Vol. 4 No. 6　　情　報　処　理　　Nov. 1963

記憶装置の多重構成*

西　野　博　二**

まえがき

計算機を設計するに当って，記憶装置の種類，速度および容量をどのように選ぶかは，計算機の性能に最も大きな影響を与えるから，方式設計上の重要な課題である．記憶装置技術の歴史的な発達につれて，その時点において高速な主記憶装置と，その記憶容量の不足を補うために，低速ではあるが大容量の補助記憶装置という構成が，多くの計算機に採用されている．これが最も単純な記憶装置の多重構成であり，記憶装置の多重構成そのものは，特別に新しいものではない．

ただ最近になって主記憶装置よりなお高速な各種のレジスタ記憶装置や固定記憶装置が設けられるようになるなど，ひとつの計算機のなかで，記憶装置を何種類も積極的に使用しようとする傾向が増加していること，またそのために種類の異なる記憶装置を全体として調和のとれたものにするための方法が，強く求められていることが指摘できる．ここでは，このような記憶装置の多重構成に関する問題点と，これに関して最近開発された新しい技術について解説してみよう．

1. 多重構成の原理

記憶装置の多重構成（hierarchy of memories）という言葉に，筆者が始めて接したのは J. von Neumann の遺著“The Computer and the Brain”[1]によってである．Neumannは「多重構成の原理」を次のように述べている．

計算機である問題を解く場合に，N語の記憶容量を呼出時間 t で必要とすると仮定する．しかしN語のすべてを，必ず呼出時間 t で得る必要はなく，かなり少ない N_1 語だけ t 時間で利用できれば，残りの $N-N_1$ 語は t より長い呼出時間であってもよいということは十分考えられる．このような考え方を続けてゆけば一般的に記憶装置の容量の系列 $N_1, N_2 \cdots\cdots N_k$ と呼出時間の系列 $t_1, t_2 \cdots\cdots t_k$ を対応させ，記憶容量および呼出時間とも，系列の後の方を大きくして最初の必要な条件を満足させることができる．（ここで添字は記憶装置のレベルを意味する）

Neumann の定義によれば，上述の系列の最初のレベルは論理素子からなるレジスタであり，最後のレベルは，カードや紙テープなどの外部記憶*である（その中間のレベルに，いわゆる主記憶装置や補助記憶装置がある）．したがって，彼によると現在の計算機では，少なくとも3レベルの記憶装置を持つことになる．

Neumann の簡単な説明を敷衍したものとしてC.F. King の論文[2]がある．King は，記憶装置が実に10レベルからなる仮想的な計算機について，各レベルの記憶装置の役割や，概略の仕様について説明し，かつある特殊な目的の計算機では，このような多重構成が必要であると述べている．

しかし，このような多重構成の説明はあまりに概念的であり，今日の計算機で採用されている記憶装置の構成を，十分説明できるような具体性に欠けている．

我々が常識的な意味で記憶装置の多重構成を問題にする場合は，論理素子で作られたレジスタや外部記憶は記憶装置の範疇から除外して考える．また磁気テープ装置も入出力装置とみて多重構成の問題には含めない．あくまで内部記憶装置の多重構成が問題である．

記憶装置を多重構成にしなければならない直接の原因は，なによりもまず内部記憶装置の製作に関する技術的要求か経済的要求である．必要とする呼出時間 t 容量Nの記憶装置が技術的に実現できないか，またはたとえ実現できるにしても，その代償として払わねばならない価額が高すぎるということである．

また内部記憶装置が多重構成である場合に memory allocation の問題がプログラマにとって厄介なものであることは周知のことである．特に，熟練しないプログラマにとっては，著しい計算時間の非能率か，または memory space の無駄使いになる．また計算機設計者にとっても，計算機の構成を複雑にするための不利益がないわけではない．

* Hierarchy of Memories, by Hiroji Nishino (Electrotechnical Laboratory, Tokyo)
** 電気試験所電子計算機部

* 計算機から切り離されているが，計算機が受け とれる形で情報を保持する媒体．たとえば，磁気テープ，紙テープ，カードなどがある．〔JIS Z 8112 計数形計算機用語〕外部記憶は しばしば 補助記憶装置と混同される．

それにもかかわらず内部記憶装置の多重構成が多くの計算機に採用されているということは，如何に前述の技術的制約と経済的要求が強いかということを示すものであろう．また別の観点に立てばプログラマの負担をなくするための解決方法の発達に，最も重要な鍵があるといえる．

