

# 知識ベースマシンMu-X (5)

## — 知識ベース演算エンジン —

浅野 滋博, 柴山 茂樹, 伊藤 英則  
(株)東芝 総合研究所 ICOT

### 1. はじめに

知識ベースマシンMu-Xでは、「項」を演算の基本要素としている。項に対する処理は演算装置にとって重い。また、関係代数の等号条件を単一化操作に拡張したRBU演算ではユニフィケーションを演算の基本要素としているため、演算装置がボトルネックとなる。

このようなボトルネックに対処するため、Mu-Xの要素プロセッサの一つ一つに付加されるハードウェアとして知識ベース演算エンジン(KBE: Knowledge Base Engine)を開発している。

ICOTではすでにユニフィケーションを高速に行う専用プロセッサとしてUE(Unification Engine)[1]を提案している。KBEの狙うところはUEと比べて比較的少量のハードウェア量で、しかもユニフィケーション以外の演算にも柔軟に対処しようということにある。従って、KBEでは高速性を狙いながら同時に処理の柔軟性を目指して、ハードワイヤードのロジックで組むのではなく、マイクロプログラム方式を採用した。

なお、KBEはMu-XのMPPM内のデータをその演算の対象としているので、ディスクアクセスの問題は議論しない。

### 2. その設計思想

知識ベース演算では大量のデータに対して同一の演算を施すのでパイプライン的な処理をすれば並列の効果により処理の高速化が図れる。また、データに対する前処理、後処理を、実際の処理の時間とオーバーラップさせるという観点から、KBEでは読み込み、演算、書き込みの三つの処理ユニットを直列に配置することにした。

ところで、パイプライン的な処理といっても、数値計算を目的とするスーパーコンピュータに見られるようにクロック同期でパイプをデータが流れるような処理をするわけではない。知識ベース演算では処理の最小単位は項であり、この項は固定長のデータではない。しかも、ユニフィケーション等の演算では演算時間も一定ではない。従って三つ

のユニット間の同期は、クロック同期よりも緩い同期をとる。この様な、パイプラインよりも緩い同期の形態を取る制御の方式をフロースルー制御と呼ぶことにする。

各ユニットの間には処理速度のばらつきに対処するためにバッファを置き、これが各ユニットの同期をとる動きをする。

### 3. KBEの構成

読み出しユニット(MRU: Memory Read Unit)、演算ユニット(OPU: Operation Unit)、書き込みユニット(MWU: Memory Write Unit)はそれぞれ独立のマイクロプログラムを持ったユニットで、主記憶からのデータがMRU、OPU、MWUと流れるうちに処理が行われ、加工されたデータが主記憶に書き込まれる。以下では図1を参照してKBE各部の働き、構成について述べる。

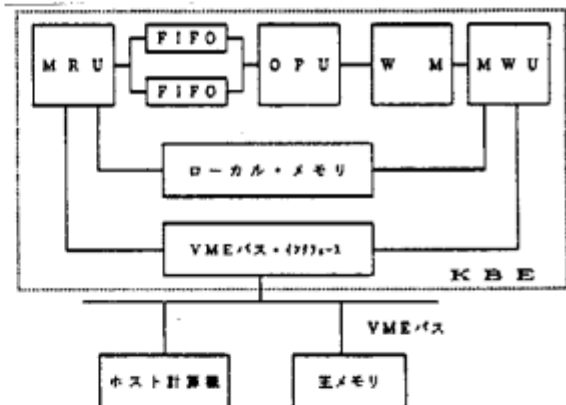


図1 KBEの構成

図1

#### 3. 1. MRU (Memory Read Unit)

MRUでは主記憶からデータを読み出し、適当な処理をしてOPUにデータを与える。MRUが分担する処理の主なものはデータの切り出しである。知識ベースの演算では演算に必要な属性はタブルの一部でありその他の属性は必要としないことも多い。そこでMRUではOPUでの実際の演算の負担を軽くするために必要なデータだけをOPUに送るフィルタの動きをする。また、データの一部はポイ

Knowledge Base Machine Mu-X(5) -Knowledge Base Engine-  
Shigehiro Asano, Shigeki Shibayama.(Toshiba R&D Center)  
Hidenori Itoh (ICOT)

