

# 磁気バブルデータベース計算機 EDCのアーキテクチャ

THE ARCHITECTURE OF ETL DATABASE COMPUTER (EDC)

岡分明男, 大妻良一, 宇場敏嗣, 植村俊亮  
A. KOKUBU R. OOMOTE T. YUBA S. UEMURA

電子技術総合研究所  
ELECTROTECHNICAL LABORATORY

## 1. はじめに

データベース処理のための専用プロセッサは一般にデータベースマシンと呼ばれてゐる。データベースマシンに必要とする人々のイメージは必ずしも一致してゐる訳ではなく、そのアーキテクチャにはいろいろなものがある。たとえば、XDS<sup>1)</sup>のように中央処理装置との間で機能を分担したもののものから、RAP<sup>2)</sup>などのようにデータベースマシン側に並列処理をばらばらしたもので多種多様である。

電子技術総合研究所で開発中の磁気バブルデータベース計算機EDC<sup>注)</sup>は、アーキテクチャから見ればデータベース処理のためのマイクロプロセッサに磁気バブルメモリを統合した“磁気バブルデータベース”<sup>3)</sup>と呼ばれ、基本単位を並列に接続した形のデータベースマシンであり、<sup>4)</sup>最新のハードウェアを駆使して製作されてゐる実験機であるといふ特色がある。

EDCを開発する目的は、データベースマシンのようにデータベースを応用向きの計算機アーキテクチャを研究することであり、将来の実用機のエンジニアリングモデルを開発しようとする訳ではない。しかしながら、今後5年から10年間のLSI技術の進歩、特にマイクロプロセッサ、半導体メモリ、磁気バブルメモリの集積度向上を考えると、EDCのようなシステムが実用的に意味を持つようになることが注) EDCはETL Database Computer または Electronic-disk oriented Database Complex の略称である。

性も十分に大きいといえる。

以下では、

EDCの構成

データベースモジュールの構成

データベースモジュール間通信方式

データベースモジュール間通信方式

の順に構成を中心に述べよう。

## 2. EDCの構成

### 2.1 設計思想

EDCを設計する際の基本思想として次の3点を重視する。

モジュラリティ

拡張性

柔軟性

モジュラリティは、データベースのような大量生産の小規模システムへの応用<sup>5)</sup>が期待される基本単位をつなぎ合わせることで構成することを実現される。各データベースモジュールはそれぞれが論理的には異なるように使われようとしても、物理的には同一である。このように構成することによって、EDCに経済的観点からの現実性が与えられる。

拡張性はデータベースの最大個数や各データベースモジュールのアドレス空間を十分に大きくとることによって実現される。EDCは完成された実用機ではなく、あくまでも実験機であるので、将来のLSI技術の進歩に対応して拡張できるようにしておく必要がある。

柔軟性はマイクロプロセッサやバブルインタ

フェース部分をマイロプロダラム制御とし、データモジュール間の結線に際してもプロダラムバブルにしていくことと実現される。EDC開発の目的はEDC上へのせりあげシステムを研究することにもあり、それが必ずしも固まっていた訳ではないことから、ハードウェアに近いレベルでの柔軟性が必要であることは明らかである。

## 2.2 全体の構成

全体の構成は、現在製作中の図1(a)の構成と来年度に予定している図1(b)の構成の2段階に分かれ、それぞれは図1(c)に示す構成のサブセットとして位置づけられている。以下では主として図1(a)の構成を中心に述べる。

EDCはインタフェースプロセッサ(IP)を介してホストシステムに接続される。現在製作中の図1(a)の構成では、ホストシステムはTOSBAC5600であり、IPは見かけ上ホストシステムのTSS端末のようになっている。また、IPはデータモジュール#0のコンソール端末の代りにもなっている。

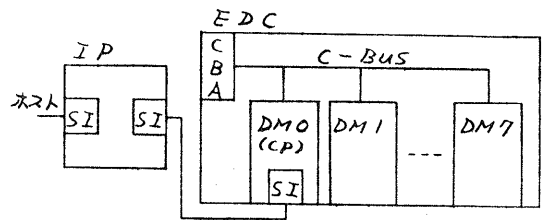
IPの役割りはホストシステムとEDCを直接することにより生じるEDC側の負担を軽減することにある。また、ホストシステムが使用できない回路にEDCを動かさせたことのできるようにするためのものでもある。これらの接続は9600ボートのシリアルインタフェースで行なわれている。

