

可変構造型並列計算機のメモリ・アーキテクチャ

3T-2

蒲池恒彦 岩田英次 村上和彰 福田 晃 末吉敏則 富田眞治
(九州大学)

1 はじめに

現在我々は、128 台の PE (Processing Element) を 128×128 の多重化クロスバー網 (MC-net : Multiplexed Crossbar Network) で接続するマルチプロセッサ・システム「可変構造型並列計算機」を開発している。^[1] 本稿では、まず、本システムが採用した“可変構造”のメモリ・アーキテクチャについて述べた後、各 PE にキャッシュを備えたことにより生じるマルチキャッシュ・コンシステンシ問題への対処方法について述べる。

2 “可変構造”メモリ・アーキテクチャ

2.1 原理

本システムでは 128 台の各 PE に 4MB のローカル・メモリを分散させるメモリ構成を採る。この分散メモリ構成自体は“固定構造”であり、“可変構造”ではないが、各ローカル・メモリを各 PE に単に専有 (dedicate) させるのではなく、複数 PE 間で共有 (shared) させることを可能としている。したがって、メモリの共有/専有を制御することにより、プロセッサが直接アクセス可能な実アドレス空間構成として様々な形態を提供することができる。このように、本システムではこの実アドレス空間

構成を“可変構造”とすることで“可変構造”のメモリ・アーキテクチャを実現している。

2.2 アドレス空間の写像

本システムでは 4GB の実アドレス空間を次の 2 つに分割している。

- ① プライベート (private) 空間 : 下位 2GB の空間であり、それぞれの PE の私有領域である。
- ② コモン (common) 空間 : 上位 2GB の空間であり、全 PE の共通領域である。コモン空間は、各 PE に対応する 256 個の共有メモリ・ウィンドウ (SMW : Shared Memory Window) からなっている (現在 SMW128~SMW255 は未使用)。各 PE はこの SMW にアクセスすることで、MC-net を介して他 PE のローカル・メモリへ直接アドレスによりアクセスすることができる。

SMW の導入により、本システムでは仮想、実、物理の 3 種のアドレスが存在することになるが、これらのアドレス変換は図 1 に示すように 2 段階の変換過程を経ることになる。

- ① 仮想アドレス→実アドレス変換 : 32 ビットマイクロプロセッサ SPARC が出力する仮想アドレスを MMU (Memory Management Unit) のページング機構により実アドレスに変換する。実アドレスがプライベート空間を指していれば

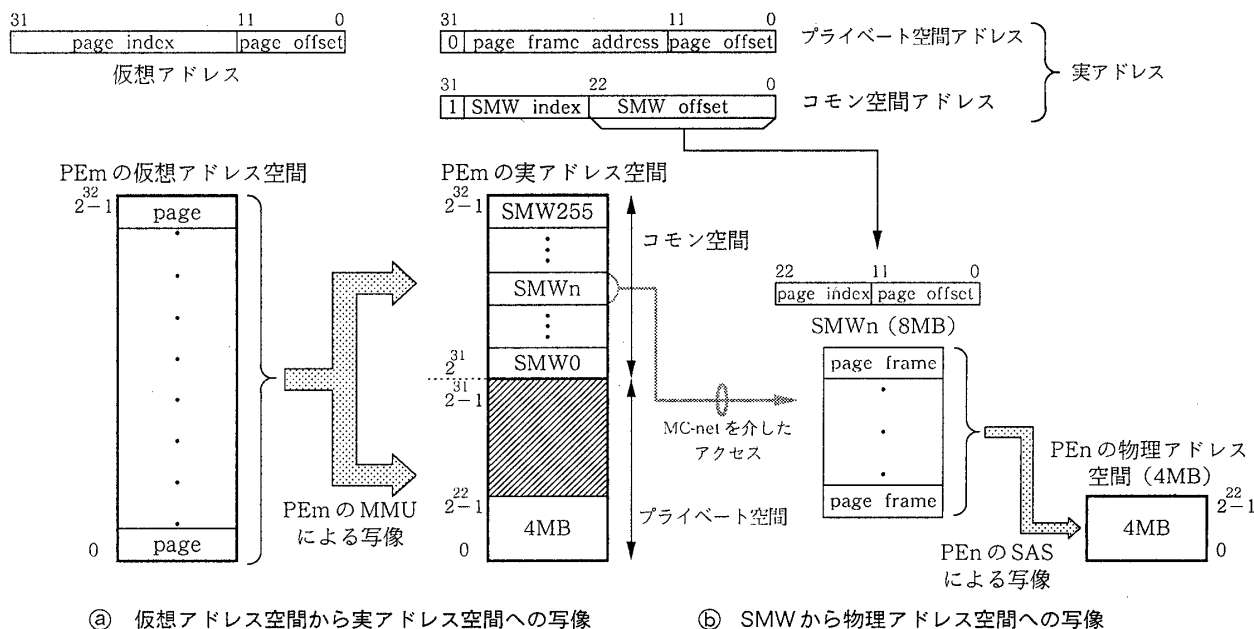


図 1 アドレス空間の写像

Memory Architecture of the Reconfigurable Parallel-Processor

Tsunehiko KAMACHI, Eiji IWATA, Kazuaki MURAKAMI, Akira FUKUDA, Toshinori SUEYOSI, and Shinji TOMITA
Kyushu University

