

## KPRのORリダクション・プロセッサ(ORP) における単一化処理

3N-8

加納 健、柴山 潔、萩原 宏

(京都大学・工学部)

### 1. はじめに

論理型言語で記述されたプログラム(論理プログラム)の実行は、「質問」として入力されたゴール列を根とするAND/OR木の各ノードにおける単一化処理に相当する。KPRでは、AND/OR木の各ノードにプロセスを割り付け、AND/OR木の展開・縮退および単一化処理を並列に実行する。

AND関係にあるノードにはS(Stream)プロセスを、OR関係にあるノードにはO(Or)プロセスを割り付ける。また、OR関係にあるノードのうち、「事実」に対応するノードにはD(Database)プロセスを割り付ける。このうち、O/Dプロセスでは単一化計算が主な処理となる。

プロセス間通信は、主に親子関係にあるプロセス間で行われる。通過メッセージとしては、子プロセスの起動を要求する「デマンド」、親プロセスに解を返す「successイベント」、親プロセスに解がないことを伝える「failイベント」の3種類がある。

### 2. KPRにおける論理プログラムの実行

KPRは、2分木状のネットワークの葉の部分に要素プロセッサを結合したマルチプロセッサ・システムである。要素プロセッサは、Sプロセスを処理するANDリダクション・プロセッサ(ARP)とOプロセスを処理するORリダクション・プロセッサ(ORP)から構成されている。ネットワークの根の部分にはスーパーバイザ・プロセッサ(SVP)が結合されており、システム全体の管理や入出力処理などのフロント・エンドとしての処理を行う。

ユーザ・プログラムは、SVPでARP用のマイクロプログラムとORP用のマシン命令列にコンパイルされ、各要素プロセッサ内のTMM-A (Template Memory Module)とTMM-Oにそれぞれロードキャストされる。SVPは、ユーザの入力した質問をOプロセスへのデマンドに変換し、ORPに送出する。ユーザの質問が複数のゴールから構成される場合は、SVPがそれらをSプロセスとみなして自分自身で処理する。ORPでは、節頭部の単一化処理と、本体ゴール中最も左にあるユーザ定義述語よりもさらに左にある組み込み述語の処理を行う。ARPでは、残りの本体ゴールの処理を行う。

### 3. ORPによる単一化処理

ORPは、Oプロセスの管理を行うプロセス制御ユニット(PCU-O)と、ORリダクション処理を行うORリダクション・ユニット(ORU)から構成されている。ORUには4個のユニフィケーション・ユニット(UU)が装備されており、4個の候補節を同時に処理できる。UUへの候補節の割り付けは、コンパイラが作成する節テーブル順に行われる。最初の4個の節は、その順序通りUU<sub>0</sub>~UU<sub>3</sub>に割り付けられる。残りの候補節は、処理の終了したUUに順次割り付けられる。

ORUは、PCU-Oからの処理要求を受け取ると、EB (Environment Buffer)への環境情報の書き込みと、ARF-R (Argument Register File)への引数情報の書き込みをPCU-Oに依頼する。環境情報と引数情報の書き込み終了後、デマンド内の述語IDをもとに各候補節の処理プログラムのTMM-O内の先頭番地を読み出す。そして、それを各UUのTMM-Oアドレス・カウンタ(TAC)にセットし、各UUを起動する。

UUにおける単一化処理が成功して終了した場合、そのUUはデマンドまたは、イベントの送出をPCU-Oに依頼する。すべての候補節の処理が終了した時点で、子プロセスが存在していなければ、PCU-Oが親プロセスに対しfailイベントを送出する。子プロセスが存在する場合には、すべての子プロセスからのfailイベントの到着を待って、親プロセスへfailイベントを送出する。

### 4. ORPのマシン命令

KPRでは、論理型言語プログラムを一旦WAMの命令機能レベルにあるマシン命令列にコンパイルする。ARP用のマシン命令列は、ARPのマイクロ命令に変換後TMM-Aに格納され、直接実行される。ORP用のマシン命令列は、そのままTMM-Oに格納されており、これをUU内のマイクロプログラムが解釈・実行する。ORPのマシン命令は、UU内のマイクロプログラムのアドレスの一部となるオペレーション部と、処理に必要な情報を格納するオペランド部からなる。

UU内のマイクロプログラムはユーザ・プログラムに依存しない。従って、プログラムで使われているアトムや整数などの定数リテラルはマシン命

令によってUU内に持ち込む必要がある。UUでは、定数リテラル用のレジスタ・ファイル(CRF)を装備し、そのアドレスをマシン命令のオペランド部で指定することで、定数リテラルをUU内に持ち込む方法を採用した。