IPはCPUにZ-80Aマイロプロセッサを使用し、ファイルメモリに140Kバイトの磁気バブルメモリを使用したマイロコンピュータ・システムであり、バブルベースのOSで制御されたシステムとしては世界的に見て最初の数台のうちに入数される。IPの構成、OS、特長については次の機会に述べる。

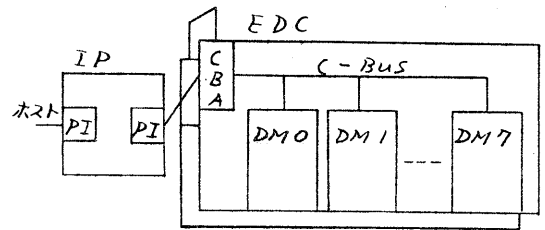
データモジュール#0のプロセッサは図中で示されたようにCP(Control Processor)と呼ばれることもある。これは設計の初期の段階でデータモジュール単体を管理するために、ハードウェアが異なる上位のプロセッサを用意することから採られた設計あり、これをCPと呼んだことからきている。しかし、設計の最終段階ではすべてのデータモジュールのハードウェア構成はほぼ同一になり、現在ではCPと呼ぶことはハードウェアの観点からは慣習に

すぎない。

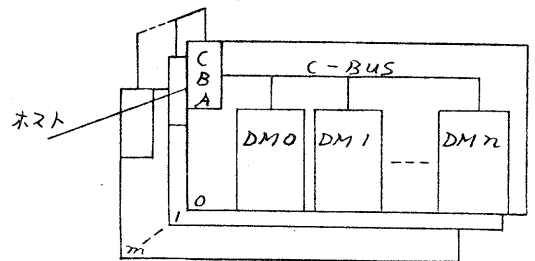
現在製作中の図1(a)の構成ではデータモジュール8台がC-Bus(Common BusまたはCommunication Busの略称)と呼ばれるバスで結ばれており、C-Busはバス上で各データモジュールからのアクセスが競合しないようにするため、CBAと呼ばれるバスアービタの管理下に置かれている。



(a) 現在の構成



(b) 来年度の構成



(c) 構想

## 図1 全体の構成

- IP : Interface Processor
- CP : Control Processor
- DM : Data Module
- CBA : Common Bus Arbiter
- SI : Serial Interface
- PI : Parallel Interface

### 3. データモジュールの構成

#### 3.1 構成

図2にデータモジュールのブロック図を示す。データモジュールは大きく分けて、フロッピッサー部分(PULCE, MPM, MMC)、メインメモリおよびインタフェース部分(MM, MMI/IBI, CBI)、磁気バツルメモリおよびインタフェース部分(BMI, MBM)から構成された。

各部分はX-Busと呼ばれる内部バスによって結ばれている。X-Busはそれぞれ16ビット幅のX-Bus0とX-Bus1に分かれ、それぞれを個別に動作させることにより32ビットまでのデータ転送が可能となるようになっている。

#### 3.2 フロッピッサー

フロッピッサー部分は16ビット幅の汎用演算要素(PULCE<sup>6)</sup>、マイクログラウラムメモリ(MPM)、マイクログラウラムメモリコントローラ(MMC)から構成された。

PULCEはALU、シフト、レジスタファイルなどを含むSOS/n-MOSのLSIであり、およそ7,000ゲートの素子が集積され、80ピンのパッケージに入れられている。PULCEにはX-Busとの間でデータ転送を行なうために16ビットのインタフェースレジスタが2個用意されており、通常のX-Bus経由のデータ転送の際にはそれぞれがデータのレジスタとして使用されている。

MPMは、マイクログラウラムをユーザ側で自由に変更できるようにするために、アドレス空間の中にPROM領域とRWM領域がある点に特徴がある。PROM領域にはバイポーラの4KビットPROMが用いられ、RWM領域にはDSA/n-MOS<sup>7)</sup>の4KビットRAMが用いられている。MPMの1語は32ビットで構成され、容量はPROM領域とRWM領域が2K語ずつになっている。

MMCはマイクログラウラムのシーケンスを制御することの他に、PULCEで解釈実行されない命令を外部命令として再定義し、解釈実行することができる。<sup>8)</sup> たとえば、MMCは外部命令を用いて、PULCEの2つのインタフェースレジスタを経由してMPMとメインメモリ間