ば変換を終了し、ローカル・メモリ・アクセスを行う。

- ⑤ 実アドレス→物理アドレス変換：実アドレスがコモン空間を指している場合、指定されたSMWnを提供しているPEnへMC-netを介してリモート・メモリアクセス（SMWアクセスと呼ぶ）を行う。PEnのSAS（SMW Access Server）は、受け取った実アドレスをページングにより物理アドレスに変換し、メモリ参照を遂行する。

2.3 マルチプロセッサ結合形態

MMUは、アドレス変換に加えて、コモン空間へのアクセス可／不可をページ・フレーム単位で制御する機能を有する。これにより、多様なコモン空間構成が可能となり、結果として以下のようなマルチプロセッサ形態を採ることができる。

- ① メッセージ交換型疎結合マルチプロセッサ：MMUにより、コモン空間へのアクセスをすべて禁止する。PE間の通信はメッセージ交換によるのみ行える。
- ② メモリ共有型密結合マルチプロセッサ：仮想→実アドレス変換過程においてコモン空間にマッピングすることでメモリを共有する。この時、MMUにより任意のPE間のメモリ共有の有無を制御することで種々の共有形態を実現することができる。また、①でのメッセージ交換機能を同時に用いれば、ハイブリッド結合マルチプロセッサとして密結合／疎結合の両形態を同時に採ることも可能である。

3 マルチキャッシュ・コンシステンシ問題への対処

3.1 基本方針

本システムでは高速・大容量の仮想アドレス・キャッシュを各PEに備え、ローカル／リモート・メモリ・アクセスの高速化を図っている。^[2]しかし、キャッシュを備えたことにより、メモリを共有する場合マルチキャッシュ・コンシステンシ問題が生じる。

マルチキャッシュ・コンシステンシ問題への対処法として

- ④ ブロードキャスト法
⑤ スヌーピング・キャッシュ
⑥ グローバル・ディレクトリ法

などが知られているが、④では各々のキャッシュに書き込みがある度にその書き込み情報を全キャッシュにブロードキャストで知らせねばならず、無効化要求トラフィックが増大すること、⑤は最近のメモリ共有型密結合マルチプロセッサで広く用いられているが、各PE間の結合は共有バス結合であることを前提としており、本システムでは主記憶更新アルゴリズムをミスヒット・ペナルティが小さいストア・スルー方式にし、不要な無効化要求を取り除くことができる⑥グローバル・ディレクトリ法を基本的な方針として採用した。このハードウェア制御に加えて、さらにOSがページ・テーブル・エントリ中に対応するページ内データのキャッシング可／不可を指定できるようにしてある。

3.2 コンシステンシ・ディレクトリ

本システムでは、グローバル・ディレクトリを各PEのメモリ・ユニットに分散して置き、各々をコンシステンシ・ディレクトリと呼ぶ。コンシステンシ・ディレクトリの各エントリはSMW（8MB）内のコンシステンシ・ブロック（32B）に対応し、

以下のフォーマットをとる。

C キャッシング権を持つPE番号（7bit）

C：キャッシング・ビット

$$C = \begin{cases} 1: \text{キャッシング権を持つPEがコピーを持つ} \\ 0: \text{どのPEもコピーを持たない} \end{cases}$$

SASはSMWアクセスに対しアドレス変換を行うと同時に、コンシステンシ・ディレクトリをチェックする。

3.3 キャッシュパーズ・メッセージ

本システムのキャッシュは1ライン（32B）を4つのトランスファ・ブロック（8B）で分割し、各トランスファ・ブロックに対応する4bitの存在ビットを設けている。このためパーズ・メッセージには、アクセスのタイプによってトランスファ・ブロックをパーズするものと、ラインをパーズするものの2種類がある。

SASはコンシステンシ・ブロックが一時には高々1PEにのみキャッシングされるように制御し、キャッシング権を持たないPEからのアクセスに対して以下の手順に従って、キャッシング・ビットの操作、キャッシング権の移動、パーズ・メッセージの作成・送出行う。

- ① ライト・アクセス時：ストア・スルー方式を採用したことでライトアクセスは最大8Bとなるため、キャッシング権の移動、およびキャッシング・ビットの操作は行わず、C=1の時のみ、キャッシング権を持つPEにトランスファ・ブロック・パーズ・メッセージを送る。
- ② リード・アクセス時：C=1の場合は、キャッシング・ビットの操作は行わず、コンシステンシ・ディレクトリのPE番号を書き換え（キャッシング権の移動）、キャッシング権を持っていたPEにライン・パーズ・メッセージを送る。C=0の場合は、C=1にし、コンシステンシ・ディレクトリのPE番号を書き換える（キャッシング権の移動）。

なお、パーズ・メッセージは、キャッシング権を剥奪したPE番号、パーズすべき実アドレスを含んでおり、MSによってMC-netを介して当該PEのキャッシュに送られる。キャッシュはパーズ・メッセージを受け取るとRTB（Reverse Translation Buffer）を実アドレスで検索し、パーズ・メッセージのタイプに従って当該ラインを、もしくはトランスファ・ブロックをパーズする。

4 おわりに

以上、本システムにおけるメモリ・アーキテクチャ、およびマルチキャッシュ・コンシステンシを維持するためのメカニズムについて述べた。現在ハードウェアの試作段階に入っており、1989年末の完成を目指している。

参考文献

- [1] 村上ほか：“可変構造型並列計算機のシステム・アーキテクチャ”，情報処理学会「コンピュータアーキテクチャ」シンポジウム論文集Vol.88, No.3, pp.165-174（1988年5月）
- [2] 濱口ほか：“可変構造型並列計算機のプロセッサ・ユニット”，本大会論文集（1989年3月）