

JUMP-1 MBPコアの命令設計

5K-1

平木 敬

松本 尚

東京大学大学院理学系情報科学専攻

1 はじめに

本報告では現在7大学共同で研究開発を進めている超並列プロトタイプ計算機JUMP-1 [2]の基本構成要素であり、細粒度処理を担当するMemory-Based Processor (MBP) [1, 3]においてプログラムを解釈実行するプロセッサ: MBPコアの命令設計を述べる。

JUMP-1の基本アーキテクチャとして異粒度複合アーキテクチャを用い [4]、また、異粒度複合アーキテクチャの粗粒度部分にはSUN社製SuperSPARCIIを使用する。MBPはメモリに関するアクセス処理、データ転送、同期処理、コンシステンシの維持などの細粒度非局所処理を一義的に担当する。なお、成功する同期処理を含む基本的なアクセス処理、データ転送等速度を要求される機能はMemory-Based Processor内部のハードウェアにより直接実現され、MBPコアで実行される細粒度プログラムでは、同期処理、コンシステンシ維持、メッセージ処理、マクロデータフロー処理など複雑な処理が要求される部分を担当する。従って、MBPコアはコア単体として効率的細粒度・大域プロセッサであることが必要であるとともに、対となって用いられる粗粒度プロセッサおよび他のMBPコアと協調して効率的動作をすることが必須の条件である。

まず、一般的な細粒度・大域処理用プロセッサとして設計上の問題点として、以下のものが列挙される。個々の内容は省略して項目を列挙すると:

1. コンテキスト切替えによる命令オーバーヘッド
2. 命令オーバーヘッドによるネットワーク使用効率の低下
3. 細粒度同期に関する命令オーバーヘッド

MBPにおいては一般の細粒度プロセッサと比較して処理に含まれる粗粒度成分が粗粒度プロセッサに移出するため上記問題点が更に深まると共に、更に下記異粒度複合アーキテクチャの特質として協調的動作が新たな問題点となる:

1. 高レイテンシ高速大容量メモリの利用
2. 静的スケジューリングとの適合性
3. 粗粒度プロセッサとの通信・同期
4. 粗粒度プロセッサと一貫した保護機構

メモリアクセスに関して、局所性の利用による効率化が困難である状況から、データに関して、アクセス時間が大きいメモリを直接使わざるを得ない。この問題点は、内部状態、特にレジスタ個数を小さくする細粒度的要望と相入れない。MBPコアでは主メモリに直結する利点を生かし、処理に対するデータの

局所性でなく、データに対する処理の局所性を利用する方式が必要なのである。

また、MBPコアの主要目的がネットワークを介在した形態での分散共有メモリの実現にあるため、ネットワークバケットがその詳細レベルまで直接命令から操作可能であると共に、ネットワーク転送と命令実行を並行動作を可能とする方式が必要である。

MBPでは処理をハードウェアを用いて高速化するため、多くの処理項目を例外処理という形態で実現する。従って、その高速な実現方法は全体性能に対して大きな影響力を持つ。

2 MBPコアスレッドの環境

MBPコアで実行されるスレッドは、特権レベル、プリエンブションの有無、長期ステートの存在等により分類され、MBPコアスレッドが使用する環境設定方式が異なっている:

カーネル短スレッド MBP内部例外処理(バッファ溢れ、動的遠隔メモリ処理等)、粗粒度プロセッサからのカーネルコールを実現するための短いスレッドでプリエンブションを受けないもの。カーネルレジスタセットを使用し、スレッド終了時に永続的状態をメモリ内容以外残さない。従って、個々の環境設定は不要であり、MBP内に設定されたカーネル用環境およびMBPスレッドエントリ表を用いる。

ユーザ短スレッド ユーザのプログラム等が高機能メモリアクセスプリミティブとして起動する短いスレッド。ユーザのメモリアクセスに付随する同期操作、バケット授受などが含まれる。ユーザ短スレッドもプリエンブションを受けず、例外処理を誘起しない限り一定時間内で終了する必要がある。ユーザスレッドでは、環境を保護の単位であるジョブ毎に持つ必要がある。