のデータ転送を行なうことができる。

#### 3.3 メインメモリおよびインタフェース

メインメモリおよびインタフェース部分は、16ビットの語単位でもバイト単位でもアクセス可能なメインメモリ(MM)、メモリバスおよび入出力バス・インタフェース(MMI/IBI)、C-Busインタフェース(CBI)から構成された。

MMIは各データモジュールにおけるデータ操作の作業領域として使われる外に、データモジュール間の通信領域としても使われる。メモリアップには16KビットのダイナミックRAMが使用されており、容量は32Kバイトが最小単位である。

MMI/IBIはメモリや入出力装置とPULCEのインタフェースレジスタ間でデータ転送を行なう際に必要なインタフェースである。MMIにはメモリアドレスレジスタ(MAR)やメモリデータレジスタ(MDR)がある。

CBIはC-Busとメインメモリ間のインタフェースであるのみならず、MARやMDRを自身のメモリに接続するか、C-Bus経由で任意のデータモジュールのメモリに接続するかなどのメモリアクセスに関する制御を行なう。CBIには上述の他にワードキャスト受信機構、リフレッシュレジスタ、タイマ/カウンタ機能なども含まれている。

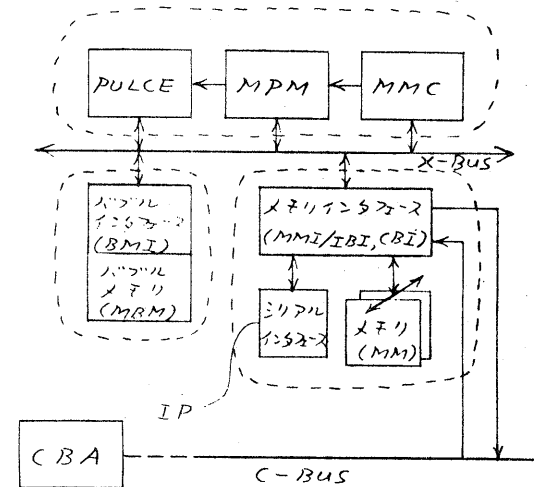


図2 データモジュールの構成

### 3.4 磁気バツルメモリおよびインタフェース

磁気バツルメモリおよびインタフェース部は磁気バツルメモリ(MBM), 磁気バツルメモリインタフェース(BMI)から構成される。

MBMは64Kビットの磁気バツルメモリチップが16個並列に接続された128Kバイトの不揮発性メモリである。アクセス時間は最大で約6ms, 転送レートは100Kバイト/sである。

BMIはMBMとX-Bus間のインタフェースである。FODセクタ部分と磁気バツルメモリ部分は独立に動作して113msで、両者の間でデータのやりとりを行なう際はBMI中のFIFOメモリを経由して行なわれる。磁気バツルメモリを制御するためのマイクDフODがBMI中のFIFOメモリを経由してMBMに与えられる。

磁気バツルメモリはマイクDフODがうへによつてチップレベルまで制御可能となるので、たとえば、READ AND WRITE動作のようなことも容易に実現することができた。また、PULSEの処理範囲が狭い場合には、READ-MODIFY-WRITE動作のようなことも実現できたことになった。

## 4. データモジュール間の結合方式

### 4.1 アドレス空間

各データモジュールはC-Busによって結合されており、24ビットで表現されるC-Bus上のメモリアドレス空間を共有する。一方、各データモジュールはL-Busという仮想バスを経由して自身のメモリに16ビットのアドレスで直接アクセスすることができた。これらのアドレスがメモリのアドレス空間上でどのような関係にあるかをここで説明する。

図3は各データモジュールのメモリバスのアドレス空間について、C-Bus側のアドレス空間とL-Bus側のアドレス空間とのような関係にあるかを示すためのものである。

各データモジュールには、C-Bus専用のリドケーションレジスタRLRCとL-Bus専用のリドケーションレジスタRLRLが用意されている。これらの内容を図4に示すように4ビット上位にずらしてアドレスに加算することによって、メモリに与えられる20ビットの

物理アドレスが生成される。メモリはバイト毎にアドレスがつけられるので、メモリバスのアドレス空間は1Mバイトであった。

図3および図4から分かるように、各データモジュールのリドケーションレジスタの内容を適切に設定することによって、メモリバスのアドレス空間上でC-BusからマッピングしたセグメントとL-Busからマッピングしたセグメントが部分的に重なったりすることも、全く重なったりすることもできる。

