

6P-4

マイクロコンピュータシステムにおける  
メモリ制御方式について

前田多可雄\*1 秋田英彦\*1 三浦明義\*2

\*1 (株)日立製作所 システム開発研究所

\*2日立マイクロコンピュータエンジニアリング(株)

1. 緒言

パソコン、ワークステーションと呼ばれる分野においては、インテル社の8086上にマイクロソフト社のMS-DOSを搭載したシステムがひとつの標準となっている。これらのシステムは1MBまでしかアドレスできず、機能の高度化に伴いメモリネットワークが問題となっている。この問題に対処するため仮想記憶制御支援機能を内蔵したCPU(Central Processing Unit), 80286, 80386が発表されたが、互換性などの問題から仮想記憶支援機能を用いずに高速8086としての利用が一般に行われている。このため互換性を重視し、しかも1MBの壁をシステムティックに越える方式の開発が望まれている。

本稿では8086を用いたシステムの上位機種を考えた場合のメモリ拡張方式について述べる。

2. 問題提起

我々が対象とするシステムは8086を用いた実記憶システムである。そのソフトウェアは図1に示すようなカーネル、ドライバ、タスクの3つの部分から構成され、マルチタスク機能を有するカーネルは他の2者を管理している。

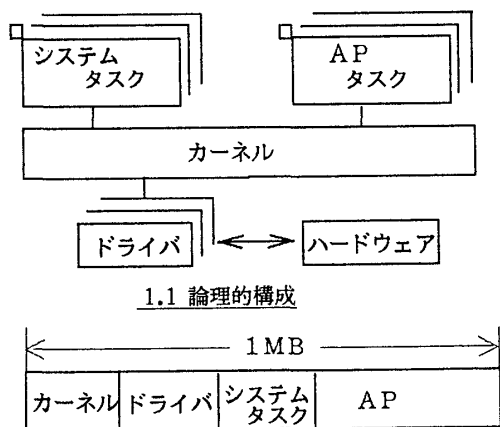


図1. ソフトウェア構成

ドライバは入出力処理を行うサービスルーチンであり、内部に割り込みに対する処理を含んでいる。このためドライバは常に主記憶上に存在することが前提となっている。

タスクはカーネルによるディスパッチの対象である。タスクにはアクセスする領域が自領域内に閉じているものと、閉じていないものがある。前者の例としてはワープロ、表計算などのようなAPがあり、後者の例としてはスプーラなどのように他のプログラムからの処理要求に応じて動作する(OSの一部を構成する)システムタスクがある。

またAPとしてVM (Virtual Machine)の考え方に基づいてMS-DOSのような標準的なシングルタスクOSを動作させることも可能とする。

このようなシステムの上位機種におけるメモリ制御方式を考えることが本稿の目的である。

3. メモリ制御方式

3.1 メモリモデルの構成

我々が採用した論理空間のモデルを図2に示す。互換性を重視する立場から、1MBのアドレス空間の一部分を多重化する方式とした。これをバンク制御方式と呼ぶことにした。多重化した部分を以後バ

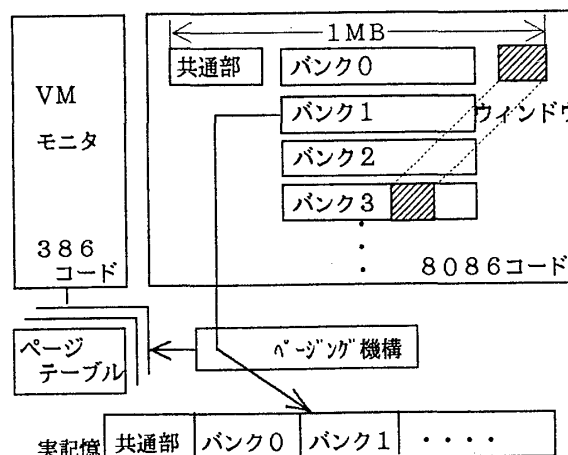


図2. メモリモデルと80386での実現方式

On the memory management of the microcomputer systems

Takao MAEDA\*1, Hidehiko AKITA\*1, Akiyoshi MIURA\*2

\*1 HITACHI Ltd. \*2 Hitachi Microcomputer Engineering Ltd.