組み込み述語の引数の受渡しは、引数を格納しているレジスタ番号をマシン命令のオペランド部で指定する方法によって実現している。

ORPでは、すべての候補節を4個のUUに割り付けて並列処理を行うため、マシン命令には引数によるインデキシングの機能を備える必要がない。また、単一化処理の失敗や、メッセージ送出要求は、UUのマイクロプログラムから直接ORUへ割り込みを発生して伝える。従って、マシン命令レベルには条件コードによる順序制御機能が不必要であり、マシン命令の単純な先読みが可能となっている。

ORUは、各UU用のマシン命令フェッチ・ユニット(IFU)内のTACに節の処理プログラムの先頭番地を格納し、UUの起動をIFUに指令する。IFUは、TMM-Oからマシン命令を読み出し、命令レジスタ(IR)に格納する。(図1参照)

IRのオペレーション部は、UU内のマイクロシーケンサの外部アドレス入力ポートの一部に接続されている。すなわち、マシン命令の命令コードが直接マイクロプログラムのアドレスの一部となっている。また、IRのオペランド部には、UU内のオペランド・レジスタ(OR0~OR2)が接続されている。マシン命令を解釈・実行するマイクロプログラムには、このオペランド・レジスタを通じて、処理するデータが格納されているレジスタ・ファイルのアドレスが渡される。新しいマシン命令の実行が開始された時、オペランド部がオペランド・レジスタにロードされる。

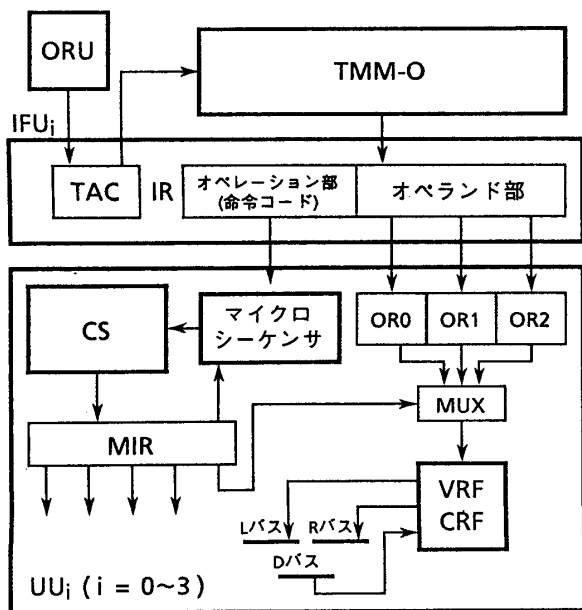


図1 マシン命令の実行機構

処理要求待ちのUUは、外部アドレス入力ポートから次に実行するマイクロ命令アドレスをロードするマイクロ命令の実行ループに入っている。IFUによって、IRにマイクロプログラムの開始番地がロードされると、UUはそのアドレスをマイクロシーケンサに取り込み、処理を開始する。

UUでの処理が終了すると、ORUへ割り込みが発生する。UUは、その割り込みが処理されるまで、ループしている。ORUは、割り込み処理を終了すると、IRをクリアし、UUに対し割り込み処理の終了を伝える。

### 5. UUのマイクロ命令とハードウェア機構

単一化処理では、ゴールの引数のデータ・タイプに基づく多方向分岐を行って、処理内容を動的に決定する操作が非常に多く現れる。従って、サイクル・タイムを短くし、命令パイプライン化を行っても、条件分岐によるパイプの乱れが頻繁に起こり、パイプライン化による高速性が発揮されないと考えられる。これらのことから、UUでは多方向分岐機能を持った水平型マイクロ命令による制御方式を採用した。

UUは、ゴールの引数が流れるLバス、節頭部の情報が流れるRバス、レジスタやメモリに書き込まれる情報が流れるDバスの3本のバスを持つ。ゴール引数の単一化処理では、ALUはほとんど用いられず、単なるデータのコピー操作が非常に多い。そこで、Lバス、RバスとDバスとの間にバイパスを設けた。また、タグの操作は、データ演算とは全く異なる操作になるので、UUでは、バスのタグ部とデータ部の各演算処理とを別々に制御できるようにした。

節内の変数や、一次的なデータの格納には、VRF (Variable Register File) という高速のレジスタ・ファイルを用いる。VRFのアドレスは、オペランド・レジスタによって指定される。

構造データや、未束縛変数の実体は、EBと呼ぶ高速のメモリに格納される。EBには2つの読み出しポートがあり、一度に2つのアドレスをアクセスすることができる。EBのアドレスは、複数のスペシャル・レジスタや、VRFの出力、ゴール引数などの中から、選ぶことができる。また、EBには変数のデレファレンスをハードウェアで直接実行する機構を備えている。

UUにおける順序制御では、取り出したデータに基づいて、16方向まで分岐することができる。タグ部は8ビットと短く、また、分岐条件を決めるタグに関する操作が性能を左右すると考えられるため、タグ操作機能のハードウェア化を進めた。

### 6. おわりに

現在、UUの論理設計を進めている。また、マシン命令レベルのシミュレータを用いて、マイクロ命令の仕様の詳細を決めている。