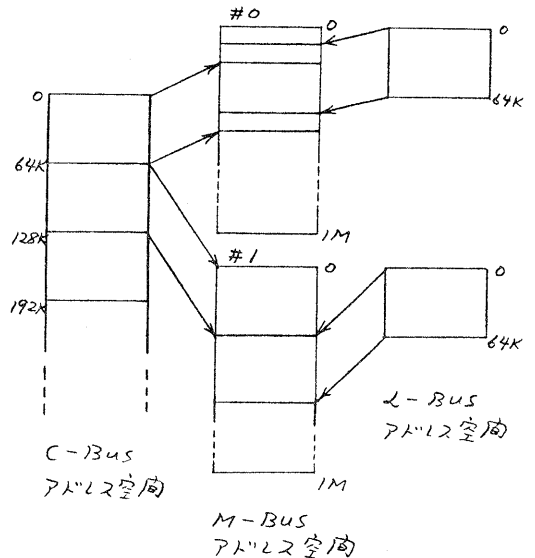
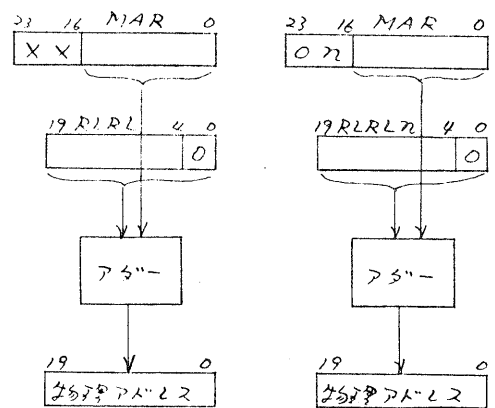


図3 アドレス空間



(a) L-Bus選択

(b) C-Bus選択

図4 物理アドレスの生成

#### 4.2 ブロードキャスト

EDCでは複数のデータモジュールのMM中の任意番地にC-Bus経由で同時にWRITE(すなわち、ブロードキャスト)することが可能である。ブロードキャストは次のように行なわれる。

各データモジュールにはそのモジュール固有のアドレスをデコードするデコーダとサブアのモジュールに共通のアドレスをデコードするデコーダの2つが用意されており、それぞれによりC-Busに出たMARの上位8ビットをデコードされた。現在動作中の構成ではモジュールアドレスとブロードキャストアドレスが図5のように決められているので、MARの上位8ビットをOF<sub>16</sub>にしてC-Bus経由でメモリにWRITEすると、自身も含めてメモリ制御レジスタのブロードキャスト・エネーブル・ビットが1のすべてのデータモジュールにMARの下位16ビットが受け入れられてWRITE動作が行なわれることになった。

#### 4.3 C-Bus制御方式

C-Bus経由でメモリにアクセスする際に各データモジュールからのアクセスがC-Bus上で競合しないようにするために、EDCでは図6に示すようにCBA(Common Bus Arbitrator)を用いたC-Bus制御方式を採用している。CBAはC-Busあたり1つしか存在しない。

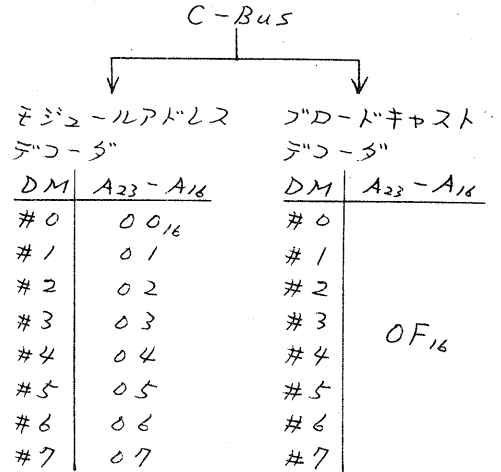


図5 ブロードキャストおよびモジュールアドレスのデコード

C-Busを使用しようとする任意のDセックは最初にCBAにリクエストを送る。(リクエストはメモリアドレス情報から自動的に作られる。)CBAはそのような情報のうちで最も優先度の高いものを選び、 Grant 信号を応答する。Grant 信号を応答したDセックは通常のメモリアクセスの場合はメモリの1サイクル分だけ、C-Busの使用権を確保する。排他制御を行なうために用意されている Lock 命令の場合は、次のメモリアクセス

