

## 連載講座



## 計算機の記憶システム-I

## 記憶システムの概観†

高橋 茂竹

## 1. はじめに

記憶装置はそこにプログラムを置いて実行できる主記憶装置と、その容量の不足を補う補助記憶装置に分かれる。ともに容量の増大と速度の向上というニーズに応じて、初期の単純な装置からそれぞれがシステムを構成するほど複雑になった。

この解説では連載講座の第1回として、まず主記憶システムのハードウェアの進歩と方式の進展を概観し、ついで補助記憶システムの動向を論ずる。ともに種々の観点での解説が別途予定されているから、ここでは記憶システム技術の現状よりは、現状に至った過程に重点を置く。

## 2. 主記憶装置ハードウェアの進歩\*

世界で最初に実用されたプログラム記憶式計算機 EDSAC (1949年) の記憶装置は、水銀が媒体の超音波遅延線であった。UNIVAC (1951年) など初期の商用機にも採用されたが、循環形でアクセスが遅く、続いて開発された静電式記憶装置がその地位を奪った。ブラウン管蛍光膜の2次電子放射による帯電を利用し、帯電スポットの形状による荷電量の差で記憶するもので、アクセスは高速だったが、電荷が漏洩するからそのリフレッシュを必要とした。均一な蛍光膜が得難いなど多くの欠点があったので、これも永続しなかった。

静電式では、たとえば 1024 b を貯えるのに、ブラウン管蛍光膜上に帯電スポットを 32×32 のマトリクス状に配置した。帯電スポットをフェライト磁心に、電荷を磁化に置き換えたのが磁心記憶装置 (コアメモリ) である。MIT と RCA が独立に開発し、1953年 MIT の Whirlwind に

接続された。商用機では IBM 704 (1954年) が最初であった。

以来 20年間 ICメモリの本格化まで、コアメモリは主記憶装置を独占し、長足の進歩を遂げた。初期には直径 80 mil (2 mm) のコアで、サイクル時間 10  $\mu$ s 程度であったが、1970年にはコア直径 20 mil, サイクル時間 0.7  $\mu$ s のものが実用された。同じ 20年間に情報技術の全分野が飛躍的に進展したが、安定なコアメモリはその原動力であった。

最初の ICメモリはバイポーラ形フリップフロップ 16個を 4×4 に配列したもので、1966年 IBM システム/360 モデル 91 のメモリ保護用メモリとして、またモデル 85 のキャッシュ (cache) (容量 32 kB, サイクル時間 80 ns) として登場した。主記憶用は IBM システム/370 モデル 145 (1970年) が最初で、チップ当たり 128 b, サイクル時間 540 ns であった<sup>1)</sup>。

主記憶用にバイポーラ、MOS のいずれを採るか、当初大いに論議があった。速度はバイポーラ、集積度、消費電力、コストは MOS となるが、記憶容量の増大により MOS の優勢が圧倒的となり、バイポーラは後述のキャッシュなど特に高速が必要なものに限定して使われるようになった。

MOS には P 形と N 形があり、キャリア移動度の差で N 形のほうが高速であるが、製造上の問題から初期には P 形が多く、商業的に成功した最初の MOSメモリ Intel 1103 (1970年)<sup>2)</sup> は P 形であった。大型機で最初に採用したのは、IBM システム/370 モデル 158 と 168 (1972年) で、チップ当たり 1 kb, サイクル時間 460 ns, N 形であり<sup>1)</sup>、以後 N 形が主流になった。このころから ROM でない ICメモリをなぜか RAM (Random Access Memory) と呼び始めた。

同じころ我が国でもチップ当たり 144 b の NMOS

† An Overview of Storage Systems by Shigeru TAKAHASHI (Tokyo Engineering University).

†† 東京工科大学

\* 以下記憶容量に関して、b, B はそれぞれビット、バイトを表し、k, M, G, T はそれぞれ 2<sup>10</sup>, 2<sup>20</sup>, 2<sup>30</sup>, 2<sup>40</sup> を表すものとする。

のメモリが通産省大型プロジェクトの一環として日本電気により開発され、1969年に完成<sup>3)</sup>、初期のDIPS-1に容量8kB、サイクル時間80nsのキャッシュとして実用された<sup>4)</sup>。

MOSメモリにはさらにスタチック(SRAM)とダイナミック(DRAM)の種別がある。上述の日本電気のものはSRAM、IntelのものはDRAMであった。SRAMはフリップフロップで、DRAMはMOSのゲートまたは接合容量に貯えた電荷で記憶する。前者はセル当りの素子が多く、チップ面積が大きい欠点があるが、高速でかつ使いやすい。後者は改良の結果、セルを1トランジスタで構成できるようになり、大容量向きであるが、かつての静電式と同様に電荷が漏洩するからリフレッシュが必要で、速度と使いやすさではSRAMに劣る。

出現以来DRAM、SRAMともに急激に集積度が向上し、それぞれ4Mb、1Mbのチップが量産、実用されている。またそれぞれ集積度4倍のものがすでに開発され、さらにその4倍のものが報告<sup>5)</sup>されている。この急速な進歩を支えているのは半導体のプロセス技術と回路技術である。図-1に集積度の向上とプロセス技術進歩の傾向を示す。DRAM、SRAMともに集積度は20カ月