ユーザ長スレッド ユーザのプログラムなどが起動する長いスレッド。途中で(1)ユーザの指定により、永続的状態を持って中断するか、(2)システムの介在により、永続的状態を持って中断するもの。MBPスレッドのスケジューリングは2段階スケジューリングを用いるためユーザレジスタの退避等はユーザ処理に委ねられる。ユーザ長スレッドでもユーザ短スレッドと同様ジョブ単位的环境管理が行なわれる。

従って、MBPコアが備えるレジスタセットはカーネル短スレッド用1組およびユーザ用最大4組(クラスタ内粗粒度プロセッサの個数に対応、JUMP-1におけるインプリメンテーションでは3組)である。なお、組み合わせ的にはカーネル長スレッドが存在するが、途中で中断する可能性のあるカーネルレベルのスレッドはスケジュール時における優先度を除きユーザ長スレッドと同等であり、ユーザ長スレッド用レジスタを使用し、専用レジスタセットは不要である。

ジョブ毎のユーザ環境はジョブ識別子を用いて管理されるが、ジョブ識別子自身は粗粒度プロセッサの持つコンテキストの一

Instruction-set design of an MBP-core processor

K. Hiraki, T. Matsumoto

Faculty of Science, University of Tokyo

7-3-1 Hongo, Bunkyo-ku, Tokyo, 113 Japan

E-mail: hiraki@is.s.u.tokyo.ac.jp

部として、MBP内に設定される。ユーザ長スレッド生起時には、ジョブ識別子が指すジョブ構造体中のMBPスレッドエントリ表により開始番地が指示される。

3 MBPコアスレッドの生起・終結

MBPコアで実行されるスレッドは、(1)粗粒度プロセッサからのクラスタ内メモリに対する高機能メモリアクセスまたは明示的MBPコア呼び出し、(2)粗粒度プロセッサからのクラスタ外メモリに対する高機能メモリアクセスまたは明示的MBPコア呼び出し、(3)到着したネットワークパケット、(4)MBP内部でのMBPコア呼び出し、(5)MBPコアスレッドからの分枝、により生起される。このなかで(5)はプログラム処理そのものであり、別段の配慮は不要である。

3.1 粗粒度プロセッサからの生起・終結

粗粒度プロセッサからのMBPスレッドの生起はすべてメモリアクセス命令で行なわれる。JUMP-1では粗粒度プロセッサにSuperSPARC IIを使用するため、メモリアクセス命令による指定は、メモリアドレスの一部(物理アドレス中上位4ビット)、メモリデータ長(4種)、書き込みデータ(swap, storeのみ)およびアクセス種(load, store, swap)だけで行なう必要がある。また、呼び出しに付随するパラメータは操作対象アドレスに加え、swap, storeの場合はデータ部分が、明示的MBPコア呼び出しの場合には粗粒度プロセッサとMBPで共有するパラメータページ上の連続領域に置かれる。共有するパラメータページへのポインタもジョブ識別子の一部としてMBPにおかれる粗粒度プロセッサコンテキストの一部を成す。

MBPスレッドの生起に用いられるメモリアクセス命令のアドレスは、操作対象を常に示す。操作対象が当該クラスタ外(メモリアドレス上位ビットで指示)の場合は、カーネル短スレッドが呼び出され、粗粒度プロセッサからのMBP生起アクセスがMBP生起パケットに転換・送出される。操作対象がクラスタ内メモリの場合は、MBPコア呼び出しのパラメータを取り出し、システムまたはジョブ構造体に含まれるユーザエントリ表からの先頭アドレスを加えてMBPコンティニューエーションを作成し、空きレジスタセットがある場合には直ちに、ない場合にはMBPコンティニューエーション・バッファに入れる。MBPコンティニューエーション・バッファに空きがない場合にはカーネル短スレッドが呼び出され、クラスタメモリに作成されるキューにつながる。

