

論理型言語向き並列マシンKPRの ANDリダクション・ユニット

4B-5

実藤隆則・加納 健・柴山 潔・萩原 宏

(京都大学 工学部 情報工学教室)

1. はじめに

本稿では、KPR¹⁾を構成する要素プロセッサである、ORP (Or Reduction Processor)、ARP (And Reduction Processor)、SVP (SuperVisor Processor)、DBP (DataBase Processor)のうち、ARP、特にその処理ユニットであるARU (And Reduction Unit)の構成について述べる。

ARPでは、KPRで実行されるAND/OR木の展開・縮退過程(並列リダクション・モデル)のうち、AND関係にある子ノードの管理を行う機能である、Sプロセス(Streamプロセス)の処理を実行する。Sプロセスでは、子プロセスの実行(解の生成)をパイプライン的に行うストリーム並列処理方式を採用している。この方式で必要となる、ストリームの状態としての環境情報は、サブプロセス情報(SPI; SubProcess Information)と呼ばれ、LEMM(Local Environment Memory Module)に保持される。

ARPにおける仕事は大きく分けて次の2つであり、それぞれに対応する機能ユニットを持っている。

(1) プロセッサ上で並行に実行されるプロセスの管理……デマンド(親プロセスからの要求)によるプロセスの生成やプロセス(プロセッサ)間通信の管理はこれに含まれ、PCU-A(Process Control Unit for Arp)が処理する。

(2) Sプロセスそのものの実行……ストリーム並列処理として、本体ゴールを流れる複数のストリームの生成・更新がARUによって実行される。

PCU-AとARUは非同期に動作する独立した機能ユニットであるが、PCU-AからARUへの起動要求や、ARUからPCU-Aへのメッセージ送出要求などにより、互いに協調して動作する。

ARPは、これらの機能ユニット、および、SVPによってコンパイルされたプログラムを格納し、ARUの制御記憶とも呼べるTMM-A(Template Memory Module for Arp)によって構成されている。

2. ARUの処理方式

2.1 ARUにおける処理の概要

ARPにおける一つの節本体の処理は、親プロセスからのデマンドによって開始される。

(a) デマンド処理……親プロセスからのinvokeデマンドは、節内の変数に関する変数束縛情報、親プロセス内での変数に関する外部変数情報、解の構成に関する解指定情報および、構造データ用ヒープからなる、環境情報を含んでいる。ARUでは、この環境情報をもとに、先頭ゴールの引数情報をCM(Communication Memory)およびLEMM上に作成し、PCU-Aにinvokeデマンドの送出を要求する。また、この環境情報は、解ストリームの出発点となるSPIとして、LEMM上に保存される。

この後は、起動した子プロセスからのイベント(success/failイベント)によって処理が進められる。failメッ

セージでは環境情報の更新は不必要であり、子プロセスの管理やLEMM上の領域管理などがPCU-Aによって実行される。ARUで処理されるのは、successイベントのみである。

(b) イベント処理……子プロセスからのsuccessイベントには、解情報が含まれている。この解情報によって、対応するSPIから、新たなSPIを作成する。また、ゴールに引き続く組み込み述語が存在すれば、新しいSPIに対してこれを実行する。

次に、イベントが最後尾ゴールからのものであれば、更新されたSPIを用いて、解情報を作成し、PCU-Aにsuccessイベントの送出を要求し、この節の処理を終了する。このとき、解情報作成のための解指定情報は、SPI中に含まれている。

それ以外のゴールからのイベントであれば、更新されたSPIを、LEMM上に割り付ける。また、デマンド処理と同様にして、次のゴールに対する子プロセスを起動し、次のイベントを待つ。

2.2 ARUの処理機能

前節で述べたように、ARUが行う処理は、各種のデータの作成・更新であり、以下の4種類のものがある。

1) 環境情報からの引数情報の作成。

[環境情報1] → [引数情報]

2) 解情報による環境情報の更新。

[環境情報0] [解情報] → [環境情報1]

3) 組み込み述語による環境情報の更新。

[環境情報1] → [環境情報1]

4) 環境情報からの解情報の作成。

[環境情報1] → [解情報]

これらのうち、1)および3)の処理は、プログラムのコンパイル時に解析され、TMM-A中に格納されている命令に基づいて実行される。一方、2)および4)の処理は、扱うデータによって動的に決定される。