で2倍、MOSのゲート長やゲート膜厚などプロセス技術関連の寸法は3年でほぼ2/3になっている。なおSRAMの集積度はDRAMの約3年遅れと考えてよい。

集積度の向上にともない、まず問題になるのが消費電力である。PMOSとNMOSを直列にしたCMOSは定常状態で一方が必ずオフで、消費電力はきわめて低い。チップ面積、製造工数、速度などに難点があり用途が限られていた。一方DRAM256kb以上のNMOSの高集積化は消費電力で行き詰まったので、CMOSが見直され、上記の欠点はプロセス技術と回路技術の向上で克服された。現在ではSRAM、DRAMともにCMOS技術に大幅に依存している。

電源電圧を下げれば消費電力はさらに下がる。初期の4kbDRAMの電源は12V、±5Vと3種類もあったが、1980年64kbDRAMでTTLと馴染みの良い5Vだけになった。以来10年を経て今度はCMOS技術に合わせ、これを3.0-3.3V<sup>6)</sup>に下げることとなる。電圧を下げれば消費電力が減るだけでなく、ゲート膜厚の減少に適合して、信頼性も一層向上しよう。

なおチップ容量を $c$ 、チップのデータ幅を $d$ 、システムのデータ幅を $m$ とすると、メモリ増設単位は $cm/d$ となる。いま $c=4M$ 、 $d=1$ 、 $m=32$ とすると、これが128Mb=16MBとなり、中小形機では実情に合わない。 $c$ の今後の増加も考えると、 $d$ は4、8、16、32などから選択できることが望ましい。

集積度の向上とともに、記憶機能に加えてシステム的な機能をチップに組み込む傾向が強くなった。集積度が4倍になれば、テスト時間は簡単なパターンでも4倍、ビット間の干渉までテストすると16倍にもなるので、並列に4または8ビットをテストする機能を組み込むようになった<sup>7)</sup>。パッケージピンの制約からアドレスを行と列に分けて2度を送る方式は、4kbDRAMでMostek<sup>8)</sup>が開発し、事実上の標準になっているが、同じ行アドレスの下で列アドレスを変えて高速に読み書きする種々のモードが64kb時代に開発された<sup>9)</sup>。ニブル(4ビット)モード、ページモードなどがあるが、行を固定してどの列のビットでも読み出せるスタチックカラムモードが最も有用とされている。このモードはCMOS技術によるDRAM

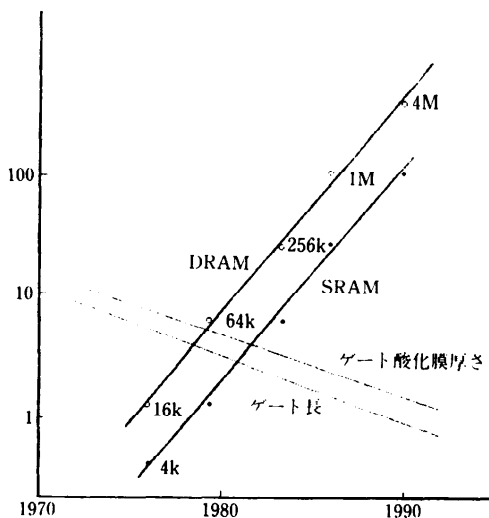


図-1 RAMの集積度の向上とプロセス技術の進歩  
縦軸目盛の単位 集積度: 10 kb, 酸化膜厚さ: 10 nm  
ゲート長:  $\mu\text{m}$

(富永四志夫, 小佐保信, 久保証治: CMOSの技術動向と応用展開, 日立評論, Vol. 66, No. 5, pp. 477-482 (1984) より許可を得て転載)

で経済的に実現できる<sup>10)</sup>。DRAM 特有のリフレッシュは馴れないシステムメーカに敬遠され勝ちであるが、単に指令だけ出せばよいように、必要なカウンタ、タイマなどを組み込むことが、これも 64 kb 時代に始まった。この機能を拡張してリフレッシュをまったく意識せずに済ますこともできる<sup>9)</sup>。さらに、キャッシュを組み込むなど、汎用から専用に移行する動きもみられる<sup>11)</sup>。

速度には種々の条件があり、一概にはいえないが、4 Mb DRAM でアクセス時間 70 ns, 1 Mb SRAM で 15 ns, バイポーラ RAM で 1.5-5 ns などというところが代表的な値であろう。

### 3. 主記憶システムの進展

主記憶システムは記憶容量の増大と速度の向上を追求して進展してきた。方式面で前者を可能にしたのはアドレス空間の拡大と仮想記憶方式であり、後者に大きく寄与したのはキャッシュメモリ方式である。以下これらの進展を概観する。