バンクと呼ぶ。各命令の実行時には1つのバンクが選択される。多重化しない部分のうち実メモリとの対応が固定的な部分を共通部と呼び、可変な部分をウィンドウと呼ぶ。ウィンドウは主記憶の任意の領域と対応付け可能にする。バンク上のプログラムはウィンドウ経由で他のバンク上の領域をアクセスする。

### 3.2 メモリモデルの実現方式

#### (1) 80386の場合

CPUの動作モードはOS部分を含め原則として仮想8086モード（以下VMモード）を用い、アドレス変換（ページング処理）に関する部分と割り込みの入り口部分の処理のみを80386モードで動作させる。80386モードで動作させる部分をVMモニタと呼ぶ。

アドレス変換部ではバンクとページテーブルを対応付けておき、各バンクが選択された状態（物理アドレスと論理アドレスの対応関係）に対応するようにページテーブルを作成しておく。バンクの切り替え（選択の変更）要求があるとページテーブルを入れ換える（80386ではCPUの1つのレジスタの書き換えで可能）。ウィンドウの対応領域の変更要求があると、ページテーブルの対応する部分を変更する。

割り込み処理の入り口では、80386として割り込みをハンドリングし、カーネル（VMモード内）が処理すべきか否かを判定し、必要があればカーネルに対し疑似的に割り込みを発生させる。VMモニタで処理する場合にはその処理を呼び出す。

#### (2) 80286以下の場合

80286以下にはVMモードやページング機構がないため、動作は全て実記憶モードとし、VMモニタのアドレス変換機構相当部分はハードウェアで実現した。図3に示すようにCPUの外部に20ビット目以降のアドレスレジスタを設け、アドレス変換をするようにした。バンクの切り替え時にはCPUはこのレジスタを入出力命令により書き換える。またウィンドウのマップの変更時にも入出力命令によりアドレス変換器にその旨を伝える。

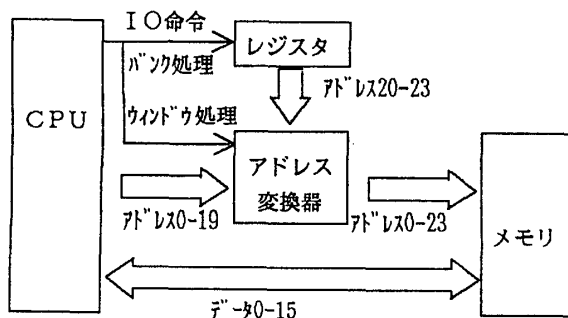


図3. 80286での実現方式

### 3.3 メモリモデルの適用

#### (1) 機能の拡張

3.1で述べたメモリモデルの適用を考える。カーネルを共通部に、ドライバ、システムタスクをバンク0に、APをバンク1以降のバンクに配置するようなソフトウェア配置とする。その際に必要となる拡張、変更点の主なものは次の通り。

- ・タスク切り替え時及び割り込み発生時のバンク切り替え
- ・割り込み処理からのタスク領域アクセス法

#### (2) 容量に関する検討

バンクの領域が大きければ多重度があがるため、カーネル部とハードウェアレジスタ領域のみを共通部に割り付け、それ以外の全ての領域をバンク上に割当てるようにした。現状で動作している実記憶制御のカーネルの容量とハードウェアレジスタのサイズを勘案すると、768KBのバンク領域を実現することが可能となった。即ち、768KBまでのタスクを動作させることが可能となった。これをVMとして動作している（カーネル下の1タスクとして実現している）標準OSからみると、OSとAPを合わせて768KBまで使用することが可能となる。

本システムの場合、スプーラなどのシステムタスクや入出力ドライバはタスクとは別のバンク上にあるため、全体として1MB以上のプログラムの動作が可能となる。

#### (3) 性能に関する検討

これらの機能拡張により従前の実記憶システムに比べオーバーヘッドが増加する。1回あたりのバンク切り換え、あるいはそれに付随する処理は数命令で実現できるため、それ自身は問題とならない。問題となるのは一連の処理の中でバンクに関係する処理の頻度である。そのため、典型的なAPの動作させオーバーヘッドの増加を求めた。

その結果カーネル、ドライバ、システムタスクの処理時間の増加（ハードウェアを用いた場合のオーバーヘッド全体）は5%程度、VMモニタによる処理時間の増加は30%程度という結果を得た。

### 4. 結言

8086を用いたシステムの上位機種におけるメモリ拡張方式について述べた。OSの一部であるドライバやシステムタスクをアプリケーションプログラムと同じ論理空間に配置できるようにしたものである。その結果、全体として1MB以上の機能を実現したプログラムを実行させることが可能となった。また80386においては仮想8086機構とページング機構を用い、特殊なハードウェアを用いずに実現した点が第2の特徴である。