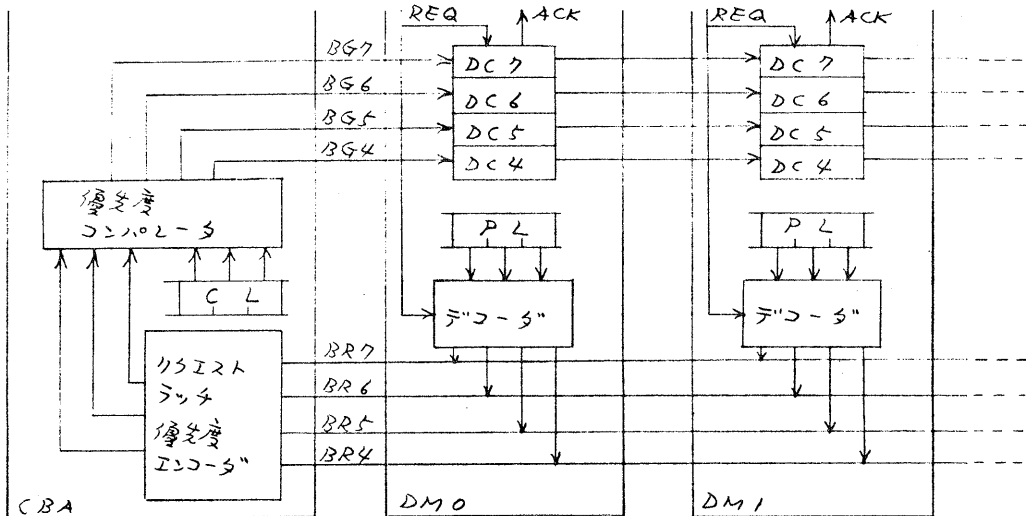


図6 C-Bus制御方式

の終了までC-BUSの使用権の解除を延期する。

リウエスタには4つのレベルがあり、それぞれを独立にCBAに送るためにBR4~BR7の4本の信号線が用いられる。各プロセッサがどのレベルでリウエスタ信号を出すかは、各データモジュール中のメモリ制御レジスタMCSRのプロセッサレベル(PL)ビットの設定値によって決められる。

CBAは受け付けたリウエスタレベルの中の最も高い信号を見て、CBA中のC-BUSレベル(CL)ラッチの内容と比較する。ここで、 $PL > CL$ の条件が満たされる場合は、CBAはリウエスタレベルに対応するデータバス線に信号を出して、C-BUSの使用権が解除されるまで新しいリウエスタ信号を受け付けないようにする。

データバス線は各レベルで独立に存在するデータ・チェーン回路を通過する。したがって、同一レベルでリウエスタを出してはいる複数のプロセッサがあり、そのレベルにデータバス線が出たときは、CBAは近いプロセッサが先にC-BUSの使用権を取ります。すなわち、プロセッサは同一レベルならばCBAに近いほど優先度が高い。

## 5. データモジュール内の通信方式

### 5.1 通信方式

各データモジュールはC-BUS経由ですべてのメモリを共有している中で、1対1の場合は相手のメモリに通常のWRITE操作で通信情報を書き込むことにより、通信が行なわれます。1対Nの場合は、プロトコルキャストによって(各データモジュールのメモリ制御レジスタによって選択された)N個のメモリに通信情報を書き込むことにより、通信が行なわれます。

通信が行なわれたことを相手に知らせるため、以下で述べる割込み機構が用意されています。

各データモジュールにはメモリのアドレス範囲のアドレス信号をコントラスタプリセットカウンタが各々2個用意されており、この内容が0になると割込みが起きようになっています。これらのカウンタのプリセット値を次のように設定しておく。

カウンタ#0 プリセット値=1

カウンタ#1 プリセット値=N

ここで、カウンタ#Cの入力はそのデータモジュールの16番地へのアドレス信号から作られ、カウンタ#1の入力はそのデータモジュールの12番地へのアドレス信号から作られる。

次に、あるプロセッサ(たとえば、CP)が複数の他のプロセッサ(MPS)にブロードキャストを用いて並列的にサーチデータのコマンドを挿付けて、結果を待つという状態を考へよう。このとき、各MPがそのコマンドを実行後CPに報告する結果は次のとおりである。

① データを検出した。

② 終端までサーチしたが見つからない。

①の場合は、各MPはCPの16番地に自分の番手をWRITEする。これによって、割り込みが起きるので、CPはデータが検出されたことを直ちに知ることができず。

②の場合は、各MPはCPの12番地に単にREADする。この結果、CPのカウンタ#1の内容は-1される。N台のMPがCPの12番地をREADしたとき、カウンタ#1の内容が0になる。この時点で割り込みが起きる。これはあるMPがデータを検出できなかった他のMPがサーチ実行中の場合もありうるので、並列的にサーチを行なうすべてのMPが終端に到達して、サーチデータが全く検出できなかったことが明らかになった時点でCPは知ることがないからである。このようにして、割り込み処理ルーチンが完了回数を増やすことができる。