#### (1) アドレス空間の拡大

主記憶は処理装置が直接にアドレッシングするので、その容量を増すにはまずアドレス空間の拡張が必要になる。いまでは当然のように思われているベースレジスタを使うアドレッシングは、1964 年 IBM システム/360 で導入されたもので、命令語長とアドレス空間の関係を切り離し、後者の拡張を容易にした。ベースレジスタはしばしば指標レジスタと兼用されて、意義が曖昧になり勝ちであるが、それ以前の命令語のアドレス部は指標レジスタと同長であった。アドレス部にもたせるディスプレースメント(オフセット)を思い切って短かくし、ベースアドレスを加えてはじめてアドレッシングできるようにしたのが優れた考えであった。

システム/360 のアドレス空間は 16 MB, アドレスは 24 ビットに設定された。当時主記憶容量は 8 kB からあったから、16 MB はかなり思い切った値であったに相違ない。システム/360 アーキテクチャの存続中に、それが狭くなるとは、有能なアーキテクトも想像しなかったのであろう。その論文<sup>12)</sup>にはベースレジスタ採用の理由は述べているが、アドレス 24 ビットについての論議は見当たらない。システム/360 アーキテクチャの本質は 28 年後の現在でも変わっていないから、

永持ちし過ぎてアドレス空間の限界がきたともいえる。

1983 年 IBM はシステム/370-XA で、24 ビットを 31 ビットに、アドレス空間を 2 GB に拡張した。1987 年 IBM の Richard Case は雑誌の対談<sup>13)</sup>で、必要なアドレス空間は 30 カ月で 2 倍になる。7 ビットの拡張では間もなく不足すると述べた。前章に述べたように、RAM の集積度は 20 カ月で 2 倍になっている。アドレス空間と集積度は比例しないにしても、Case の予測はまだ内輪であった。

事実 1988 年 IBM は ESA (Enterprise Systems Architecture)/370<sup>14)</sup>を発表し、命令、データともに収容する従来のアドレス空間とは別に、データ空間を導入した。データ専用で命令は収容できないが、命令は従来のアドレス空間から、新しく設けた 16 個のアクセスレジスタを介して、この空間のデータをオペランドとしてアドレッシングできる。データ空間は一応 2 GB 区切であるが、連結して 16 TB ( $T=M^2$ ) まで広げられることもできる。従来から主記憶に広い場所を必要としていたのは、データであったから、この拡張でシステム/360 以来のアーキテクチャはまたしばらく延命されることとなる。日立<sup>15)</sup>、富士通<sup>16)</sup>とも同様なアーキテクチャの大型機を発表している。

アドレス空間の行き詰まりは大型機だけではない。Microsoft が 1980 年代初期に開発した MS-DOS はパーソナルコンピュータのオペレーティングシステムとして広く普及しているが、目標としたマイクロプロセッサが Intel 8086/8 でアドレス空間 1 MB であった。その後マイクロプロセッサのアドレス空間は、仮想記憶の採用もあって著しく拡大され、Intel 製では 80286 で 1 GB, 80386 で 64 TB もあるが、これらの新しいプロセッサを実装しても、MS-DOS は 8086 互換モードでしか使えないので、広いアドレス空間は役に立たず、単に実行速度が向上するだけである。

もちろんオペレーティングシステムを変えれば新しいハードウェアの機能は活かせるが、優れたオペレーティングシステムが完成し、その普及の見通しがつくまでには時間がかかる。また応用プログラムなど MS-DOS ベースのソフトウェア資産が莫大であり、変更は簡単ではない。そこで 8086 アーキテクチャのまま増設メモリを使え

るようにする EMS (Expanded Memory Specification) という仕様<sup>17)</sup>が設定されている。その仕組みについての説明は次章に譲る。

大型機でも小型機でもアーキテクチャとハードウェアの変更だけなら、アドレス空間の拡張は簡単であるが、ソフトウェア資産保存の制約の下にこれを実現するのは容易ではない。

## (2) 仮想記憶

実装された主記憶より広いアドレス空間を提供するのが仮想記憶方式の目的で、その最初の試みは 1959 年マンチェスタ大学で開発された計算機 Atlas の原形 Muse<sup>18)</sup>が行われた。今日ページングといわれている技術である。

図-2 に Muse の記憶システムの原理的な構成を示す。アドレス空間は 1M 語、これを 512 語を単位とする 2048 ページに分けた。アドレスは語単位で 20 ビット、うち 9 ビットはページ内アドレス、残り 11 ビットがページアドレスである。一方主記憶は容量 16k 語のコアメモリで、これをページと同じ大きさの 32 のページフレームに分けた。補助記憶として容量 2048 ページの磁気ドラムがあり、必要なページはドラムからコアに取り出して使う。現在どのページがどのフレームにあるかが分かるように、各フレーム対応にページアドレスレジスタを設け、ページアドレス 11 ビットを貯えておく。

ある 1 語を参照するには、まずアドレスの上位 11 ビットをページアドレスレジスタと比べる。一致するものがあれば、その語を含むページがコアに取り出されている。それがどのフレームにあるかはレジスタの番号で分かる。一致しない場合はそのページをコアに取り出すために、実行中の

