

論理型言語向き並列マシンKPRの
ORリダクション・ユニット

4B-4

柴山 潔・加納 健・実藤隆則・萩原 宏

(京都大学 工学部 情報工学教室)

1. はじめに

KPRのORリダクション・プロセッサ(ORP)は、図1に示すように、ANDリダクション・プロセッサ(ARP)と対で要素プロセッサを構成し、両者がCM (Communication Memory) および LEMM (Local Environment Memory Module)、ネットワーク・インタフェースを共有する方式で密結合されている。KPRでは、並列論理型言語によるプログラムの実行をAND/OR木の展開・縮退過程とみなし(「並列リダクション・モデル」)、その木のORノードから展開されるOR関係にある子ノードを管理する機能を「Orプロセス」と呼び、これをORPが主に処理する。

ORP自身は、(1)プロセスの管理を行うPCU-O (Process Control Unit)、(2)Orプロセスにおけるリダクション操作を行うORU (Or Reduction Unit)、(3)ORPを制御するプログラムを格納しておくTMM-O (Template Memory Module) の3ユニットから構成される。本論文では、このうちORUのハードウェア構成について述べる。

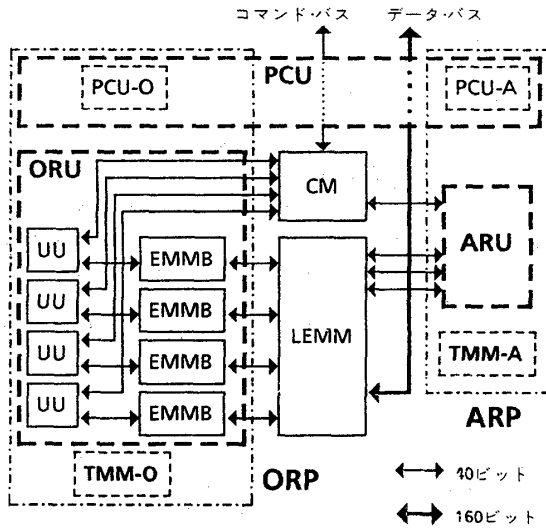


図1 KPRの要素プロセッサのハードウェア構成

2. ORUのアーキテクチャ

2.1 ORUの処理方式

ORUは、4個のユニフィケーション・ユニット(UU)と、それらの制御部から構成される。この外に、対のARUと共有しているファシリティとしてCMとLEMMがある。(図1参照) ORUは、2レベル・マイクロプログラム方式で制御され、TMM-Oには実行に先立ちあらかじめSVP (Supervisory Processor)によって並列論理型言語からコンパイルされた(主に節頭部に関する)マイクロプログラムが格納されている。このマイクロ命令は、各UUの制御記憶に格納されているナノ命令列の先頭アドレス・フィールドとマイクロ命令シーケ

ンス制御フィールドなどから成る。

ORUにおける主な処理機能は次のようなものである。

1) 並列ユニフィケーション……ORUでは、4個の節とのユニフィケーションあるいは組み込み述語の処理を並列に行う。OrプロセスにおけるOR並列性は、この並列ユニフィケーション機能によって実現されている。1個のOrプロセス処理において、最初の最大4個の候補節とのユニフィケーション命令列は、TMM-Oにあるマイクロプログラムとして4個のUUにそれぞれ静的に割り付けられているが、候補節が5個以上ある場合には、前のユニフィケーション処理が終了したUUから順に、5個目からのユニフィケーション命令列が動的に割り付けられる。このために、ORUはカウンタにより候補節数を管理する。すなわち、UUに節が1個割り付けられるごとにカウンタを1減らし、カウンタ値と定義体(同じ述語名を持つ節の集合)番号とによってマイクロプログラムの先頭番地を計算する。

2) UUの管理……ORPで処理すべきメッセージのうち、ARP/ORPと授受されるfailイベントはPCU-Oだけで処理されるが、UUが直接処理対象とする環境情報(EMMに格納されている)を核に構成されるinvokeデマンドのUU内への読み込みやUU内からの書き出しの管理と、それに関連するUUでの処理の起動や完了通知は、ORUがPCU-Oの制御下で行う。また、KPRの実行モデルでは、イベント・メッセージの飛び越し機能を実現しているため、ORPはsuccessイベントを受け取ることはない(飛び越されて祖先プロセスに直接転送される)。従ってORPでは、ARPのようにある時点でのプロセスの環境情報を保存する機能は必要ない。