2. 多重構成の現状

現在の計算機の最も標準的な主記憶装置は，磁心マトリクスであるが，新しく設計される計算機ほど，記憶容量が増加する傾向にあり，またサイクル時間の方はますます短かくなってきて，最近では 1 μs 程度のものも設計されている．

しかし，磁心記憶装置のサイクル時間の短縮は，ほぼ限界に近づいてきているようである．もちろん高速化の努力は，パーシャル・スイッチングの技術や磁心をますます小形化する製造技術の開発などによって，たゆまなく続けられているが，現在のものより 1 桁以上も高速化するのは極めて困難である．いきおい高速化の代償として，容量を減らさざるを得ない．

また最近開発されているトンネル・ダイオードや磁性薄膜素子を利用した記憶装置も，素子としては磁心よりも高速ではあるが，やはり大容量のものの実用化には多くの技術的困難がある．

したがって，現在の記憶装置技術の必然性から生まれた高速小容量の記憶装置を計算機の高速化のためにどのように効果的に利用するかが，新しく設計される計算機では重要な課題となっている．

このような方向の一つは論理素子で作られたレジスタ，たとえばインデックス・レジスタなどを，高速記憶装置で置き換えることである．さらにこのような傾向を押し進めていけば，制御演算装置の全てのレジスタを高速記憶装置にしてしまうということになる．

また別のいき方として，このような高速小容量の記憶装置の新しい利用法を積極的に考え出すことがある．たとえばプログラムを先廻りして取り込んでおくためのプログラム・スタックや last-in first-out の原理に従う push-down 記憶装置の採用などがある．このような，いわゆるレジスタ記憶装置の詳細については，この特集号では別の著者によって述べられるはずであるから，ここではこれ以上触れないことにする．

一方，主記憶装置の容量は，ますます増える傾向にあるが，記憶容量の増加に決定的な役割をになっている要因のひとつは自動プログラミング技術の発達であろう．ALGOL, COBOL, FORTRAN などの問題向き言語を採用して，このような source language での debugging を容易にするような高級なものになれば，数万語のソフトウェア・システムも珍らしくない．

また計算機の利用価値がますます認識され，その利用分野が広まるにつれ，ますます取り扱う問題も大形化していく傾向にあるのは当然である．このように考えてくると，記憶容量の増大についての要求は際限のないように思われる．

このような要求に応ずるために，大形計算機では莫大な容量の磁心記憶装置を持つものがある．例えば，IBM 7030, CDC 3600, RCA 501 などでは，それぞれ 16 K 語を基本として 262 K 語まで持つことができる．

しかし，記憶容量に対する要求が，際限のないものであっても，金物としての記憶装置の容量には以下のような制限がある．

まず，磁心記憶装置の速度は，容量が大きくなるにつれて，信号の伝送時間も長くなるために遅くなる．また容量が増すにつれて，雑音も大きくなり信号対雑音比（S/N比）が劣化する．これを避けるためには，記憶装置を幾つかに分割して，その分割された 1 単位では，信号の伝送時間があまり問題にならないで，かつ S/N 比が著しく劣化しない程度の大きさに止めねばならない．このようなスタックの大きさは速度によって異なるが，4 K から 16 K 語が標準となっている．容量を増やすには，スタックを複数個並列におくことになる．

第 2 に，当然のことながら，容量の増大につれて記憶装置の価額が高くなることが挙げられる．磁心記憶装置のビット当りの価額は，磁気ドラム記憶装置や磁気ディスク記憶装置に比較すると，1 桁から 2 桁程度も高い．したがって，磁心記憶装置は必要最小限に止め，その容量の不足は磁気ドラム，磁気ディスクなどの補助記憶装置で補うことが，経済的なシステムの構成法となる．

また命令語で指定できるアドレスのビット数には制限があり，これを増やせばそのための容量増加も生じてきて，主記憶装置の容量は無制限にふやせない．

3. ブロック転送

低速な補助記憶装置の内容を必要に応じて，高速な主記憶装置に移すには 1 語ずつでは能率が悪いから，前者の記憶場所を数十ないし数百語を単位とするブロックに分割し，ブロックごとにまとめて情報のやりと

りをするブロック転送が普通行なわれている．

しかし，このようなブロックを常に意識して，転送が能率的に行なわれるようなプログラムを作ることは，プログラマにかなりの負担がかかる．多少能率が悪くても，プログラム上ではブロックを意識しないでもよい方法が要求される．