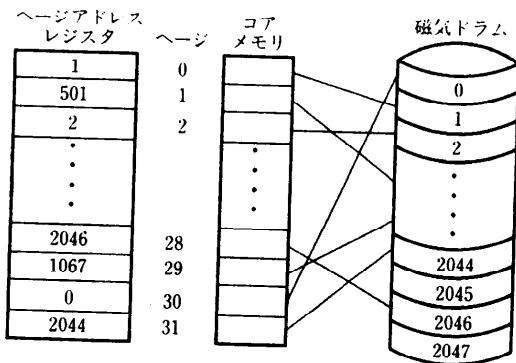


図-2 Muse の記憶システム

プログラムを一時停止し、ページ転送プログラムを起動する。このプログラムはまずコアに空きがあるかどうかを調べ、あればそこに必要なページを転送する。いっぱいならいずれかのページを選んでドラムに返し、その場所に必要なページをもってくる。

問題はどのページを追い出すかで、選択が悪いと同じページがまた必要になり、システムはページの入れ換えに忙殺される。これはスラッシング (thrashing) と呼ばれ、システムとして失敗である。Muse ではこの選択を「学習プログラム」によって行った。命令 1024 ステップを実行する時間を単位として、ハードウェアで各フレームへのアクセスを調べ、最近不活性であった時間がその前に不活性であった時間より増えているものを追い出した。現在広く使われている置き換えアルゴリズム LRU (Least Recently Used)<sup>\*</sup>以上のことをすでに考えていたのには驚かされる。ただし追い出しに際し、変更がなかったページは単に無効にすればよいが、Muse ではこれもドラムに返していた。

このページングでは 1M 語の仮想アドレス空間を 16k 語の実アドレス空間にマッピングし、容量 1M 語のコアメモリとほぼ等価な主記憶システムを経済的に実現しようとした。もちろん理想には及ばないが、メモリ参照の局所性によってその効果は十分あり、広く普及することとなった。

一方これとは異なる仮想記憶方式がある。元来データやプログラムは、分割して作った論理的には独立なセグメントの集まりである。従来はメモリに読み込む前に、アドレスを決めてこれらを結合しておかないとプログラムとして実行できなかった。大抵はアドレス空間が不足するので、用済みのセグメントを追い出し、その場所に次のセグメントを持ち込むオーバーレイの仕組みをプログラムが明確に指定しておく必要があった。これが面倒なので、セグメントを結合せず、それぞれに番号を与えてオペレーティングシステムで管理し、実行時にハードウェアの支持で実アドレスを与えるセグメンテーションと呼ぶ方式が、1961 年 Burroughs B-5000<sup>19)</sup>で提案された。図-3 にこの方式を概念的に示す。命令やオペランドのアドレス

\* ページを最近に参照されたものから順に並べておき、末尾のものを追い出すアルゴリズム。

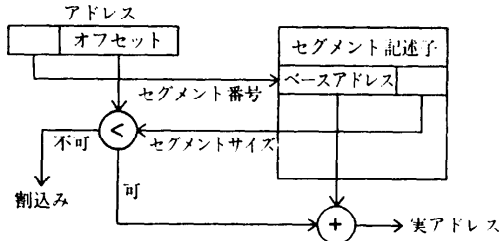


図-3 セグメンテーションの概念

はセグメント番号とセグメント内のオフセットで与え、各セグメントに対応してベースアドレス、セグメントの大きさなどを指定するセグメント記述子を設けておく。あるアドレスを参照すると、セグメント番号でセグメント記述子を引き、ベースアドレスにオフセットを加えて、実アドレスが得られる。

ページングでは、プログラムやデータを仮想アドレス空間に連続して並べたものを論理的構造とは無関係にページに分け、これらを実アドレス空間にマッピングする。一方セグメンテーションではプログラムやデータをセグメントに分けて仮想アドレス空間に配列する。したがってあるセグメントの大きさが変わっても、他のセグメントには影響がない。またセグメントの大きさや属性が分かっているからその保護が容易になるなど、アドレス空間拡張以外の利点がある。他方セグメントの大きさがまちまちで、仮想、実ともにアドレス空間に無駄な隙間ができる。MITのMULTICS<sup>20)</sup>で採用され、現在一般化した仮想記憶方式はセグメントをさらにページに分けるもので、セグメンテーションとページング両者の利点を活かしている。

仮想記憶方式は1972年IBMがシステム/370<sup>21)</sup>に採用して以来急速に普及し、いまではマイクロプロセッサにまで使われている。現在の用語では、参照したページが主記憶にない状態をページフォールト、そこで補助記憶から必要なページをもってくることをページイン、その場所を空けるのにページを追いつ出すことをページアウトという。ページアウトの際のページ置き換えアルゴリズムには種々のものがあるが<sup>22)</sup>、広く使われているのはLRUである。