3) EMMの管理……ARPやORPからのinvokeデマンドのうち、環境情報(節外部における未束縛変数に関する束縛情報である外部変数情報と、実際にどの外部変数の解を返して欲しいのかを示す解指定情報、および構造データ用ヒープ)はEMMに格納されているので、UUにおけるユニフィケーション処理に先立ち、EMMから各UUへ転送される。また、UUによって処理された環境情報はORPやARPへのデマンドを構成するためにUUからEMMに転送される。この時、ARPへのデマンドとしての環境情報には、変数束縛情報(節内部の変数に関する束縛情報)も付加される。

2.2 UUの処理方式

UUは、TMM-O内にあらかじめコンパイル・格納されているマイクロプログラムと各制御記憶内のナノプログラムの制御の下に、CM内の引数情報とEMM内の環境情報を用いて、ユニフィケーションを実行し環境情報を更新する。4個のUUは、同じ引数情報と環境情報を使うが、TMM-O内の相異なる4個のマイクロ命令によって起動された異なるナノ命令列によって、並列ユニフィケーションを実行する。

環境情報は各UUによって更新されるので、各UUごとにそのコピーを持つ必要がある。また、LEMMはARPからも頻繁にアクセスされる(ARPはLEMMを主記憶とみなしてア

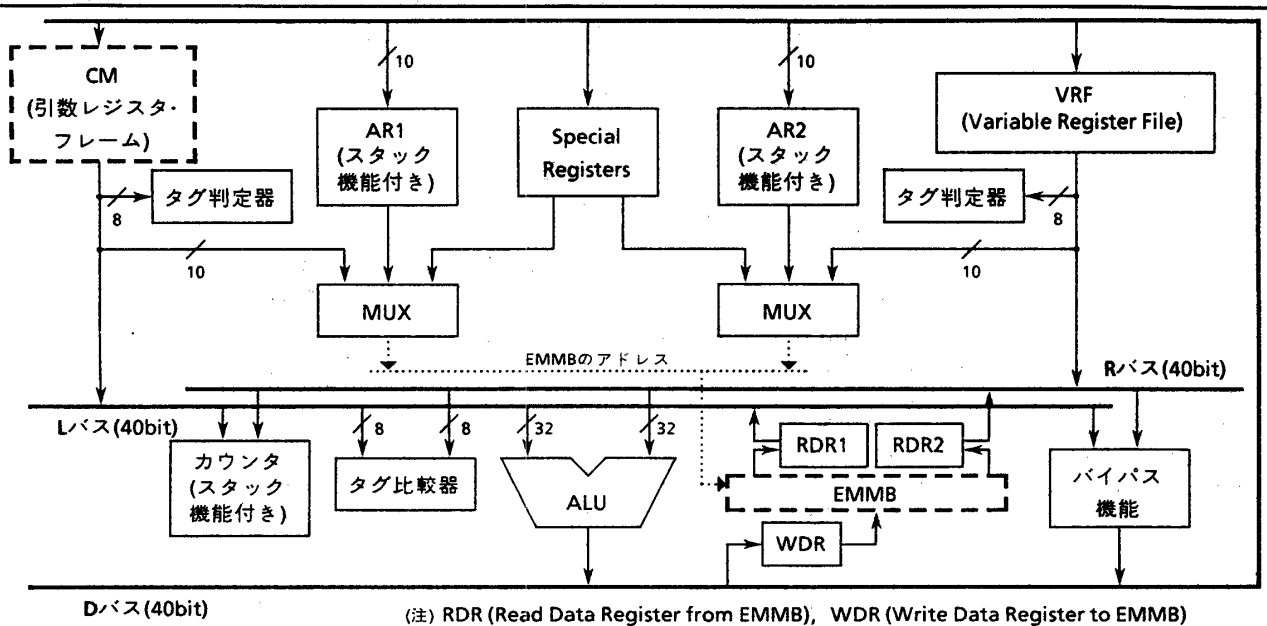


図2 UUのハードウェア構成

アクセスするので、UUは直接EMMにアクセスするのではなく、図1に示すように、バッファを介して間接的にアクセスする方式を採用している。ユニフィケーションに先立ちEMM内の必要な環境情報をEMMバッファ(EMMB)と呼ぶバッファ内にコピーし、ユニフィケーションが成功すると*invoke*デマンド送出要求のためにEMMB内のデータを所定のEMMにブロック転送する。