ンタで表現されているものもあるのでポインタで指された実体を手繰り寄せるのもMRUの働きとなる。

### 3. 2. FIFO

FIFOはMRUからOPUへ与えるデータのバッファとして使用される。また、FIFOを二本用意したのはユニフィケーション時の二つの項にそれぞれFIFOを用意したかったためである。

このFIFOはデータに加えて4ビットのタグが用意され、このタグをOPUが読むことによってデータの種類が分かるようになっている。

### 3. 3. OPU (Operation Unit)

OPUは実際の演算を行うユニットである。ユニフィケーション等の演算を考えると三つのユニットのうち一番重くなる部分と考えられる。そこで、OPUではALUをデュアル構成にして他のユニットより処理能力を高めてある。

### 3. 4. WM (Working Memory)

WMはOPUの作業領域として使用されるほか、OPUからMWUへの通信のバッファとして使用される。OPUとMWU間を単純なFIFOで結ばなかったのは次のような理由による。OPUで行われたユニフィケーションの結果は変数の束縛の情報としてポインタが張られた構造を持っている。この構造をFIFOで受渡しをするのは適当でないと考えたからである。

OPUからMWUへのバッファはこの点考えると構造を要素としたバッファが必要で、共有メモリ上にソフトウェアでバッファを構成することにした。

なお、このWMはOPU、MWU間での競合が発生するが、OPUの処理が重いことを考えてOPUが優先されるようにした。

### 3. 5. MWU (Memory Write Unit)

MWUはOPUで処理された結果をもとに、主記憶に書き込む働きを受け持つ。また、結果に必要な属性が、演算に直接関係ない属性(サイトアトリビュート)であることもあるので、OPUで演算された属性以外の属性に付いて処理をするのもMWUの役目である。

### 3. 6. LM (Local Memory)

LMはMRU、MWUで共有される高速なメモリである。LMは主記憶へのアクセスを減らすバッファとして使用されるが、MRU、MWUがその内容を管理しなければならない。LMは例えばMRUで使用されたデータをMWUで再び使用するような場合、あるいはMRUが何回も同じデータを必要とする場合(ジョイン演算のネステッドループアルゴリズムを採用した場合におけるインナーループ)等有効である。

LMはMRU、MWUの両方からアクセスされるのでアービタにより調整がとられる。

## 4. プロセッサとの接続

KBEはMu-Xの要素プロセッサに付加されるハードであるが、この章ではプロセッサとの接続方式について述べる。

### 4. 1. DMA (Direct Memory Access)

KBEのMRU、MWUはDMAにより主記憶に直接アクセス出来るようになっている。主記憶のデータはDMAを通じて読み込まれ、演算された後DMAを通じて主記憶に書き込まれる。

### 4. 2. コプロセッサインタフェース

KBEはMu-Xの要素プロセッサよりコマンドをもらいそのコマンドを実行する。コマンドを受け渡す方法としてはコプロセッサ方式を考えている。コプロセッサインタフェースを採用すれば、要素プロセッサのプログラマに、あたかも要素プロセッサの命令セットが知識ベース演算用に拡張されたかのような環境を提供できる。

## 5. 実装

KBEの三つのユニットはマイクロプログラムにより動作するが、マイクロシーケンサ、ALU、レジスタファイルにはAMD社のAM293xxシリーズを採用した。内部の処理は32ビット単位で行われる。また、クロックの周波数は10MHzである。WMの容量は64Kワード、LMの容量は256Kワードである。

KBEは評価用にホストとしてMC68020に接続されており、メモリとのインタフェースとしてVMEバスを採用している。

KBEの全ての部品はVMEのトリプルハイトのボード二枚に納められている。

## 6. 現状と今後の課題

現在、ハードウェアの調整は完了して、安定に動作している。評価のためウィスコンシンベンチマークテストをおこなったが、1000件のジョイン演算でMC68020単独で処理を行った場合に比べて、KBEで実行した場合には10倍程度の速度が出た。

今後は各種の演算をインプリメントして、評価をすすめていく。

## 参考文献

- [1] Morita, Y. et al: Retrieval-By-Unification Operation on a Relational Knowledge Base. Proceedings Twelfth International Conference on Very Large Data Bases pp.52-59 (1986)