ここで先に触れたEMSの説明をしておこう。

図-4にその概要を示す。MS-DOSが使う8086

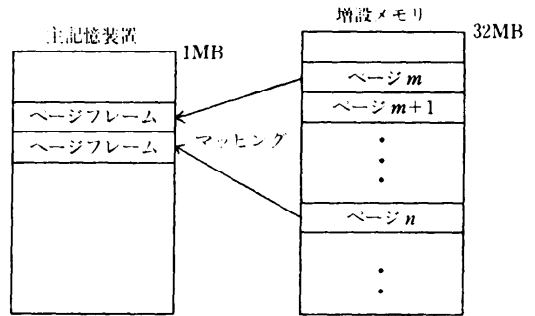


図-4 EMSの仕組み

モードのアドレス空間は1MBに限られているから、EMSでは最大32MBの増設メモリ(RAM)を16kB単位のページに区切り、これを1MBの空間に設けた若干のページフレームにマッピングして使う。仮想記憶でのページングに似ているようであるが、ページングが論理的なアドレス空間のページを物理的な実体であるフレームにマッピングするのに対し、EMSは物理的な実体であるページを論理的なアドレス空間の一部であるフレームにマッピングする点で本質的に異なる。EMSはフレームがもつアドレス空間を、RAMで構成された実体である自分のメモリバンクを一時はめてむために借用する苦肉の策である。

(3) キャッシュメモリ

主記憶の速度は処理装置に比べて常に遅い。これに対応する種々の工夫が古くからあるが、アーキテクチャに無関係に広く用いられているのは、1968年IBMシステム/360モデル85が始めたキャッシュメモリ方式<sup>23)</sup>である。図-5に示すように、処理装置と主記憶との間に高速小容量のメモリ(キャッシュ)をおく。処理装置は必要な主記憶の内容をすべてキャッシュから取り出して使う。初めて使うものはキャッシュになく、主記憶

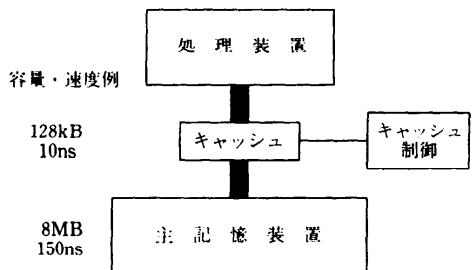


図-5 キャッシュメモリの概念

■: ハードウェア制御による双方向転送

からもってくるが、2度目からはキャッシュにあり、すぐ使うことができる。キャッシュがいっぱいになると既存のものを追い出してそこに新しく必要になったものを入れる。キャッシュの目的は、主記憶の容量で速度はキャッシュに近いメモリを、参照の局所性を利用して等価的に実現することである。

キャッシュと主記憶との関係は、Atlas のページングでの磁気ドラムとコアメモリの関係に似ているが、大きな相違がある。ページングではページフォールトが起こるとただちに制御プログラムが介入し、ページアウト、ページインなどの処理はプログラムが行う。一方キャッシュでは必要なものがないと、それに続く処理はすべてハードウェアが行う。補助記憶装置と主記憶の速度の差は、プログラムが介入し、要すればプロセスを切り換えるほど大きい。キャッシュと主記憶の速度の差はそれほど大きくないからである。このようにキャッシュの制御にはプログラムは一切関係しないので、キャッシュ (cache, 隠された袋) という名のとおり、その存在はプログラムには見えない。

キャッシュを参照して必要なものがキャッシュにある率をヒット率と呼ぶ。キャッシュは主記憶の部分的コピーである。コピーする単位をブロック、キャッシュを分割してブロックを収容する場所をフレームと呼ぶと、ブロックの大きさ、ブロックとフレームとの対応、ブロックの置き換えアルゴリズムなどがヒット率に関係する。種々の条件でヒット率を調べた結果、ブロックとフレームの対応はセットアソシアティブ方式、置き換えアルゴリズムは LRU が良いとされている。

さらにキャッシュに書き込む場合、キャッシュと主記憶の双方に書くストアスルー (ライトスルー) 方式と、キャッシュにだけ書いておき、ブロックを追い出すときこれを主記憶に転送するストアイン (ライトバック) 方式がある。前者は原本とコピーを常時一致させる方式、後者はあとで一括訂正する方式で、前者のほうが常に一致が取れている利点があるが、ストア命令の場合にキャッシュの効果がない。高速化の要求から大型機では後者が多くなっている。

半導体技術の進歩にともなうスケールメリットの消失<sup>24)</sup>により、数台の処理装置が主記憶を共用

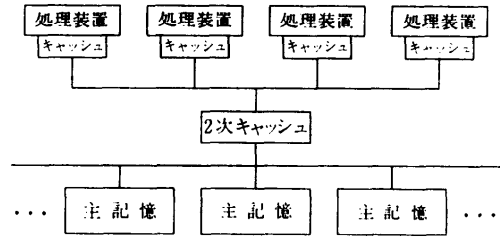


図-6 多重プロセッサシステムとキャッシュ

するシステムが普通になった。この場合図-6に示すように、各処理装置がそれぞれのキャッシュをもつ。さらにこれらの高速キャッシュとは別に、やや低速な2次キャッシュを主記憶側に設けたシステムもある。コピーが方々にできて、これらと主記憶との一致制御が複雑になる<sup>25)</sup>。キャッシュと2次キャッシュとの間はストアスルー、2次キャッシュと主記憶との間はストアインというのが一つの妥協であろう。