3. ORUのハードウェア構成

3.1 UUとCMのインタフェース

UUはCMを引数レジスタ・フレームとして直接アクセスする。(図2参照) CMは4個の同一コピーを持つ(4重構成)方式により、4個のUUごとに独立した読み出しポートを実現している。4個のCMの読み出しポートは各UUのLバスに直結され、また書き込みは各Dバスからマルチプレクサを介して、すべてのCMにブロードキャスト転送される(書き込み競合はアービトレーションする)。CMは固定長ページごとにフレーム化され、プロセス起動時にPCUが割り付けたページ番号をベース・アドレスとしている。CMのページ管理(アドレス変換を含む)はPCUが行う。

なお、CMは4Kワード(1ワード=40ビット)を実装する予定である。

3.2 UUとLEMMのインタフェース(EMMB)

環境情報は、ORUがEMMBに読み出し(ブロック転送)、各UUに渡す。UUは、EMMB内のデータをCM内に引数情報として格納されているアドレスをもとに、データ・レジスタ(図2に示したRDRやWDR)を介してアクセスする。従って、UUからCMは直接アクセス用、EMMBは間接アクセス用のそれぞれ大容量レジスタ・ファイルに見える。ユニフィケーションが成功した場合には、UUはORUにEMMB内のデータをEMMに書き込むことを要求する。ユニフィケーションが失敗したEMMB内のデータはそのまま捨てられる。

EMMBは各UUごとに備えられているダブルバッファであり、UUが一方のバッファを使用してユニフィケーションを行うのと並行して、EMMとEMMB間のデータ転送を行うことができる。また、EMMBはバンク化(4バンク)され、各バンクごとに40ビット、EMMB全体では160ビット(4ワード)幅のポートを備えている。

また、ユニフィケーションの性質上、各UUは同時にEMMBの相異なるアドレスの2個のデータを読み出すことが

できることが望ましい。このため、各EMMBは2個の同一コピーを持つ2重構成としている。

ダブルバッファとすることによりUUとEMMによるデータ・アクセスを並行して行うことを可能にし、さらにマルチコピー構成方式で各UUにおけるユニフィケーション処理時の並列オペランド・フェッチを、それぞれ実現している。

なお、EMMは256Kワードを、EMMB(1バッファ分)は1Kワードを、それぞれ実装する予定である。

3.3 UUのハードウェア構成

ORUは4個のUUを単一クロックで同期制御するが、各UUはORUのマイクロ命令によって起動されたナノプログラムを独立に実行する。ORUはタグ・アーキテクチャを採用しているため、UUの順序制御も動的に決められることが多い。このために、UUによるタグの組み合わせにより必要な処理プログラムを高速に判定する機能(タグ比較器)を持たせている。(図2参照)

また、変数のデレファレンス時にはタグやデータの多重間接アクセスが頻繁に生じるので、EMMBにはデレファレンスのための専用ハードウェア機構(タグによる読み出しデータのアドレス・バスへのバイパス機構)を備えた。

ネストしたリストや複合項間のユニフィケーションのために、EMMBのアドレス・レジスタ(AR)やカウンタにはスタック機能が付けてある。(図2参照)

UU内のファンクション・ユニットとしては、タグ比較器の他に、汎用ALUや、バスのバイパス機能などがあり(図2参照)、これらを水平型ナノ命令により同時に制御する低レベル並列処理方式を採用している。

4. おわりに

KPRの並列リダクション・モデルが対象としている言語はOR並列型言語である。しかし、実用化を目指して提案されている並列論理型言語のほとんどはOR並列性を何らかの形で制限するAND並列型言語である。従って、これらの言語へのKPRの適応性についてもソフトウェア/ハードウェアのトレードオフの観点から考察を加える必要がある。

UUは、ORP内に4個あるので、できるだけ小規模なハードウェアで実現しなければならない。また、要素プロセッサの一部として、そのVLSI化の可能性についても考察する必要がある。