実行後に実行可能となることを表す。左側からそれぞれPU、HU、MUのプログラムである。ここで、HUとMUで用いる命令を定義する。

・HU 操作命令

LDR：高速メモリの内容をレジスタにロードする。

STH：レジスタの内容を高速メモリに格納する。

・MU 操作命令

LHM：主メモリから高速メモリにロードする。

SHM：高速メモリから主メモリに格納する。

また、EXEはデータを必要とする命令である。

3. 基本動作

PUで命令が実行される前に、MUによる8の命令LHMで主メモリからの高速メモリへデータを転送し、そのデータをHUが4の命令LDRで高速メモリからレジスタへ転送する。次にPUがそのデータを処理する。その後、必要であればデータはHUによる6のSTHの

*Program Controled Hierachical Memory System

†Graduate School of Information Systems, The University of Electro-Communications

‡MAKI Nobuhiro

§SOWA Masahiro

命令で高速メモリに書き戻され、MUによる命令10のSHMで主メモリに書き戻される。

通常はレジスタにPUの必要とするデータがあるが、問題によってはデータが無く、PUが待状態になることがある。転送するための十分な時間がない場合である。処理プログラムを解析する時点で、PUが必要とするデータを決定することができない場合、処理プログラム実行時に、必要なデータが決定されしだいレジスタにそのデータを転送することになる。図では、命令9、7、3がそのことを表している。命令7、3で示すように、データの使用時間と更新時間の間隔が短くなると、PUのデータ待ち状態になる。このPUがデータ待ち状態になることをデータミスと呼んでいる。

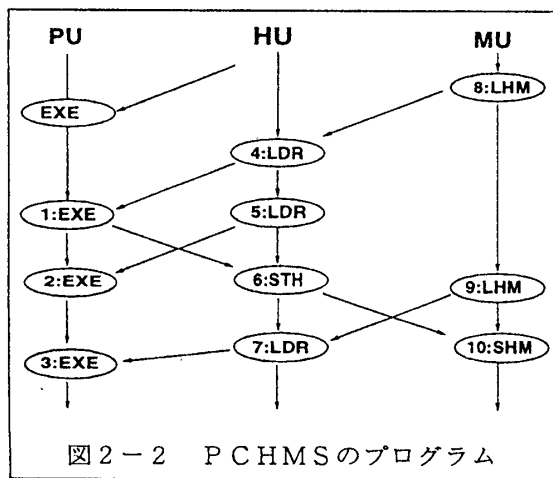


図2-2 PCHMSのプログラム

4. PCHMSについて

PCHMSは、静的に実行プログラムを解析することにより操作プログラムを生成するために、従来のキャッシュシステムと比べて以下のようなすぐれた特徴をもつ。

[1] キャッシュミスという概念がない。

コンパイラにより静的に解析し、極力レジスタにデータがあるようにプログラムを作ることができる。そのため、キャッシュメモリ参照時にデータが存在しないということをほとんどなくすることができる。

[2] PUのデータ待ち遅延時間を低減することができる。

従来のキャッシュシステムは、キャッシュにアクセスした時に初めてデータの有無を判定することができ、そのためキャッシュミス時にはプロセッサが待ち状態に入り、処理が止まってしまった。それに対し、PCHMSは、キャッシュミスが起こってからデータ転送を始めるのではなく、PUの必要とするデータが分かり次第、主メモ

りからのデータ転送をすることができる。これは、PUのデータ待ちの遅延時間を低減することができる可能性が大きい。

[3] 処理プログラムが短縮される

このことは、それぞれのメモリが独立したアドレスを持っているため、現行アーキテクチャーの処理プログラムで行っていた主メモリのアドレスに関する計算を行う必要がなくなり、その分処理プログラムは短くなる。

[4] 分岐をすかししないかを前もって知ることができる。静的に処理プログラムを解析するため、分岐するかどうか分岐の前にわかることが多い。

5. おわりに

必要なデータを超高速メモリであるレジスタに置き、それをプロセッサで処理し、レジスタへのデータの転送を、処理プログラムを解析して得られたデータ転送用プログラムの実行によって行う、プログラム制御メモリ階層システムについて述べた。本方式ではデータの使用時期はコンパイル時に調べられているので、キャッシュミスのようなデータが無いという状態は原則的に起こらない。必要なデータがレジスタに無いのは、低速メモリから超高速メモリにデータを移動するのに十分な時間が無い場合だけである。キャッシュメモリとの性能を比較するためシミュレーションを行った。結果は、数列計算の場合、キャッシュメモリのヒット率61%に対してPCHMSでは、100% $((1 - \text{データミス率}) * 100)$ 、バブルソートでは、キャッシュメモリのヒット率53.8%に対してPCHMSは、87.5%が得られている[4]。いずれの場合においてもキャッシュ以上の性能が得られることは確かである。現在大規模なシミュレーションの準備をしている。

参考文献

- [1] 佐藤 正樹、'キャッシュメモリの操作明示に関する基礎研究'、昭和63年度名古屋工業大学卒業論文
- [2] 佐藤 正樹、'並列処理によるキャッシュ操作の明示化'、JSPP'90論文記、情報処理学会(1990)
- [3] JOHN L HENNESSY & DAVID A PATTERSON、"COMPUTER ARCHITECTURE A QUANTITATIVE APPROACH"、MORGUN KAUFMAN PUBLISERS INC. (1990)
- [4] 牧 晋広、'階層化メモリの能動的構成法'、平成4年度名古屋工業大学卒業論文