なお各処理装置のキャッシュを命令用とデータ用に分ける方式もある。

#### (4) メモリの誤り制御と保護

各メモリバンクでのハードウェアによる誤り訂正検出は大型機では不可欠で、小型機でもメモリ容量の増大にともない必要性が増している。大型機では1ビット誤り訂正・2ビット誤り検出が普通であり、後述の IBM ESA/370 拡張記憶では2ビット誤り訂正・3ビット誤り検出を行っている<sup>25)</sup>。一般にメモリ幅が大きいほど誤り制御のための冗長度は少なく済むが、たとえば主記憶のメモリ幅がキャッシュの2倍だと、ストアスルーでは単に書き込むだけではなく、一度全体を読み出した上で改めて冗長ビットを作り出すことが必要になり、この点では必ずしも有利ではない。

記憶容量増大の目的の一つは多数のプロセスを並行して進めることであるから、各プロセスが使うメモリ領域を他のプロセスによる書き込みや読み出しから保護することが当然必要になる。

物理的なメモリ領域にメモリーキーを与え、プログラムがもつアクセスキーと比較する保護方式はシステム/360の時代からあり、領域の単位が2kBから4kBに変わったが、現在でも行われている。限界レジスタによる保護も古くからあり、セグメンテーションに引き継がれている。セグメンテーションではさらにセグメントの属性を指定

することによる保護もできる。一方仮想記憶方式は多数のユーザにそれぞれ独立した仮想アドレス空間を与えるように拡張され、自動的に保護が強化されていたが、ESA/370 などでは逆にこれらのユーザ間で資源を共有する機能が強化され、そのための保護機能が必要になっている<sup>20)</sup>。データ空間は属性が異なるので、記憶保護の観点からも望ましい拡張である。

#### 4. 記憶階層

記憶システムの進展を支配してきた指導的な原理は、図-7 に一例を示すようなピラミッド形の階層構造によって、理想的な高速大容量メモリシステムを経済的に実現しようというものである。この指導原理を支持する経験則として下記のメモリ参照の局所性がある。

(1) 一度参照したものは、間もなくまた参照する可能性が高い (時間的局所性)。

(2) ある場所にアクセスすると、その付近にまたアクセスする可能性が高い (空間的局所性)。置き換えアルゴリズムが(1)に、ブロックやページの大きさの選定が(2)に関係する。仮想記憶、キャッシュメモリなどは、これらの指導原理と経験則に沿って開発された方式である。

図-7 の各層に対して容量と速度を例示したが、これらはあくまで例に過ぎない。またこの構造では主記憶と高速キャッシュの間に2次キャッシュを、主記憶の下位に拡張記憶を、また磁気ディスクにディスクキャッシュを設けているが、これらの有無はシステムによる。

層間で行われるデータの授受にはハードウェア制御のものとプログラム制御のものがある。前者はキャッシュを含むもので、すべてハードウェア

によって行われ、プログラムには見えない。後者はオペレーティングシステムの介入によって行われる。図-7 の例では層 3-4、3-5 間のデータの授受は後者である。

図-7 の例で第 4 層の拡張記憶は ESA/370 で導入されたものである。システム/370 アーキテクチャでは、仮想アドレス空間にも増して、実アドレス空間の限界が問題であった。この限界は 2GB のままであるが、ESA/370 ではオペレーティングシステムで制御する RAM を拡張記憶として主記憶の下位に設けて、この問題を解決しようとしている。拡張記憶の容量の上限は 16 TB で、主記憶とのデータの授受は 4 kB ブロックで行う。主記憶からみると高速ではあるが補助記憶の一つである。

拡張記憶は別として、従来の補助記憶と主記憶との間のデータの授受は、機械的に動作する磁気ディスクが含まれるため、非同期的に行う点で他の層間のものとは異なる。図-7 の例ではオペレーティングシステムには第 5 層は見え、5-6 層で一つの磁気ディスクシステムに見えるが、この磁気ディスクシステムと主記憶との間のデータの授受は非同期的な入出力操作である。これに 2 種類あり、仮想記憶制御の一環として行うものと、ファイルの入出力として行うものがある。いずれもオペレーティングシステムの制御下ではあるが、前者が完全にその管理下にあるのに対して、後者はユーザが主体で、オペレーティングシステムは実行を制御しているに過ぎない。

記憶階層と仮想記憶の考え方をさらに進めて、仮想アドレス空間を十分広く取り、データをファイルとして収容している補助記憶領域にまでこれを拡大すれば、ファイル入出力をこのアドレス空間でのデータの移動として実行できる。結果として記憶領域の概念が単純化され、プログラムが容易になるだけでなく、補助記憶を含む入出力を記憶階層間のデータの授受として、ユーザではなくオペレーティングシステムによって一元的に管理できて効率が向上するとされている。しかしこれによって、従来のファイル入出力がなくなるわけではない。

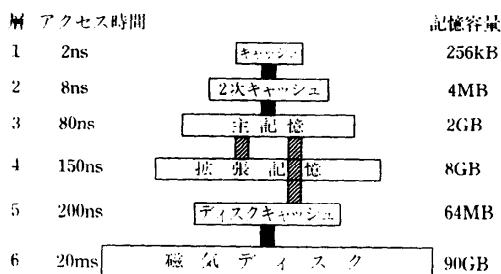


図-7 記憶階層の一例  
 ■: ハードウェア制御の転送  
 ▨: プログラム制御の転送

## 5. 補助記憶装置

主記憶装置の容量は半導体 RAM の進歩により飛躍的に増大したが、記憶容量に対するニーズも、データベースなどの応用の進展にともないこれを上回る勢いで増え、計算機システムにおける補助記憶装置の重要性はますます増大している。もちろんそのビット当たりコストの低下も著しい。