### 5.2 排他制御

排他制御はC-BUSをロックする命令を用いて、各データモジュールの通信領域の先頭語にTest and Set操作を繰り返すことにより行なわれる。すなわち、

(1) C-BUSにロックをかけた、先頭語を読む。

(2) 先頭語が0ならば、そこに1を書くことによりその領域を論理的にロックする。

(1)を書くことにより、C-BUSのロックは解除される。

(3) 先頭語が1ならば、排他制御用のキューに領域#とプロセッサ#を積み(これによってC-BUSのロックは解除される)、アンドゥの通知が終了までループし続ける。

を待つ。

(4) 領域をアンロックするとき、そのFODセツカはアンロック・ルーチンに飛んで次のことを行なう。

- ① 先頭語をクリアする。
- ② キューを調べて、その領域が空くのを待ってFODセツカに通知する。

## 6. 討論

### 6.1 磁気バブルメモリの将来性

磁気バブルメモリをデータベースのような大容量のメモリとして使用することについては、ビットあたりの価格がダイナミック装置などと比較して大きいので、議論のあふるところである。一方、磁気バブルメモリのチップは1Mビットのものが出そろったことにより、将来は100Mビットのものも可能になると見方があり、筆者らは5年から10年のレンジで見れば、磁気バブルメモリもその程度はダイナミック装置と価格的に競争できるようなになると予想している。

### 6.2 データのリアルタイム処理

データベースでは磁気バブルメモリから読み出されたデータをリアルタイムで処理する方式を採用している。リアルタイム処理は効率が良いけれども、FODセツカと補助メモリ間で速度のバランスがとれなければならないことが重要である。現在のマイコンFODセツカと磁気バブルメモリの速度はリアルタイムで処理に適している。

## 7. おわりに

磁気バブルデータベース計算機EDCの構成を中心に述べた。EDCは昭和53年10月現在で3台のデータベースの製作が完了しており、近日中に3台の5台の製作も完了する予定になっている。部分的修正はしばらくの間続くとしてもアーキテクチャは固まったと考えているので、今後はハードウェアとバランスのとれたソフトウェアの開発に重点を移したと考えている。

最後に、本研究の機会を与えてくれた西野博二パターン情報部長、石井浩三ソフトウェア部長、種々のマニピュレーションと通じてサポートした新田松雄先生システム研究室長、熱心な討議頂いたEDC開発グループの諸氏に感謝する。

## 参考文献

- 1) R. H. Canaday et al, "A Back-end Computer for Data Base Management", Comm. ACM, Vol. 17, No. 10, pp. 555-582, Oct. 1974.
- 2) E. A. Ozkanakan et al, "RAP - An Associative Processor for Data Base Management", Proc. AFIPS Conf., Vol. 44, pp. 379-387, 1975.
- 3) 国分明男他, "磁気バブルデータベース", 信学技報, EC76-58, pp. 43-50, Nov. 1976.
- 4) 植村俊亮他, "磁気バブルによるデータベースマシンの構想", 信学技報, EC76-78, pp. 65-74, Jan. 1977.
- 5) G. E. Juliusen, "Bubbles and CCD memories - Solid state mass storage", Proc. AFIPS Conf., Vol. 47, pp. 1067-1075, 1978.
- 6) H. Iizuka et al, "Development of a high-performance universal computing element - PULCE", Proc. AFIPS Conf., Vol. 47, pp. 1255-1264, 1978.
- 7) Y. Tanui et al, "Diffusion Self-Aligned Enhance-Depletion MOS-IC (DSA-ED-MOS-IC)", Proc. 2nd Conf. on Solid State Devices, Tokyo, 1970; Supplement J. Jap. Soc. Appl. Phys., Vol. 40, pp. 193-198, 1971.
- 8) ハードウェア情報処理システムプロジェクト, マイコンコンピュータシステムPMCS, マイコン命令説明書, 東京電研, 昭和53年5月.
- 9) 見利男他, "磁気バブルメモリデータベース", 信学技報, EC77-53, pp. 35-40, Dec. 1977.
- 10) H. Chang, "On Bubble Memories and Relational Data Base", Proc. Fourth Int. Conf. on Very Large Data Bases, pp. 207-229, West Berlin, 1978.