MBPスレッド終了時の返値は、パラメータと同様パラメータページを用いて行なわれる。

3.2 ネットワークからの生起・終結

ネットワークから起動される場合にはヘッダ部分によりジョブ識別子を介して実行されるプログラムが指定されMBPコンティニューエーションに転換される。その後の処理は粗粒度プロセッサから直接生起される場合と同様である。また、返値はネットワークへのパケットを経由してもとのパラメータページへと戻される。

3.3 MBP内部からの生起・終結

MBP内部から生起されるスレッドはデッドロックを避けるため全てカーネル短スレッドとして処理される。MBPスレッド生起の原因に従ってMBPコンティニューエーションが生成され、現在動作中の短スレッド終了時にスケジュールされる。

なお、MBPスレッド生起・終結の概要を図??に示す。

4 MBPコアの命令設計

MBPコアは作業用レジスタファイルとして、ユーザ用に64ビット8本のレジスタセットを2組、カーネル短スレッド用に4本のレジスタセットを1組持つ。ユーザ用およびカーネル

用ともにレジスタセットは、プログラムで陽に指定しないかぎりMBPスレッドに固定され、レジスタが利用できない限り新たなMBPスレッド実行を開始することは無い。なお、1本のレジスタは70ビット構成であり、64ビットのデータおよび同期タグ、保護タグを保持する。

一方、作業用レジスタファイルではないが、スレッド環境としてレジスタアドレスが付加されているものとして(1)MBPコンティニューエーションレジスタ、(2)MBPジョブ構造体レジスタ(フリーリストポインタ、スレッドエントリ表へのポインタ、実行キューポインタを含む)、(3)2組のメモリアクセスバッファ、および(3)MBPステートレジスタが使用可能である。また、レジスタとしてはアクセス不可能であるが、命令からインプリシットにアクセスできるレジスタとしてネットワークパケットレジスタがあり、パイプライン的なネットワーク操作に利用される。

MBPの命令形式は所謂RISC系形式に並行して動作するメモリアクセス制御、同期制御、および条件制御を並列指定する形式であり、簡略化したLIW形式である。命令セットは月並なものに加え、MBPコアに特徴的なものとして、次のものを含む。

パケット送信命令 メモリバッファ、コンティニューエーションレジスタ等からパケットを合成・送出する

条件分岐付きload/store full/empty bitに基づく条件分岐。ヒープ領域アクセスに用いる

ビットスキャン命令 語内のビット位置の走査。priority queue 実現に用いる

ハッシュ命令 高速なハッシュ表アクセスを実現する。分散共有メモリ管理等に使用

特殊レジスタ転送命令 各種バッファおよび表とのデータ転送用。カーネル短スレッド専用

カーネル短スレッド呼び出し命令 カーネル用コンティニューエーションを作成し、カーネル短スレッドを生起

5 おわりに

JUMP-1のMBPコアの方式および命令設計を述べた。汎用細粒度処理装置と比較し、メモリ、ネットワークの操作と同期操作を円滑に接続する。MBPは現在設計・製作中である。なお、本研究の一部は文部省科学研究費(重点領域研究「超並列原理に基づく情報処理基本体系」)による。研究機会の授与とMBPおよびSMS開発のサポートに対して、重点領域研究研究代表者の東京大学工学部田中英彦教授ならびにハードウェア班班長の京都大学工学部富田真治教授に深く感謝の意を表します。また、お世話になっている同重点領域研究プロジェクトの関係者の皆様に感謝いたします。

参考文献

- [1] 松本 尚: 局所処理と非局所処理を分離並列処理するアーキテクチャ. 第43回情報処理学会全国大会論文講演集(6), pp.115-116 (October 1991).
- [2] 文部省重点領域研究「超並列原理に基づく情報処理基本体系」第2回シンポジウム予稿集, 1993年3月.
- [3] 松本 尚, 平木 敬: Memory-Based Processorによる分散共有メモリ. 並列処理シンポジウム JSP'93 論文集, pp.245-252 (May 1993).
- [4] 平木 敬, 松本 尚: 対称粒度の異なる処理装置を複合した並列アーキテクチャ. 信技報, CPSY 93-13, pp.1-8 (August 1993).