最も原理的な方法は，補助記憶装置の1ブロックに相当する記憶場所のバッファを主記憶装置の一部に設けて，情報のやりとりはこのバッファを介して行なう方法である．

たとえば，磁気ドラムのN語の内容を磁心に読み出すようなマクロ命令を考える．まず必要なことは，磁気ドラムのアドレスが，どのブロックにあり，またそのブロック内のどの位置にあるかを計算することである．次に必要なブロックの内容がバッファに移され，さらにバッファから磁心記憶装置の最終の記憶場所に移される．

磁心の内容を磁気ドラムに逆に書き込むマクロ命令の場合には，読出しの場合よりやや複雑である．ブロック内に転送に無関係な記憶場所があるときには，あらかじめ該当するブロックの内容を磁気ドラムからバッファに持ってきておいてから，バッファの必要な記憶場所だけを書きかえることが必要である．連続したアドレスを持つ数ブロックの転送をするときには，一番最初と最後の二つのブロックで，このような手順が必要になる．

このような方法を拡張していけば，2レベルの記憶装置を1レベルに近い取り扱いのできる解釈ルーチンができる[3]．ただし，バッファと主記憶装置との間の情報のやりとりのためのむだ時間は避けられない．

しかし，このような方法も，機械に固有なアセンブラ言語を使って，プログラムを書く場合であって，ALGOLのような問題向き言語を使う場合には，問題は一層困難になる．

J. Jensenなどは，ALGOLのcommentを利用して，磁気ドラムと磁心の記憶場所を割りつける方法[4]について述べているが，二つの記憶装置の間の情報のやりとりをoptimumにするようなことは始めから問題の対象にしていない．むしろプログラマにとって，多重構成の記憶装置を有効に使用することが，困難である実例を示しているものと考えられる．

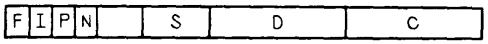
ソフトウェアだけで，多重構成の難点を解決するには，次節に述べるように相当大がかりなソフトウェア・システムを必要とする．

4. 制御語の利用

あるプログラムの実行が実際の記憶場所に無関係であるためには，そのプログラム自身やデータの収容場所が，基準となるベース・アドレスに相対的に指定されていることが必要である．これはプログラミングの段階では記号アドレスを用いることによって，極めて初期の頃から実用されているし，計算の実行段階でやる場合には，相対アドレス方式として知られている．

また直接アドレスがoperandの記憶場所を直接示すのに対して，operandのアドレスを含む記憶場所のアドレスを示す間接アドレスがある．間接アドレス方式によっても，媒介となる記憶場所の内容を変えることによって，実際のアドレスを計算の実行段階で変更することが可能である．

このような考え方を拡張して，Burroughs社のB-5000[5]では制御語を利用して，プログラムをもっと自由に実際の記憶場所したがって記憶装置の多重構成と無関係にしようとしている．B-5000にはprogram reference table（PRT）と称する1,024語の領域があり，プログラムの実行時に必要な種々の制御語が収容されている．現在の説明に必要なものは，プログラムとデータのための制御語で，その内容は**第1図**に示



F: Flag　　S: Size Field
I: Identification　　D: Drum Address
P: Presence　　C: Core Address
N: Drum Number

第1図

すようなものである．

IとSの情報はコンパイルの段階で挿入され，磁気ドラムの番号とそのアドレスは主制御プログラムによって，loadingのとききめられる．磁心のベース・アドレスは，その領域が使用可能なときに主制御プログラムが予約する．

またPのビットは，プログラムやデータが実際に磁心にあれば0に，磁気ドラムにあれば1にしておく．計算の実行段階で，PRTの中のこの制御語を調べてみて，もしこのビットが1ならば，割り込みが起って磁心と磁気ドラムの間でブロック転送が実行される．

プログラマは，強力な主制御プログラムの介入によって，記憶装置の金物を意識する必要はないが，このような徹底した間接方式によるかなりの速度の低下は

避けられない．

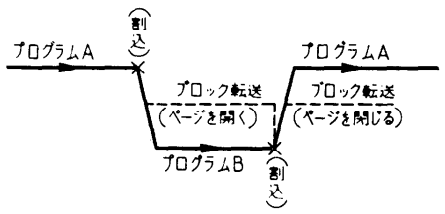
5. ページ・アドレス方式

プログラマにとって，記憶装置は全て一様である，すなわち1レベルであるようにするために，かなりの金物を設ける方法の代表的なものが，Manchester 大学と Ferranti 社の Atlas で採用されているページ・アドレス方式[6]である．