ARUにおいて、これらの処理を行うために必要となる基本操作としては、以下のようなものがある。

(1) ポインタによる高速多重間接参照。

(2) データの種類(タグの値)に応じた多方向分岐。

(3) 構造データのたどり(スタック操作)。

(4) 複数のデータに対する同時アクセス。

(5) 複数の基本操作の並列実行。

(6) 組み込み述語の処理のための算術論理演算。

(7) PCU-Aとの通信。

ARUではこれらの操作が、効率良く実行可能でなくてはならない。

3. ARUのハードウェア構成

3.1 バス構成

ARUのハードウェア構成を、図1に示す。

ARUは、Dバス、Aバス、DAバスの3系統のバスを持

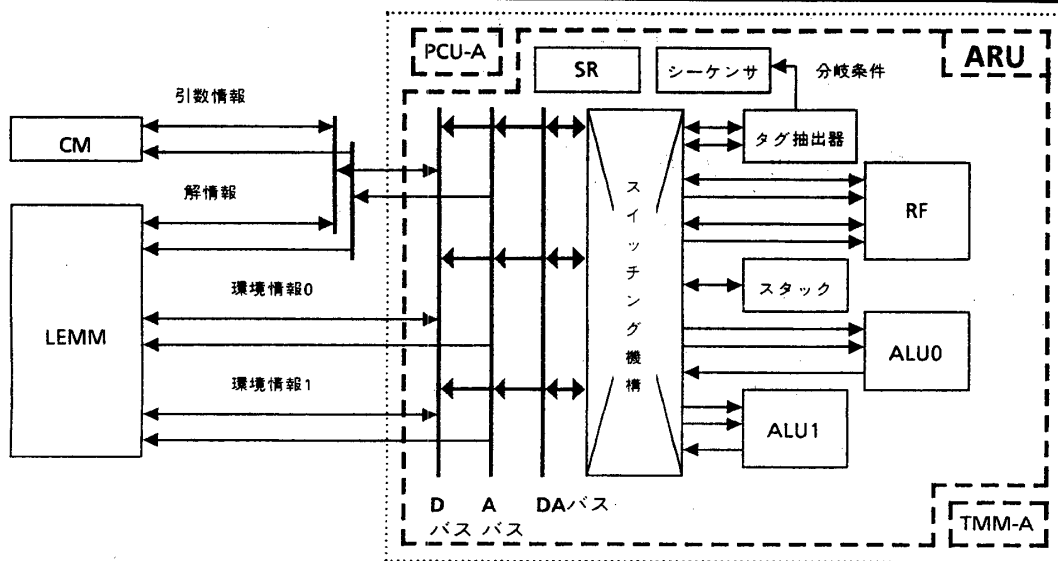


図1 ARUのハードウェア構成

つ。各種のファシリティは、これらのバスに、スイッチング機構を通して接続されている。

ARUの操作するデータとしては、CM上の引数情報、EMM上の(新・旧)環境情報および解情報の4種類がある。ARUでの処理では、これらのうち同時に3個までをアクセスする場合がある。これに対応して、各バスは、3重の構成になっている。すなわち、新・旧の環境情報用および引数情報・解情報用に1個ずつである。

Dバスには、これらのデータをアクセスするためのデータレジスタが接続され、Aバスには、アドレスレジスタが接続される。DAバスは、データのアクセスとは関係なく、ファシリティ間のデータバスを提供するものであり、従って特に3重構成にする必要はないが、ファシリティが有効に並列動作するためには、少なくとも2重構成にする必要があると考えている。

3.2 ハードウェア・ファシリティ

前節で述べたARUの基本機能を実現するために、以下のようなハードウェア・ファシリティが、バスに接続される。

・RF(Register File)・・・汎用レジスタ・ファイルであり、データ操作における各種のポインタ・データの値を保持する。デュアル・ポートRAMを使用し、同時に2個のレジスタにアクセス可能にする。

多重化されたバスの1本をデータ・アクセスに、1本をアドレス計算に用いることにより、ポインタのたどりを高速に(サイクルごとに)行うことができる。

・タグ抽出器・・・バスのタグ部の値に、指定された論理演算などを行うもので、PALにより構成する。その結果を用いて、シーケンサにより、多方向分岐を行う。

・ハードウェア・スタック・・・レジスタの値の待避用である。このデータ用スタックと、シーケンサに内蔵されているコントロール・スタックを用いることで、再帰的な構造を持ったデータに対して、効率の良いアクセスを行うことができる。

・ALU・・・アドレス計算や組み込み述語の計算を行うもので、2個装備する。多重化されたバス構成と組み合わせることにより、複数のデータを並列に処理することが可能となり、これによって処理効率の向上を図る。

・SR(Special Registers)・・・PCU-Aとの通信に関して、実行制御(ARUの起動など)は、PCU-Aが直接ARUのシーケンサを制御することによって行う。それ以外の情報(送出する

メッセージの種類など)については、SRを通してPCU-Aと通信する。

以上の各ファシリティは、PALによって構成されるスイッチング機構によって、入力・出力とも、すべてのバスに任意に接続可能となっている。このようにすることで、ARUの処理に内在する並列性を、ファシリティ・レベルで抽出することを狙っている。

なお、RF、ALUおよびシーケンサには、AMDの29300シリーズを用いる予定である。

これらのファシリティは、TMM-Aに格納したプログラムによって制御する。TMM-Aに格納されるプログラムの形式としては、基本機能に対応したマクロ命令セットを設定して、それによるプログラムを格納する方式をとることも可能である。しかしその場合、命令のデコードやディスパッチの処理がオーバーヘッドになりかねないこと、マクロ命令ごとに処理が分割されることにより、ハードウェア機能としての並列性が十分に引き出せなくなるおそれがあること、などの理由により、KPRのTMM-Aには、定義体テンプレートとして表現された水平型マイクロ命令を直接格納するという方式を採用する。

4. おわりに

KPRは、現在開発を行っているプロトタイプではプロセッサ5台(要素プロセッサ-ARPとORPの対-2対4台、DBP兼用のSVP1台)の構成となっているが、最終的には、要素プロセッサを512対まで増やすことを目標にしている。これを可能とするためには、個々の要素プロセッサのハードウェア・コストの削減が必要不可欠である。特に、ARUでは、バスとファシリティとのスイッチング機構の部分がかなり大きなウエイトを占めると考えられるが、この部分を、処理効率をあまり低下させない範囲でいかにハードウェア量を減少させるかが問題となる。さらに、要素プロセッサとしてのVLSI化の可能性の検討も含め、これらのための考察を、今後行う予定である。

参考文献

- [1] 柴山ほか:「論理プログラミング指向並列リダクション・マシン KPRのアーキテクチャ」, 電子通信学会・技術研究報告, EC85-70 (1986).