### (1) 磁気テープ装置

1960年代中ごろまでは磁気テープ装置が補助記憶の主役であったが、その後急速に発達した磁気ディスクにその地位を譲り、その役割は情報交換媒体である磁気テープの入出力と磁気ディスクのバックアップに変わった。

### (2) 磁気ディスク装置<sup>26)</sup>

補助記憶装置として現在最も重要な磁気ディスク装置は、技術的には磁気ドラムの延長である。ドラムの円筒を円板にし、ヘッドをその表面に沿って移動できるようにして容量を大きくしたもので、広く用いられるようになったのは1963年IBMが1311形を発表してからである。動圧空気軸受技術によりヘッドを表面からわずかに浮上させ、直径14インチの6枚のディスクに2.68MBを貯えた。駆動装置から取り外せるディスクで、ディスクパックと呼ばれ、IBMが1973年に発表した3330-11まではこの方式であった。記憶容量はこの時点でパック当たり200MBになっていた。

IBMは同年ウインチェスタと呼ばれる3340形を発表した。この技術では静止状態でヘッドはディスク表面に接触していて、ディスクが回転始めると風圧で浮上する。ディスクの表面に傷がつかないようにヘッドは十分軽くしておく。また従来は別々の部品であったヘッドとディスクを一体にして、HDA (Head Disk Assembly) とし、装置から取り外さないことを原則とし、取り替えるときには一体として取り替えることによって、互換性の問題を一挙に解決した。

この方式になってからの進歩はさらに著しく、最近のものではHDA当りの容量2.92GBに達した。もちろん媒体、磁気ヘッドなどの材料、ヘッドの構成、ヘッド位置決め機構などの基盤技術の進歩があつてのことであるが、記憶方式の進歩と信頼性を高めるための誤り訂正・検出方式の進

歩によるところもきわめて大きい。

またウインチェスタ技術は大型だけではなく、小型ディスクに広く用いられ、その進歩と価格の低下はパーソナルコンピュータやワークステーションの普及と相俟って特に著しい。

### (3) キャッシュつき磁気ディスク装置

半導体RAMのキャッシュを磁気ディスク制御装置に設ければ、主記憶と磁気ディスクとの間の大きな速度ギャップを埋めることができる。処理装置のキャッシュと異なり、一度に読み書きするブロックが大きいから、キャッシュとして最低50MB程度の容量が必要であるが、半導体技術の進歩により、コストの問題も解決されつつある。

処理装置の場合と同じく、書き込みの場合ストアインか、ストアスルーかという問題がある。磁気ディスクは不揮発性が建前であるから、電源断などの場合にキャッシュに書き込んだものが失われるのでは、システムの信頼性が著しく低下し、使いものにならない。ストアスルーなら問題はないが、平均5回に1回はある書き込みでキャッシュの効果がまったくないことになる。そこでSRAMによるバックアップメモリを設け、その電源を電池でバックアップしてストアインを可能にするようになった<sup>27)</sup>。

## 6. おわりに

半導体技術の進歩に支えられた記憶素子にしても、また磁気ディスクにしても、ハードウェアの性能の向上はまさに目覚ましいものがある。一方これを利用する計算機システムのほうは、量的な進歩はみられるものの、方式的にはアドレス空間の行き詰まりをなんとか切り抜けている程度で、仮想記憶にしても、キャッシュにしても1960年代の方式をハードウェア技術の進歩に応じて形を変えて適用しているに過ぎない。

ソフトウェアの資産が方式面での進歩の足枷になっているが、今までのように動きのとれない、しかもシステムによって異なるインタフェースではなく、ハードウェア、ソフトウェアともにもう少し自由度をもってシステムを開発できるインタフェースを設定できないものだろうか。