Atlas のページ・アドレス方式では，磁気ドラムと磁心をそれぞれ512語を単位とするブロック（磁心のブロックを特にページという）に分割し，磁心の各ページに対応して，ページのアドレスを収容するページ・アドレス・レジスタ（PAR）を設ける．計算機の命令が記憶装置を参照するごとに，全ての PAR の内容を調べて該当するアドレスが，どのページに含まれているかを照合する．もし該当するページがあればそのまま磁心記憶装置の内容の読出しまたは書込みが行なわれる．もし該当するページがなければ，割り込みが生じて主制御プログラムにより磁気ドラムと磁心の間で自動的にブロック転送が行なわれる．

この方式の特徴は，磁心記憶装置は常に磁気ドラム記憶装置の“写し”である点である．固有のアドレスが付いているのは磁気ドラム記憶装置だけで，磁心記憶装置の方には固有のアドレスがない．制御装置からみれば，磁心にあるブロックはページが“開いて”おり，磁心にない場合はページが“閉じて”いる．

閉じているページを開くのは，主制御プログラムの役割であるが，このために磁心記憶装置の1ページ分をブロック転送のためのバッファとして常に空けておく．このバッファを利用して，必要なページを開ければ，プログラムは先に進むことができ，不要なページを閉じて，新しいバッファを設けておくことは磁心記憶装置の空いている時間を利用してプログラムの進行とは無関係に自動的に行なわれる．また**第2図**に示すようにページを開いている時間も計算機は遊んでいないで別のプログラムを実行できる．



第 2 図

開いているページを閉じるには，過去におけるページの利用度の統計を基礎にして判断する．Atlas の場合は，各ページにこの統計のデータをとるために一つのフリップ・フロップがあり，記憶装置を1,024回参照する間に，1回でもそのページが呼び出されればそのページのフリップ・フロップがオンになる．主制御プログラムにはこのデータを基礎にして，ページの切り替えを最少にするような，学習プログラムがある．

ページ・アドレス方式の付随的な利点に，PAR を利用して memory protection が容易に行なえることがある．また PAR が機能的には連想記憶装置 associative memory そのものである点も興味がある．

しかし，ページ・アドレス方式にも欠点がないわけではない．ページの切り替えのためのむだ時間以外に PAR の内容を調べるために記憶装置の呼出時間が長くなる．またページが開いていないための割り込みが最優先になるから，先廻り制御装置の設計は，このために相当な制限をうける．

あとがき

Atlas のページ・アドレス方式，B-5000 の制御語を利用した方式のいづれも，プログラムの原籍が磁気ドラムにあるから，原理的には磁気ドラムを主記憶装置とみることができる．

このような観点からすれば，今まで主記憶装置とそれより低いレベルの補助記憶装置の間の問題として説明してきたことが，実は主記憶装置とそれより高いレベルの記憶装置の間の問題としてみることもできる．

これは命令のアドレス部が十分大きくて，磁気ドラムの全容量を含むことによってのみ可能である．磁気ドラムの記憶場所の容量が，命令のアドレス部によって指定できる範囲を超えるときには，磁気ドラムは入出力装置と同じ扱いをうける．

いずれにしろ，プログラミング cost を除外すれば，多重構成のために生ずる memory allocation の困難さを，金物によるかソフトウェアによって償えば，結局その方式の評価は速度と金物の経済性で判断することになる．

参考文献

1) J. von Neumann: The Computer and the Brain, Yale Univ. Press, 1958
2) C.F. King: Factors Affecting the Choice of Memory, WJCC (May 1961) pp. 405～409
3) R.A. Brooker: Some Techniques for Dealing with Two-level Storage, Computer Journal, **2**, No. 4 (Jan. 1960) pp. 189～194
4) J. Jensen, P. Mondrup and P. Naur: A Storage Allocation Scheme for ALGOL 60, Comm. ACM, **4**, (Oct. 1961) pp. 441～445
5) The Descriptor, Bulletine 5000—20002-p. (Feb. 1961) Burroughs Corp.
6) T. Kilburn, D.B.G. Edwards,: One-Level Storage System, IRE Trans. EC-11, (April 1962) pp. 223～235