## 参 考 文 献

- 1) Pugh, E. W., Critchlow, D. L., Henle, R. A. and Russell, L. A.: Solid State Memory Development in IBM, IBM J. Res. Dev., Vol. 25, No. 5, pp. 585-602 (1981).
- 2) Regitz, W. M. and Karp, J.: A Three-Transistor-Cell, 1024 bit 500 ns MOS RAM, ISSCC '70 Digest of Technical Papers, pp. 42-43(1970).
- 3) Tarui, Y., Hayashi, Y., Koyanagi, T., Yamamoto, H., Shiraishi, M. and Kurosawa, T.: A 40 ns 144 bit N-channel MOS-LSI Memory, IEEE J. Solid-State Circuits, SC-1, No. 5, pp. 271-279 (1969).
- 4) Takashima, K., Toda, I., Arai, K. and Yamada, M.: A Large-Scale Data Processing System: DIPS-1, First USA-Japan Computer Conference Proceedings, pp. 193-202 (1972).
- 5) ISSCC '92, ポスト 5 V時代の先端プロセッサを探った, 日経エレクトロニクス, No. 548, pp. 127-161 (1992).
- 6) さらに, 5 V単一電源, ボード設計が混乱期に, 日経エレクトロニクス, No. 527, pp. 143-199 (1991).
- 7) 宮沢一幸, 石原政道, 下東勝博, 清水真二: 4 Mビットダイナミック RAN 技術, 日立評論, Vol. 69, No. 7, pp. 63-66 (1987).
- 8) 選択範囲がますます広がるメモリー, 日経エレクトロニクス, No. 158, pp. 58-90 (1977).
- 9) 収穫期を迎える 64 K ダイナミック RAM, 日経エレクトロニクス, No. 298, pp. 147-188 (1982).
- 10) 石原政道, 宮沢一幸, 酒井 修: サイクル時間 50 ns のスタック・コラム・モード付き 256 K CMOS ダイナミック RAM, 日経エレクトロニクス, No. 362, pp. 243-263 (1985).
- 11) 松田吉雄, 日高秀人, 朝倉幹雄, 藤島一康: キャッシュ・メモリ内蔵 DRAM を試作, 主記憶が 1チップに収まる時代をにらむ, 日経エレクトロニクス, No. 475, pp. 129-144 (1989).
- 12) Amdahl, G. M., Blaauw, G. A. and Brooks, Jr. F. P.: Architecture of the IBM System/360, IBM J. Res. Dev., Vol. 8, No. 2, pp. 87-101 (1964).
- 13) Gifford, D. and Spector, A.: Case Study: IBM's System/360-370 Architecture, Comm. ACM, Vol. 30, No. 4, pp. 292-307 (1987).
- 14) Aken, Jr., B. R.: Large Systems and Enterprise Systems Architecture, IBM Systems J., Vol. 28, No. 1, pp. 4-14 (1989).
- 15) 吉沢康文, 新井利明, 原田 晃, 旭 寛沢: OS VOS 3/AS の高性能・大容量化方式, 日立評論, Vol. 73, No. 2, pp. 175-184 (1991).
- 16) 中島幸雄, 竹内 誠, 沖原宣行: FUJITSU M-1800 モデルグループのソフトウェア: MSP-EX, FUJITSU, Vol. 42, No. 3, pp. 139-150 (1991).
- 17) 拡張進む MS-DOS, 日経バイト, No. 11, pp. 151-177 (1988).
- 18) Kilburn, T., Edwards, D. G. B., Lanigan, M. J. and Sumner, F. H.: One-Level Storage System, IRE Trans. EC-11, No. 2, pp. 223-235 (1962).
- 19) The Descriptor—A Definition of the B 5000 Information Processing System: Bulletin 5000-20002-P, Burroughs Corp. (1961).
- 20) Corbato, F. J. and Vyssotsky, V. A.: Introduction and Overview of the Multics System, Proc. F. J. C. C., pp. 185-196 (1965).
- 21) System/370 Principles of Operation, GA-22-7000, IBM Corp. (1973).
- 22) 益田隆司, 亀田壽夫: オペレーティングシステムの性能解析, 128 p., 情報処理学会, 東京 (1982).
- 23) Conti, C. J., Gibson, D. H. and Pitkowsky, S. H.: Structural Aspects of the System/360 Model 85, 1, General Organization, IBM Systems J., Vol. 7, No. 1, pp. 2-14 (1968).
- 24) 高橋 茂: 計算機の性能と価格, 情報処理学会論文誌, Vol. 29, No. 5, pp. 529-538 (1988).
- 25) Scalzi, C. A., Ganek, A. G. and Schmalz, R. J.: Enterprise Systems Architecture/370: An Architecture for Multiple Virtual Space Access and Authorization, IBM Systems J., Vol. 28, No. 1, pp. 15-38 (1989).
- 26) Harker, J. M., Brede, D. W., Pattison, R. E., Santana, G. R. and Taft, L. G.: A Quarter Century of Disk File Innovation, IBM J. Res. Dev., Vol. 25, No. 5, pp. 677-689 (1981).
- 27) 宮崎道生, 久野 潔, 北島弘行, 加藤善久, 川村哲士: 新キャッシュ付きディスク制御装置 "H-6581-C3", 日立評論, Vol. 73, No. 2, pp. 211-216 (1991).
- 28) 浦城恒雄: キャッシュメモリの一致性について, 情報処理, Vol. 32, No. 1, pp. 64-73 (1991).

(平成4年5月27日受付)



高橋 茂 (名誉会員)

1921年生。1944年慶應義塾大学工学部電気工学科卒業。同年電気試験所に入る。1956年同所電子部回路課長。わが国最初のトランジスタ計算機 ETL Mark 3, Mark 4 を開発。1962年(株)日立製作所に移り, 同社神奈川工場開発部長, 設計部長, 副工場長, 同社コンピュータ事業本部次長を歴任。HITAC 8000 シリーズ, DIPS, M シリーズなどの製品計画並びに開発を担当。1980年筑波大学教授。1986年東京工科大学教授, 1989年同大学副学長。1979-1980年本会副会長, 1988年本会情報規格調査会長。1988年本会功績賞。電気学会, 電子情報通信学会, 米国 IEEE 各会員。工学博士。